A Deep Dive into MobileNet: Mathematical Explanation and Experiments

Xin Huang, Qimeng Tao, Andreas Knaupp, Ling Sun

Columbia University
New York, USA

May 4, 2022

Introduction

- MobileNet is a type of convolutional neural network designed for mobile and embedded vision applications.
- MobileNet is based on a streamlined architecture that uses **depthwise separable convolutions**, **inverted residual block** and **squeeze-and-excitation block** to build lightweight deep neural networks that can have low latency for mobile and embedded devices.

Outline

- Architecture of MobileNet
- Mathematics Explanation of Key Elements in MobileNet
  - Depthwise Separable Convolution
  - Inverted Residual Block
  - Squeeze-and-excitation
- Experiments on Image Classification
- Conclusion and Discussion

Architecture of MobileNet

- Depthwise Separable Convolution
- Inverted Residual Block
- Squeeze-and-excitation

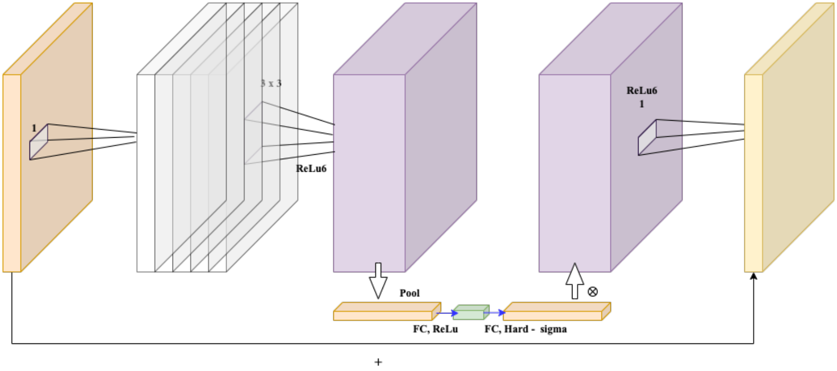


Figure 1: Architecture of MobileNet(V3)

Depth-wise Separable Convolution - Intuitions

Inception in 2014, and Xception, MobileNet in 2017

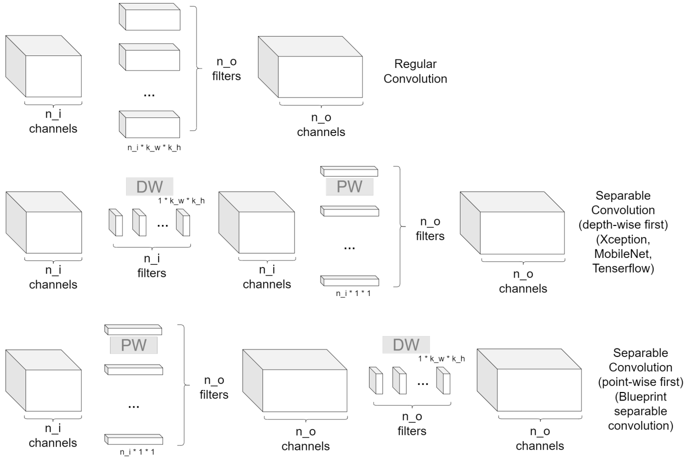
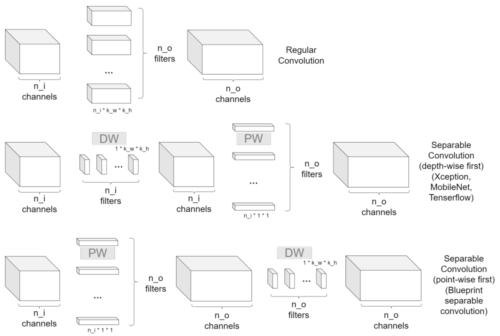


Figure 2: Two versions of depth-wise separable convolutions

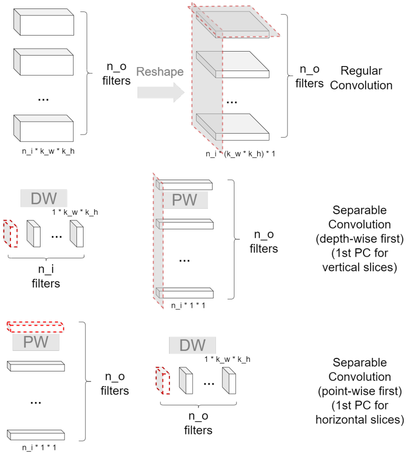
Depth-wise Separable Convolution - Parameter Reduction



params. in regular conv. $O\left(n_i \times k_h \times k_w \times n_o\right)$
params. in depth-wise first $O\left(k_h \times k_w \times n_i + n_i \times n_o\right)$
params. in point-wise first $O\left(n_i \times n_o + k_h \times k_w \times n_o\right)$
approx. $\frac{1}{k_h \times k_w}$ of params. in regular conv., if $n_i, n_o \gg k_h \times k_w$
point-wise first slightly more than depth-wise first, if $n_o > n_i$

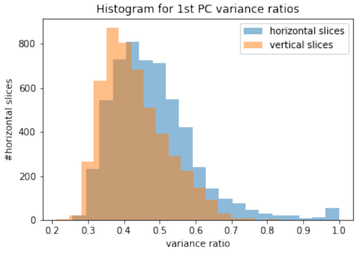
Depth-wise Separable Convolution - PCA Decomposition

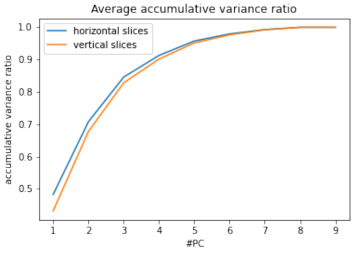
(Guo et al., 2018) A block of regular convolutions $(W_1, ..., W_{n_o})$ $\in \mathbb{R}^{n_o \times n_i \times k_h \times k_w}$ is equivalent to the sum of at most $k_h \times k_w$ blocks of depth-wise separable convolutions. (holds for both version)



Depth-wise Separable Convolution - PCA Decomposition

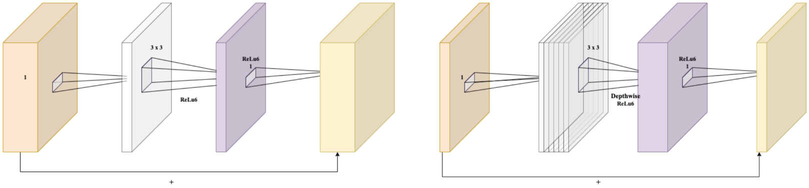
Example VGG19: 16 convolution layers, 4995 vertical slices, 5504 horizontal slices
Vertical slices: 43.15% variance on 1st PC (depth-wise first)
Horizontal slices: 48.24% variance on 1st PC (point-wise first)





Inverted Residual Block

- General Residual Block and Inverted Residual Block
- Residual Block: *wide* $\rightarrow$ *narrow* $\rightarrow$ *wide*
- Inverted Residual Block: *narrow* $\rightarrow$ *wide* $\rightarrow$ *narrow*



(a) Residual Block (a) Inverted Residual Block

Figure 3: General residual block and inverted residual block.

Inverted Residual Block as ODE

Inverted Residual Block shares the same layer transformation with general residual network.

$$\mathbf{h}^{(t+1)} = \mathbf{h}^{(t)} + \sigma\left(\mathbf{h}^{(t)}, \theta^{(t)}\right), \tag{1}$$

where the $\sigma(*)$ is a differentiable layer transformation function (CNN, in this case) and $h^{(t)}$ is the hidden state at time t.

By Euler's rule, we have:

$$\lim_{\tau \to 0} \frac{h^{(t+\tau)} - h^{(t)}}{\tau} = \frac{\mathrm{d}h(t)}{\mathrm{d}t} = \sigma(h(t), t). \tag{2}$$

Inverted Residual Block as ODE

Consider a process that $d\mathbf{z}/dt = \sigma(\mathbf{z}(t), t)$ which we want to approximate, and we have some observations $\{(z_0, t_0), (z_1, t_1), \ldots, (z_M, t_M)\}$, we pick two points $(z(t), t)$, $(z(t+\tau), t+\tau)$

$$L\left(\mathbf{z}\left(t+\tau\right)\right) = L\left(\mathbf{z}\left(t\right) + \int_t^{t+\tau} \sigma(\mathbf{z}(t), t, \theta) dt\right) \tag{3}$$

where $t + \tau$ refers to a small change in time corresponding to t, $L(*)$ is the loss function and $\mathbf{z}(*)$ denotes the hidden state.

Parameters that we want to update during backward propagation.

- $\frac{\partial L}{\partial \mathbf{z}(t)}$ and $\frac{\partial L}{\partial \theta}$

Calculating Gradients

The adjoint:

$$\mathbf{a}(t) = \frac{\partial L}{\partial \mathbf{z}(t)}. \tag{4}$$

From the work of Pontrjagin et al., we have:

$$\frac{d\mathbf{a}(t)}{dt} = -\mathbf{a}(t)^{\top} \frac{\partial \sigma(\mathbf{z}(t), t, \theta)}{\partial \mathbf{z}}. \tag{5}$$

Now we take the integral of equation (5) to calculate adjoint $\mathbf{a}(t)$,

$$\frac{\partial L}{\partial \mathbf{z}\,(t+\tau)} = -\int_{t}^{t+\tau} \mathbf{a}(t)^{\top} \frac{\partial \sigma(\mathbf{z}(t), t, \theta)}{\partial \mathbf{z}}\ \mathrm{d}t. \tag{6}$$

Compute the gradients corresponding to θ, by chain rule, we have:

$$\frac{dL}{d\theta} = -\int_{t}^{t+\tau} \mathbf{a}(t)^{\top} \frac{\partial \sigma(\mathbf{z}(t), t, \theta)}{\partial \theta} dt. \tag{7}$$

ResNet and Neural ODE

Ricky et al. have experiments on Neural ODE and achieve 99.58% accuracy on the MNIST dataset with a relative small network model. Here we show the work on CIFAR-10 dataset and compare the performance between Neural ODE and a small network built up with general residual block.

Table 1: Performance on CIFAR-10.

	Validation Accuracy	# Params
ResNet	85.14%	0.27M
Neural ODE	84.61%	0.27M

As shown in table above, the Neural ODE model achieves similar performance for a small ResNet with general residual block.

Squeeze and Excitation

MobileNet(V3) applies the squeeze-and-excitation block which is originally an outstanding work from Hu et al.

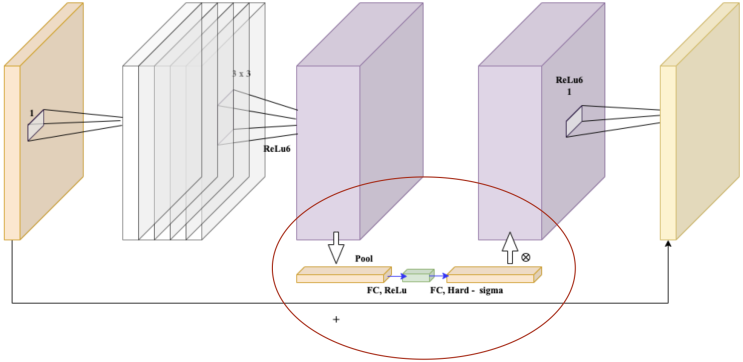


Figure 4: Squeeze-and-excitation block in MobileNet

Squeeze and Excitation

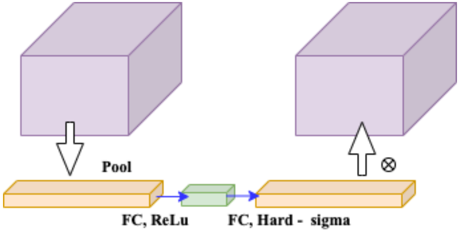


Figure 5: Squeeze-and-excitation block

Squeeze and Excitation include Squeeze (compression) and Excitation (activation) two parts. Among them, Squeeze obtains the global compressed feature vector of the current feature map by performing Global Average Pooling on the feature map, and Excitation obtains the weight of each channel in the feature map through two-layer full connection, and uses the weighted feature map as the input of next layer of network.

Squeeze and Excitation: Squeeze

We can write the model of squeeze-and-excitation (Jie Hu et al., 2019), which we can build up a transformation $\mathbf{F}_{tr}$ from the input to output, firstly, we have:

$$\mathbf{u}_c = \mathbf{v}_c * \mathbf{X} = \sum_{s=1}^{C'} \mathbf{v}_c^s * \mathbf{x}^s \tag{8}$$

where $\mathbf{U} = [u_1, u_2, \cdots, u_C]$ is the output and X is the input, and $\mathbf{V} = [\mathbf{v}_1, \mathbf{v}_2, \ldots, \mathbf{v}_C]$ is earned set of filter kernels, $*$ denotes the convolution operation.

We can denote **global information embedding** (squeeze), formally, a statistic $\mathbf{z} \in \mathbb{R}^C$ is generated by shrinking U through its spatial dimensions $H \times W$, such that the c-th element of z is calculated by:

$$z_c = \mathbf{F}_{sq}(\mathbf{u}_c) = \frac{1}{H \times W} \sum_{i=1}^{H} \sum_{j=1}^{W} u_c(i,j) \tag{9}$$

Squeeze and Excitation: Excitation

How?

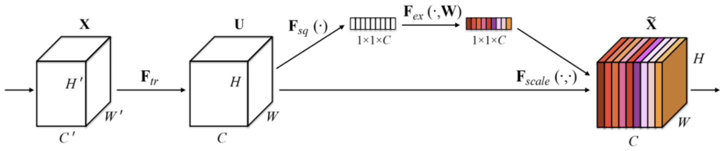


Figure 6: Squeeze and excitation (image src: Jie Hu et al., 2019)

$$\tilde{\mathbf{x}}_c = \mathbf{F}_{\text{scale}}\,(\mathbf{u}_c, s_c) = s_c \mathbf{u}_c \tag{10}$$

Excitation is implemented with **2 fully-connected layers** (FC).

- The first FC compresses C channels into C/r channels to reduce the amount of computation.
- The second FC restores back to C channels.

Squeeze and Excitation in MobileNet

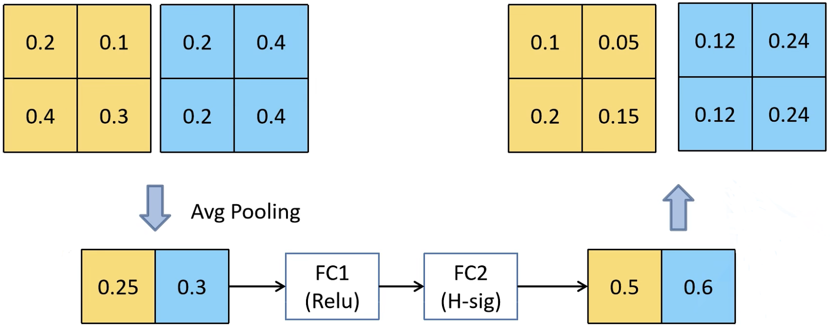
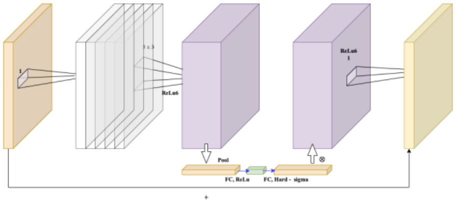


Figure 7: Calculation process

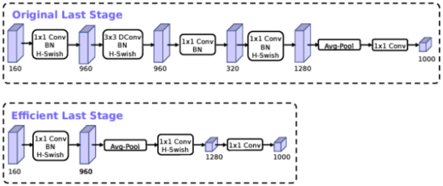
Architecture: Detailed Network Architecture

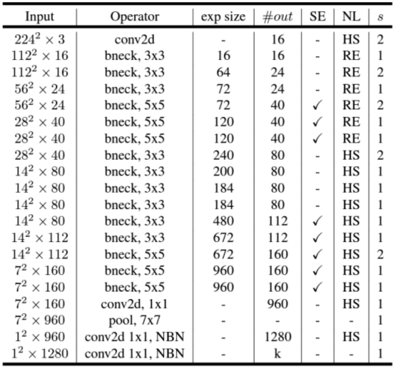


Input	Operator	exp size	$\#out$	SE	NL	s
$224^2 \times 3$	conv2d	-	16	-	HS	2
$112^2 \times 16$	bneck, 3x3	16	16	-	RE	1
$112^2 \times 16$	bneck, 3x3	64	24	-	RE	2
$56^2 \times 24$	bneck, 3x3	72	24	-	RE	1
$56^2 \times 24$	bneck, 5x5	72	40	✓	RE	2
$28^2 \times 40$	bneck, 5x5	120	40	✓	RE	1
$28^2 \times 40$	bneck, 5x5	120	40	✓	RE	1
$28^2 \times 40$	bneck, 3x3	240	80	-	HS	2
$14^2 \times 80$	bneck, 3x3	200	80	-	HS	1
$14^2 \times 80$	bneck, 3x3	184	80	-	HS	1
$14^2 \times 80$	bneck, 3x3	184	80	-	HS	1
$14^2 \times 80$	bneck, 3x3	480	112	✓	HS	1
$14^2 \times 112$	bneck, 3x3	672	112	✓	HS	1
$14^2 \times 112$	bneck, 5x5	672	160	✓	HS	2
$7^2 \times 160$	bneck, 5x5	960	160	✓	HS	1
$7^2 \times 160$	bneck, 5x5	960	160	✓	HS	1
$7^2 \times 160$	conv2d, 1x1	-	960	-	HS	1
$7^2 \times 960$	pool, 7x7	-	-	-	-	1
$1^2 \times 960$	conv2d 1x1, NBN	-	1280	-	HS	1
$1^2 \times 1280$	conv2d 1x1, NBN	-	k	-	-	1

Figure 8: bneck and architecture

Architecture: Redesigning Expensive Layers





Input	Operator	exp size	$\#out$	SE	NL	s
$224^2 \times 3$	conv2d	-	16	-	HS	2
$112^2 \times 16$	bneck, 3x3	16	16	-	RE	1
$112^2 \times 16$	bneck, 3x3	64	24	-	RE	2
$56^2 \times 24$	bneck, 3x3	72	24	-	RE	1
$56^2 \times 24$	bneck, 5x5	72	40	✓	RE	2
$28^2 \times 40$	bneck, 5x5	120	40	✓	RE	1
$28^2 \times 40$	bneck, 5x5	120	40	✓	RE	1
$28^2 \times 40$	bneck, 3x3	240	80	-	HS	2
$14^2 \times 80$	bneck, 3x3	200	80	-	HS	1
$14^2 \times 80$	bneck, 3x3	184	80	-	HS	1
$14^2 \times 80$	bneck, 3x3	184	80	-	HS	1
$14^2 \times 80$	bneck, 3x3	480	112	✓	HS	1
$14^2 \times 112$	bneck, 3x3	672	112	✓	HS	1
$14^2 \times 112$	bneck, 5x5	672	160	✓	HS	2
$7^2 \times 160$	bneck, 5x5	960	160	✓	HS	1
$7^2 \times 160$	bneck, 5x5	960	160	✓	HS	1
$7^2 \times 160$	conv2d, 1x1	-	960	-	HS	1
$7^2 \times 960$	pool, 7x7	-	-	-	-	1
$1^2 \times 960$	conv2d 1x1, NBN	-	1280	-	HS	1
$1^2 \times 1280$	conv2d 1x1, NBN	-	k	-	-	1

Figure 9: Redesigning Expensive Layers

Experiments show that the accuracy of the adjusted model does not change significantly, but saves 7 milliseconds, which 11% of the training time.

Experimental Setup: Dataset

We use the "flowers" dataset from TensorFlow, a dataset comprised of 3,670 images of 5 types of flowers (daisies, dandelions, roses, sunflowers, tulips)

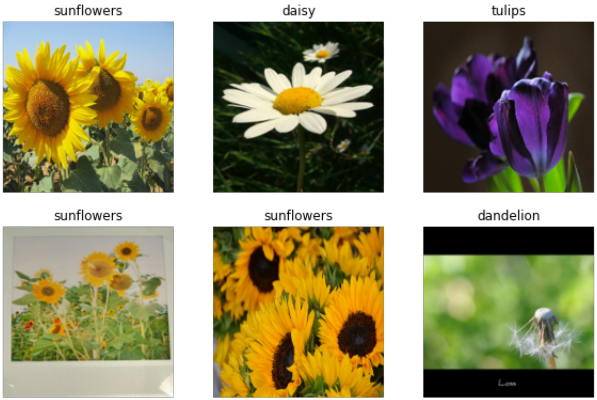


Figure 10: Sampling of dataset images and classes

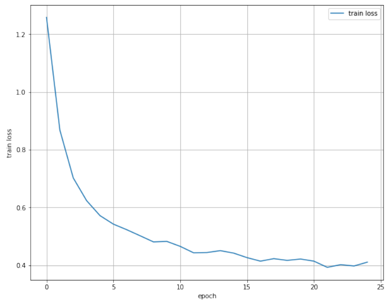
Experimental Setup: Testing Parameters

We aim to compare the performance of MobileNetV2 and MobileNetV3, both when training from scratch and when using pre-trained weights derived from classification of ImageNet.

Training Hyperparameters (consistent across all tests):

- Optimizer: Adam
- Learning Rate: .0001
- Batch Size: 16
- No data augmentation or early stopping

Experiments: Training



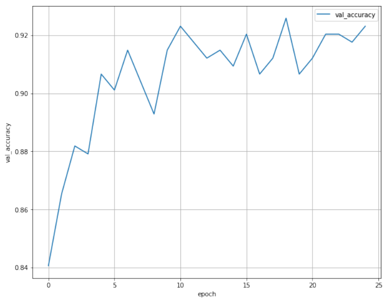
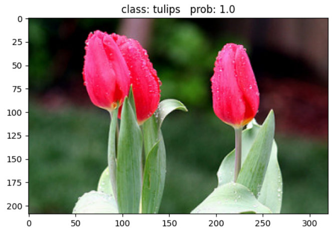


Figure 11: Training Loss (L) and Validation Accuracy (R) for pre-trained MobileNetV2, initialized with ImageNet weights

Experiments: Sample Classification Results



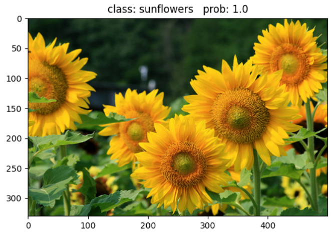


Figure 12: Sample Predictions from MobileNetV3 Large learn from scratch

Experiments: Analysis

Model Accuracy versus Parameters

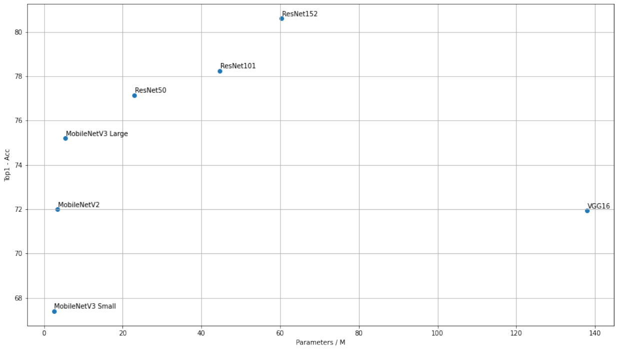


Figure 13: Comparative Results of tested models

Conclusion

- MobileNet architectures provide significant advantages in efficiency, making them an ideal choice for mobile and embedded applications
- The efficiency benefits of MobileNet architectures are afforded by novel architectural features like depthwise-separable convolutions, inverted residual connections, and squeeze-excitation blocks
- These benefits can be seen in the performance of MobileNet architectures in tasks such as classification, where they are competitive with, and sometimes better, than established architectures like VGG, with a vastly reduced set of parameters

Thanks

Thanks for listening!

Reference

[1] Y. LeCun, Y. Bengio, and G. Hinton, “Deep learning,” nature, vol. 521, no. 7553, p. 436, 2015.
[2] K. Simonyan and A. Zisserman, “Very deep convolutional networks for large-scale image recognition,” in International Conference on Learning Representations, 2015.
[3] K. He, X. Zhang, S. Ren, and J. Sun, “Deep residual learning for image recognition,” 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 770–778, 2016.
[4] A. G. Howard, M. Zhu, B. Chen, D. Kalenichenko, W. Wang, T. Weyand, M. Andreetto, and H. Adam, “Mobilenets: Efficient convolutional neural networks for mobile vision applications,” 2017. cite arxiv:1704.04861.
[5] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, and L.-C. Chen, “Mobilenetv2: Inverted residuals and linear bottlenecks,” 2018. cite arxiv:1801.04381.

Reference

[6] A. G. Howard, M. Sandler, G. Chu, L.-C. Chen, B. Chen, M. Tan, W. Wang, Y. Zhu, R. Pang, V. Vasudevan, Q. V. Le, and H. Adam, “Searching for mobilenetv3,” 2019 IEEE/CVF International Conference on Computer Vision (ICCV), pp. 1314–1324, 2019.
[7] F. Chollet, “Xception: Deep learning with depthwise separable convolutions,” in Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 1251–1258, 2017.
[8] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna, “Rethinking the inception archi- tecture for computer vision,” in Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 2818–2826, 2016.
[9] J. Guo, Y. Li, W. Lin, Y. Chen, and J. Li, “Network decoupling: From regular to depthwise separable convolutions,” arXiv preprint arXiv:1808.05517, 2018.

Reference

[10] D. Haase and M. Amthor, “Rethinking depthwise separable convolutions: How intra-kernel correlations lead to improved mobilenets,” in Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 14600–14609, 2020.
[11] A. Krizhevsky, I. Sutskever, and G. E. Hinton, “Imagenet classification with deep convolutional neural networks,” in Advances in Neural Information Processing Systems 25 (F. Pereira, C. J. C. Burges, L. Bottou, and K. Q. Weinberger, eds.), pp. 1097–1105, Curran Associates, Inc., 2012.
[12] K. He, X. Zhang, S. Ren, and J. Sun, “Identity mappings in deep residual networks,” in Computer Vision – ECCV 2016 (B. Leibe, J. Matas, N. Sebe, and M. Welling, eds.), (Cham), pp. 630–645, Springer International Publishing, 2016.

Reference

[13] R. T. Q. Chen, Y. Rubanova, J. Bettencourt, and D. K. Duvenaud, “Neural ordinary differential equations,” in Advances in Neural Information Processing Systems (S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett, eds.), vol. 31, Curran Associates, Inc., 2018.
[14] L. Pontrjagin, V. Boltyanskii, R. Gamkrelidze, E. Mishchenko, and D. Brown, The Mathe- matical Theory of Optimal Processes. International series of monographs in pure and applied mathematics, Wiley, 1962.
[15] N. Ganaba, “Deep learning: Hydrodynamics, and lie-poisson hamilton-jacobi theory,” 05 2021.
[16] J. Hu, L. Shen, and G. Sun, “Squeeze-and-excitation networks,” CoRR, vol. abs/1709.01507, 2017.
[17] R. M. Errico, “What is an adjoint model,” Bulletin of the American Meteorological Society, vol. 78, pp. 2577–2591, 1997.