Title: MobileNet

URL: https://en.wikipedia.org/wiki/MobileNet

PageID: 78148688

Categories: Category:Computer vision, Category:Google software, Category:Machine learning

---

MobileNet is a family of convolutional neural network (CNN) architectures designed for image classification , object detection , and other computer vision tasks. They are designed for small size, low latency, and low power consumption, making them suitable for on-device inference and edge computing on resource-constrained devices like mobile phones and embedded systems . They were originally designed to be run efficiently on mobile devices with TensorFlow Lite .

The need for efficient deep learning models on mobile devices led researchers at Google to develop MobileNet. As of October 2024 [update] , the family has four versions, each improving upon the previous one in terms of performance and efficiency.

Features

V1

MobileNetV1 was published in April 2017. [1] [2] Its main architectural innovation was incorporation of depthwise separable convolutions . It was first developed by Laurent Sifre during an internship at Google Brain in 2013 as an architectural variation on AlexNet to improve convergence speed and model size. [3]

The depthwise separable convolution decomposes a single standard convolution into two convolutions: a depthwise convolution that filters each input channel independently and a pointwise convolution (1×1 {\displaystyle 1\times 1} convolution) that combines the outputs of the depthwise convolution. This factorization significantly reduces computational cost.

The MobileNetV1 has two hyperparameters: a width multiplier α {\displaystyle \alpha } that controls the number of channels in each layer. Smaller values of α {\displaystyle \alpha } lead to smaller and faster models, but at the cost of reduced accuracy, and a resolution multiplier ρ {\displaystyle \rho } , which controls the input resolution of the images. Lower resolutions result in faster processing but potentially lower accuracy.

V2

MobileNetV2 was published in March 2019. [4] [5] It uses inverted residual layers and linear bottlenecks .

Inverted residuals modify the traditional residual block structure. Instead of compressing the input channels before the depthwise convolution, they expand them. This expansion is followed by a 1×1 {\displaystyle 1\times 1} depthwise convolution and then a 1×1 {\displaystyle 1\times 1} projection layer that reduces the number of channels back down. This inverted structure helps to maintain representational capacity by allowing the depthwise convolution to operate on a higher-dimensional feature space, thus preserving more information flow during the convolutional process.

Linear bottlenecks removes the typical ReLU activation function in the projection layers. This was rationalized by arguing that that nonlinear activation loses information in lower-dimensional spaces, which is problematic when the number of channels is already small.

V3

MobileNetV3 was published in 2019. [6] [7] The publication included MobileNetV3-Small, MobileNetV3-Large, and MobileNetEdgeTPU (optimized for Pixel 4). They were found by a form of neural architecture search (NAS) that takes mobile latency into account, to achieve good trade-off between accuracy and latency. [8] [9] It used piecewise-linear approximations of swish and

sigmoid activation functions (which they called "h-swish" and "h-sigmoid"), squeeze-and-excitation modules, [10] and the inverted bottlenecks of MobileNetV2.

V4

MobileNetV4 was published in September 2024. [11] [12] The publication included a large number of architectures found by NAS.

Inspired by Vision Transformers , the V4 series included multi-query attention . [13] It also unified both inverted residual and inverted bottleneck from the V3 series with the "universal inverted bottleneck", which includes these two as special cases.

See also

Convolutional neural network

Deep learning

TensorFlow Lite

References

External links

"mobilenet" . GitHub . Retrieved 2024-10-18 .

"Keras documentation: MobileNet, MobileNetV2, and MobileNetV3" . Keras . Retrieved October 18, 2024 .

v

t

e

Google

Google Brain

Google DeepMind

AlphaGo (2015)

Master (2016)

AlphaGo Zero (2017)

AlphaZero (2017)

MuZero (2019)

Fan Hui (2015)

Lee Sedol (2016)

Ke Jie (2017)

AlphaGo (2017)

The MANIAC (2023)

AlphaFold (2018)

AlphaStar (2019)

AlphaDev (2023)

AlphaGeometry (2024)

AlphaGenome (2025)

Inception (2014)

WaveNet (2016)

MobileNet (2017)
Transformer (2017)
EfficientNet (2019)
Gato (2022)
Quantum Artificial Intelligence Lab
TensorFlow
Tensor Processing Unit
Assistant (2016)
Sparrow (2022)
Gemini (2023)
BERT (2018)
XLNet (2019)
T5 (2019)
LaMDA (2021)
Chinchilla (2022)
PaLM (2022)
Imagen (2023)
Gemini (2023)
VideoPoet (2024)
Gemma (2024)
Veo (2024)
DreamBooth (2022)
NotebookLM (2023)
Vids (2024)
Gemini Robotics (2025)
" Attention Is All You Need "
Future of Go Summit
Generative pre-trained transformer
Google Labs
Google Pixel
Google Workspace
Robot Constitution
Category
Commons
v
t
e
Differentiable programming

Information geometry

Statistical manifold

Automatic differentiation

Neuromorphic computing

Pattern recognition

Ricci calculus

Computational learning theory

Inductive bias

IPU

TPU

VPU

Memristor

SpiNNaker

TensorFlow

PyTorch

Keras

scikit-learn

Theano

JAX

Flux.jl

MindSpore

Portals Computer programming Technology

Computer programming

Technology