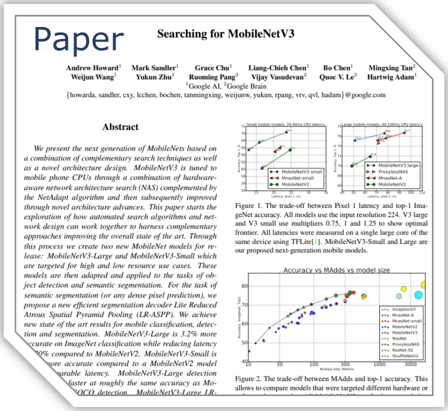


Searching for MobileNetV3

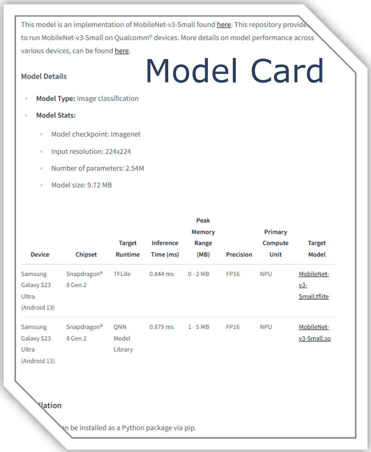
Andrew Howard[1] Mark Sandler[1] Grace Chu[1] Liang-Chieh Chen[1] Bo Chen[1] Mingxing Tan[2]
Weijun Wang[1] Yukun Zhu[1] Ruoming Pang[2] Vijay Vasudevan[2] Quoc V. Le[2] Hartwig Adam[1]
[1]Google AI, [2]Google Brain
{howarda, sandler, cxy, lcchen, bochen, tanmingxing, weijunw, yukun, rpang, vrv, qvl, hadam}@google.com

Abstract

We present the next generation of MobileNets based on a combination of complementary search techniques as well as a novel architecture design. MobileNetV3 is tuned to mobile phone CPUs through a combination of hardware-aware network architecture search (NAS) complemented by the NetAdapt algorithm and then subsequently improved through novel architecture advances. This paper starts the exploration of how automated search algorithms and network design can work together to harness complementary approaches improving the overall state of the art. Through this process we create two new MobileNet models for release: MobileNetV3-Large and MobileNetV3-Small which are targeted for high and low resource use cases. These models are then adapted and applied to the tasks of object detection and semantic segmentation. For the task of semantic segmentation (or any dense pixel prediction), we propose a new efficient segmentation decoder Lite Reduced Atrous Spatial Pyramid Pooling (LR-ASPP). We achieve new state of the art results for mobile classification, detection and segmentation. MobileNetV3-Large is 3.2% more ccurate on ImageNet classification while reducing latency 20% compared to MobileNetV2. MobileNetV3-Small is ore accurate compared to a MobileNetV2 model arable latency. MobileNetV3-Large detection faster at roughly the same accuracy as Mo-OCO detection. MobileNetV3-Large LR-

Figure 1. The trade-off between Pixel 1 latency and top-1 ImageNet accuracy. All models use the input resolution 224. V3 large and V3 small use multipliers 0.75, 1 and 1.25 to show optimal frontier. All latencies were measured on a single large core of the same device using TFLite[1]. MobileNetV3-Small and Large are our proposed next-generation mobile models.

Figure 2. The trade-off between MAdds and top-1 accuracy. This allows to compare models that were targeted different hardware or



This model is an implementation of MobileNet-v3-Small found here. This repository provide to run MobileNet-v3-Small on Qualcomm® devices. More details on model performance across various devices, can be found here.

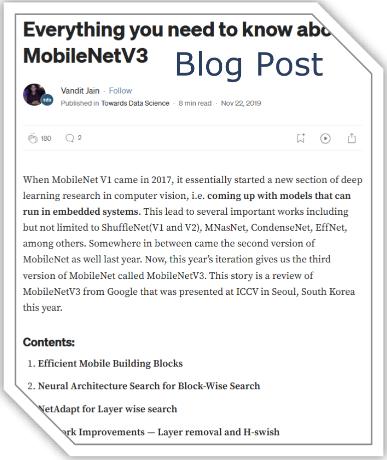
Model Details

- **Model Type:** Image classification
- **Model Stats:**
  - Model checkpoint: Imagenet
  - Input resolution: 224x224
  - Number of parameters: 2.54M
  - Model size: 9.72 MB

Device	Chipset	Target Runtime	Inference Time (ms)	Peak Memory Range (MB)	Precision	Primary Compute Unit	Target Model
Samsung Galaxy S23 Ultra (Android 13)	Snapdragon® 8 Gen 2	TFLite	0.844 ms	0 - 2 MB	FP16	NPU	MobileNet-v3-Small.tflite
Samsung Galaxy S23 Ultra (Android 13)	Snapdragon® 8 Gen 2	QNN Model Library	0.879 ms	1 - 5 MB	FP16	NPU	MobileNet-v3-Small.so

llation

n be installed as a Python package via pip.



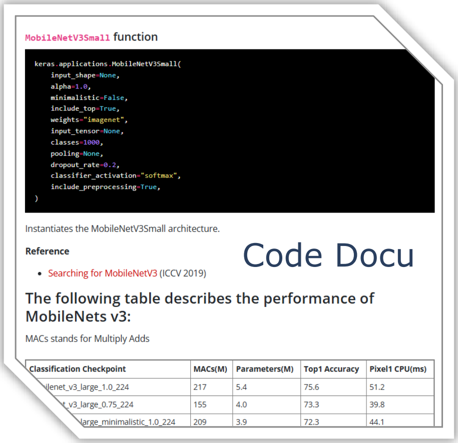
Everything you need to know abo MobileNetV3

Vandit Jain · Follow
Published in Towards Data Science · 8 min read · Nov 22, 2019

When MobileNet V1 came in 2017, it essentially started a new section of deep learning research in computer vision, i.e. **coming up with models that can run in embedded systems.** This lead to several important works including but not limited to ShuffleNet(V1 and V2), MNasNet, CondenseNet, EffNet, among others. Somewhere in between came the second version of MobileNet as well last year. Now, this year's iteration gives us the third version of MobileNet called MobileNetV3. This story is a review of MobileNetV3 from Google that was presented at ICCV in Seoul, South Korea this year.

Contents:

1. Efficient Mobile Building Blocks
2. Neural Architecture Search for Block-Wise Search
NetAdapt for Layer wise search
rk Improvements — Layer removal and H-swish



`MobileNetV3Small` function

```
keras.applications.MobileNetV3Small(
    input_shape=None,
    alpha=1.0,
    minimalistic=False,
    include_top=True,
    weights="imagenet",
    input_tensor=None,
    classes=1000,
    pooling=None,
    dropout_rate=0.2,
    classifier_activation="softmax",
    include_preprocessing=True,
)
```

Instantiates the MobileNetV3Small architecture.

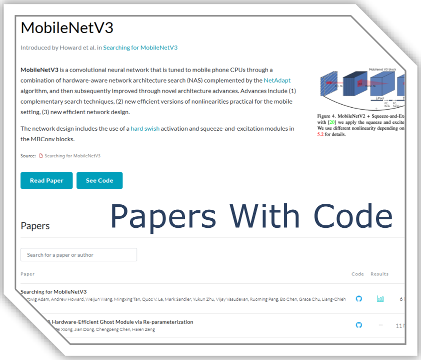
Reference

- Searching for MobileNetV3 (ICCV 2019)

The following table describes the performance of MobileNets v3:

MACs stands for Multiply Adds

Classification Checkpoint	MACs(M)	Parameters(M)	Top1 Accuracy	Pixel1 CPU(ms)
ilenet_v3_large_1.0_224	217	5.4	75.6	51.2
t_v3_large_0.75_224	155	4.0	73.3	39.8
large_minimalistic_1.0_224	209	3.9	72.3	44.1



MobileNetV3

Introduced by Howard et al. in Searching for MobileNetV3

MobileNetV3 is a convolutional neural network that is tuned to mobile phone CPUs through a combination of hardware-aware network architecture search (NAS) complemented by the NetAdapt algorithm, and then subsequently improved through novel architecture advances. Advances include (1) complementary search techniques, (2) new efficient versions of nonlinearities practical for the mobile setting, (3) new efficient network design.

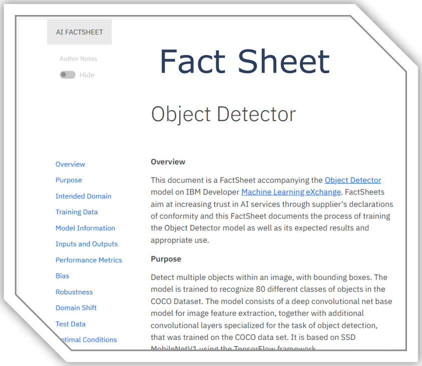
The network design includes the use of a hard swish activation and squeeze-and-excitation modules in the MBConv blocks.

Figure 4. MobileNetV2 + Squeeze-and-Ex with [20] we apply the squeeze and excite We use different nonlinearity depending on 5.2 for details.

Source: Searching for MobileNetV3

Papers

Paper	Code	Results	
Searching for MobileNetV3 rtwig Adam, Andrew Howard, Weijun Wang, Mingxing Tan, Quoc V. Le, Mark Sandler, Yukun Zhu, Vijay Vasudevan, Ruoming Pang, Bo Chen, Grace Chu, Liang-Chieh			6
A Hardware-Efficient Ghost Module via Re-parameterization ei Xiong, Jian Dong, Chengpeng Chen, Haien Zeng		–	11



Object Detector

Overview

This document is a FactSheet accompanying the Object Detector model on IBM Developer Machine Learning eXchange. FactSheets aim at increasing trust in AI services through supplier's declarations of conformity and this FactSheet documents the process of training the Object Detector model as well as its expected results and appropriate use.

Purpose

Detect multiple objects within an image, with bounding boxes. The model is trained to recognize 80 different classes of objects in the COCO Dataset. The model consists of a deep convolutional net base model for image feature extraction, together with additional convolutional layers specialized for the task of object detection, that was trained on the COCO data set. It is based on SSD MobileNetV1 using the TensorFlow framework