Accelerate ML inferences on mobile devices

with Android Neural Networks API (NNAPI)

Agenda

- What is NNAPI?
- Current features
- Performance and Power
- How to use NNAPI

What is NNAPI

Introduction

NDK API for a neural networks inference on hardware accelerators

- C API
- Fast evolving
- Backward compatible

NNAPI 1.0 (Android O-MR1)

- 29 operators, float32 or unsigned asymmetric quantization

NNAPI 1.1 (Android P)

- 38 operators

NNAPI 1.2 (Android Q)

- 94 operators
- float16 and signed per-channel quantization
- Introspection API
- Vendor extension

NNAPI 1.3 (Android R)

- 101 operators
- Signed asymmetric quantization
- Control Flow, QoS, memory domains, async command queue
- Runtime is an updatable APEX module

GPU

Graphics processing unit

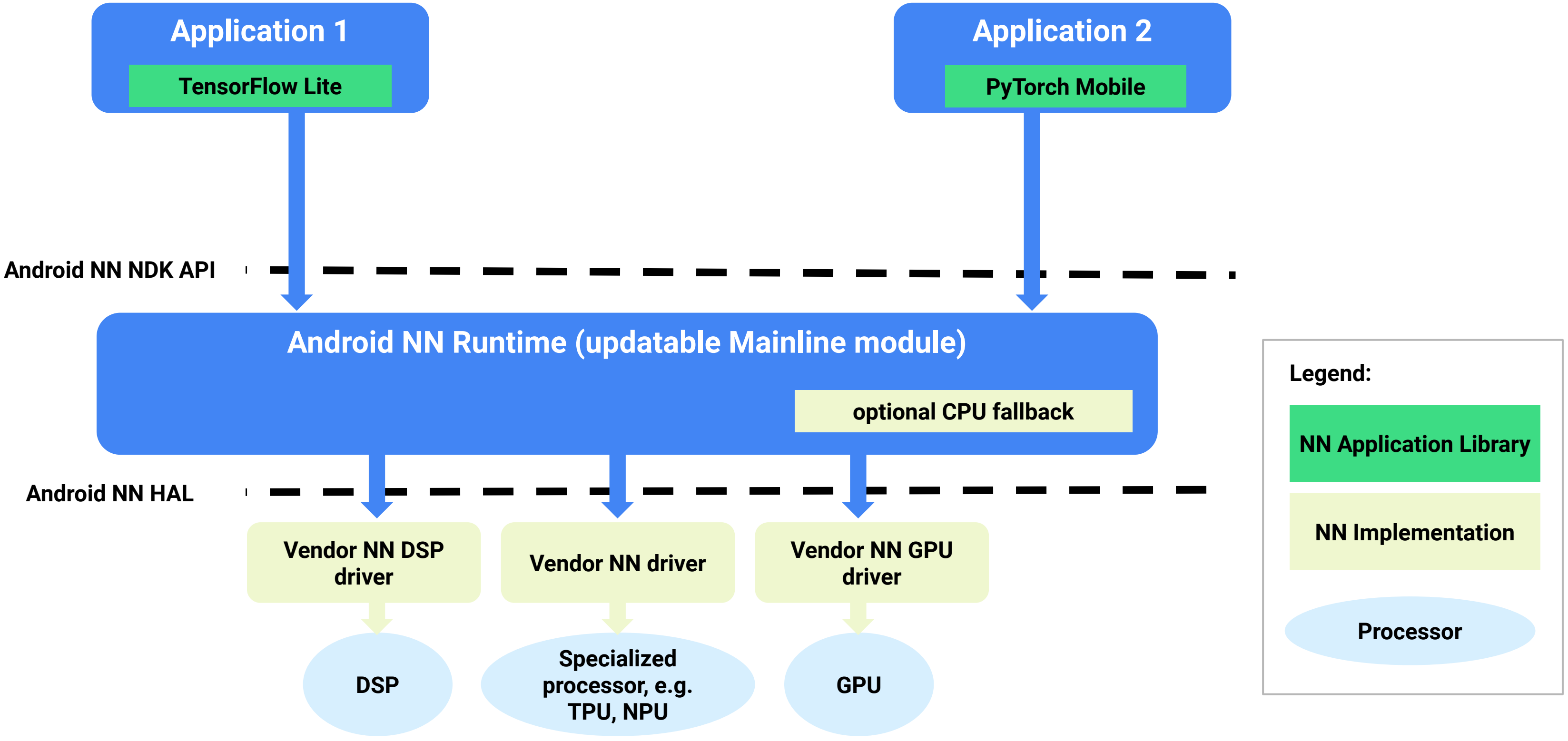
DSP

Digital signal processor

NPU

Neural processing unit

Architecture

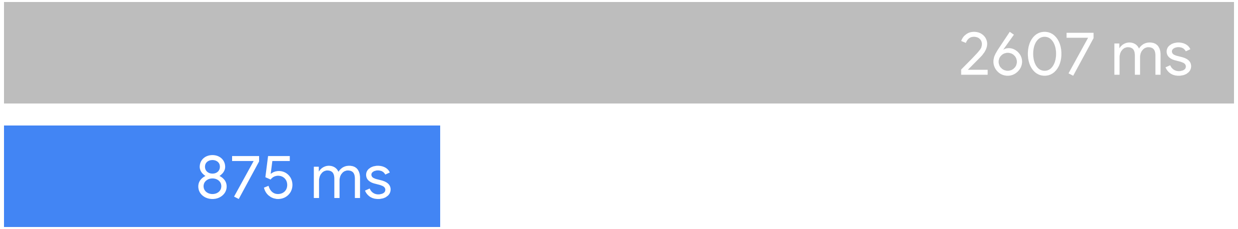


https://source.android.com/devices/neural-networks

Performance and Power

3x

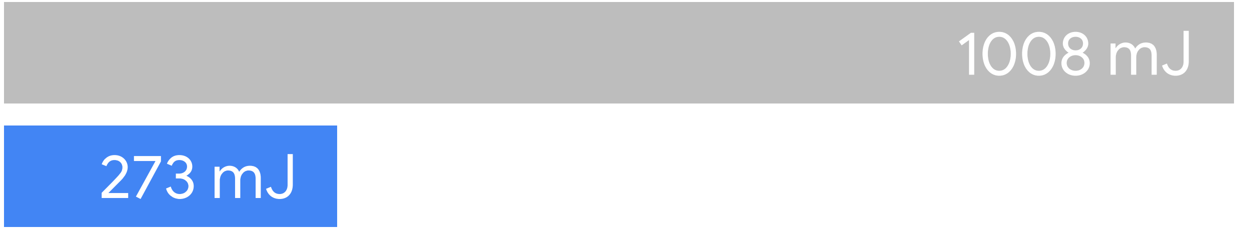
latency reduction*



Note: Worst case latency @ 500mW power cap

3.7x

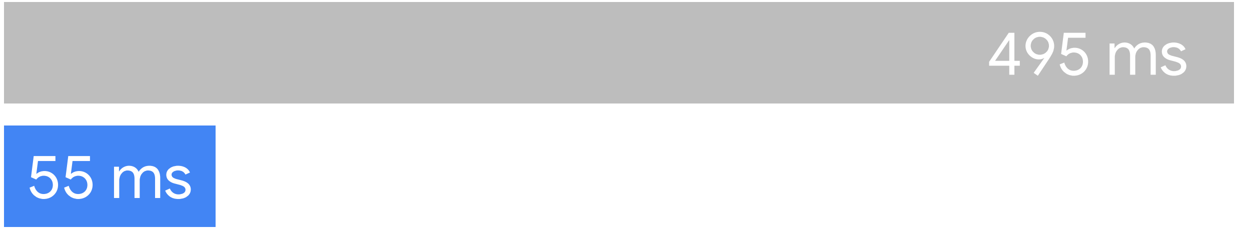
power reduction*



* Source: Google. Based on Google Lens OCR running on the AI Engine in Qualcomm Snapdragon 855 with Android Q

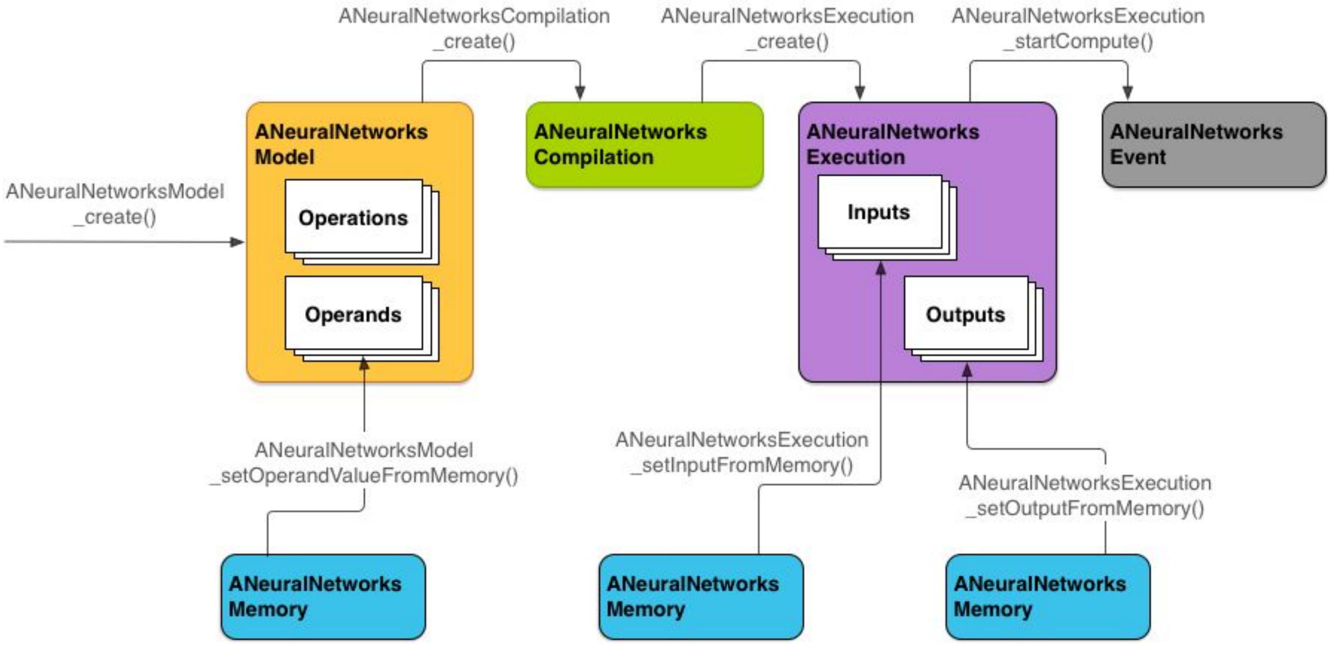
9x

latency reduction*



* Source: Google. Based on ML Kit Face Detection running on MediaTek Helio P90

How to use NNAPI



or in TFLite: `ModifyGraphWithDelegate(NnApiDelegate());`

https://developer.android.com/ndk/guides/neuralnetworks
https://www.tensorflow.org/lite/performance/nnapi

Thanks!