The key steps behind Neural Networks

1. Capture data
2. Train NN Model
3. Clean, label data
   Build NN topology
4. Process & analyze new data using trained NN
5. Convert NN into optimized code for MCU
ST toolbox for Neural Networks

- Capture data
- Clean, label data
- Build NN topology
- Process & analyze new data using trained NN
- Convert NN into optimized code for MCU
Input your framework-dependent, pre-trained Neural Network into the STM32Cube.AI conversion tool.

Automatic and fast generation of an STM32-optimized library.

STM32Cube.AI offers interoperability with state-of-the-art Deep Learning design frameworks.

Any framework that can export models in ONNX open format can be imported.

Train NN Model

Convert NN into optimized code for MCU

Process & analyze new data using trained NN.
Possible conversion strategies: 
Network code generation and interpreter

More Flexible: 
TensorFlow Lite interpreter mode

More optimized: 
Optimized C code generated by

Pre-trained model

Model is pre-compiled and linked only with used ops

Pre-trained model

Model is interpreted and executed by pre-built ops

Flat buffer .tflite

Flat buffer .tflite

TFLite interpreter

User app

User app

built-in ops

custom ops

STM32 device

STM32 device

STM32 BSD

STM32 BSD

STM32.AI lib

STM32.AI lib

STM32 device

STM32 device

STM32 run-time

NN C files

NN C files

User app

User app

STM32 BSP

STM32 BSP

STM32 device
Collecting data & architecting a NN topology

Services provided by Partners

Capture data

Clean, label data
Build NN topology

Selected partners
Neural Networks engineering services support.
Data scientists and Neural network architects.

ST tools to support

ST BLE Sensor mobile phone application
Collect and label data from the SensorTile.

ST BLE Sensor

Partner Program

ST toolbox for neural networks
more than just a conversion tool

- Function packs for quick prototyping
- Audio, Motion and Vision examples
- STM32 Community with dedicated Neural Networks topic
- For support and idea exchange

Process & analyze new data using trained NN

Convert NN into optimized code for MCU
STM32 solutions for AI
More than just the STM32Cube.AI

An extensive toolbox to support easy creation of your AI application

1. **Software examples** for Quick prototyping
   Audio, Motion and Vision Function packs
   On ST development Hardware

2. **STM32 Community** with dedicated
   Neural Networks topic

3. **Trainings, hands on, MOOCs and**
   partners **videos**

4. **STM32 AI Partner Program** with dedicated Partners providing
   **Machine or Deep Learning engineering services**

- **AI extension for STM32CubeMX** to **map pre-trained Neural Networks**
Example form factor hardware to capture and process data

- Capture data
- Process & analyze new data using trained NN

STM32L4

Motion MEMS

MP34DT04: Microphone

Balun Filter

BlueNRG-MS: Bluetooth low-energy

STM32L476: Cortex-M4

LPS22HB: Barometer

Motion MEMS

LSM6DSM: 3DAcc+3Dgyro

Motion MEMS

LSM303AGR: 3DAcc+3DMag

www.st.com/SensorTile
www.st.com/SensorTile-edu
Fast go to market module to capture data with more accuracy

More advanced, high accuracy and low power sensors

- First Inertial module with Machine Learning capabilities.
- Motion (accelerometer and gyroscope, magnetometer) and slow motion (inclinometer)
- Altitude (pressure), environment (pressure, temperature, humidity, compass) and sound (sound and ultrasound analog microphone)
- Microsoft IoT services ready to make available on a web dashboard the result of the embedded processing

www.st.com/SensorTileBox
Distributed AI: sensor + STM32
Optimize performance and power consumption

Smart Sensor
with Machine Learning Core

- Best ultra-low-power sensing at high performance
  - 550µA (gyroscope and accelerometer)
    - 200µA less than closest competitor
  - 20~40µA (Accelerometer only for HAR)
- Efficient Finite State Machines: 2µA
- Configurable Machine Learning Core: 4~8µA

Smart STM32
Second level of AI processing

- More advanced and complex NNs
- Decisions on multiple sensors
- NN input can be sensor data and/or sensor Machine Learning decisions
- Multiple Neural Networks support
- Actuation & communication

FSM up to 16
MLC up to 8

Inertial Sensor
New LSM6DSOX

Raw Data
Event Decision

FSM and MLC
Re-configuration

Deep Learning
Neural Networks
Machine Learning

550µA (gyroscope and accelerometer)
200µA less than closest competitor
20~40µA (Accelerometer only for HAR)
Efficient Finite State Machines: 2µA
Configurable Machine Learning Core: 4~8µA

➔
200µA less than closest competitor
20~40µA (Accelerometer only for HAR)
Form factor hardware

AI IoT node for more connectivity

IoTNode

Capture data

Process & analyze new data using trained NN

More debug capabilities
  - Integrated ST-Link/V2.1
  - PMOD extension connector
  - Arduino Uno extension connectors

Sub-1GHz

Wi-Fi

Sub-1GHz

Dynamic NFC Tag

https://www.st.com/IoTnode
OpenMV integration
Fast machine vision prototyping

Configure Machine Vision in real-time over USB in Python

Run and validate optimized Neural Network

OpenMV CAM
Running MicroPython over STM32

https://github.com/openmv/openmv
Function Packs

Simple, fast, optimized
Audio scene classification (ASC)
Audio example in FP-AI-SENSING1 package

Audio Data capture
Labelling controlled by smartphone application
Data stored on the device SD card for future learning
Indoor, Outdoor, In vehicle labelling

Embedded audio pre-processing
Inferences running on the microcontroller
Inference result displayed on mobile app

Demo available
Motion Data Capture

Labelling controlled by smartphone application

Data stored on the device SD card for future learning

Stationary, walking, running, biking, driving labelling

Embedded motion pre-processing

Inferences running on the microcontroller

NN & example dataset provided

Inference result displayed on mobile app

Demo available
Enjoy the food classification demo
- Default demo based on 18 classes (224x224 RGB pictures)
- Several camera image output size possible

Full end-to-end optimized software example
- from camera acquisition to image pre-processing before feeding the NN
- Multiple memory mapping possibilities to optimize and test impact on performances
- Retrain this NN with your own dataset
- Quantize your trained network to optimized inference time and memory usage

Embedded image pre-processing (SW) on the STM32H747

Inferences running on the microcontroller

Pizza 99% 150ms
## Leader in Arm® Cortex®-M 32-bit General Purpose MCU

<table>
<thead>
<tr>
<th>Arm® Cortex® core</th>
<th>-M0</th>
<th>-M0+</th>
<th>-M3</th>
<th>-M33</th>
<th>-M4</th>
<th>-M7</th>
<th>dual -A7 &amp; -M4</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>MPU</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>High Perf</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>STM32F0</strong></td>
<td>106 CoreMark</td>
<td>48 MHz</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>STM32G0</strong></td>
<td>142 CoreMark</td>
<td>64 MHz</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>STM32F1</strong></td>
<td>177 CoreMark</td>
<td>72 MHz</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>STM32F2</strong></td>
<td>398 CoreMark</td>
<td>120 MHz</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>STM32F3</strong></td>
<td>245 CoreMark</td>
<td>72 MHz</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>STM32F4</strong></td>
<td>608 CoreMark</td>
<td>180 MHz</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>STM32F5</strong></td>
<td>245 CoreMark</td>
<td>72 MHz</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>STM32H7</strong></td>
<td>1024 CoreMark</td>
<td>216 MHz</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>STM32F7</strong></td>
<td>1082 CoreMark</td>
<td>216 MHz</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Mainstream</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>STM32L0</strong></td>
<td>75 CoreMark</td>
<td>32 MHz</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>STM32L1</strong></td>
<td>93 CoreMark</td>
<td>32 MHz</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>STM32L1</strong></td>
<td>424 CoreMark</td>
<td>110 MHz</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>STM32L5</strong></td>
<td>273 CoreMark</td>
<td>80 MHz</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>STM32L4</strong></td>
<td>409 CoreMark</td>
<td>120 MHz</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>STM32L4+</strong></td>
<td>409 CoreMark</td>
<td>120 MHz</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Ultra-low Power</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>STM32L0</strong></td>
<td>75 CoreMark</td>
<td>32 MHz</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>STM32L1</strong></td>
<td>93 CoreMark</td>
<td>32 MHz</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>STM32L4</strong></td>
<td>409 CoreMark</td>
<td>120 MHz</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>STM32L4+</strong></td>
<td>409 CoreMark</td>
<td>120 MHz</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Wireless</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>STM32WL</strong></td>
<td>161 CoreMark</td>
<td>48 MHz</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>STM32WB</strong></td>
<td>216 CoreMark</td>
<td>64 MHz</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Compatible with Machine Learning Partner ecosystems
Compatible with Deep Learning STM32Cube.AI ecosystem

- More than 40,000 customers
- Over 4 Billion STM32 shipped since 2007
AI solutions for STM32MP1

Running AI on ST Microprocessors
STM32MP1 microprocessor
Augmented intelligence

- STM32Cube.AI to convert pre-trained NNs for the Cortex-M4 core
- TensorFlow Lite STM32MP1 support up streamed for native NN inferences support on the dual Cortex-A side
Inferences running on the microprocessor in 80ms for image classification

USB camera or built-in camera module

Applicaton examples in C/C++ and Python
- Image classification: 1000 objects classified
- Multiple object detection: 90 classes

Includes code for camera acquisition and image pre-processing

Displayed on STM32MP1-DK2, STM32MP1-EV1 and Avenger96 board

► 2x demos available
www.st.com/STM32CubeAI

Capture Data

Label Data

Train NN

Run on STM32

For more information
Thank you