Artificial Intelligence at the Edge

Moving part of intelligence closer to the data acquisition

- Better user experience
- Realtime, no latency
- Reliable

Add new functions and services with Embedded AI

- Optimized Cloud usage
- Privacy by design (GDPR compliant)
- Sustainable on energy

Contact us at edge.ai@st.com
Computer Vision for STM32

Give vision to your STM32 product for new features and add-on services

- Food classification
- Person presence detection
- Face recognition
- Multiple object detection

FP-AI-VISION1 v1.0
FP-AI-VISION1 v2.0
FP-AI-FACEREC v1.0

Q2 2021*

*available for alpha customers

Keras 🐍 TensorFlowLite 🕒 ONNX

PyTorch And more

STM32 Cube.AI ↔ STM32 CubeMX

Optimized NN files
STM32.AI lib

Customer application
STM32.Vision lib

run-time

*available for alpha customers
Condition monitoring for STM32

Monitor STM32 equipment health for improved uptime and lower maintenance cost

- Vibration monitoring for in-field retrofit of existing systems
- Condition monitoring with current for build-in systems

Get started using dedicated
And industrial boards

STM32 Cube.AI
STM32 Cube MX

- Optimized NN files
- STM32.AI lib
- ML models
- Customer application

FP-AI-NANOEDG1 v1.0 Q1 2021*

*available for alpha customers
AI tools for STM32
The key steps behind Neural Networks

1. Capture data
2. Train NN Model
3. Process & analyze new data using trained NN
4. Convert NN into optimized code for MCU

- Clean, label data
- Build NN topology
ST toolbox for Neural Networks

Capture data
Wide board ecosystem

Clean, label data
Build NN topology

Process & analyze new data using trained NN

Convert NN into optimized code for MCU

STM32 Cube.AI

STM32

7
Easily implement Neural Networks on STM32

Train Neural Network using any major AI frameworks

- K
- TensorFlow Lite
- ONNX
- MATLAB
- PyTorch
- and more...

Convert NN into optimized code

- Select most appropriate MCU
- Review computation and memory consumption per layer

Run on optimized runtime

- Validate code directly on target
- Get accuracy and inference time
- Optimize memory usage
STM32Cube.AI main features

- Generate C-code for pre-trained model
- Support quantized models to reduce RAM, flash and latency with minimal loss of accuracy
- Use light run-time libraries
- Optimize for performance
- Quickly assess model footprint requirements
- Select and configure MCU in STM32CubeMX
- Review model layers in STM32Cube.AI
- Optimize memory allocation
- Fine control of weight mapping
- Split between internal and external memory
- Update model without full FW update

And quickly iterate thanks to on-target validation

STM32Cube.AI is available both as graphical and command line interface
STM32Cube.AI, an STM32CubeMX expansion

- Power Consumption Calculator
- MCU Selector
- Code Generation
- Pinout Configuration
- Middleware Parameters
- Clock Tree Initialization
- Peripherals Configuration
Collecting data & architecting a NN topology

Services provided by Partners

Capture Data

Clean, label data
Build NN topology

ST tools to support

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

Selected partners
Neural Networks engineering services support.
Data scientists and Neural network architects.
Example form factor hardware to capture and process data

SensorTile

Capture Data

Inference on **STM32L476**

STM32L4

STM32L476: Cortex-M4

Balun Filter

BlueNRG-MS: Bluetooth low-energy

MP34DT04: Microphone

LPS22HB: Barometer

Motion MEMS

Motion MEMS

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

- FSM up to 16
- MLC up to 8
- Inertial Sensor
  New LSM6DSOX
- Raw Data
- Event Decision
- FSM and MLC Re-configuration

Smart STM32
Second level of AI processing

- Deep Learning
  Neural Networks
  Machine Learning

- 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

• 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
Form factor hardware
AI IoT node for more connectivity

B-L475E-IOT01A

Capture Data

Inference on STM32L4

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

www.st.com/IoTnode
Wireless Industrial node to capture data at industrial grade

STWIN

Capture Data

Industrial-grade sensors
- Industrial scale 9-DoF motion sensors including accelerometer, gyrometer and an ultra wide-bandwidth vibrometer with ultra low noise
- Very high frequency audio and ultrasound microphone
- High precision temperature and environmental monitoring
- Micro SD card for standalone data logging
- BLE5.0 connectivity and WiFi expansion board
- USART

Inference on STM32L4R9

www.st.com/stwin
STM32H7 discovery boards with camera

- STM32H747I-DISCO with B-CAMS-OMV
- Computer Vision on microcontroller
  - STM32H747 high-performance and DSP with DP-FPU, Arm Cortex-M7 at 480 MHz + Cortex-M4 MCU with 2MB internal Flash, 1MB internal RAM, Chrom-ART Accelerator
  - External memory 2x64MB Quad-SPI NOR Flash and 32MB SDRAM
  - 4” capacitive touch LCD display module with MIPI® DSI interface
  - Camera module adapter board and camera module based on OV5640 5MPx 8b color rolling shutter
  - ST-MEMS digital microphones
  - Ethernet RJ45 and Wi-Fi / cellular expansion boards
- Capture Data
- Inference on STM32H747
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
### Making AI Accessible Now

#### Leader in Arm® Cortex®-M 32-bit General Purpose MCU

<table>
<thead>
<tr>
<th>MPU</th>
<th>STM32MP1</th>
<th>4158 CoreMark</th>
<th>650 MHz Cortex-M7</th>
<th>209 MHz Cortex-M4</th>
</tr>
</thead>
<tbody>
<tr>
<td>High Perf MCUs</td>
<td>STM32F2</td>
<td>398 CoreMark</td>
<td>120 MHz</td>
<td></td>
</tr>
<tr>
<td>STM32G0</td>
<td>142 CoreMark</td>
<td>64 MHz</td>
<td></td>
<td></td>
</tr>
<tr>
<td>STM32F1</td>
<td>177 CoreMark</td>
<td>72 MHz</td>
<td></td>
<td></td>
</tr>
<tr>
<td>STM32F4</td>
<td>608 CoreMark</td>
<td>180 MHz</td>
<td></td>
<td></td>
</tr>
<tr>
<td>STM32H7</td>
<td>3224 CoreMark</td>
<td>240 MHz Cortex-M4</td>
<td>480 MHz Cortex-M7</td>
<td></td>
</tr>
<tr>
<td>STM32F7</td>
<td>1082 CoreMark</td>
<td>216 MHz</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Mainstream MCUs</td>
<td>STM32F0</td>
<td>106 CoreMark</td>
<td>48 MHz</td>
<td></td>
</tr>
<tr>
<td>STM32G0</td>
<td>142 CoreMark</td>
<td>64 MHz</td>
<td></td>
<td></td>
</tr>
<tr>
<td>STM32F1</td>
<td>177 CoreMark</td>
<td>72 MHz</td>
<td></td>
<td></td>
</tr>
<tr>
<td>STM32F3</td>
<td>245 CoreMark</td>
<td>72 MHz</td>
<td></td>
<td></td>
</tr>
<tr>
<td>STM32G4</td>
<td>550 CoreMark</td>
<td>170 MHz</td>
<td></td>
<td></td>
</tr>
<tr>
<td>STM32L0</td>
<td>75 CoreMark</td>
<td>32 MHz</td>
<td></td>
<td></td>
</tr>
<tr>
<td>STM32L1</td>
<td>93 CoreMark</td>
<td>32 MHz</td>
<td></td>
<td></td>
</tr>
<tr>
<td>STM32L5</td>
<td>424 CoreMark</td>
<td>110 MHz</td>
<td></td>
<td></td>
</tr>
<tr>
<td>STM32U5</td>
<td>651 CoreMark</td>
<td>160 MHz</td>
<td></td>
<td></td>
</tr>
<tr>
<td>STM32L4</td>
<td>273 CoreMark</td>
<td>80 MHz</td>
<td></td>
<td></td>
</tr>
<tr>
<td>STM32L4+</td>
<td>409 CoreMark</td>
<td>120 MHz</td>
<td></td>
<td></td>
</tr>
<tr>
<td>STM32U5</td>
<td>651 CoreMark</td>
<td>160 MHz</td>
<td></td>
<td></td>
</tr>
<tr>
<td>STM32WL</td>
<td>161 CoreMark</td>
<td>48 MHz</td>
<td></td>
<td></td>
</tr>
<tr>
<td>STM32WB</td>
<td>216 CoreMark</td>
<td>64 MHz</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Ultra-low Power MCUs</td>
<td>STM32F0</td>
<td>106 CoreMark</td>
<td>48 MHz</td>
<td></td>
</tr>
<tr>
<td>STM32G0</td>
<td>142 CoreMark</td>
<td>64 MHz</td>
<td></td>
<td></td>
</tr>
<tr>
<td>STM32F1</td>
<td>177 CoreMark</td>
<td>72 MHz</td>
<td></td>
<td></td>
</tr>
<tr>
<td>STM32F3</td>
<td>245 CoreMark</td>
<td>72 MHz</td>
<td></td>
<td></td>
</tr>
<tr>
<td>STM32G4</td>
<td>550 CoreMark</td>
<td>170 MHz</td>
<td></td>
<td></td>
</tr>
<tr>
<td>STM32L0</td>
<td>75 CoreMark</td>
<td>32 MHz</td>
<td></td>
<td></td>
</tr>
<tr>
<td>STM32L1</td>
<td>93 CoreMark</td>
<td>32 MHz</td>
<td></td>
<td></td>
</tr>
<tr>
<td>STM32L5</td>
<td>424 CoreMark</td>
<td>110 MHz</td>
<td></td>
<td></td>
</tr>
<tr>
<td>STM32U5</td>
<td>651 CoreMark</td>
<td>160 MHz</td>
<td></td>
<td></td>
</tr>
<tr>
<td>STM32L4</td>
<td>273 CoreMark</td>
<td>80 MHz</td>
<td></td>
<td></td>
</tr>
<tr>
<td>STM32L4+</td>
<td>409 CoreMark</td>
<td>120 MHz</td>
<td></td>
<td></td>
</tr>
<tr>
<td>STM32U5</td>
<td>651 CoreMark</td>
<td>160 MHz</td>
<td></td>
<td></td>
</tr>
<tr>
<td>STM32WL</td>
<td>161 CoreMark</td>
<td>48 MHz</td>
<td></td>
<td></td>
</tr>
<tr>
<td>STM32WB</td>
<td>216 CoreMark</td>
<td>64 MHz</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Wireless MCUs</td>
<td>STM32F0</td>
<td>106 CoreMark</td>
<td>48 MHz</td>
<td></td>
</tr>
<tr>
<td>STM32G0</td>
<td>142 CoreMark</td>
<td>64 MHz</td>
<td></td>
<td></td>
</tr>
<tr>
<td>STM32F1</td>
<td>177 CoreMark</td>
<td>72 MHz</td>
<td></td>
<td></td>
</tr>
<tr>
<td>STM32F3</td>
<td>245 CoreMark</td>
<td>72 MHz</td>
<td></td>
<td></td>
</tr>
<tr>
<td>STM32G4</td>
<td>550 CoreMark</td>
<td>170 MHz</td>
<td></td>
<td></td>
</tr>
<tr>
<td>STM32L0</td>
<td>75 CoreMark</td>
<td>32 MHz</td>
<td></td>
<td></td>
</tr>
<tr>
<td>STM32L1</td>
<td>93 CoreMark</td>
<td>32 MHz</td>
<td></td>
<td></td>
</tr>
<tr>
<td>STM32L5</td>
<td>424 CoreMark</td>
<td>110 MHz</td>
<td></td>
<td></td>
</tr>
<tr>
<td>STM32U5</td>
<td>651 CoreMark</td>
<td>160 MHz</td>
<td></td>
<td></td>
</tr>
<tr>
<td>STM32L4</td>
<td>273 CoreMark</td>
<td>80 MHz</td>
<td></td>
<td></td>
</tr>
<tr>
<td>STM32L4+</td>
<td>409 CoreMark</td>
<td>120 MHz</td>
<td></td>
<td></td>
</tr>
<tr>
<td>STM32U5</td>
<td>651 CoreMark</td>
<td>160 MHz</td>
<td></td>
<td></td>
</tr>
<tr>
<td>STM32WL</td>
<td>161 CoreMark</td>
<td>48 MHz</td>
<td></td>
<td></td>
</tr>
<tr>
<td>STM32WB</td>
<td>216 CoreMark</td>
<td>64 MHz</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Compatiable with **Machine Learning** Partner ecosystems

Compatiable with **Deep Learning** STM32Cube.AI ecosystem

More than **40,000 customers**  
Over **4 Billion STM32 shipped since 2007**
Function Packs
AI Solutions on STM32

A full development ecosystem to create your AI application

STM32 Cube.AI

AI extension for STM32CubeMX to map pre-trained Neural Networks

STM32 Community with dedicated Neural Networks topic and AI expert partners

Trainings, hands on, MOOCs and partners videos

Person presence detection
Food classification

FP-AI-VISION1

People activity recognition
Audio scene classification

FP-AI-SENSING1

Condition-based monitoring

FP-AI-NANOEDG1
Audio scene classification (ASC)

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

NN & example dataset provided

Inference result displayed on mobile app

Demo available
Human activity motion recognition (HAR)

Motion Data Capture

Labelling controlled by smartphone application

Data stored on the device SD card for future learning

Stationary, walking, running, biking, driving labelling

5 classes example

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
One-class image classification demo
- Models from tensorflow.org (L4R and H7) and MobileNet v2 (H7 only)
- QVGA 320x240 color image on the LCD
- Can adapt camera flipping depending on which side camera is placed

Full end-to-end optimized software example
- from camera acquisition to image pre-processing before feeding the NN
- Multiple models fitting STM32L4R to STM32H7 depending on required performance and cost
- Visual wake word for Smart home or cities security cameras
- Reduce false alarms due to object movement detection

Inferences running on the microcontroller

STM32L4R display accelerations

Unused 9.4 %
Person 34.8 %
No Person 87.1 %
Embed face recognition in your IoT project

**STM32H7**

**User-personalized services / features**

Adjust automatically per user
- Device *preferred settings* or *ergonomics* per user
- Customized device *behavior* / action for registered user
- Customize *alerts*
- *Prevent child injury* with underage appliance lock
- Create user-specific *automations*

**Features**
- On-device face enrollment of multiple users
- Real-time face recognition, display enrolled image
- Displays match accuracy and inference speed

Available on-demand as a software library
Contact edge.ai@st.com to get access
Condition monitoring on STWIN

Get straight to proof-of-concept with full anomaly detection system without deep Data Science knowledge

Collect dataset from industrial-grade vibration sensor

Generate free ML library

Integrate and deploy

Install on premise

Incremental learning on-target

Monitor anomalies on-target

Download the dedicated SW package
AI solutions for STM32MP1
STM32MP1 microprocessor
Augmented intelligence

- STM32Cube.AI to convert pre-trained NNs for the Cortex-M4 core
- TensorFlow Lite STM32MP1 support upstreamed 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

Applications 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 STM32MP157-DK2, STM32MP157-EV1 and Avenger96 board

2x demos available
ST co-development and partnerships
Leverage the power of Edge AI

AI co-development partnerships
Contact us at edge.ai@st.com

Meet our expert AIS partners
Visit st.com/stm32cubeai

Multiple object detection with thermal imager

Predictive maintenance of reflow oven

www.st.com/STM32CubeAI
Contact us at edge.ai@st.com
Releasing your creativity

/STM32

@ST_World

community.st.com

www.st.com/STM32CubeAI
Thank you