



# 「Nano Edge AI Studio」 「STM32 Cube.AI」 を用いたマイコンへの AI実装方法のご紹介

2026年 2月10日

STマイクロエレクトロニクス  
マイクロコントローラ・デジタルIC・RF製品グループ

市岡 徹郎

# TRONプログラミングコンテスト2026 対象STM32ボードと組み込みAI開発環境

## 対象STM32ボード



初中級者 / 時系列データ処理 / 低消費電力向け  
“NUCLEO-H533RE”



上級者 / コンピュータ・ビジョン / 高度なAI実行向け  
“STM32N6570-DK”



## 組み込みAI開発環境



時系列データ処理など  
初心者でも扱いやすい  
機械学習ライブラリ生成ツール



ディープラーニングなど  
ニューラルネットワークの  
マイコン用コード最適化ツール



# Nano Edge AI Studio



# AI初心者でも扱える機械学習モデル開発ツール

NANOEDGE AI  
STUDIO 



データさえ用意できれば、**AIモデル設計やプログラミングなし**でマイコン向け機械学習ライブラリ生成可能

入力データに**最適なエッジAIモデル / ハイパー・パラメータ / 前処理**を**自動探索**

**オンデバイス学習**により、再学習無しで実装済みソリューションを調整 & 最適化

# NanoEdge AI Studioで解決できる3つの事例

## AutoMLを使用することで改善できる事例

異常検出  
Anomaly Detection →



**機器異常の予兆を検出したい**

機器が自身の正常状態に対しての類似度を算出

状態分類  
Classification →



**機器の動作状態、動作環境、使用状況を分類したい**

機器が外部機器との接続なしで自身の状態をあらかじめ定義された状態のどれかに分類

回帰予測  
Extrapolation →



**未知の状態を予測したい**

機器の各パラメータを予測したり、将来の状態を予測

# NanoEdge AI Studio 組み込みAI開発フロー

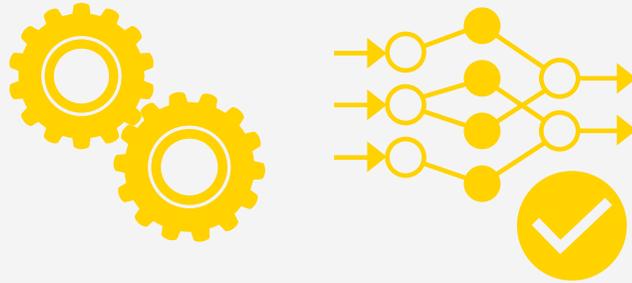
## 1. データ準備



データ取得

データ処理

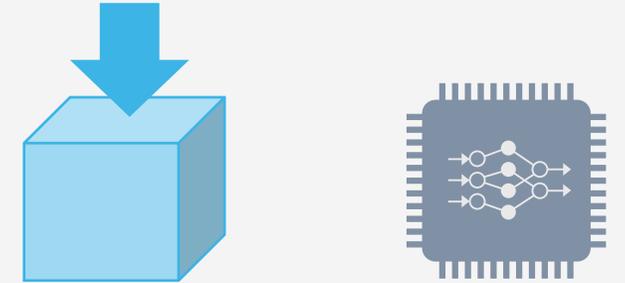
## 2. モデル開発



モデルの  
選択と学習

モデルのテスト

## 3. モデル実装



モデルの  
ライブラリ生成

モデル推論

自動組み込みAIライブラリ生成

データ・ロガー  
機能

NANOEDGE AI  
STUDIO 



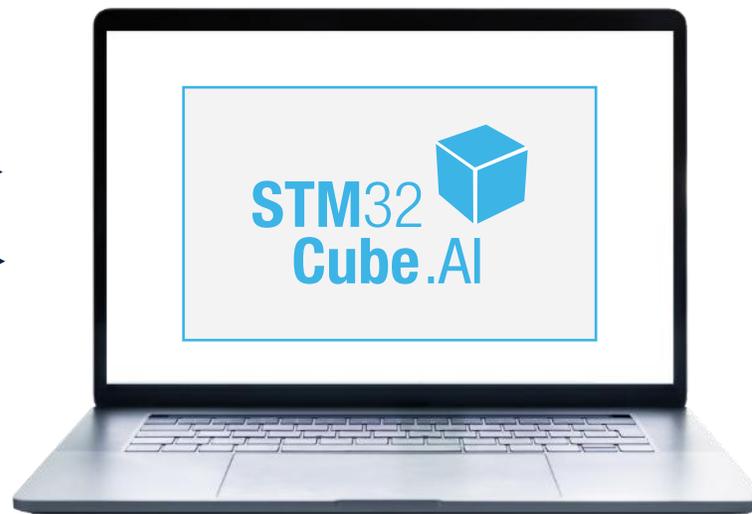
# STM32Cube.AI

# 独自のAIモデルをSTM32向けに最適化

機械学習



ディープ・ラーニング



- 最適なマイコン選択 & AIモデルの読み込み
- 最適化 & 検証
- プロジェクト生成



STM32マイコンへの実装



# STM32 Model Zoo : エッジAI最適化モデルから開発

## STM32向けに開発されたアプリケーション・モデル・ライブラリ

### アクティビティ認識



モーション・センシング

### 画像分類



コンピュータ・ビジョン

### オーディオ検知



オーディオ分類

### 物体検知



コンピュータ・ビジョン

**Github**でホスティング

### モデルの学習スクリプト

- 生成 & 検証のスクリプト

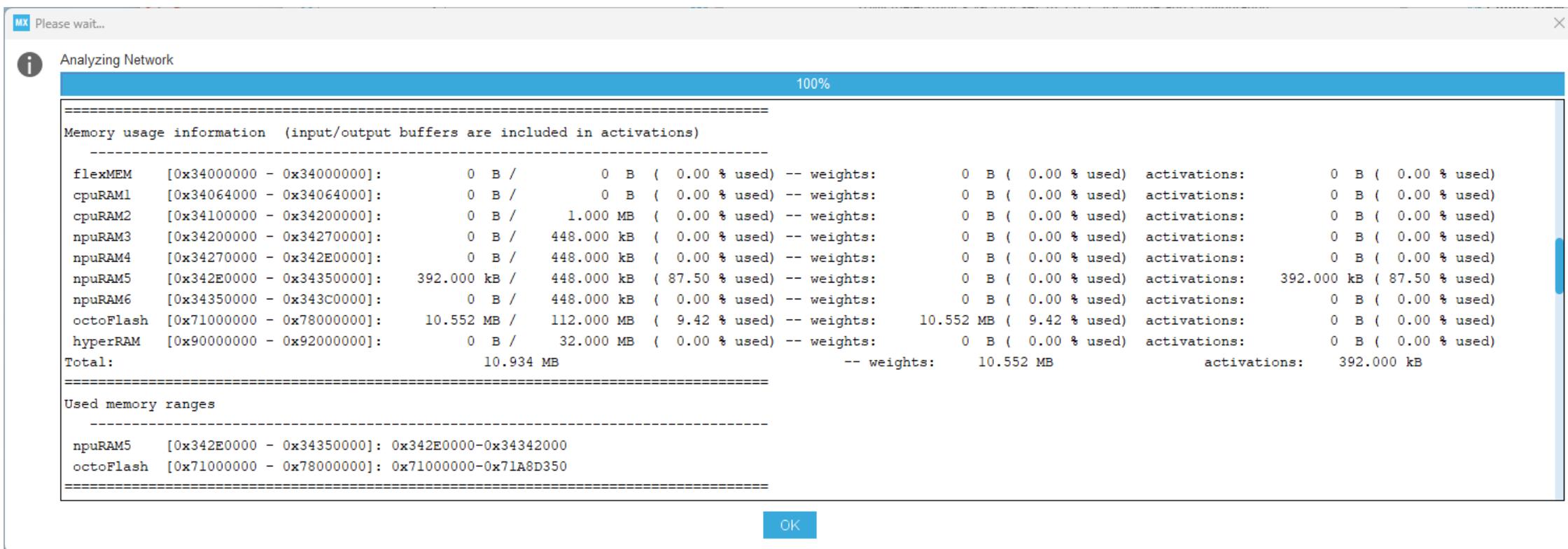
### すぐに始められるアプリケーション・パッケージ

- 準備済みのAIモデルから自動的にコード生成
- エンド・ツー・エンド評価を簡単に実現

# STM32Cube.AI デモンストレーション

## 解析 (Analyze) レポートの内容 (1)

- STM32N6の内蔵/外部メモリの各領域毎の使用量



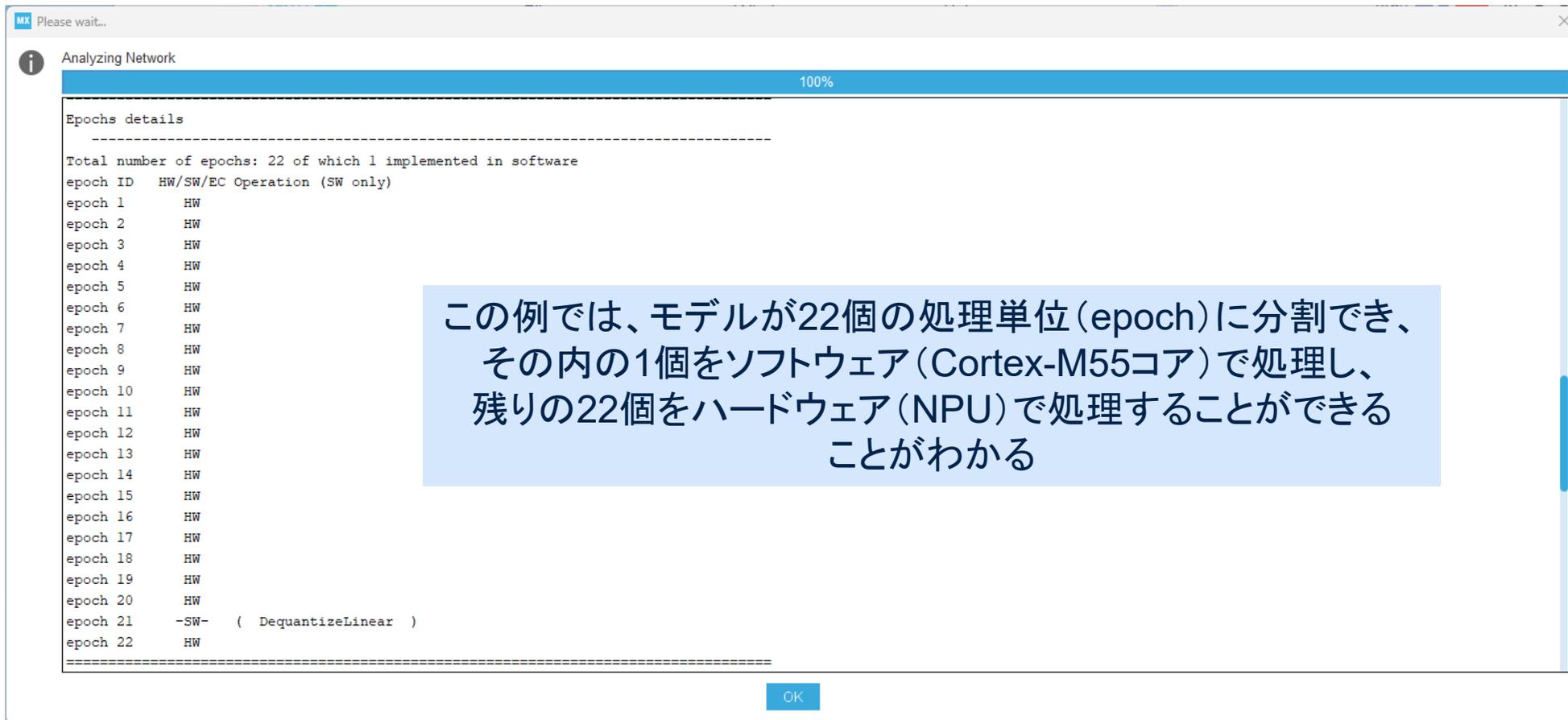
The screenshot shows a dialog box titled "Analyzing Network" with a progress bar at 100%. The main content is a table of memory usage information for various components. The table includes columns for component name, address range, used memory, and usage percentage for weights and activations. A summary row at the bottom shows the total memory usage. Below the table, there is a section for "Used memory ranges" listing specific memory addresses for npuRAM5 and octoFlash.

```
Memory usage information (input/output buffers are included in activations)
-----
flexMEM [0x34000000 - 0x34000000]: 0 B / 0 B ( 0.00 % used) -- weights: 0 B ( 0.00 % used) activations: 0 B ( 0.00 % used)
cpuRAM1 [0x34064000 - 0x34064000]: 0 B / 0 B ( 0.00 % used) -- weights: 0 B ( 0.00 % used) activations: 0 B ( 0.00 % used)
cpuRAM2 [0x34100000 - 0x34200000]: 0 B / 1.000 MB ( 0.00 % used) -- weights: 0 B ( 0.00 % used) activations: 0 B ( 0.00 % used)
npuRAM3 [0x34200000 - 0x34270000]: 0 B / 448.000 kB ( 0.00 % used) -- weights: 0 B ( 0.00 % used) activations: 0 B ( 0.00 % used)
npuRAM4 [0x34270000 - 0x342E0000]: 0 B / 448.000 kB ( 0.00 % used) -- weights: 0 B ( 0.00 % used) activations: 0 B ( 0.00 % used)
npuRAM5 [0x342E0000 - 0x34350000]: 392.000 kB / 448.000 kB ( 87.50 % used) -- weights: 0 B ( 0.00 % used) activations: 392.000 kB ( 87.50 % used)
npuRAM6 [0x34350000 - 0x343C0000]: 0 B / 448.000 kB ( 0.00 % used) -- weights: 0 B ( 0.00 % used) activations: 0 B ( 0.00 % used)
octoFlash [0x71000000 - 0x78000000]: 10.552 MB / 112.000 MB ( 9.42 % used) -- weights: 10.552 MB ( 9.42 % used) activations: 0 B ( 0.00 % used)
hyperRAM [0x90000000 - 0x92000000]: 0 B / 32.000 MB ( 0.00 % used) -- weights: 0 B ( 0.00 % used) activations: 0 B ( 0.00 % used)
Total: 10.934 MB -- weights: 10.552 MB activations: 392.000 kB
-----
Used memory ranges
-----
npuRAM5 [0x342E0000 - 0x34350000]: 0x342E0000-0x34342000
octoFlash [0x71000000 - 0x78000000]: 0x71000000-0x71A8D350
-----
```

# STM32Cube.AI デモンストレーション

## 解析 (Analyze) レポートの内容 (2)

- モデル内の処理単位 (=epoch) ごとのHW / SWの割り当て状況



The screenshot shows a dialog box titled "Analyzing Network" with a progress bar at 100%. Below the progress bar, the text "Epochs details" is followed by a dashed line. The main content is a table with the following data:

epoch ID	HW/SW/EC Operation (SW only)
epoch 1	HW
epoch 2	HW
epoch 3	HW
epoch 4	HW
epoch 5	HW
epoch 6	HW
epoch 7	HW
epoch 8	HW
epoch 9	HW
epoch 10	HW
epoch 11	HW
epoch 12	HW
epoch 13	HW
epoch 14	HW
epoch 15	HW
epoch 16	HW
epoch 17	HW
epoch 18	HW
epoch 19	HW
epoch 20	HW
epoch 21	-SW- ( DequantizeLinear )
epoch 22	HW

At the bottom of the dialog box, there is an "OK" button.

この例では、モデルが22個の処理単位 (epoch) に分割でき、その内の1個をソフトウェア (Cortex-M55コア) で処理し、残りの22個をハードウェア (NPU) で処理することができることがわかる

# STM32Cube.AI デモンストレーション

## 解析 (Analyze) レポートの内容 (3)

The screenshot shows a dialog box titled "Analyzing Network" with a progress bar at 100%. The main content area displays the following text:

```
-----  
TOTAL          11,122,663          401,437  
-----  
*  rt/total  
Creating txt report file C:\Users\...\.stm32cubemx\network_output\network_analyze_report.txt  
elapsed time (analyze): 129.355s  
Model file:      tiny yolo v2 224 int8.tflite  
Total Flash:    11122663 B (10.61 MiB)  
  Weights:      11064129 B (10.55 MiB)  
  Library:       58534 B (57.16 KiB)  
Total Ram:      401437 B (392.03 KiB)  
  Activations:  401408 B (392.00 KiB)  
  Library:       29 B  
  Input:         150528 B (147.00 KiB included in Activations)  
  Output:        5880 B (5.74 KiB included in Activations)  
Done  
Analyze complete on AI model
```

Annotations in the image:

- A blue box highlights the file path: `C:\Users\...\.stm32cubemx\network_output\network_analyze_report.txt`, with the text "生成された解析レポートの保存場所" (Storage location of the generated analysis report).
- A yellow box highlights the memory usage statistics (Total Flash, Weights, Library, Total Ram, Activations, Library, Input, Output), with the text "解析で得られたメモリ使用量の見積もり結果" (Estimated memory usage results obtained from analysis).

Buttons: "OK" and "Please wait..."

# STM32Cube.AI デモンストレーション

## Validate on desktop 実行レポートの内容(1)

- 入力データを乱数で生成

```
MX Please wait...  
Validation on desktop  
100%  
Starting AI validation on desktop with random data...  
Setting validation data...  
generating random data, size=10, seed=42, range=(0, 1)  
I[1]: (10, 224, 224, 3)/float32, min/max=[-0.996077, 1.003918], mean/std=[0.003833, 0.577072]  
c/I[1] conversion [Q(0.00784314,127)]-> (10, 224, 224, 3)/uint8, min/max=[0, 255], mean/std=[127.488716, 73.577912]  
m/I[1] conversion [Q(0.00784314,127)]-> (10, 224, 224, 3)/uint8, min/max=[0, 255], mean/std=[127.488716, 73.577912]  
no output/reference samples are provided  
Exec/report summary (validate)  
OK
```

# STM32Cube.AI デモンストレーション

## Validate on desktop 実行レポートの内容(2)

- モデルの概要と実行環境の情報

The screenshot shows a dialog box titled "Validation on desktop" with a progress bar at 100%. The content is divided into two sections:

```
Summary 'network' - ['network']
-----
I[1/1] 'input_1' : uint8[1,224,224,3], 150528 Bytes, QLinear(0.007843138,127,uint8), in activations buffer
O[1/1] 'output_1' : f32[1,7,7,30], 5880 Bytes, in activations buffer
n_nodes          : 22
activations      : 224236
weights          : 11034600
macc             : 777482634
hash             : 0xe8b6a5b60582ba023d3dc8f10633aa6e
compile datetime : May 28 2025 14:26:46
-----
protocol         : DLL Driver v2.0 - Direct Python binding
tools            : ST.AI (legacy api) v2.1.0
runtime lib     : v10.1.0
capabilities    : IO_ONLY, PER_LAYER, PER_LAYER_WITH_DATA
device.desc     : AMD64, Intel64 Family 6 Model 142 Stepping 12, GenuineIntel, Windows
-----
```

Annotations in the image:

- A yellow box highlights the model summary section, with a callout: "入出力の書式やサイズ、ノード数、演算量など" (Input/output format or size, number of nodes, number of operations, etc.).
- A blue box highlights the execution environment section, with a callout: "実行環境" (Execution environment).

At the bottom of the dialog is an "OK" button.

# STM32Cube.AI デモンストレーション

## Validate on desktop 実行レポートの内容(3)

- 実行結果

Validation on desktop

100%

STAI.IO:	Progress	Count	Time	Speed
0%		0/10	[00:00<?, ?it/s]	
40%	####	4/10	[00:00<00:00, 8.67it/s]	8.67it/s
50%	#####	5/10	[00:00<00:00, 5.02it/s]	5.02it/s
60%	#####	6/10	[00:01<00:01, 3.85it/s]	3.85it/s
70%	#####	7/10	[00:01<00:00, 3.41it/s]	3.41it/s
80%	#####	8/10	[00:02<00:00, 3.10it/s]	3.10it/s
90%	#####	9/10	[00:02<00:00, 2.78it/s]	2.78it/s]

NOTE: The duration and execution time per layer are just indications. They depend on the host machine's workload.  
ST.AI Profiling results v2.0 - "network"

---

nb sample(s)	: 10
duration	: 417.094 ms by sample (384.017/487.846/34.673)
macc	: 777482634

Statistic per tensor

---

tensor #	type[shape: size]	min	max	mean	std	name
----------	-------------------	-----	-----	------	-----	------

OK

異なる入力データを使用して10回実行

実行回数(10回)と実行時間、演算量

# STM32Cube.AI デモンストレーション

## Validate on desktop 実行レポートの内容(4)

- 比較結果

```
MX Please wait...
Validation on desktop
100%
Computing the metrics...
Cross accuracy report #1 (reference vs C-model)
-----
notes: - ACC metric is not computed ("--classifier" option can be used to force it)
       - the output of the reference model is used as ground truth/reference value
       - 10 samples (1470 items per sample)
acc=n.a. rmse=0.153796345 mae=0.055611547 l2r=0.058213457 mean=-0.001270 std=0.153796 nse=0.995950 cos=0.998306
Evaluation report (summary)
-----
Output      acc      rmse      mae      l2r      mean      std      nse      cos      tensor
-----
X-cross #1  n.a.    0.153796345  0.055611547  0.058213457  -0.001270  0.153796  0.995950  0.998306  'conversion_24', 10 x f32(1x7x7x30), m_id=[24]
-----
acc : Accuracy (class, axis=-1)
rmse : Root Mean Squared Error
mae  : Mean Absolute Error
l2r  : L2 relative error
mean : Mean error
std  : Standard deviation error
nse  : Nash-Sutcliffe efficiency criteria, bigger is better, best=1, range=(-inf, 1]
cos  : COsine Similarity, bigger is better, best=1, range=(0, 1]
Creating txt report file C:\Users\nobuo suzuki\.stm32cubemx\network_output\network_validate_report.txt
elapsed time (validate): 110.667s
Validation ended
OK
```

# STM32Cube.AI デモンストレーション

## Validate on target 実行レポートの内容(1)

- モデルの概要とマイコンでの実行条件の情報

MX Please wait...

Automatic Validation on the target

100%

Summary 'network' - ['network']

I[1/1] 'Input\_0\_out\_0' : uint8[1,224,224,3], 150528 Bytes, QLinear(0.007843138,127,uint8), activations  
O[1/1] 'Transpose\_55\_out\_0' : f32[1,7,7,30], 5880 Bytes, activations  
n\_nodes : 20  
activations : 401408  
compile\_datetime : May 26 2025 21:32:36

protocol : Proto-buffer driver v2.0 (msg v3.1) (Serial driver v1.0 - COM5:921600)  
tools : ST Neural ART (LL\_ATON api) v1.1.0  
runtime lib : atonn-v1.1.0-31-g27f5d5bc (optimized SW lib v10.1.0-fbaabb46 G  
capabilities : IO\_ONLY, PER\_LAYER, PER\_LAYER\_WITH\_DATA  
device.desc : stm32 family - 0x486 - STM32N6xx @800/400MHz  
device.attrs : fpu,core\_icache,core\_dcache,npu\_cache=1,mcu\_freq=800MHz,noc\_freq=400MHz,npu\_freq=1000MHz,nic\_freq=900MHz

Warning: C-network signature checking is skipped on ST Neurla-ART NPU  
STAI.IO: 0% | 0/10 [00:00<?, ?it/s]

OK

入出力の書式やサイズ、ノード数、演算量など

実行条件  
コア/NPUの動作周波数など

# STM32Cube.AI デモンストレーション

## Validate on target 実行レポートの内容(2)

- 処理単位 (epoch) ごとの処理時間などの情報

Automatic Validation on the target

100%

Inference time per node

c_id	m_id	type	dur (ms)	%	cumul	CPU cycles	name
0	-	epoch	0.157	0.5%	0.5%	[ 2,924 121,182 1,357 ]	EpochBlock_2
1	-	epoch	0.825	2.6%	3.1%	[ 12,637 644,959 2,639 ]	EpochBlock_3
2	-	epoch	0.104	0.3%	3.5%	[ 1,892 80,910 762 ]	EpochBlock_4
3	-	epoch	0.844	2.7%	6.2%	[ 15,797 656,262 3,242 ]	EpochBlock_5
4	-	epoch	0.054	0.2%	6.4%	[ 1,925 40,822 723 ]	EpochBlock_6
5	-	epoch	0.458	1.5%	7.8%	[ 25,019 336,726 4,546 ]	EpochBlock_7
6	-	epoch	0.029	0.1%	7.9%	[ 1,817 20,862 723 ]	EpochBlock_8
7	-	epoch	0.493	1.6%	9.5%	[ 38,934 349,928 5,154 ]	EpochBlock_9
8	-	epoch	0.017	0.1%	9.6%	[ 1,825 11,020 723 ]	EpochBlock_10
9	-	epoch	0.826	2.6%	12.2%	[ 63,698 591,778 5,141 ]	EpochBlock_11
10	-	epoch	0.011	0.0%	12.2%	[ 1,851 6,417 762 ]	EpochBlock_12
11	-	epoch	3.089	9.9%	22.2%	[ 108,102 2,360,520 2,887 ]	EpochBlock_13
12	-	epoch	0.049	0.2%	22.3%	[ 1,935 36,392 723 ]	EpochBlock_14
13	-	epoch	0.055	0.2%	22.5%	[ 3,878 38,650 1,690 ]	EpochBlock_15
14	-	epoch	0.019	0.1%	22.5%	[ 1,968 12,234 727 ]	EpochBlock_16
15	-	epoch	12.094	38.8%	61.3%	[ 10,789 9,660,697 3,525 ]	EpochBlock_17
16	-	epoch	0.033	0.1%	61.4%	[ 1,892 23,914 760 ]	EpochBlock_18
17	-	epoch	11.937	38.3%	99.7%	[ 108,496 9,438,367 2,943 ]	EpochBlock_19
18	-	epoch	0.068	0.2%	99.9%	[ 17,440 32,077 4,581 ]	EpochBlock_20
19	-	epoch (SW)	0.017	0.1%	100.0%	[ 45 66 13,112 ]	EpochBlock_21
n/a	n/a	Inter-nodal	0.000	0.0%	100.0%		n/a
total			31.179			[ 422,864 24,463,783 56,720 ]	
			32.07 inf/s			[ 1.7% 98.1% 0.2% ]	

OK

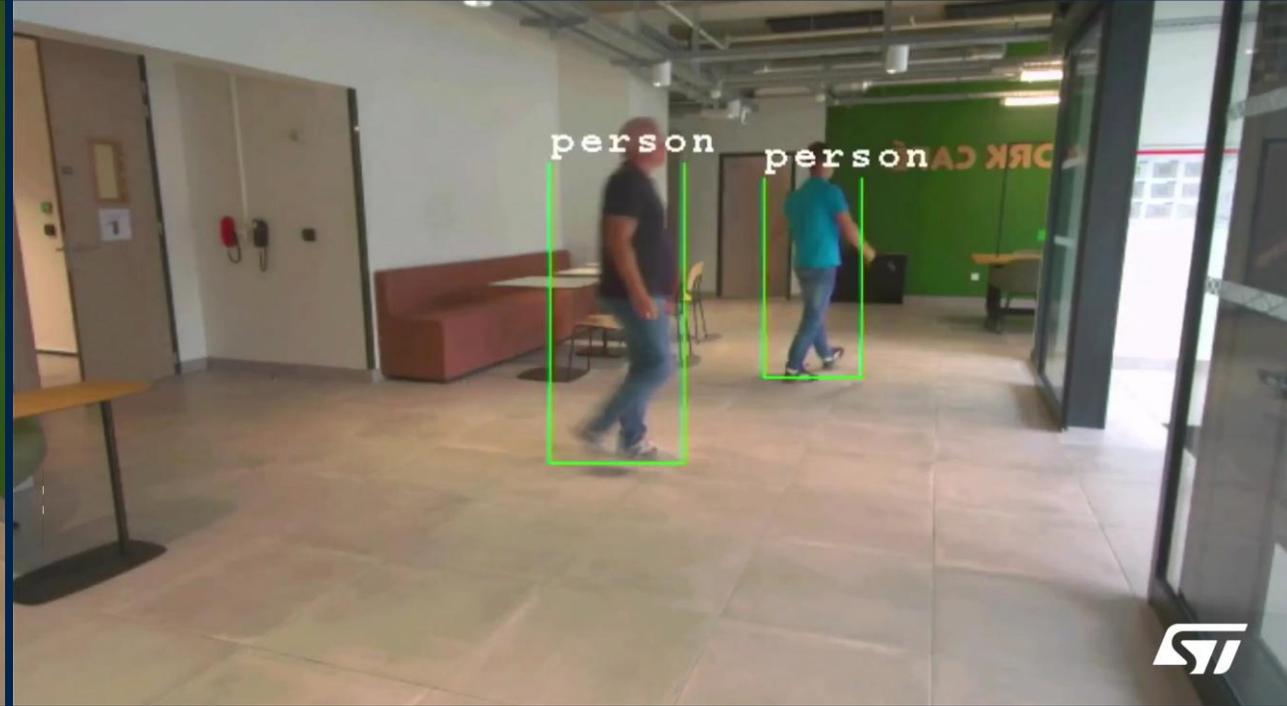
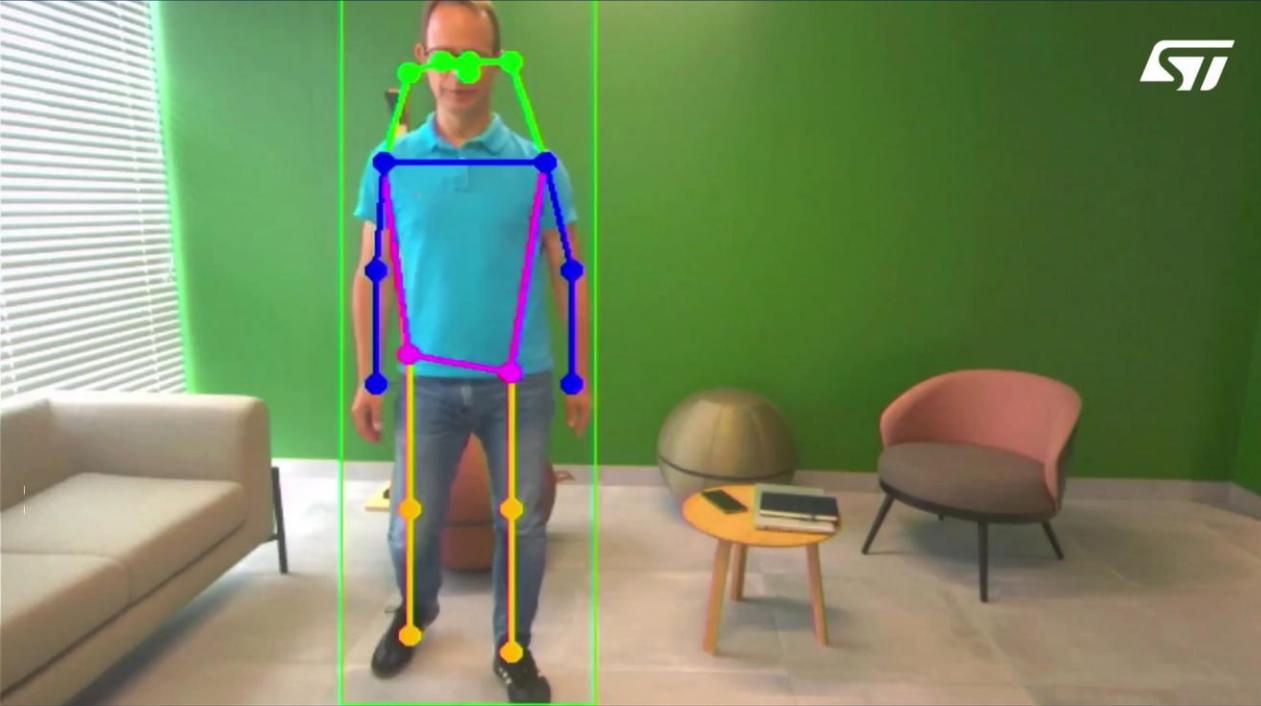
# STM32Cube.AI デモンストレーション

## Validate on target 実行レポートの内容(3)

- 比較結果

```
MX Please wait...
Automatic Validation on the target
100%
Computing the metrics...
Cross accuracy report #1 (reference vs C-model)
-----
notes: - ACC metric is not computed ("--classifier" option can be used to force it)
       - the output of the reference model is used as ground truth/reference value
       - 10 samples (1470 items per sample)
acc=n.a. rmse=0.197166875 mae=0.089913994 l2r=0.074466638 mean=-0.004562 std=0.197121 nse=0.993344 cos=0.997223
Evaluation report (summary)
-----
Output      acc      rmse      mae      l2r      mean      std      nse      cos      tensor
-----
X-cross #1  n.a.    0.197166875  0.089913994  0.074466638  -0.004562  0.197121  0.993344  0.997223  'conversion_24', 10 x f32(1x7x7x30), m_id=[24]
-----
acc : Accuracy (class, axis=-1)
rmse : Root Mean Squared Error
mae : Mean Absolute Error
l2r : L2 relative error
mean : Mean error
std : Standard deviation error
nse : Nash-Sutcliffe efficiency criteria, bigger is better, best=1, range=(-inf, 1]
cos : COsine Similarity, bigger is better, best=1, range=(0, 1]
Creating txt report file C:\Users\nobuo suzuki\.stm32cubemx\network_output\network_validate_report.txt
elapsed time (validate): 138.441s
Validation ended
OK
```

コサイン類似度 = 0.997223

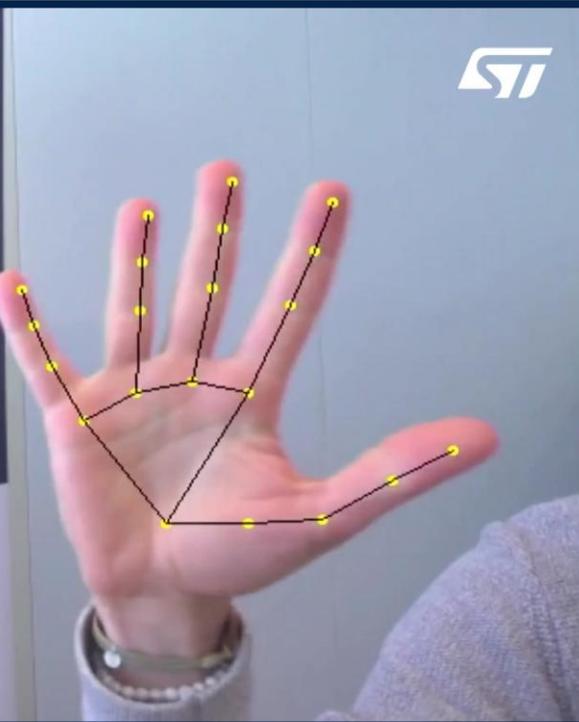


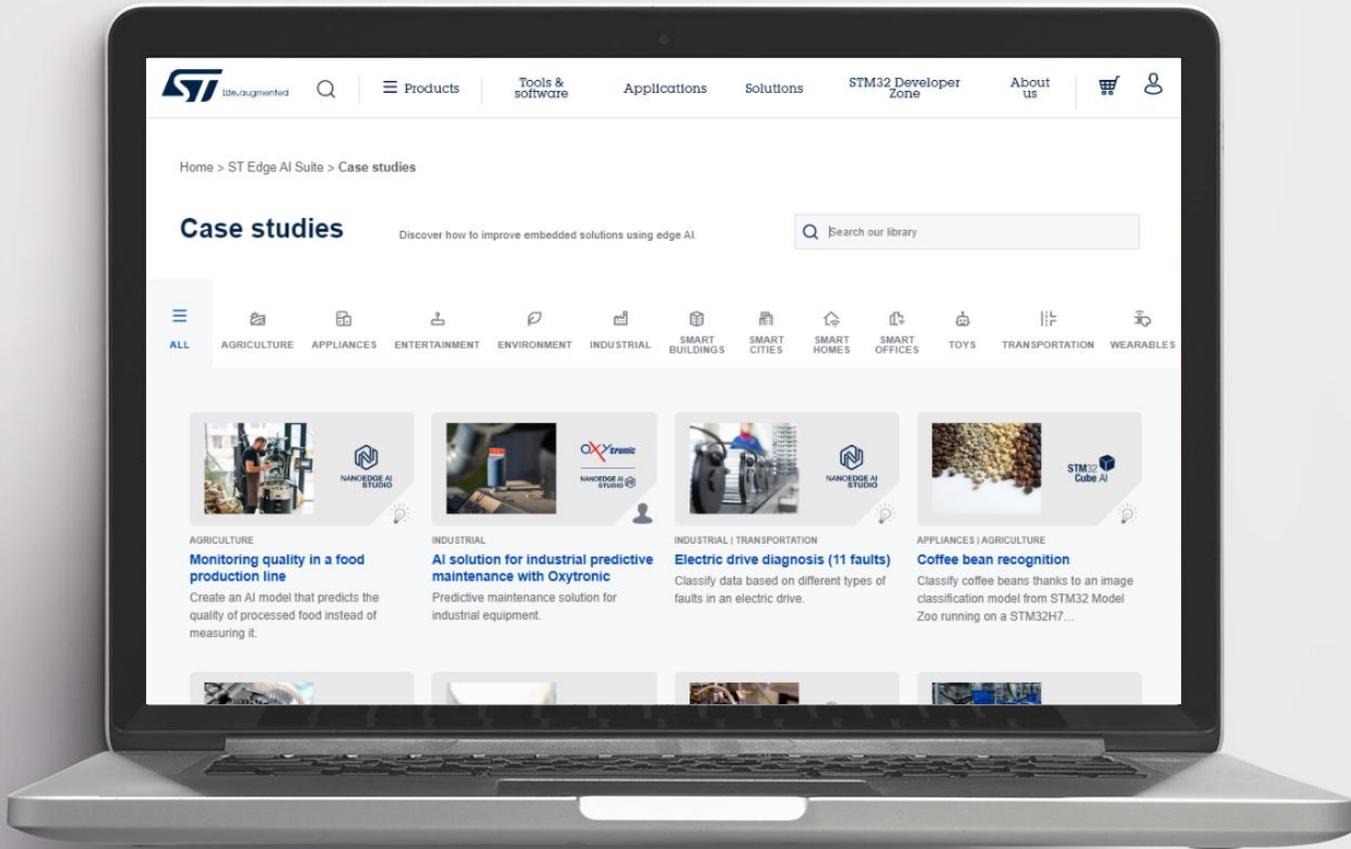
ST

Advance your product offering with edge AI

ST Edge AI Suite

Start your journey now





実際の採用事例を含めた  
今すぐ使えるエッジAIによる  
組み込みシステムの  
改善アイデアを  
多数掲載



[Edge AI case studies](#)

# 参考リンク

- STM32N6マイコン製品ページ: <https://www.stmcu.jp/stm32/stm32n6/>
- STM32H5マイコン製品ページ: <https://www.stmcu.jp/stm32/stm32h5/>
- 統合開発環境STM32CubeIDE: <https://www.st.com/ja/development-tools/stm32cubeide.html>
- 初期化コード自動生成ツールSTM32CubeMX: [https://www.stmcu.jp/design/sw\\_dev/pc\\_soft/52798/](https://www.stmcu.jp/design/sw_dev/pc_soft/52798/)
- NNモデル最適化ツールSTM32Cube.AI: [https://www.st.com/content/st\\_com/ja/st-edge-ai-suite/tools.html#tools=stm32cube-ai](https://www.st.com/content/st_com/ja/st-edge-ai-suite/tools.html#tools=stm32cube-ai)
- STM32N6 AIサンプルコード: <https://www.st.com/ja/development-tools/stm32n6-ai.html>
- オンラインAIツールST Edge AI Developer Cloud: [https://www.st.com/content/st\\_com/ja/st-edge-ai-suite/tools.html#tools=st-edge-ai-developer-cloud](https://www.st.com/content/st_com/ja/st-edge-ai-suite/tools.html#tools=st-edge-ai-developer-cloud)
- ST Model Zoo: <https://github.com/STMicroelectronics/stm32ai-modelzoo/>
- **STM32Cube.AI Wiki**: <https://wiki.st.com/stm32mcu/wiki/Category:STM32Cube.AI>
- 機械学習コード自動生成ツールNanoEdge AI Studio: [https://www.st.com/content/st\\_com/ja/st-edge-ai-suite/tools.html#tools=nanoedge-ai-studio](https://www.st.com/content/st_com/ja/st-edge-ai-suite/tools.html#tools=nanoedge-ai-studio)
- **NanoEdge AI Studio Wiki**: <https://wiki.st.com/stm32mcu/wiki/Category:NanoEdgeAI>
- ST Edge AI Suite: [https://www.st.com/content/st\\_com/ja/st-edge-ai-suite.html](https://www.st.com/content/st_com/ja/st-edge-ai-suite.html)
- 開発ボードSTM32N6570-DK: <https://www.stmcu.jp/design/hwdevelop/discovery/112317/>
- 開発ボードNUCLEO-H533RE: <https://www.stmcu.jp/design/hwdevelop/nucleo/106898/>

# Our technology starts with You



Find out more at [www.st.com/stm32](http://www.st.com/stm32)

© STMicroelectronics - All rights reserved.

ST logo is a trademark or a registered trademark of STMicroelectronics International NV or its affiliates in the EU and/or other countries.

For additional information about ST trademarks, please refer to [www.st.com/trademarks](http://www.st.com/trademarks).

All other product or service names are the property of their respective owners.

