



Contest problems

1. Introduction

Human activity recognition is a well-known applied machine learning task across many fields. This technology leverages data from multiple sensors to support applications such as health monitoring, detecting unusual behaviors for security purposes, tracking exercise routines, and more.

However, implementing tiny, low-power human activity recognition comes with several challenges. To deliver the required accuracy on low-power devices like wearable sensors, embedded developers shall devise tiny machine learning models to run efficiently on sensors with limited memory and computational capabilities. This requires designing tiny neural networks that balance footprint and accuracy, tailored to the specific task and the constraints of memory-limited sensors.

2. Objective

The participants, being them single applicants or a team, are expected to design and implement an edge AI model for human activity recognition optimized for deployment on an Intelligent Sensor Processing Unit (ISPU) development kit and applicable to STMicroelectronics MEMS sensors and deployed using ST Edge AI tools. Higher scores are awarded to entries with the most complex to solve and creative use case and balancing the highest accuracy with the least amount of memory usage and the lowest execution time. We encourage the participants to provide real-time demos at the IEEE COINS 2026 conference.

We invite participants to create an TinyML multiclass model with any of the frameworks supported by ST Edge AI (Keras, QKeras, TensorFlow Lite, any framework that can export to ONNX format, such as PyTorch and MATLAB) to detect different types of daily human activities in real time using accelerometer and gyroscope data and the ISPU for processing. The model conversion to C code shall be performed using ST Edge AI tools.

The following examples illustrate possible daily activities, but the participants are encouraged to think creatively in building a use case specific dataset. The essential point is that the classes correspond to human activity recognition tasks. For example:

- Brushing hair
- Brushing teeth
- Washing hands
- Washing face
- Cleaning a countertop
- Keeping hands still
- Using a razor
- Dicing vegetables
- Using an iron

Other examples:

- Individual & Solo Sports
 - Running
 - Jogging.
 - Swimming.
 - Cycling.
 - Walking.
 - Weightlifting.
- Group & Team Sports
 - Football (Soccer).
 - Tennis.
 - Basketball.
 - Badminton.
 - Volleyball
- Mind-Body & Flexibility
 - Yoga.
 - Tai Chi.
 - Pilates.
 - Stretching
- Home-Based & Everyday Activities
 - Dancing.
 - Bodyweight Exercises:
 - Push-ups,
 - squats,
 - planks.
 - Climbing Stairs.
 - Gardening
 - Housework:
 - Vigorous cleaning
 - washing the car.

3. MCU and ISPU platform

Participants should use the following development kits:

- NUCLEO-F401RE (<https://www.st.com/en/evaluation-tools/nucleo-f401re.html>) or NUCLEO-U575ZI-Q (<https://www.st.com/en/evaluation-tools/nucleo-u575zi-q.html>) STM32 Nucleo board.
- X-NUCLEO-IKS4A1 (<https://www.st.com/en/evaluation-tools/x-nucleo-iks4a1.html>) or X-NUCLEO-IKS5A1 (<https://www.st.com/en/evaluation-tools/x-nucleo-iks5a1.html>) STM32 Nucleo expansion board for MEMS sensors, which embed, respectively the LSM6DSO16IS and ISM330IS sensors.

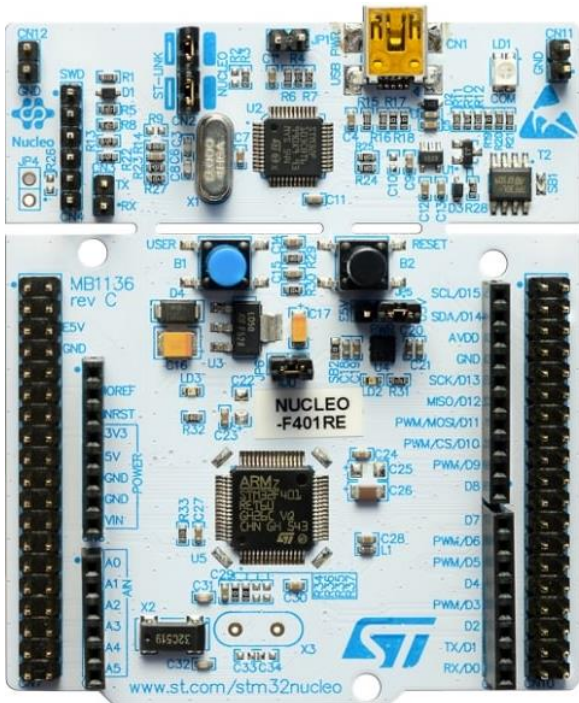


Figure 1. NUCLEO-F401RE

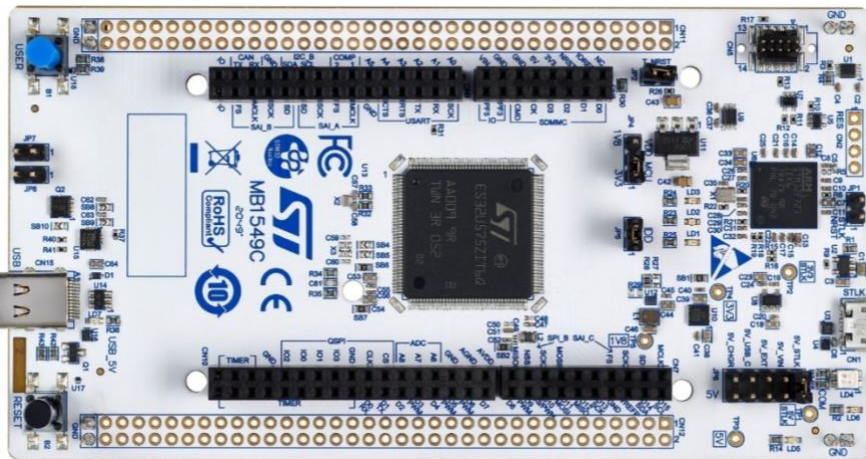


Figure 2. NUCLEO-U575ZI-Q



Figure 3. X-NUCLEO-IKS4A1

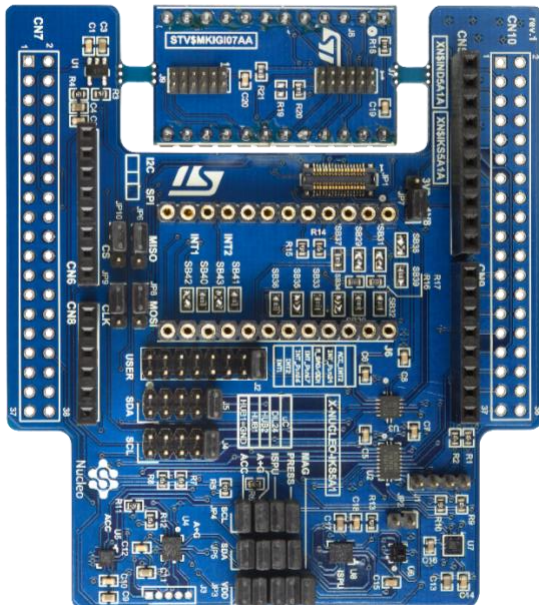


Figure 4. X-NUCLEO-IKS5A1

4. Data

Participants shall independently collect sensor data to train the model. MEMS Studio (<https://www.st.com/en/development-tools/mems-studio.html>) and the X-CUBE-MEMS1 (<https://www.st.com/en/embedded-software/x-cube-mems1.html>) expansion package can be used for this purpose. We suggest collecting both accelerometer and gyroscope measurements and then evaluating during the model design phase which data is significant to distinguish the activities to detect. We recommend collecting at least 10-15 minutes of raw sensor data for each activity. After collecting the data, participants must use it for training the model with the preferred AI framework.

Applicants shall make available the datasets for reproducibility tests and under CC BY-NC-SA license and specify the authorship.

5. Sensor setup

The sampling rate of the data must be selected depending on the target activities, selecting the lowest rate possible while preserving the relevant frequency components of each activity to be detected.

It is recommended to set the full scale of the accelerometer and gyroscope sensors to values that prevent the signals from saturating, thus preserving the dynamic range of each activity to be detected.

There is no constraint in terms of device placement. However, it is required to provide information about the placement and orientation of the sensor.

6. ST Edge AI

Once the model has been trained, it must be converted to C code using one of the following tools:

- ST Edge AI Core (<https://www.st.com/en/development-tools/stedgeai-core.html>) for command line usage.
- MEMS Studio (<https://www.st.com/en/development-tools/mems-studio.html>) for graphical interface usage with its ISPU Model Converter tool. ST Edge AI Core must also be installed and linked in MEMS Studio's settings.

ST Edge AI Core and MEMS Studio's ISPU Model Converter also allow to analyze and validate the model:

- The "analyze" command allows verifying if the model architecture is supported and estimating the memory footprint of the converted model.
- The "validate" command allows checking the output of the converted model vs the original model and measuring the inference time. The "validate" must be executed in "target" mode with the Nucleo board + expansion board attached to the PC and flashed with the dedicated firmware (https://github.com/STMicroelectronics/st-mems-ispu/tree/master/host_firmware/nucleo_ispu_stedgeai_validate).

The converted code can be generated using the "generate" command and can then be integrated into an ISPU project starting from the template available on GitHub (https://github.com/STMicroelectronics/st-mems-ispu/tree/master/examples/ism330is_ism6dso16is/template_stedgeai). As indicated on the README file of the template, the sensor data preprocessing and buffering to prepare to input for the network, and the writing of predicted activity in the ISPU output registers must be implemented by the participants.

For further information on how to use ST Edge AI tools, refer to the ST Edge AI Core documentation (available in the installation folder in HTML format), MEMS Studio's documentation (<https://www.st.com/en/development-tools/mems-studio.html#documentation>), and to the README files and tutorials on the ISPU GitHub repository (<https://github.com/STMicroelectronics/st-mems-ispu>).

To test the final ISPU program integrating the model, there are two options:

- Use MEMS Studio with the Nucleo board flashed with the DataLogExtended firmware (available in X-CUBE-MEMS1) and load the ISPU program from MEMS Studio using JSON file generated when building the project.

- Use the test firmware (https://github.com/STMicroelectronics/st-mems-ispu/tree/master/host_firmware/nucleo_ispu_test_header) and connect to the UART interface of the Nucleo board to visualize the results. Refer to the firmware README file for details on how to use it.

7. Deliverables

The following material must be delivered:

- Use case description, including the list of activities recognized.
- Sensor configuration:
 - Sensors used (accelerometer, gyroscope, both).
 - Output data rate.
- Trained model in one of the formats supported by ST Edge AI.
- Test set (".npz" or ".csv" format) to reproduce the accuracy evaluation with ST Edge AI containing both the inputs of the model and the corresponding activity to be detected. The test set must be tested with ST Edge AI Core or MEMS Studio's ISPU model converter (using the "validate" command in "target" mode) and must be confirmed working.
- ISPU project integrating the model for runtime execution.

8. Scoring

Our scoring metric will assess the delivered solution, according to the following criteria:

- **Test set accuracy score:** S_A (defined as value between 0 and 1) is computed using ST Edge AI tools on the provided test dataset.
- **ISPU execution time score:** The average execution time of the final ISPU program integrating the model and executing on LSM6DSO16IS/ISM330IS. The execution time score is computed as follows:

$$S_T = \left(1 - \min \left(1, \frac{T_{AVG}}{T_{ODR}} \right) \right)$$

where $T_{ODR} = \frac{1}{ODR}$.

- **ISPU data RAM footprint score:** The data memory footprint of the final ISPU program integrating the model will be measured using the ISPU toolchain. The data RAM footprint score is computed as follows:

$$S_D = \left(1 - \min \left(1, \frac{DATA + BSS}{RAM_{DATA}} \right) \right)$$

where $RAM_{DATA} = 8 \text{ KB}$.

- **ISPU program RAM footprint score:** The program memory footprint of the final ISPU program integrating the model will be measured using the ISPU toolchain. The program RAM footprint score is computed as follows:

$$S_P = \left(1 - \min \left(1, \frac{TEXT}{RAM_{PROGRAM}} \right) \right)$$

where $RAM_{PROGRAM} = 32 \text{ KB}$.

- **Jury score:** The jury score S_j (defined as value between 0 and 1) is a score assigned by the jury based on their evaluation of the complexity of the use case, the creativity of the solution.

The final score is calculated as follows:

$$S = 2.5 \times S_A + 1 \times S_T + 1.5 \times S_D + 2 \times S_P + 3 \times S_j$$

Terms and conditions:

Conference date: 7-9th September

Notification date: during the conference (will be announced later)

Start date: 15th April 2026

Submission date: 15th June 2026

Registration:

Interested parties must email the ST contacts listed below.

Contact persons:

ST: lorenzo.bracco@st.com

ST: giacomo.turati@st.com

COINS: farshad.firouzi@duke.edu

Awards:

One registration fee waiver will be granted to the winning team. In case the team will consist of more than one applicant, the participants can select only one representative who will make use of the voucher.

