



Tiny ML Contest

1. Introduction

The Tiny ML Contest at IEEE COINS is a challenging, multi-month, research and development competition, focusing on advanced, real-world problems in the fields of AI, Embedded Systems, and IoT. It is open to multi-person teams worldwide. The focus of this year's contest is on Human activity recognition. Human activity recognition is a common machine learning task that is widely used in multiple domains. This technology uses data from various sensors to help monitor health conditions, recognize abnormal human activities for surveillance, track fitness workouts, and much more.

There are, of course, various challenges associated with implementing complex human activity recognition. To achieve the required accuracy on low-power devices like wearables, embedded engineers have to find ways to run machine learning models on platforms with limited memory and processing capacity. Experts, therefore, need to design neural networks with a size-accuracy ratio that is appropriate for a proposed machine learning task on a memory-constrained device.

2. Objective

We ask participants to design and implement a Tiny ML solution for human activity recognition using the Neuton.ai AutoML platform on an STMicroelectronics ISPU development kit. Higher scores are awarded to entries that support the highest number of classes and the best confusion matrix that the ISPU memory can easily accommodate with the lowest inference latency. We encourage real-time demos at the conference and will recognize creativity in the class list definition during evaluation.

We invite participants to create a Tiny ML multiclass model with the Neuton platform to detect different types of daily human activities in real-time using ISPU accelerometer and gyroscope data.

The following activities represent examples, but these should not limit your imagination. The most important thing is to ensure is that the classes fit human activity recognition concepts:

- Brushing hair
- Brushing teeth
- Washing hands
- Washing face
- Wiping the table
- Hand still

- Shaving face
- Chopping
- Ironing

3. MCU and ISPU platform

Participants should use the following development kits:

- NUCLEO-F401 STM32 Nucleo board.
- X-NUCLEO-IKS01A3 STM32 Nucleo expansion board for MEMS sensors.
- DIL24 adapter board with appropriate IMU (with embedded ISPU) based on sample availability on the ST eStore and through distributors:
 - STEVAL-MKI229A DIL24 with LSM6DSO16IS or
 - STEVAL-MKI230KA DIL24 with ISM330IS

NUCLEO-F401



X-NUCLEO-IKS01A3



STEVAL-MKI230KA

or

STEVAL-MKI229A



4. Data

Participants must independently collect sensor data to train the model. You can use Unicleo-GUI (<https://www.st.com/en/development-tools/unicleo-gui.html>) and the X-CUBE-ISPU (<https://www.st.com/en/embedded-software/x-cube-ispu.html>) expansion package for this purpose. The data should represent a 6-axis accelerometer and gyroscope measurements collected in **float** or **int16** format. We recommend collecting at least 10-15 minutes of raw sensor data for each activity. After collecting data, participants must prepare a dataset for training the model on the Neuton AutoML platform (<https://neuton.ai/>).

Applicants must render datasets available for reproducibility tests and under CC BY-NC-SA license and specify the authorship. The aggregated dataset may eventually be published through a Data In Brief Elsevier publication.

All activities should be merged in one .csv dataset with an appropriate target (label) column added to each row of the dataset. Here is an example of how a dataset header might look:

ax	ay	az	gx	gy	gz	target
----	----	----	----	----	----	--------

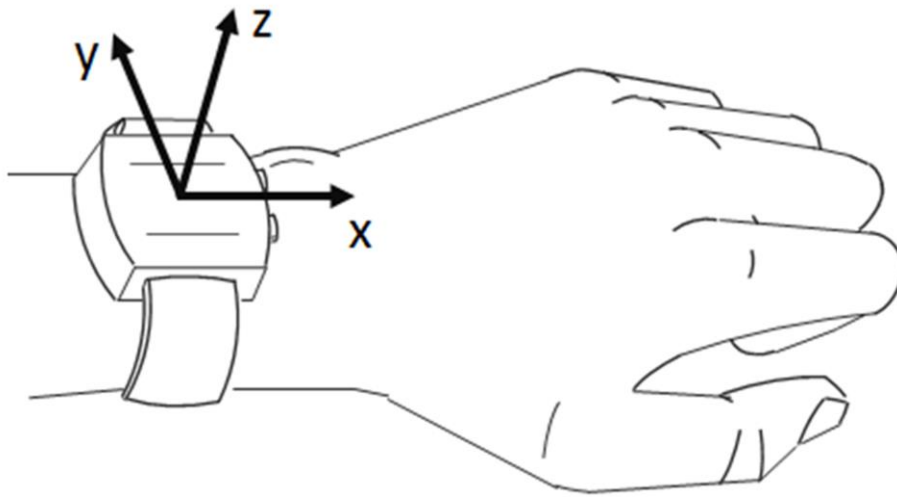
Where:

- **ax** accelerometer X axis;
- **ay** accelerometer Y axis;
- **az** accelerometer Z axis;
- **gx** gyroscope X axis;
- **gy** gyroscope Y axis;
- **gz** gyroscope Z axis;
- **target** activity label in numeric format.

5. Sensor setup

We can offer advice to less experienced participants regarding data collection. The sampling rate of the data acquisition should allow qualitative determination of the difference in activity patterns; we recommend starting with 52 Hz. Sensor sensitivity is very important because a wider dynamic range incurs noisier sensor data, but you also need to avoid sensor data saturation. We recommend starting your experiments with medium sensitivity values for the gyroscope and accelerometer; for example, 8 g for the accelerometer and 1000 dps for the gyroscope.

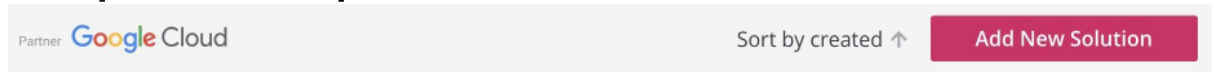
To collect data, the ISPU sensor **must be** attached to the participant's wrist.



6. Neuton Tiny ML platform

Participants that are not familiar with the Neuton platform can follow these steps to create a Tiny ML model. To build a Tiny ML solution, participants should register for a free Neuton account. The registration manual can be downloaded [here](#).

1. Log in to lab.neuton.ai with your registered google account.
2. Select [Add New Solution] in the header



3. Upload your dataset, this may take some time, depending on the size of the dataset and connection speeds.

New Solution

Data Type

Audio data

Sensor data

Solution Name

hand_activities

25 Character Limit 15 / 25

Description

140 Character Limit 0 / 140

Cancel

Next →

Training Dataset

Upload dataset

Select dataset from storage

Preloaded dataset

CSV UTF-8

Drag and drop or click to upload CSV file

OK

Dataset Requirements

4. Once the data is uploaded, select the target variable. The generated 'train.csv' will contain the 'target' column you selected. Click [Next].

Training Dataset

data_short_75712.csv
/data_short_75712.csv

Change

Holdout Validation 

Features Specification

Remove variables (if necessary) 

Search

☐ ax

☐ ay

☐ az

Select target variable

Search

☐ gy

☐ gz

☒ target



Please make sure that your data is suitable for machine learning. [Dataset Requirements](#)

- Excludes text data, missing values, categorical data represented as strings
- Date/time columns are in epoch time representation or relative date format

Back

Next →

5. In the [Training] tab, select [Task type]: Multiclass Classification, as we need to define several different activities. [Input data type]: INT16 or FLOAT, and enable [Windowing and Feature Extraction].

Training dataset: data_short_75712.csv


Target variable: target


Task type:

Multi Classification


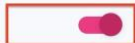
Metric:

Accuracy

Input data type:  INT16

Normalization type:  AutoSelect

Digital Signal Processing

Windowing and Feature Extraction  

To qualitatively determine activities, we must make predictions on data windows. Depending on your sampling rate, select a window in which activity should be identified; for example 1-3 seconds of collected sensor data.

You can choose which features you want to extract from your data window or can [Select All] and let the platform choose features for you. Click [Edit] for each single axis (ax, ay, az, gx, gy, gz)

Windowing and Feature Extraction ?

☒ In Number of Rows/Samples

☐ In Duration

☐ Auto Determination ?

72

Estimated SRAM

1.1
kB

ax

Mean, Max, Min, Root mean square, Skewness, Kurtosis, Global peak to peak of hi ...

Edit

Remove

ay

Mean, Max, Min, Root mean square, Skewness, Kurtosis, Global peak to peak of hi ...

az

Mean, Max, Min, Root mean square, Skewness, Kurtosis, Global peak to peak of hi ...

gx

☐ Lag features ?

The setting is common for all variables.

In this example, we chose the following features:

Mean; Max; Min; Root mean square; Mean Absolute Deviation; Standard Deviation Mean-crossing rate; Zero-crossing rate.

ax

Select features

Select all Remove all

— Statistical (4)

☒ Mean

☒ Max

☒ Min

☒ Root mean square

☐ Skewness

☐ Kurtosis


☐ Range

Cancel

OK

6. Finally, you can enable or disable [Feature Selection] and set the [Bit depth] parameter of the model.


Feature Selection



Model Settings

Model calculation parameters: ?


Float Datatype Support



Bit depth

8

Maximum Number of Coefficients

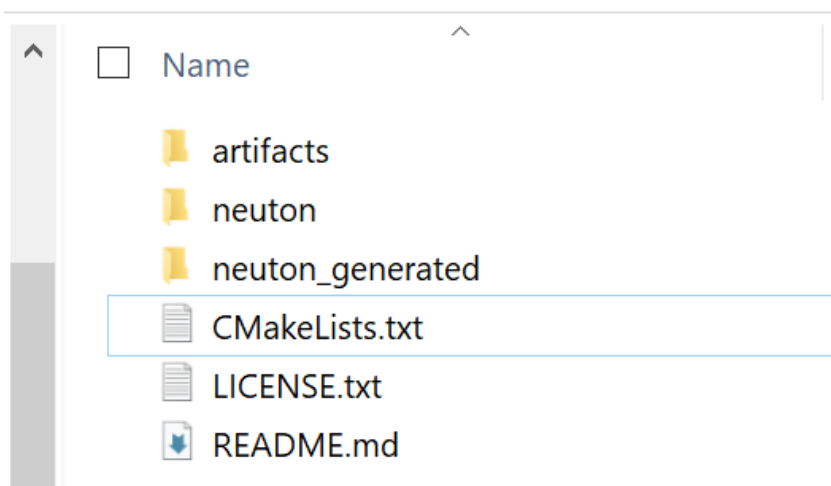


Maximum Training Duration (hours)

480

▶▶ Start Training

7. Click [Start Training] and the model will proceed to train for some time. The platform will then redirect you to the [Prediction] tab where you can access the cross-validation scores and download the model archive with everything necessary for embedding the model.
8. When your Neutron Tiny ML solution is ready, the downloaded archive will have the following structure:



7. Embedding Neutron Tiny ML

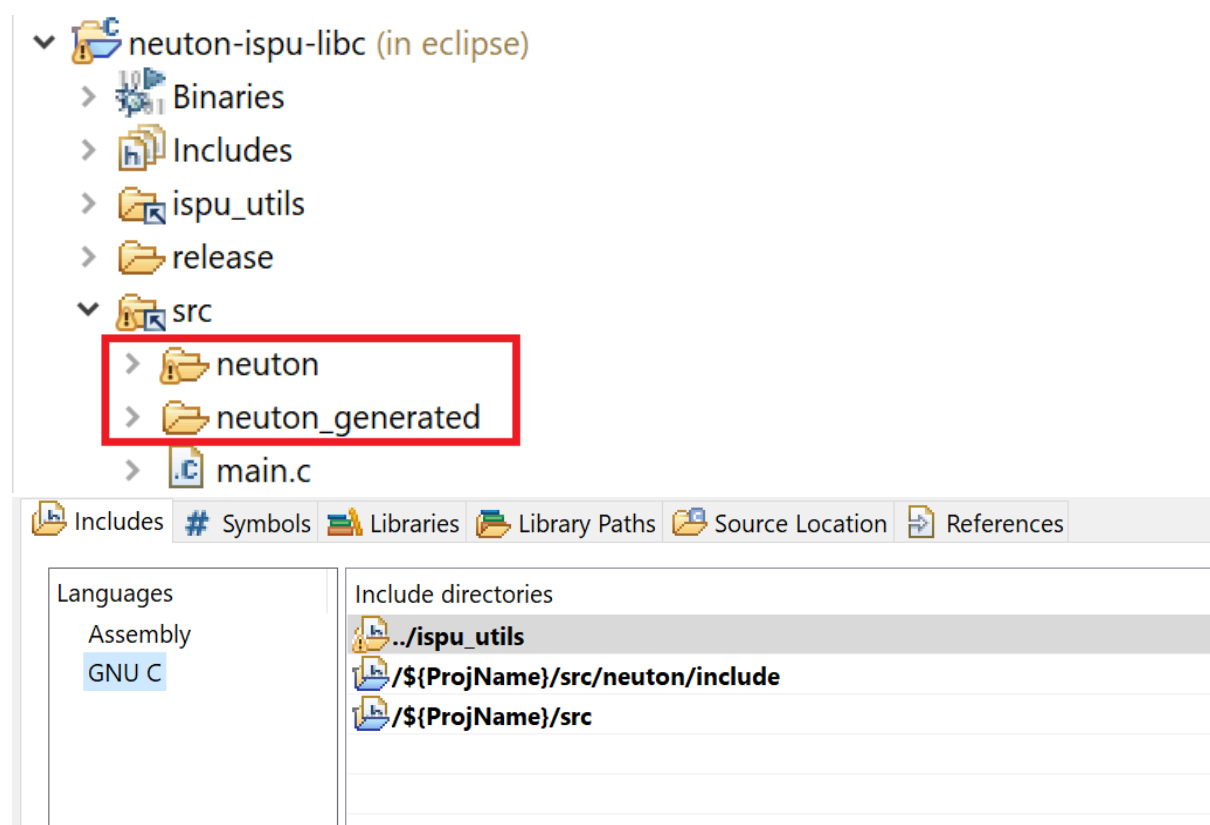
Follow *README.md* file in X-CUBE-ISPU expansion package (en.x-cube-ispu.zip\lspu\l) for the development environment setup.

In the X-CUBE-ISPU (<https://www.st.com/en/embedded-software/x-cube-ispu.html>) expansion package you can find examples for building ISPU firmware (en.x-cube-ispu\lspu\ism330is_lsm6dso16is\template\).

There are only two folders that should be added to your ISPU project for integrating Neutron:

- ``neutron`` main framework folder in which all Neutron algorithms and utilities are located;
- ``neutron_generated`` folder containing all user solution-specific files (neural network model, data preprocessing configuration, etc.).

Also, you should add the folder where ``./neutron_generated`` is located and ``./neutron/include`` folder to your project INCLUDE paths.



Refer to the *README.md* in the Neutron archive folder for integration tips and instructions.

8. Scoring

Our scoring metric for submitted ISPU firmware solutions will assess the overall performance in terms of detection and practical considerations, according to the following criteria:

- **Number of recognized classes:** The number of recognized classes is one of the most important metrics in the final **Score(S)**. The number of recognized classes will be normalized by $NCn = (1 - ((NC - NCmin) / (NCmax - NCmin)))$, where $NCmin = 1$ class and $NCmax = 15$ classes.
- **Neuton holdout balanced accuracy:** Ah is computed using the Neuton Inference Runner tool on the validation dataset. You can find this tool in every download archive at <your_neuton_project>\artifacts\inference_runner\l. Participants should also submit their validation dataset together with the ISPU firmware project.
- **ISPU inference latency:** The average latency L (in ms) executed on ISM330IS recorded in real time will be measured. The latency score is normalized by $Ln = (1 - ((L - Lmin) / (Lmax - Lmin)))$, where $Lmin = 1$ ms, $Lmax = 1/ODR$ (sensor output data rate). The latency score (Ln) will be used only for those participants who limit the ISPU inference latency to lower than 1/ODR.
- **ISPU data RAM occupation:** The memory occupation (in bytes) will be measured based on ISPU data RAM occupation on the ISM330IS for the storage of the deep neural network model and the program. Specifically, it is $Md = data + bss$ section occupation reported by the ISPU toolchain when building the project. Md is normalized by $Mdn = (1 - ((Md - Mdmin) / (Mdmax - Mdmin)))$, where $Mdmin = 100$ bytes, $Mdmax = 8192$ bytes.
- **ISPU program RAM occupation:** The memory occupation (in bytes) will be measured based on ISPU program RAM occupation on the ISM330IS for the storage of the deep neural network model and the program. To be more specific, it is $Mp = text$ section occupation reported by ISPU toolchain when building the project. Mp will be normalized by $Mpn = (1 - ((Mp - Mpmmin) / (Mpmmax - Mpmmin)))$, where $Mpmmin = 1000$ bytes, $Mpmmax = 32768$ bytes.
- **Realtime demo:** If a real time demo is provided by participants, the **Score (S)** will be incremented by adding $RT = 100 * [0.1 - 1]$, where 0.1-1 is assigned depending on jury evaluation of the real time demo demonstrated during the conference.

The final score is calculated as follows:

$$S = (200 * NCn) + (100 * Ah) + (80 * Ln) + (100 * Mdn) + (50 * Mpn) + RT$$

Terms and conditions:

Tentative Schedule:

Submission date: 31st May 2023

Conference date: 23-25th July

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

Registration:

Interested parties must email the ST, Neuton, and COINS contacts listed below.

Contact persons:

ST: marco.bianco@st.com

Neuton.AI: roman.rusak@neuton.ai

IEEE COINS: farshad.firouzi@duke.edu

Awards donated by ST Microelectronics to the top three entries:

1st place wins \$1,500

2nd place wins \$1,000

3rd place wins \$500

