



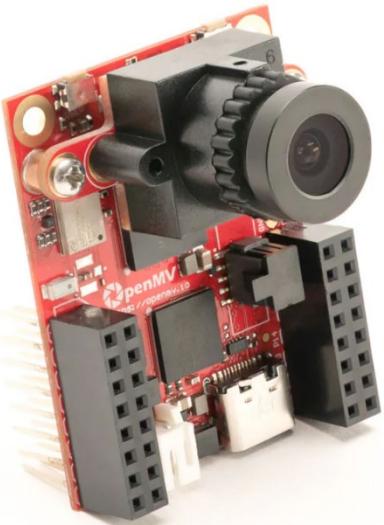
CONCEPTION NN POUR
SYSTÈMES EMBARQUÉS

LAB SESSION

Introduction

- L'objectif du projet est d'intégrer des modèles de réseaux de neurones sur la carte OpenMV H7 Plus, qui comprend une caméra et une interface dédiée pour la gérer, afin d'analyser un flux vidéo.
- Le TP se divise en 3 parties :
 - Prise en main d'OpenMV IDE pour la création de base de données
 - Création d'un premier projet en utilisant le site Edge Impulse pour intégrer un modèle de reconnaissance Pierre-Feuille-Ciseaux sur la carte OpenMV H7 Plus.
 - En utilisant les méthodes vues précédemment, développer un modèle pour répondre à un cahier des charges défini plus bas.

OpenMV H7 Plus



Specs

- **Processor:**

- ARM® 32-bit Cortex®-M7 CPU w/ Double Precision FPU
- 480 MHz (1027 DMIPS)
- Core Mark Score: 2400 (compare w/ Raspberry Pi 2: 2340)

- **RAM Layout (33MB Total):**

- 256KB .DATA/.BSS/Heap/Stack
- 32MB Frame Buffer/Stack
- 512KB SDRAM Cache
- 256KB DMA Buffers

- **Flash Layout (34MB Total):**

- 128KB Bootloader
- 32MB Embedded Flash Drive
- 1792KB Firmware

- **Supported Image Formats:**

- Grayscale
- RGB565
- JPEG (and BAYER/YUV422)

- **Maximum Supported Resolutions:**

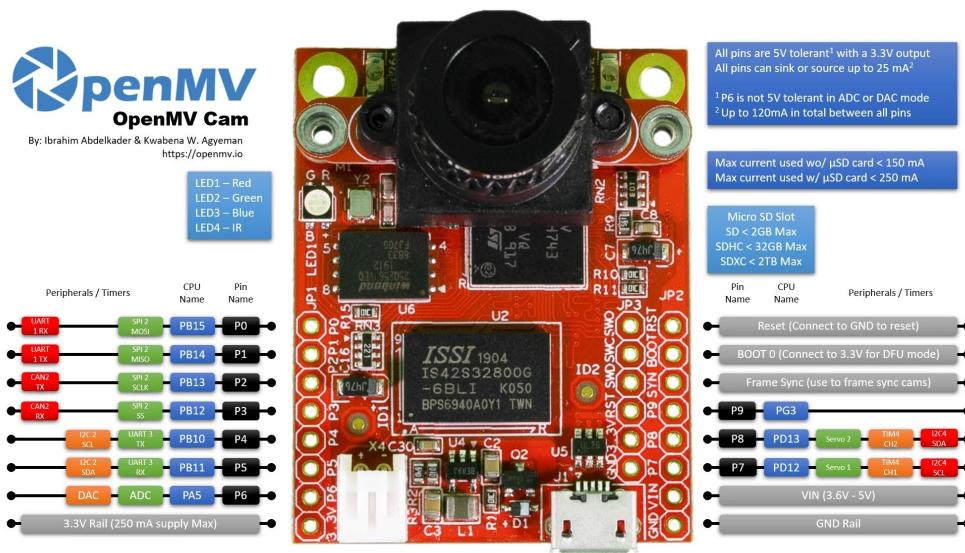
- Grayscale: 2952x1944 (5MP) and under
- RGB565: 2952x1944 (5MP) and under
- Grayscale JPEG: 2952x1944 (5MP) and under
- RGB565 JPEG: 2952x1944 (5MP) and under

- **Electrical Info:**

- All pins are 5V tolerant with 3.3V output.
All pins can sink or source up to 25mA.
P6 is not 5V tolerant in ADC or DAC mode.
Up to 120mA may be sunked or sourced in total between all pins. VIN
may be between 3.6V and 5V. Do not draw more than 250mA from your
OpenMV Cam's 3.3V rail.

- **Power Consumption:**

- Idle - No µSD Card: 140mA @ 3.3V
- Idle - µSD Card: 140mA @ 3.3V
- Active - No µSD Card: 230mA @ 3.3V
- Active - µSD Card: 240mA @ 3.3V





LET'S GO !

- Lancer OpenMV IDE.
- Cet environnement de développement est spécifiquement conçu pour exploiter pleinement les cartes OpenMV, avec en particulier des panneaux fournissant des informations sur le flux vidéos entrant.

Connexion/programmation

Gestion de la base de données

Pas d'image

Script exécuté sur la carte

Tampon de trame

Balance des couleurs de l'image caméra

The screenshot shows the OpenMV IDE interface. On the left, there's a file browser titled "éditeur de jeu de données" showing files like homs.class, index.class, palm.class, dataset_capture_script.py, and labels.txt. Below it is an "Aperçu d'image" (image preview) window. The central part is a code editor with a red border, displaying Python code for a sensor dataset capture script. The right side features a "Tampon de trame" (frame buffer) window showing a camera feed of a product in a plastic bag, and a "Histogramme" (histogram) window showing three color channels (RGB) with statistical data below them. The status bar at the bottom provides hardware and software information.

```

Fichier Édition Outils Fenêtre Aide
main.py dataset_capture_script.py
Éditeur de jeu de données
└── dataset_capture_script.py
    ├── homs.class
    ├── index.class
    └── palm.class
    └── dataset_capture_script.py
    └── labels.txt

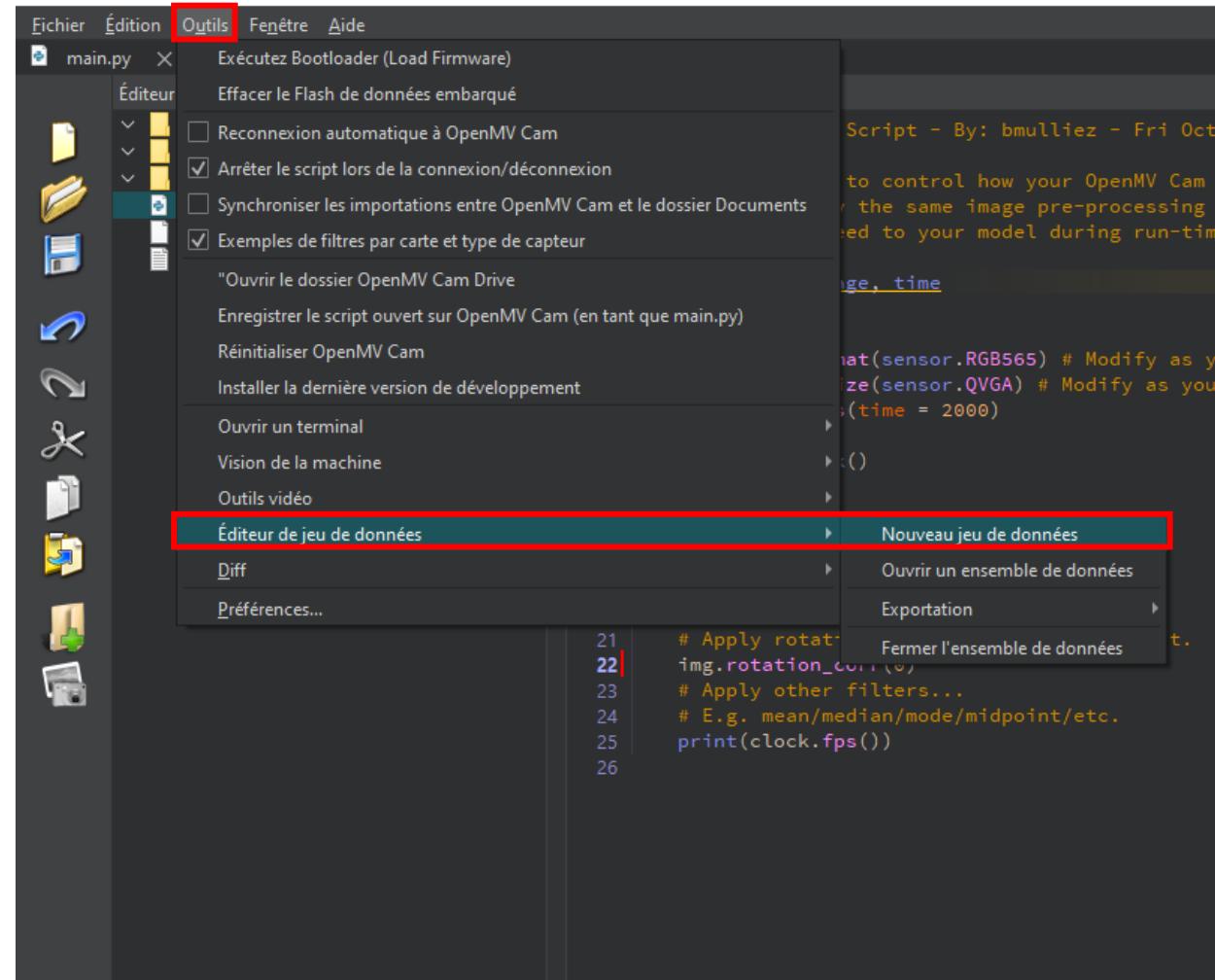
::: sensor
1 # Dataset Capture Script - By: bmulliez - Fri Oct 17 2025
2
3 # Use this script to control how your OpenMV Cam captures images for your dataset.
4 # You should apply the same image pre-processing steps you expect to run on images
5 # that you will feed to your model during run-time.
6
7 import sensor, image, time
8
9 sensor.reset()
10 sensor.set_pixformat(sensor.RGB565) # Modify as you like.
11 sensor.set_framesize(sensor.QVGA) # Modify as you like.
12 sensor.skip_frames(time = 2000)
13
14 clock = time.clock()
15
16 while(True):
17     clock.tick()
18     img = sensor.snapshot()
19     # Apply lens correction if you need it.
20     #img.lens_corr()
21     # Apply rotation correction if you need it.
22     #img.rotation_corr(0)
23     # Apply other filters...
24     # E.g. mean/median/mode/midpoint/etc.
25     print(clock.fps())
26

```

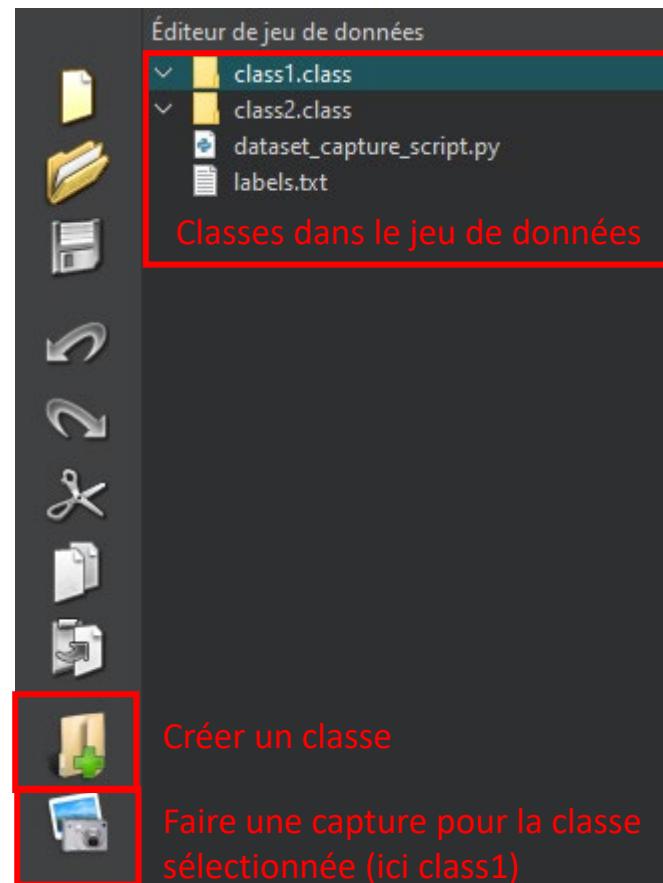
Résultat de la recherche Terminal série

Planche: OpenMV Cam H7 Plus Capteur: OV5640 Version du micrologiciel : 4.7.0 - [la plus récente] Port série : COM9 Lecteur : E/ FPS : 16,1

- Il est possible de créer une base de données directement avec la carte OpenMV H7 Plus et sa caméra embarquée.
- Pour ce faire, cliquer sur Outils -> Editeur de jeu de données -> Nouveau jeu de données et choisissez un répertoire pour créer votre base de données.



- En utilisant le script fourni sur Moodle, il est alors possible de développer un jeu de données à plusieurs classes :
 - Créer une classe
 - Dans le panneau d'éditeur de jeu de données, sélectionner la classe souhaitée. Il est alors possible de faire une capture des données issues de la caméra pour ajouter un échantillon à la classe sélectionnée.



- Vous savez maintenant constituer rapidement une base de données contenant des images issues de la caméra présente sur la carte OpenMV H7 Plus.
- A présent, vous allez développer des modèles simples pour exploiter le microcontrôleur de la carte. Une technique rapide, simple est l'utilisation du site [Edge Impulse](#).
- L'objectif de ce site est le développement de modèles préconçus, leur entraînement, leur optimisation pour des cibles matérielles et leur déploiement sur une seule et même plateforme. C'est un outil simple d'utilisation et pratique pour le prototypage ou la preuve de concept. L'optimisation fine de modèles y est toutefois très limitée.

 **EDGE IMPULSE**

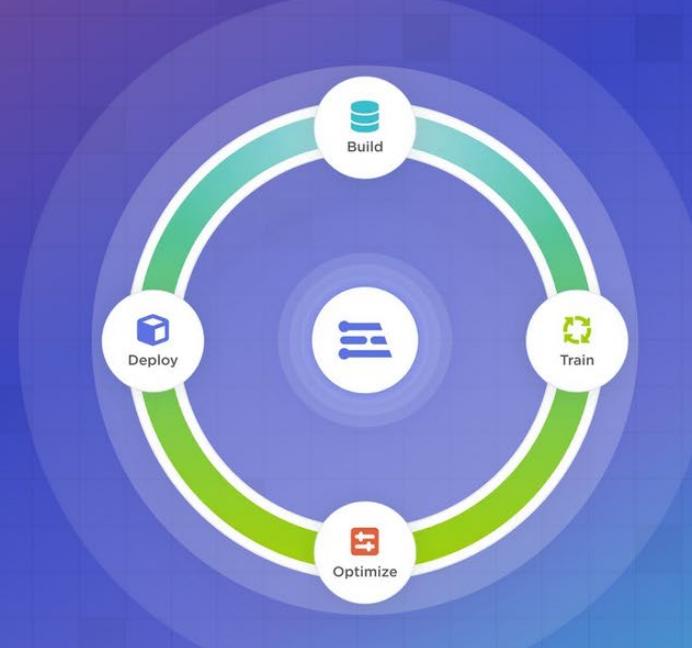
Product ▾ Solutions ▾ Developers ▾ Pricing ▾ Company ▾ Blog Login [Get started](#)

AI for Any Edge Device

MCUs, NPUs, CPUs, GPUs

Build datasets, train models, and optimize libraries to run directly on device; from the smallest microcontrollers to gateways with the latest neural accelerators (and anything in between).

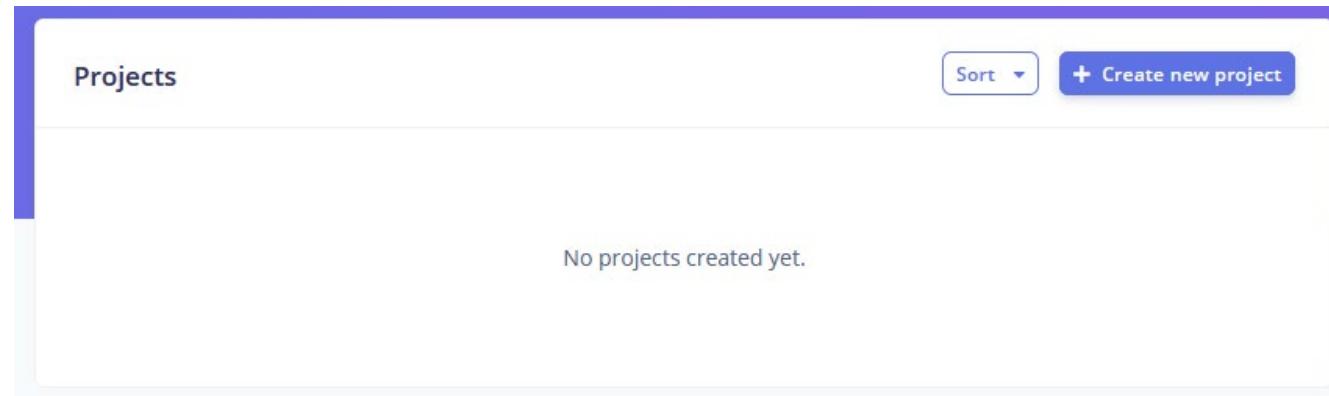
[Get started](#) [Book a demo](#)



```

graph TD
    Build((Build)) --> Train((Train))
    Train --> Deploy((Deploy))
    Deploy --> Optimize((Optimize))
    Optimize --> Build
  
```

- Pour tester la méthodologie de Edge Impulse, vous allez créer un modèle de reconnaissance de position de la main à 3 classes : pierre, feuille, ciseau.
- Créez un compte sur Edge Impulse puis créez un projet.



- Vous pouvez importer des données issues de collection en ligne ou importer vos propres données.
- Téléchargez la base de données fournies sur Moodle et importer là sur Edge Impulse via **Add Existing Data**. Dans la fenêtre qui apparaît, choisissez **Upload data**, puis **Select a folder** et sélectionnez le dossier contenant la base de données que vous venez de télécharger. Laissez les options. Edge Impulse vous propose un type d'application qui correspond à la nature de la base de données que vous importez, ici **Object Detection**.

The screenshot shows the Edge Impulse web interface for a project titled "Rock_paper_scissors". The left sidebar includes links for Dashboard, Devices, Data acquisition, Experiments, Impulse design, Create impulse, Live classification, Model testing, Deployment, Versioning, Documentation, Forums, and Discord. A "Upgrade Plan" section at the bottom offers higher job limits and more collaborators, with a "View plans" button. The main content area has tabs for Project info, Keys, Export, and Jobs, with "Project info" selected. It displays the project name "Rock_paper_scissors" and a brief description: "This is your Edge Impulse project. From here you acquire new training data, design impulses and train models." Below this is a "Getting started" section with three buttons: "Add existing data" (highlighted with a red box), "Collect new data", and "Upload your model". Further down, there's a "Start with a tutorial" section featuring three cards: "Motion: Gesture recognition" (blue card with a hand icon), "Images: Object detection" (teal card with a camera icon), and "Audio: Audio classification" (orange card with a sound wave icon).

- Dans l'application proposée par Edge Impulse, il est nécessaire de labeliser les données.

The screenshot shows the Edge Impulse AI labeling interface. At the top, there are tabs for Dataset, Data sources, Synthetic data, Labeling queue (299) (which is highlighted with a red box), and AI labeling. Below the tabs, there's a section for DATA COLLECTED showing 299 items. A TRAIN / TEST SPLIT section indicates a ratio of 1:1. To the right, there's a Collect data section with a note to connect a device. The main area is titled 'Dataset' and contains a table for Training (236), Test (63), and Post-processing (0). The table columns are SAMPLE NAME, LABELS, and ADDED. Two rows are listed: 'scissors_99' and 'scissors_98', both added today at 10:59:56. To the right of the table, there's a preview window showing a hand holding a piece of paper. A red box highlights the 'Save labels' button at the bottom right of the preview window. The overall interface is clean and modern, designed for efficient data annotation.

- Dessiner alors des *bounding boxes* pour labeliser vos données et sauver les pour chaque échantillon. Après quelques tracés, Edge Impulse est capable de déterminer à peu près les masques lui-même.
- Cette étape est fastidieuse...



Dashboard

Devices

Data acquisition

Experiments

EON Tuner

Impulse design

Create impulse

Retrain model

Live class

Model testing

Post-processing

Deployment

Versioning

- Cliquez sur **Create Impulse** pour commencer à développer un modèle.
- Dans la liste des cibles matérielles, sélectionnez la carte **OpenMV H7 Plus**, préconfigurée par Edge Impulse.
- Sélectionnez ensuite un bloc de prétraitement et un modèle prédéfini de détection d'objets.

Impulse #1

An impulse takes raw data, uses signal processing to extract features, and then uses a learning block to classify new data.

Image data

Input axes: image
Image width: 96
Image height: 96
Resize mode: Fit shortest axis

Image

Name: Image
Input axes (1): image

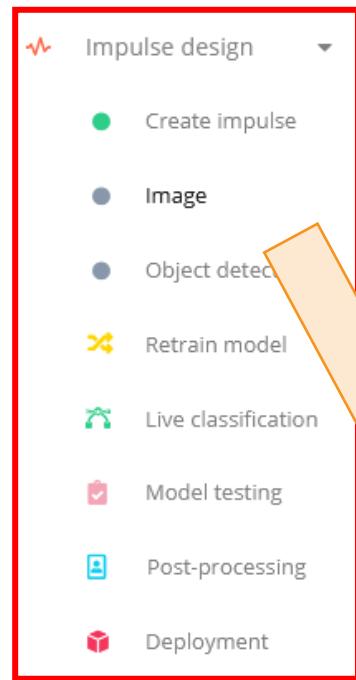
Object Detection (Images)

Name: Object detection
Input features: image
Output features: 3 (Paper, rock, scissors)

Output features

3 (Paper, rock, scissors)

Save impulse



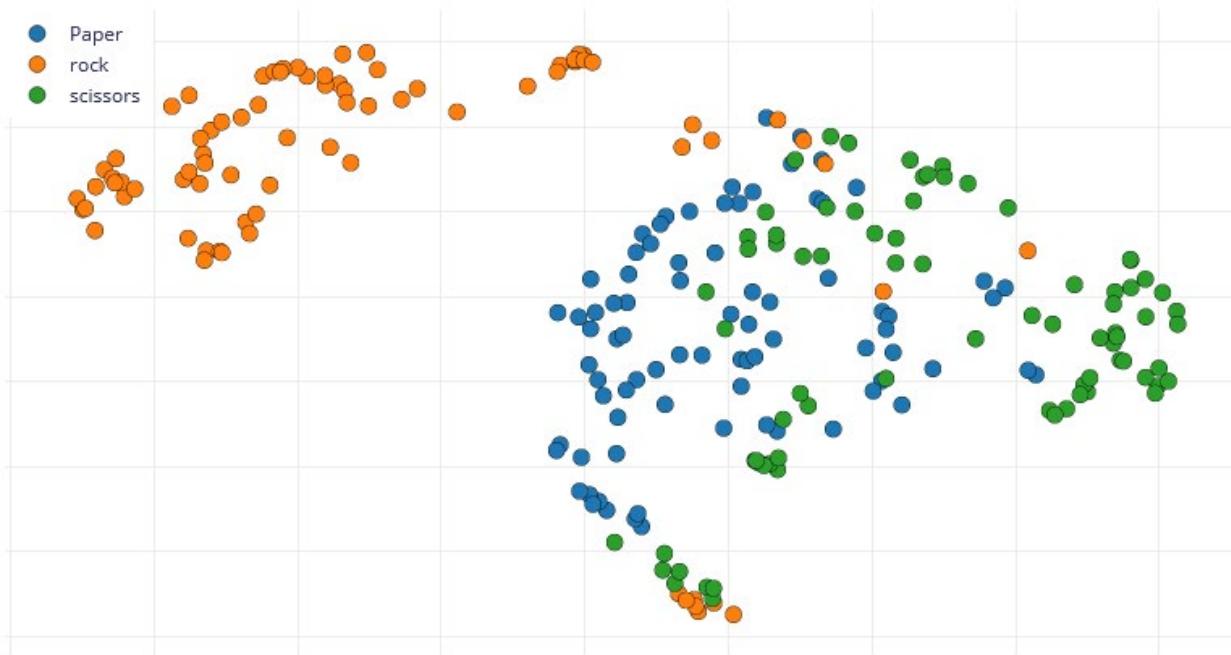
- Le panneau de gauche présente les différentes étapes à réaliser dans la méthodologie Edge Impulse afin de pouvoir déployer un modèle sur une cible matérielle.
- Dans la partie Image, enregistrer les paramètres avec lesquels les images sont traités, puis faites générer les *features* à Edge Impulse.
- A ce moment, Edge Impulse sépare la base de données en une base d'entraînement et une base de validation.

This screenshot shows the 'Image' workspace in Edge Impulse. At the top, there are tabs for 'Parameters' and 'Generate features'. Below that, the 'Raw data' section displays an image of a hand making a 'scissors' gesture. In the bottom left, the 'Raw features' section shows a list of binary values. The 'Parameters' section includes a dropdown for 'Color depth' set to 'RGB'. A red box highlights the 'Save parameters' button at the bottom right of the parameters panel.

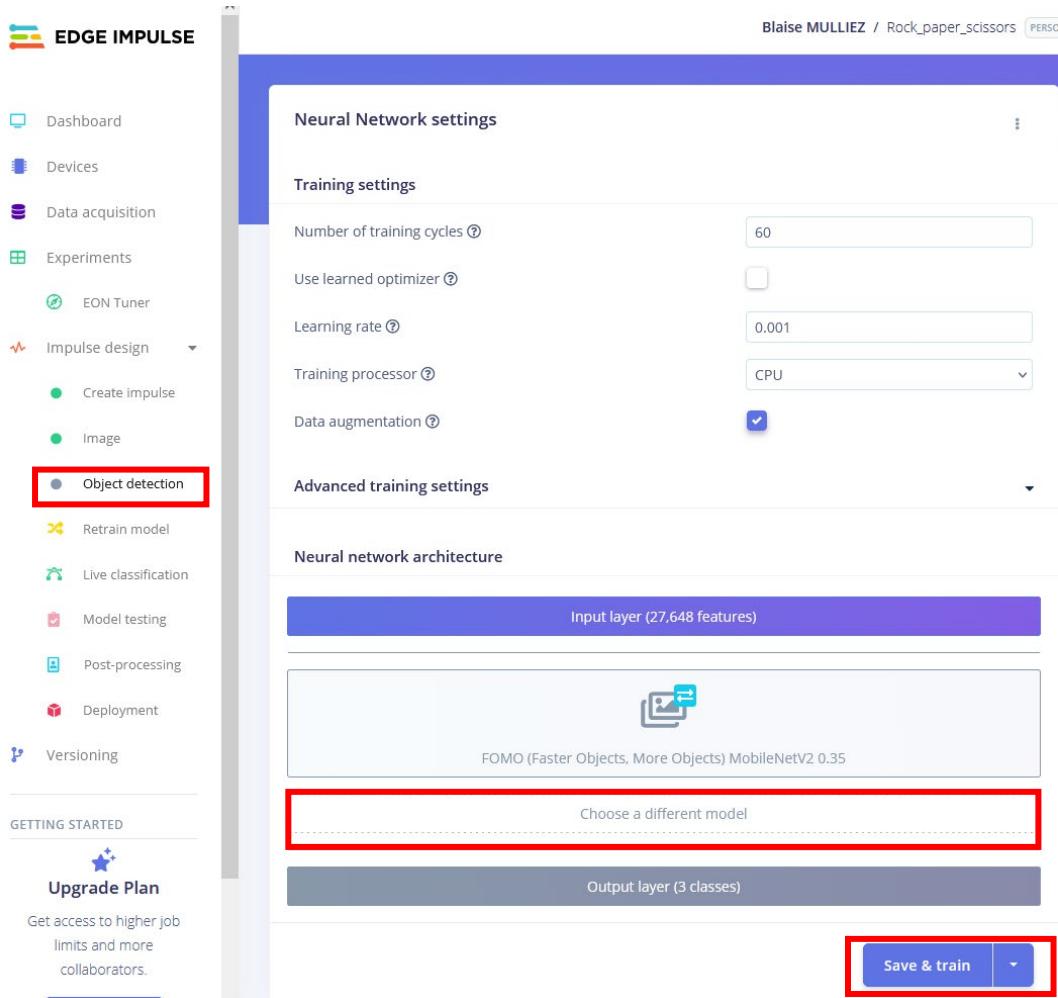
This screenshot shows the 'Generate features' workspace. It starts with a 'Training set' summary: 'Data in training set' (236 items) and 'Classes' (3: Paper, rock, scissors). A large orange arrow points from the 'Generate features' button in the top right towards the 'Output' section. The 'Output' section shows a log of the feature extraction process, starting with 'Job scheduled' and ending with 'Job completed (success)'. The 'On-device performance' section indicates a processing time of '1 ms.'.

- Edge Impulse détermine des *features* afin de séparer au mieux les classes. Dans le cas présent, la classe *rock* est bien séparée dans deux autres. La frontière entre *paper* et *scissors* est moins nette. La base de données contient également quelques *outliers*.

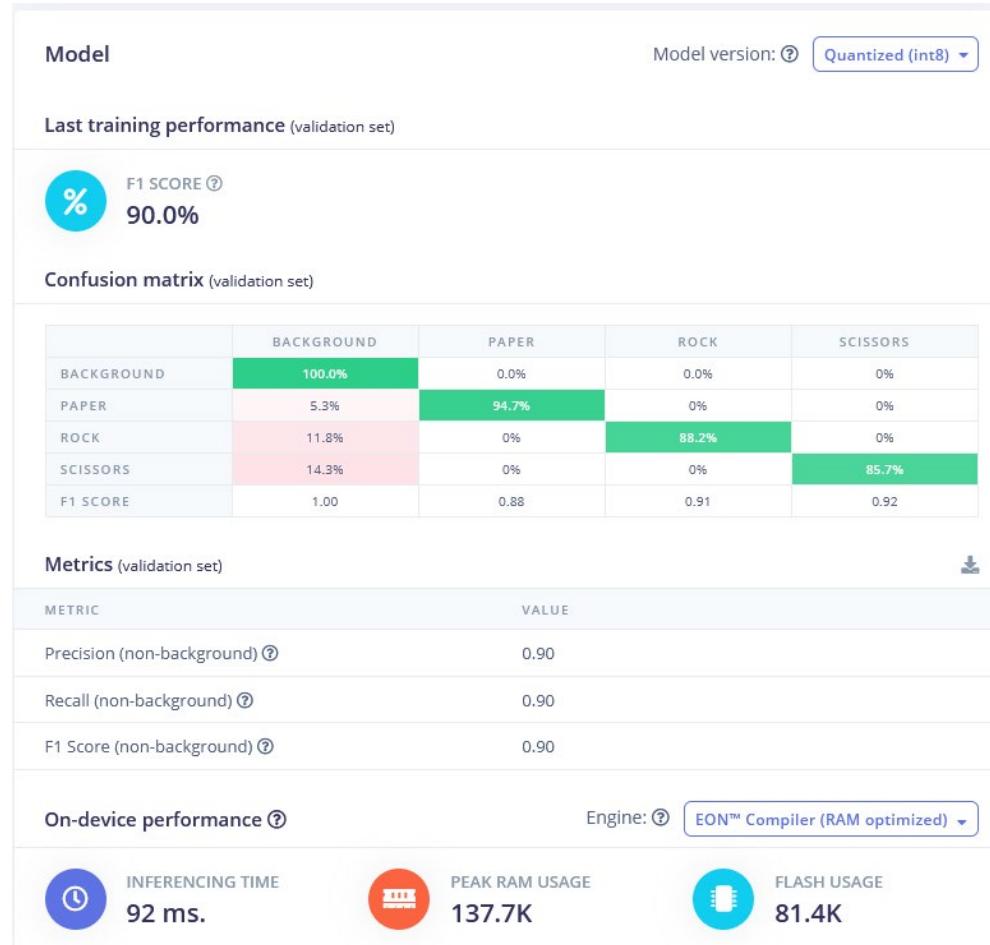
Feature explorer



- Vous pouvez maintenant entraîner un modèle. Vous pouvez choisir un modèle différent de celui par défaut si vous le souhaitez pour comparer les performances des différentes architectures.
- Vous pouvez également augmenter le nombre de cycles d'entraînements, changer la taille du set de validation, changer la taille des batch...



- En fin d'entraînement, Edge Impulse vous propose une matrice de confusion, afin de rendre compte de la qualité de l'entraînement.
- Ici, malgré une base de données limitée (100 échantillons par classe), la reconnaissance est efficace.



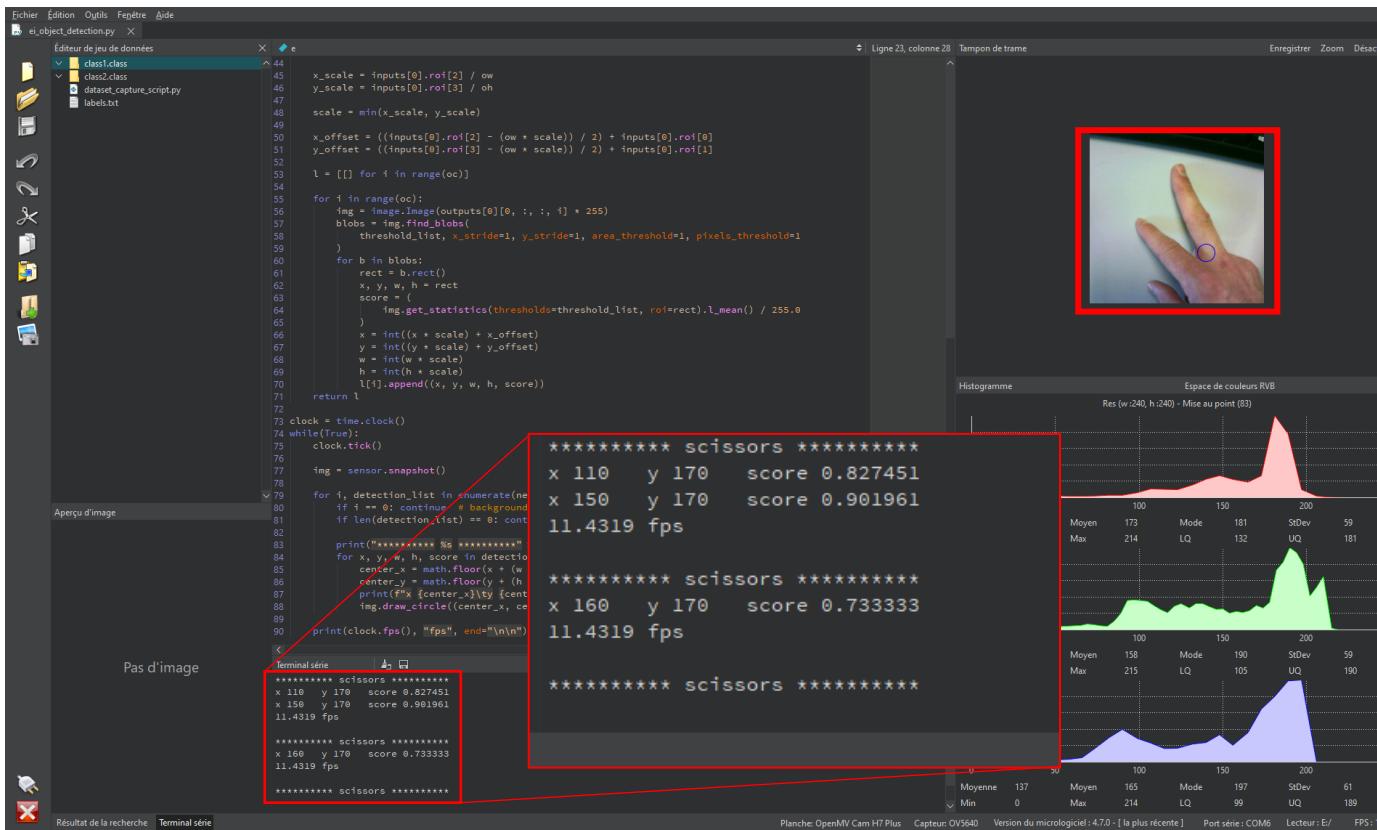
- Vous pouvez maintenant déployer votre modèle en recherchant **OpenMV** dans les options de déploiement.
- Cliquez sur **Build**.

The screenshot shows the Edge Impulse web interface. On the left is a sidebar with the following menu items:

- Dashboard
- Devices
- Data acquisition
- Experiments
- EON Tuner
- Impulse design
 - Create impulse
 - Image
 - Object detection
 - Retrain model
 - Live classification
 - Model testing
 - Post-processing
- Deployment

The main area is titled "Configure your deployment". It contains a search bar with the text "OpenMV library" highlighted by a red box. Below the search bar, there is a section titled "SELECTED DEPLOYMENT" with the "OpenMV library" option selected. A description below it states: "A library that runs your impulse on OpenMV cameras." At the bottom of this section is a large blue "Build" button. At the very bottom of the page, there is a copyright notice: "© 2025 EdgeImpulse Inc. All rights reserved".

- Dans l'archive générée se trouvent 3 fichiers :
 - ei_object_detection.py : script à exécuter dans OpenMV IDE pour observer les résultats d'inférence dans le terminal.
 - labels.txt : contient les labels.
 - trained.tflite : votre modèle, au format TfLite.
- Placer les deux derniers fichiers dans la mémoire de la carte OpenMV et exécuter le script Python dans OpenMV IDE.
- Dans OpenMV IDE, vous pourrez alors observer dans la console le résultat de la détection effectuée par votre inférence ainsi qu'un cercle de couleur dans la fenêtre vidéo, correspondant au label de l'objet détecté.



- Maintenant que vous savez comment construire une base de données grâce à OpenMV IDE, comment entraîner un modèle sur Edge Impulse et le déployer sur OpenMV, votre mission est la suivante :

Développer un modèle de décodage de l'alphabet de la langue des signes

- Votre modèle ne doit pas reconnaître les signes avec mouvement (« J », « Y » et « Z »).
- Vous pouvez limiter votre nombre de classes, mais devez au moins être capables de reconnaître 10 signes (choisissez judicieusement).
- Vous pouvez développer votre propre base de données ou importer des bases de données en ligne (par ex. sur [kaggle.com](https://www.kaggle.com)).

