



# détection des moustiques avec Arduino Nicla Vision et des données open source

Jose Bagur (UVG), João Vitor Freitas da Costa (UERJ), Silvia A. Sotelo López (UPB),  
Peter Ing (Arm SAAIA) et David J. Cuartielles Ruiz (Arduino/Université de Malmö)

L'équipe s'est confrontée à la problématique critique liée à la prolifération des moustiques et à leur rôle dans la propagation de maladies. Elle a donc développé un système de détection qui utilise le son pour identifier les moustiques et activer les dispositifs répulsifs. Face à l'augmentation alarmante des cas de dengue en Amérique, le projet vise à offrir une solution efficace pour la détection des moustiques en conditions réelles. Découvrez comment l'association de la technologie et de tinyML peut favoriser le développement durable !





Durant l'été 2024, le CIPT a collaboré avec l'Université de Harvard, Barnard College, UNIFEI, IBM Research Brazil et Arduino pour organiser le *Workshop on TinyML for Sustainable Development* à São Paulo, Brésil. L'atelier a réuni des personnes venant principalement des Amériques et passionnées par le développement de systèmes intelligents en périphérie. Pendant une semaine, les participants ont eu l'occasion de travailler avec des experts et d'acquérir des connaissances sur des sujets variés tels que la collecte de données, la formation des appareils Edge, les LLM DIY, l'apprentissage avec tinyML et les exemples d'application des technologies. Chaque participant était également invité à présenter ses propres travaux dans le domaine.

Au cours de cette activité de cinq jours, nous avons eu l'occasion de nouer des liens, d'assister à une série de conférences et de séances pratiques, de visiter l'université de São Paulo et son usine de fabrication de circuits imprimés, ainsi que de collaborer en équipe pour développer diverses solutions techniques. Nous, les auteurs de cet article, avons décidé de collaborer à la recherche de solutions potentielles aux populations croissantes de différents types de moustiques, vecteurs de maladies considérées comme éradiquées. Dans certaines régions, il y a plusieurs cycles de reproduction par saison, ce qui est inédit. Chaque été, les zones touchées par les maladies transmises par les insectes, telles que la malaria, s'élargissent. Nous avons eu l'idée de concevoir un système capable de détecter la présence de certains types de moustiques par l'analyse sonore de leurs battements d'ailes et de répondre par l'émission d'un brouillard répulsif, pour éloigner ces nuisibles.

L'augmentation du nombre de cas de dengue dans les Amériques est la principale motivation de ce projet. En 2024, 11 517 728 cas ont été enregistrés, soit une augmentation alarmante de 230 % par rapport au nombre de cas signalés en 2023 (rapport de l'OPS, septembre 2024 [1]).

Pour reproduire ce projet, vous aurez besoin de :

- Un Arduino Nicla Vision [2]. Nous nous concentrerons uniquement sur les capacités de détection sonore de l'appareil.

- Le jeu de données qui contient des sons de moustiques qui permettront de détecter leur présence.
- Un compte sur Edge Impulse, la plate-forme en ligne qui créera le moteur d'inférence après l'avoir alimenté avec le jeu de données mentionné.
- Une carte relais pouvant être déclenchée par un signal de 3,3 V (directement depuis le Nicla Vision).
- Un piézo-humidificateur
- Des fils de connexion
- Un fer à souder (le Nicla Vision est livré avec des pastilles crénelées, mais n'a pas de broches pour connecter quoi que ce soit. Nous avons dû souder quelques fils pour que la carte puisse contrôler le reste du dispositif).

Arduino a fait don des cartes nécessaires à l'organisation de l'atelier et Edge Impulse propose des comptes gratuits pour les makers, qui peuvent créer le code pour les cartes sur la base de l'ensemble de données. Cependant, nous avons dû acheter localement la carte relais et l'humidificateur. São Paulo compte plusieurs magasins où il est possible de se procurer des composants électroniques du commerce.

## Contexte

Les moustiques ont une durée de vie courte mais intense. Plusieurs espèces existantes sont très résistantes aux conditions climatiques, et leurs œufs fécondés peuvent survivre dans des environnements très hostiles jusqu'à ce que les niveaux de température et d'humidité adéquats facilitent leur éclosion. Les moustiques ont été largement étudiés par les scientifiques. Ils se nourrissent du sang d'autres animaux, ce qui les rend très aptes à transmettre des maladies grâce au mélange de sanguins différents qu'ils transportent dans leur système. Certaines expériences sur l'ADN (comme Prometeo de Microsoft) ont démontré que le sang d'un seul moustique pouvait contenir des traces d'ADN de 3 ou 4 espèces différentes de mammifères, incluant les humains, les chiens, les vaches et les moutons. Cette capacité à passer rapidement d'un hôte à l'autre rend la transmission de maladies rapide.

Il est possible de distinguer le type et le sexe des moustiques en observant la taille

de leurs ailes. Cela signifie également qu'il est possible de classer ces insectes selon la signature sonore qu'ils produisent, due à leurs ailes mécaniquement distinctes. Bien que cela soit très difficile pour les humains, un système tinyML correctement entraîné pourrait y parvenir. C'est dans ce contexte que l'accès à une vaste base de données, contenant des milliers d'enregistrements sonores de moustiques, devient utile. Elle pourrait servir à développer un moteur d'inférence, soit une minuscule machine d'intelligence artificielle capable de distinguer les sons des moustiques.

Un défi inattendu rencontré lors de notre projet est qu'au Brésil, deux types différents de courant alternatif coexistent : 110 et 230 V. Lors de l'achat des composants, nous avons également acheté un fer à souder bon marché, aucun de nous n'ayant pensé à en apporter un. Nous devions souder des fils de liaison aux broches de la carte Nicla Vision afin de la connecter à la carte relais. Nous avons réalisé ensuite que notre fer à souder ne fonctionne pas sous 110 V AC. Malgré nos tentatives, le manque de puissance rendait impossible son utilisation, et nous avons dû abandonner. Finalement, nous avons dû nous résoudre à utiliser la prise de la machine à raser de notre chambre d'hôtel pour réaliser les soudures, ajoutant ainsi une anecdote mémorable à notre projet. À l'avenir, nous nous souviendrons toujours d'emporter un fer à souder USB-C !

## Idée et réalisation

Lors de l'atelier, nous avons conçu une preuve de concept pour un dispositif capable de détecter la présence de moustiques et de produire un brouillard qui pourrait contenir différents types d'essences (telles que, la citronnelle) pour nous avertir de la présence du moustique ou simplement le faire fuir. L'idée était de vérifier si les outils permettaient de construire rapidement ce dispositif intelligent. En termes de technologie, l'Arduino Nicla Vision est équipé d'une caméra couleur, d'un microphone numérique, d'un IMU, d'un processeur double cœur (Arm Cortex M7 + M4), des capacités de communication (Wifi, BLE), d'un port d'extension I<sup>2</sup>C, d'une batterie/un connecteur d'alimentation externe, et de plusieurs d'E/S numériques et d'entrées analogiques. Pour notre projet,

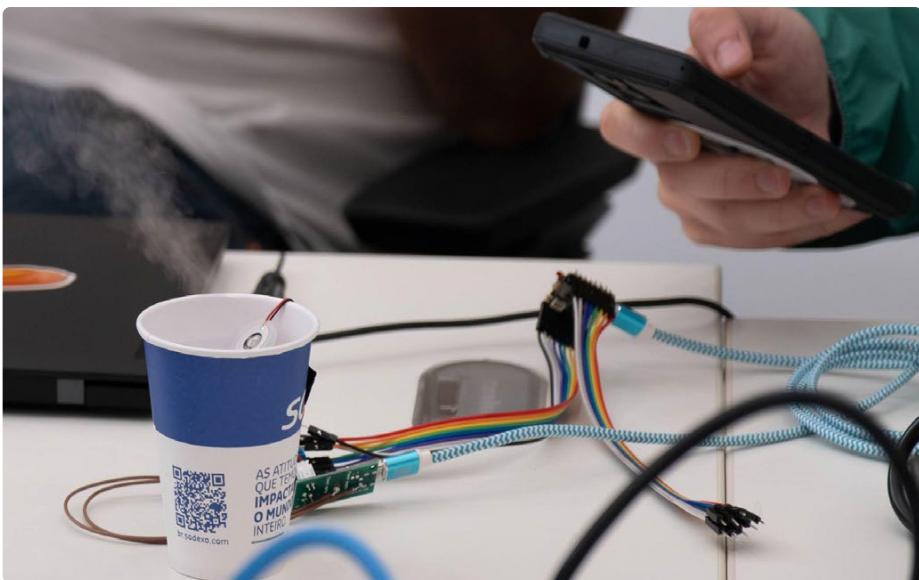


Figure 1. Évaluation en direct de l'humidificateur basé sur tinyML, à l'aide d'un enregistrement de l'ensemble de test lors de la clôture du Workshop on TinyML for Sustainable Development.

nous avons uniquement utilisé le microphone et le microcontrôleur Cortex M7, ainsi qu'une broche numérique pour contrôler l'humidificateur. Les autres fonctionnalités n'étaient pas nécessaires pour cette application. Tout le code (moteur d'inférence, communication série pour la surveillance, contrôle des broches numériques et collecte de données par le microphone) était exécuté avec un seul processeur, laissant le second libre pour d'autres tâches.

Pour générer le brouillard, nous avons utilisé un humidificateur non (**figure 1**). Ce dernier était équipé d'un élément piézoélectrique, activé par un oscillateur et un amplificateur, avec un interrupteur tactile offrant trois réglages : arrêt, demi-puissance et pleine puissance. Nous avons essayé deux méthodes différentes pour le faire fonctionner. La première, qui consistait à couper un des fils du piézo et à le connecter via un relais, s'est avérée infructueuse car l'impédance du relais perturbait le fonctionnement du piézo, empêchant la projection de la brume. La seconde méthode consistait à utiliser le relais pour actionner l'interrupteur tactile de l'humidificateur. Cela signifiait également que nous devions cliquer deux fois sur le relais pour activer l'humidificateur, et une seule fois pour le désactiver. Cette solution s'est révélée efficace, le système fonctionnant parfaitement sans aucun problème de rebond.

## Méthodologie

Pour détecter efficacement la présence de moustiques et activer un répulsif approprié, nous avons opté pour l'utilisation d'un modèle d'apprentissage profond déployé sur l'Arduino Nicla Vision. Cette méthode permet à l'appareil d'« apprendre » et d'identifier les sons émis par les moustiques, offrant ainsi une

solution plus flexible et adaptative par rapport aux méthodes algorithmiques ordinaires. Ces approches conventionnelles deviennent souvent trop complexes dans les environnements réels, tandis que l'apprentissage profond excelle en s'adaptant à la variabilité et aux subtilités des données, garantissant ainsi une détection plus fiable.

tinyML a révolutionné les applications à la périphérie en permettant de déployer des modèles d'apprentissage profond tels que les réseaux neuronaux convolutifs sur les CPU que l'on trouve dans les microcontrôleurs, au lieu de recourir à des GPU gourmands en énergie.

Le Nicla Vision est doté d'un microcontrôleur à double cœur, combinant le puissant Arm Cortex M7 avec le cœur Arm Cortex M4. Cette configuration permet un équilibre entre un traitement puissant et une faible consommation d'énergie, ce qui la rend idéale pour les applications basées sur la vision qui

nécessitent à la fois une efficacité de calcul et une faible consommation énergétique, tout en restant en ligne avec les principes de durabilité.

## Créer le modèle avec Edge Impulse

Edge Impulse a été le pionnier des applications tinyML en fournissant aux développeurs embarqués une plateforme MLOps qui facilite le processus de collecte et d'utilisation des données de capteurs pour construire et déployer des modèles sur des microcontrôleurs et des environnements Edge. Le processus général avec Edge Impulse est (voir **figure 2**):

- 1. Collecte des données :** Utiliser des capteurs pour collecter des données de l'environnement, ou importer des ensembles de données existants, à utiliser pour entraîner le modèle.

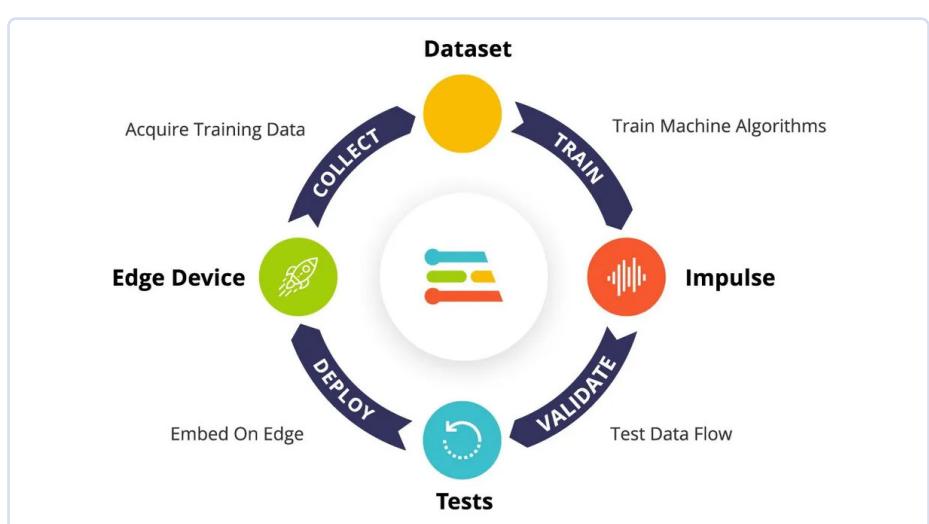


Figure 2. Le processus général de création de modèles avec Edge Impulse. (Source : Edge Impulse)

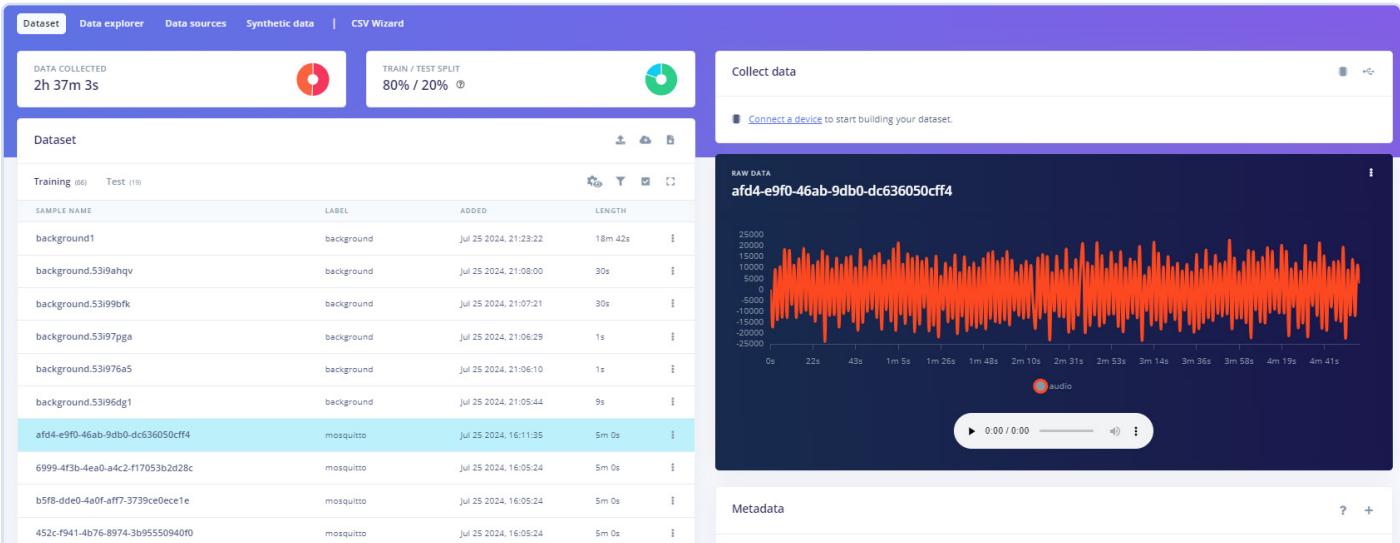


Figure 3. Travail avec des ensembles de données en ligne et une capture audio en direct avec Edge Impulse.

2. **Prétraitement des données :** Exploiter les outils d'Edge Impulse pour étiqueter, prétraiter et préparer les données, les rendant ainsi prêtes pour l'apprentissage des modèles ML.
3. **Entraînement des modèles :** Créer et entraîner des modèles d'apprentissage automatique en utilisant un catalogue d'architectures préconçues ou personnalisées, basées sur les données collectées.
4. **Déploiement des modèles :** Installer ces modèles sur des appareils Edge, avec des déploiements optimisés et une intégration simplifiée adaptée à Arduino, y compris le Nicla Vision, pour une analyse et des actions en temps réel.

Pour développer un prototype fonctionnel et réaliser une démonstration efficace lors de l'atelier, nous avions besoin d'un moyen de démontrer le concept tout en le rendant possible à utiliser dans un cadre réel. Nous avons atteint cet objectif en combinant un ensemble de données de sons de moustiques réels avec quelques sons synthétisés dérivés d'enregistrements trouvés en ligne. De plus, nous avons capturé le bruit ambiant (**figure 3**), directement avec l'Arduino Nicla Vision à un taux d'échantillonnage de 8 kHz. Nous avons collecté 2 heures et 37 minutes de données au total, qu'Edge Impulse a automatiquement

DESCRIPTION	AUTHOR	RECOMMENDED
<b>Audio (MFCC)</b> OFFICIALLY SUPPORTED Extracts features from audio signals using Mel Frequency Cepstral Coefficients, great for human voice.	Edge Impulse	★ Add
<b>Audio (MFE)</b> OFFICIALLY SUPPORTED Extracts a spectrogram from audio signals using Mel-filterbank energy features, great for non-voice audio.	Edge Impulse	★ Add
<b>Spectrogram</b> OFFICIALLY SUPPORTED Extracts a spectrogram from audio or sensor data, great for non-voice audio or data with continuous frequencies.	Edge Impulse	Add

Figure 4. Edge Impulse propose des blocs DSP préconstruits adaptés à divers types de traitement audio.

divisées en 80% pour l'entraînement et 20% pour les tests. Cette méthode illustre bien comment Edge Impulse gère à la fois les ensembles de données en ligne et la capture en direct de données audio avec des cartes prises en charge telles que la Nicla Vision. En utilisant le bruit de fond de la salle, nous avons pu ajuster le modèle pour notre environnement de test, mettant en évidence l'adaptabilité de l'apprentissage automatique.

Cette étape était cruciale car les données initiales avaient été recueillies dans divers environnements intérieurs et extérieurs, avec des niveaux de bruit de fond très différents de ceux rencontrés lors de notre atelier. La capacité du modèle à s'adapter à ces nouvelles conditions a mis en évidence la capacité de l'apprentissage automatique à s'adapter à des environnements variés. Pour résoudre le problème de la détection de

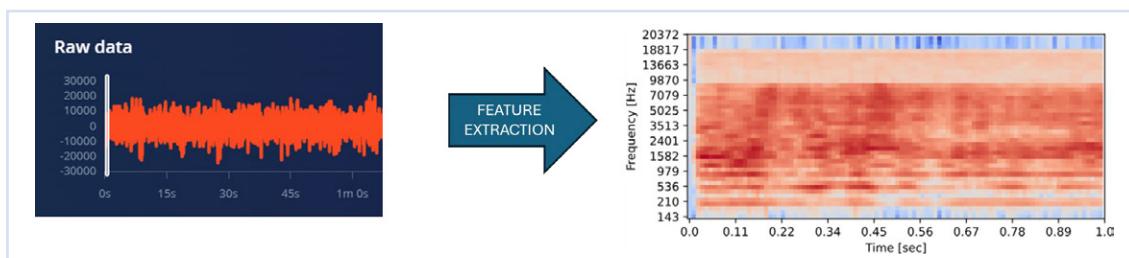


Figure 5. Le Mel-filterbank Energy Spectrogram convertit les sons de moustiques en une image 2D, en utilisant un modèle de classificateur similaire à la classification d'images.

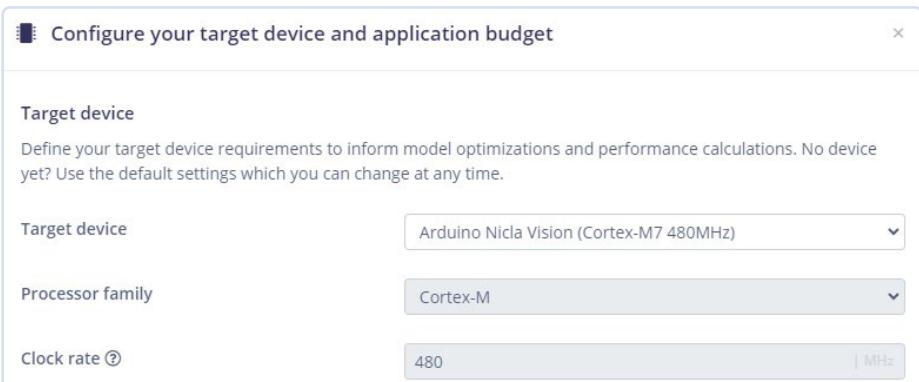


Figure 6. Configuration du pipeline d'entraînement en sélectionnant le cœur Cortex M7 du Nicla Vision.



Figure 7. L'explorateur de caractéristiques visuelles montre des classes bien séparées

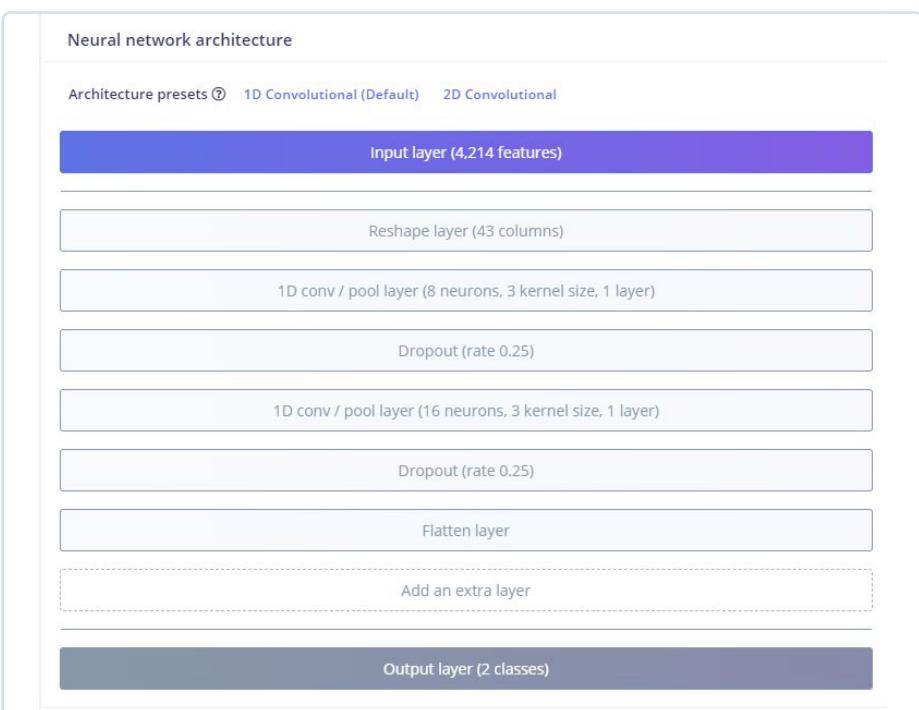


Figure 8. Architecture du réseau neuronal utilisé.

deux classes distinctes de sons, nous avons choisi un modèle *Classifieur* pour effectuer une classification binaire entre le bruit de fond et les moustiques. Ce type de modèle n'est pas entraîné directement sur l'entrée audio brute, mais plutôt sur des caractéristiques extraites grâce à des techniques de traitement numérique du signal (DSP) est utilisée pour dériver ces caractéristiques et les introduire dans le modèle. Edge Impulse fournit des blocs DSP préconstruits adaptés à divers types de traitement audio (**figure 4**) et optimisés pour le déploiement sur microcontrôleurs de la série Arduino Pro.

Dans notre projet, nous avons utilisé le Mel-filterbank Energy Spectrogram, qui convient parfaitement à l'extraction de caractéristiques non vocales telles que les bruits émis par les moustiques. À noter que l'architecture du modèle de classificateur que nous avons choisi ressemble à celle utilisée pour la classification d'images, et en utilisant les spectrogrammes, nous convertissons en fait l'audio en une image 2D (**figure 5**).

Edge Impulse prend en charge le traitement DSP automatiquement en arrière-plan, à la fois pendant l'entraînement du modèle et lors du déploiement au moment de l'exécution. Nous avons configuré cela dans le pipeline d'entraînement en sélectionnant le cœur Cortex M7 de Nicla Vision (**figure 6**). Cela a permis à Edge Impulse de configurer les optimisations requises pour la carte Arduino, assurant ainsi que le code généré soit adapté en termes de mémoire et capable de réaliser la classification en temps réel avec une faible latence.

Edge Impulse offre un explorateur visuel de caractéristiques, qui utilise des techniques de réduction de dimensionnalité pour illustrer les relations entre les caractéristiques et comment elles sont regroupées par classe. Les deux classes, représentées dans la **figure 7** par différentes couleurs, sont clairement distinctes et bien séparées, ce qui nous a confortés dans la suite du processus sans nécessité de collecter davantage de données. Le temps d'exécution du code DSP a été mesuré à 181 ms sur la Nicla Vision, un résultat rapide pour un microcontrôleur et qui répond aux exigences de temps réel du projet, avec une utilisation de la mémoire vive modérée de 41 Ko. Nous étions prêts à finaliser la sélection et l'entraînement du modèle.

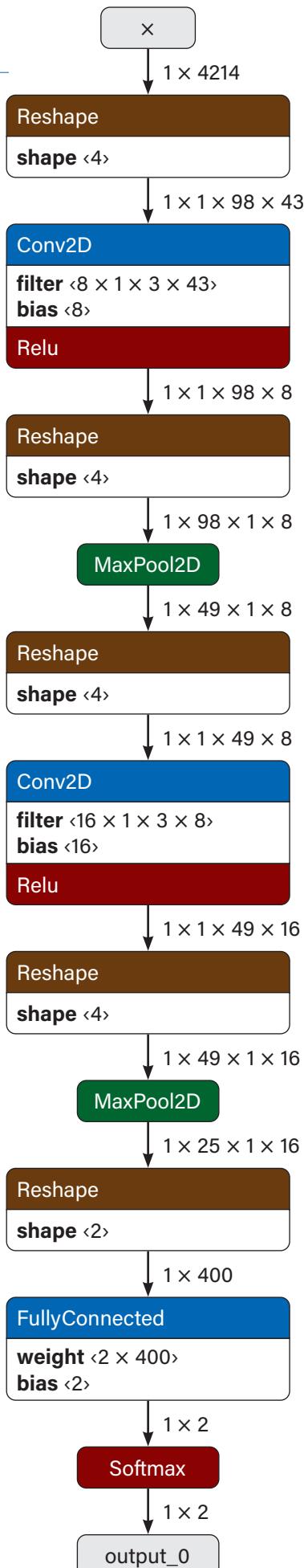


Figure 9. Schéma du réseau simple de type feed-forward.

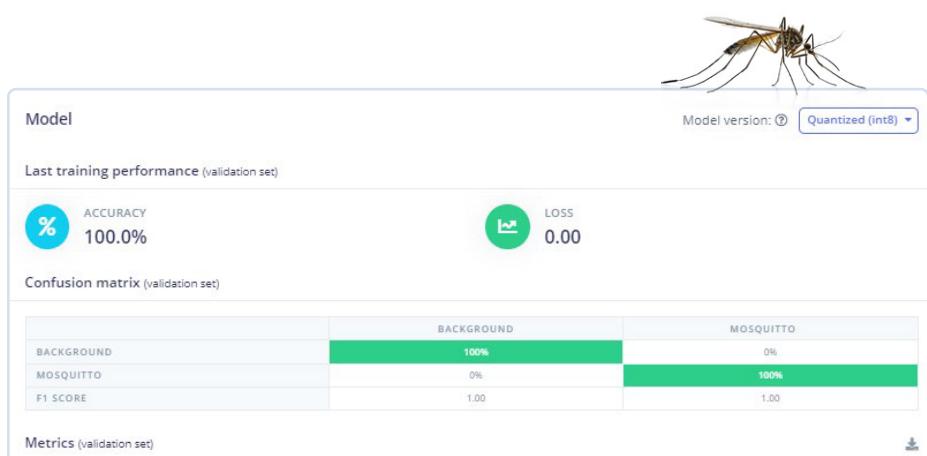


Figure 10. Le modèle a atteint une précision de 100 % et un score F1 de 1,00, classant avec succès à la fois le bruit de fond et les sons de moustiques de l'ensemble de test.



Figure 11. Les optimisations du compilateur EON d'Edge Impulse ont réduit l'utilisation de la RAM pendant l'exécution du modèle.

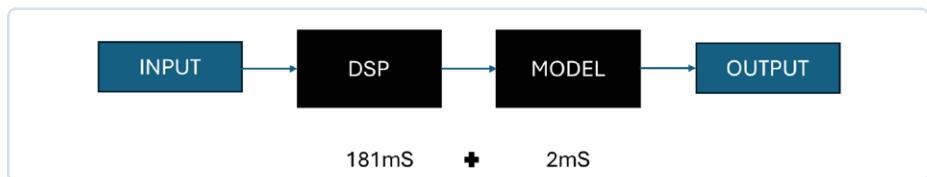


Figure 12. Le bloc DSP s'exécute en temps réel, avec une latence totale de 183 ms par inférence sur le Nicla Vision.

L'architecture du modèle par défaut fournie par Edge Impulse consiste en 2 couches de convolution 1D. Nous avons décidé d'opter pour ce mode (**figure 8**), car il s'est avéré idéal pour notre application. Graphiquement, ce modèle est un simple réseau feed-forward, comme présenté à la **figure 9**. Edge Impulse en collaboration avec Arduino, a facilité et simplifié le processus de construction de modèles, offrant aux ingénieurs un accès simplifié aux résultats - en ligne avec la philosophie d'Arduino. Le résultat a été un modèle qui a fonctionné avec une précision de 100 % (**figure 10**) et un score F1 de 1,00 avec un ensemble de données relativement petit, classant avec précision à la fois le bruit de fond et les sons de moustiques de l'ensemble de test.

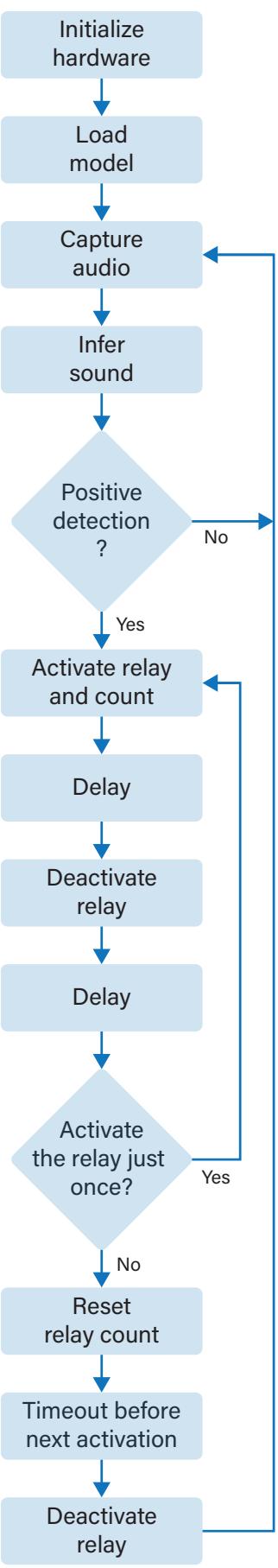
Les optimisations fournies par le compilateur EON d'Edge Impulse ont également réduit l'utilisation de la RAM du modèle à seulement 11,1 K lors de l'exécution et l'utilisation de la ROM à 32,6 K pour le stockage des poids et

du graphe d'exécution, comme le montre la **figure 11**.

En exécutant notre modèle sur la vision Nicla, nous pouvons atteindre une latence totale de 183 ms par inférence (**figure 12**). Cela offre également assez de temps pour que la machine à états commande l'actionneur, détail que nous aborderons dans la section suivante. Avec 2 MB de Flash et 1 MB de RAM partagés entre les coeurs, le Nicla Vision gère aisément ces opérations, ce qui ouvre des perspectives pour l'ajout d'autres fonctionnalités futures. Cette partie centrale du problème étant résolue, nous avons procédé à la construction finale.

## Réalisation

La réalisation était assez simple en termes de connexions : nous avons utilisé l'Arduino Nicla Vision connecté à un ordinateur via USB pour alimenter la carte et surveiller les résultats du moteur d'inférence, une carte de relais abordable activée par des signaux de



240556-014

Figure 13. Organigramme illustrant le processus.

3,3 V, et un humidificateur. Le diagramme de la **figure 13** illustre cette configuration. Le code (lien de téléchargement [3]) a été généré à l'aide de Edge Impulse. Le processus peut être expliqué comme suit :

➤ Dans Edge Impulse, la base de données de sons de moustiques [4] a été téléchargée, y compris les enregistrements de bruit de fond, dans le but de construire un classificateur binaire pour détecter la présence de moustiques. Les signaux ont été enregistrés à une fréquence d'échantillonnage de 44 100 Hz. Une fenêtre d'échantillonnage

de 1000 ms a été configurée avec un décalage de fenêtre de 500 ms.

- Pour le prétraitement, le bloc Audio (MFE) (EdgelImpulse) a été utilisé, qui extrait des caractéristiques du Mel Spectrogram en intégrant les domaines temporel et fréquentiel, en mettant en évidence les basses fréquences qui sont les plus reconnaissables pour l'oreille humaine.
- Un réseau neuronal convolutionnel 1D a été construit pour la classification binaire pendant la formation, avec un taux d'abandon de 0,25.



Figure 14. Le modèle a atteint une précision de 100 % et un score F1 parfait de 1,00, avec des groupes bien séparés pour les sons de moustiques et les sons de fond, ce qui le rend prêt pour une utilisation pratique.

Les résultats de l'entraînement de notre modèle sont incroyablement encourageants (**figure 14**). Nous avons atteint une précision de 100%, ce qui signifie que le modèle a parfaitement identifié chaque son de moustique par rapport au bruit de fond dans l'ensemble de validation, sans aucune erreur. La valeur de perte est de 0,00, indiquant aucune erreur significative dans les prédictions du modèle. Les mesures telles que la précision, le rappel et le score F1 ont toutes atteint une valeur parfaite de 1,00, ce qui montre que le modèle est non seulement précis, mais aussi constamment fiable. L'explorateur visuel de données en bas de page le confirme, en montrant des groupes clairs et bien séparés pour les bruits de moustiques et les bruits de fond, confirmant la capacité du modèle à distinguer les deux avec une grande précision. Ceci est crucial pour une détection efficace des moustiques dans le monde réel avec l'Arduino Nicla Vision. Globalement, cette performance suggère que notre modèle est prêt à être déployé dans des contextes pratiques, où il peut aider à détecter efficacement les moustiques sur la base de leurs signatures sonores.

Vous pouvez en savoir plus sur le processus de programmation de l'Arduino Nicla Vision

avec Edge Impulse sur le site Web Arduino Docs [5]. Le code provenant d'Edge Impulse s'exécute comme thread principal dans le processeur, tandis que tout le reste du code dépend des décisions du moteur d'inférence. L'humidificateur ne se déclenchera qu'en cas de détection positive de moustiques. Une minuterie sera activée pour stopper l'humidificateur après un certain temps, et une seconde minuterie empêchera l'appareil de fonctionner en continu si le son persiste. Cette approche vise à éviter une diffusion excessive de citronnelle dans l'air en peu de temps.

Comme notre focus était sur la détection, nous avons activé l'appareil avec tout son ressemblant à celui d'un moustique. Nous avons réalisé des tests avec différents échantillons présents dans notre base de données, tous couronnés de succès.

### Perspectives

La démonstration de faisabilité que nous présentons offre une solution rapide et économique pour la détection des moustiques. Ce même ensemble de données pourrait également servir à entraîner le système à reconnaître des espèces spécifiques de moustiques ou même à distinguer leur sexe, sachant que

seules les femelles sont vectrices de maladies, les mâles ne se nourrissant pas de sang. Il y a donc encore des améliorations à apporter au moteur d'inférence pour affiner la classification. Nous invitons les lecteurs d'Elektor à expérimenter et à partager leurs optimisations, contribuant ainsi à enrichir la communauté des développeurs et passionnés de tinyML ! 

240556-04

### Questions ou commentaires ?

Contactez Elektor (redaction@elektor.fr).



### Produits

› **Arduino Pro Nicla Vision**  
[www.elektor.fr/20152](http://www.elektor.fr/20152)

› **Elektor Special: Guest-Edited by Arduino**  
[www.elektor.com/EP-0518](http://www.elektor.com/EP-0518)

› **Elektor Special: Arduino Guest Edition 2022 PDF**  
[www.elektor.com/ED-0518](http://www.elektor.com/ED-0518)



### LIENS

- [1] PAHO Report, September 2024 : <https://tinyurl.com/PAHO-report>
- [2] Arduino Nicla Vision : <https://www.arduino.cc/pro/hardware-product-nicla-vision/>
- [3] Téléchargement du code : [https://github.com/dcuartielles/ictp\\_brazil\\_24](https://github.com/dcuartielles/ictp_brazil_24)
- [4] Mosquito sound database: <https://github.com/HumBug-Mosquito/ZooniverseData>
- [5] S. Romero, "Image Classification with Edge Impulse", Arduino Docs, September 2024 :  
<https://docs.arduino.cc/tutorials/nicla-vision/image-classification/>