Salma Aghrizen

Edge Impulse Utilisation, Génération du code.

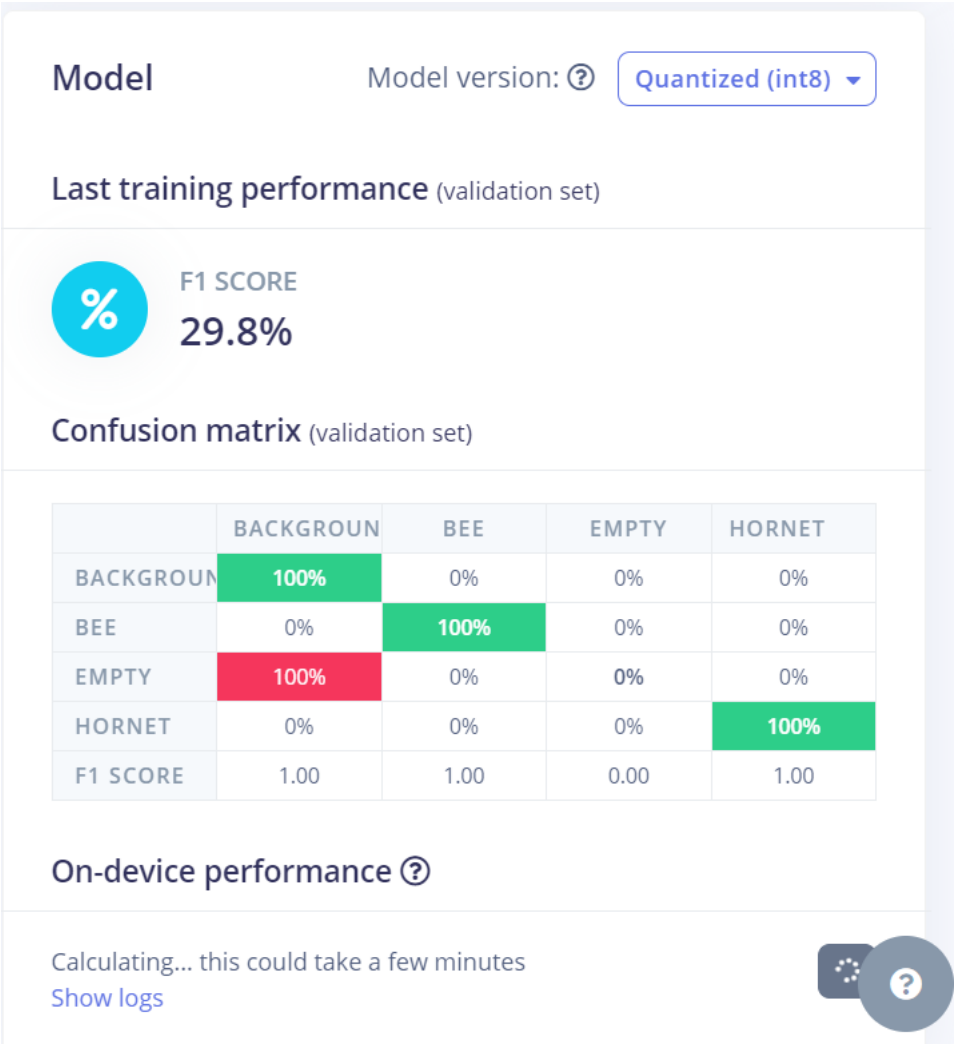
L'objectif :

Edge Impulse est un outil de développement qui permet de construire un programme permettant d'effectuer des différentes fonctionnalités en fonction du besoin.

Dans notre cas, on vise d'implémenter un programme permettant de distinguer entre un frelon et une abeille, et surtout de détecter les frelons parce que notre objectif c'est de les tuer.

Edge Impulse utilisation :

- La première étape c'est de créer un projet sur l'outil.
- Puis, il faut ajouter la base des données sur notre projet. Une fois que c'est bon, on commence à modifier les labels de chaque image : c'est-à-dire, d'encadrer les frelons, les abeilles et le background dans chaque donnée.
- Une fois que c'est bon, il faut cliquer sur Impulse Design, on choisit une détection des images en GrayScale, après la création de l'impulse bien sûr (Object Detection). On choisit un cycle de training de 60, et un learning rate de 0,006.
- Après ça, dans le deployment, on choisit Arduino, et on fait le build du modèle.



- Puis on le test.

Remarques :

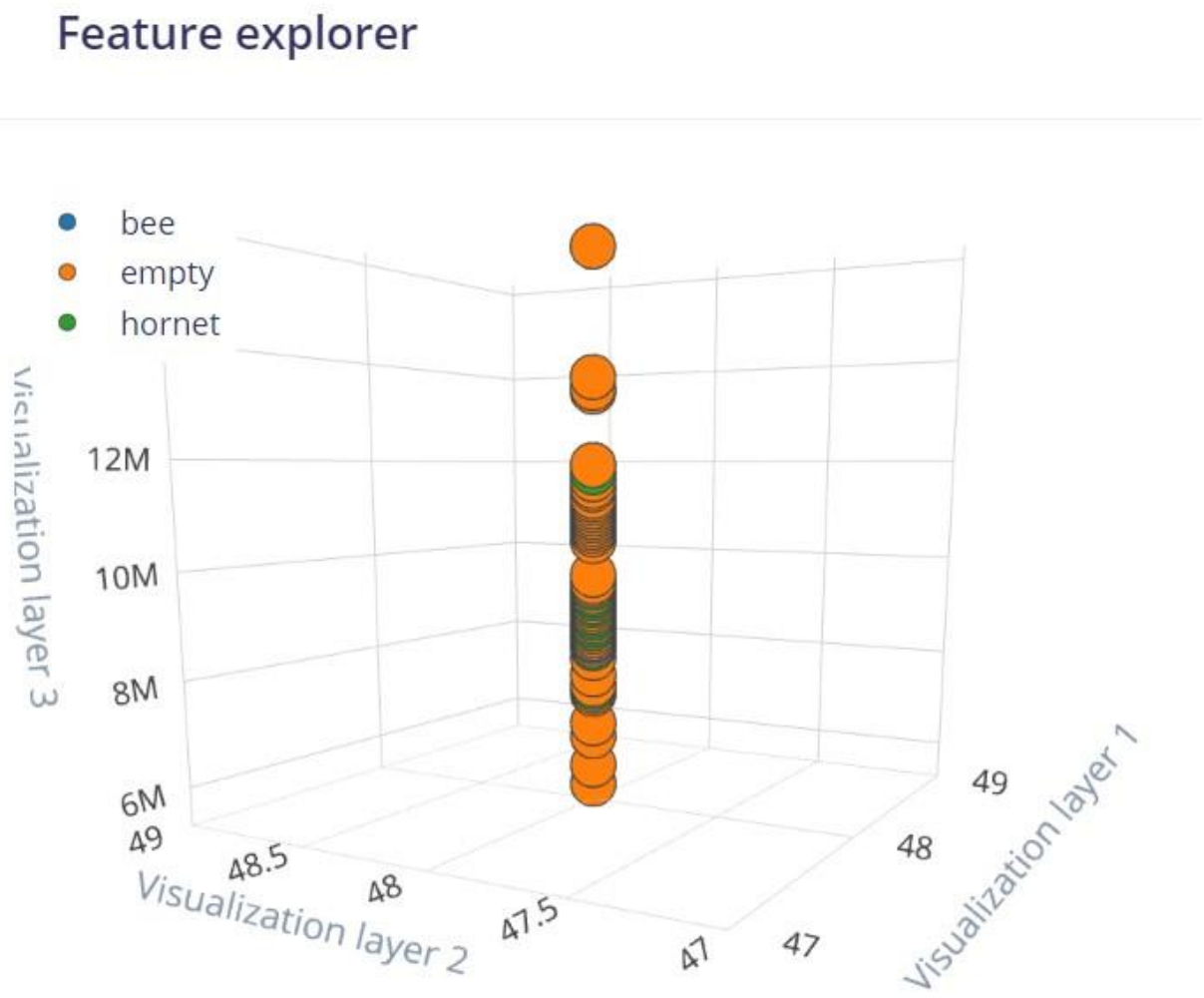
- Par défaut, l'outil Edge Impulse, va générer un programme qui peut marcher sur deux types des caméras : AI_THINKER et ESP_EYE, pour le modèle de WROVER, il faut le redéfinit et ajouter les pins de la carte aussi.
- Si on veut travailler par exemple avec le modèle AI_THINKER, il faut bien le choisir en définie et de mettre les autres en commentaire :

```
//#define CAMERA_MODEL_ESP_EYE // Has PSRAM
#define CAMERA_MODEL_AI_THINKER // Has PSRAM
```

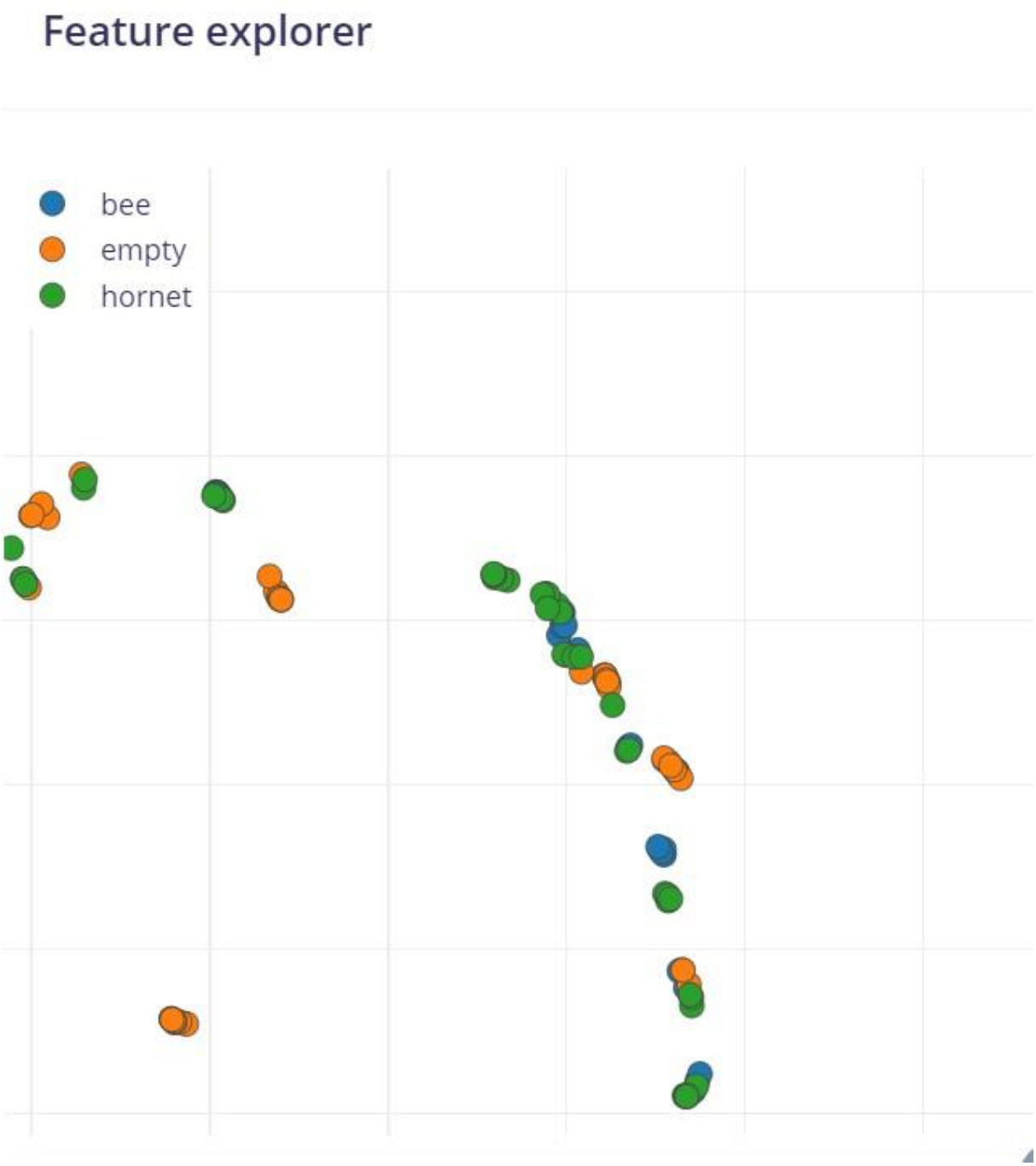
- Concernant les modèles de Edge Impulse qu'on puisse utiliser, il y'en a deux :
    - Modèle par défaut de training
    - Modèle FOMO (First Object Most Object).

Exemples :

- Probabilité des classes sur Edge Impulse :



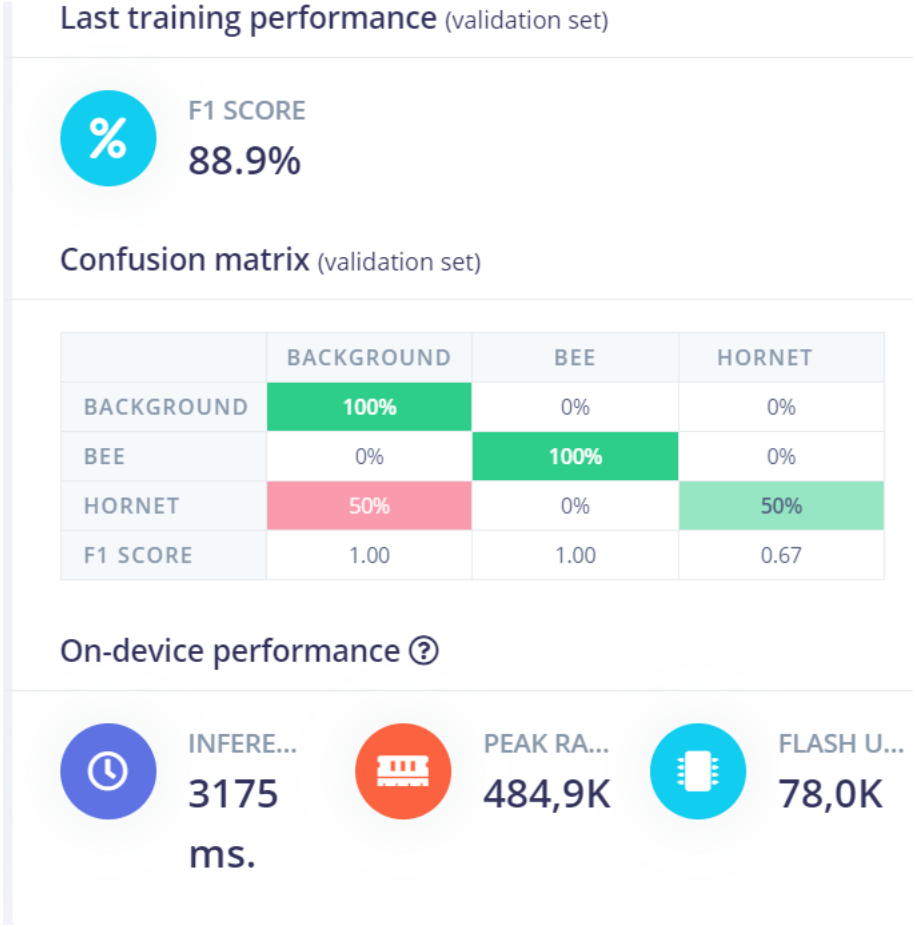
-



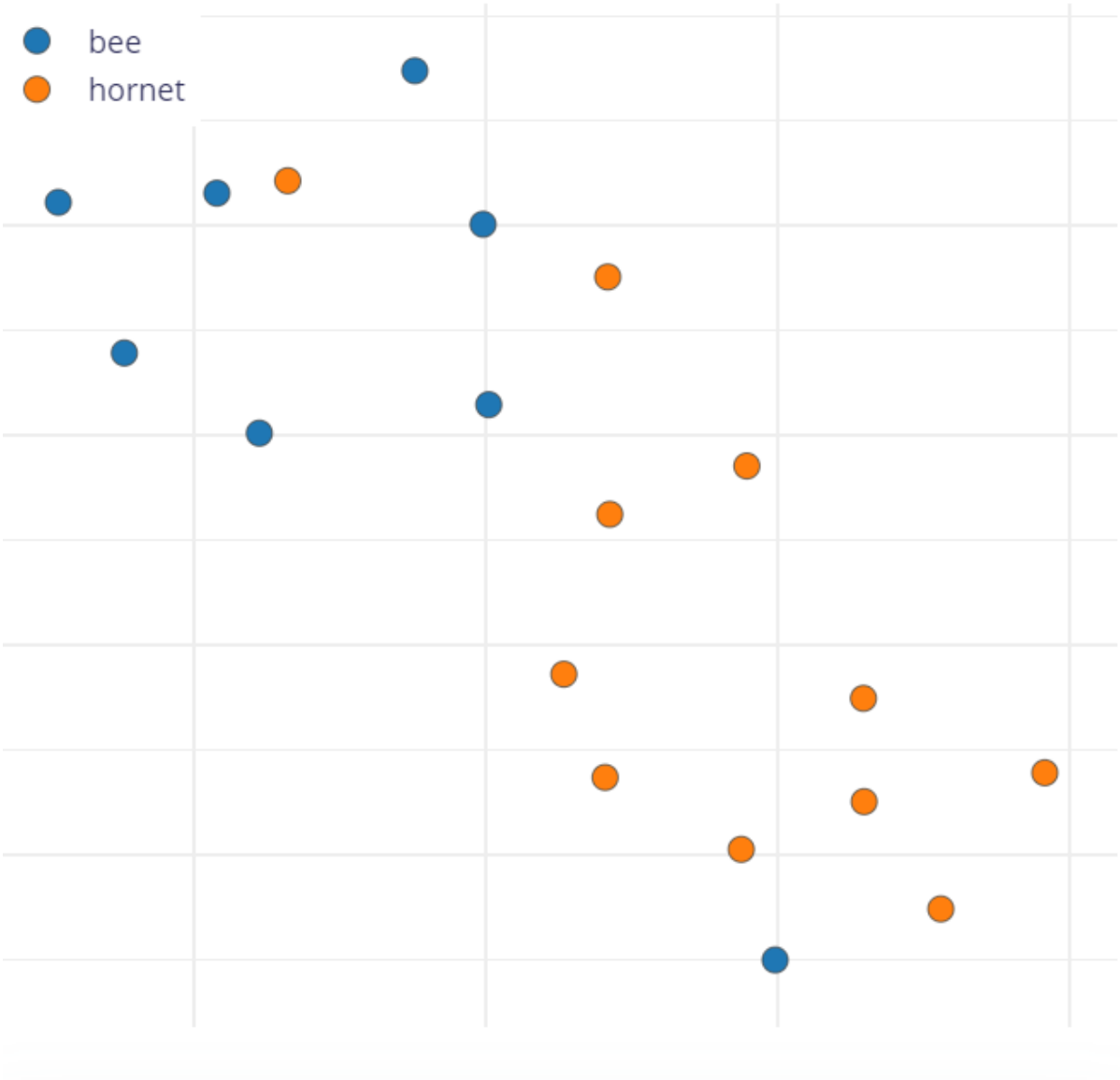
-

- Empty sera considéré comme Background. L'outil va directement le considérer comme background lors de l'étude des classes définies et de la précision des data.

- Cette version, en sélectionnant le background, n'était pas une très bonne solution, surtout que la data que j'ai utilisé avait souvent des backgrounds au lieu de frelon, abeille. Donc, j'ai réduit dans une autre version la data et j'ai gardé que ceux qui sont bien net, en faisant que deux classes : hornet et bee.
- La performance était mieux par rapport au cas précédant :



- Les résultats statistiques de la probabilité de chacune des deux classes :


bee
hornet