

Présentation JRAF - Wildcount

Développement d'un piège photo pour le comptage d'animaux dans les espaces naturels



Sommaire

1. Contexte du projet
2. Etat de l'art
3. Conception du prototype
4. Résultats et suite du projet

1

Contexte du projet

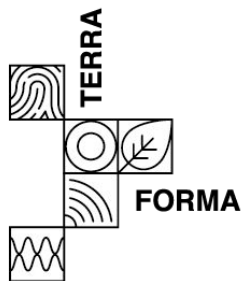
Partenaires du projet



Laboratoire d'informatique de Grenoble



Université Grenoble Alpes



Projet Terra Forma CNRS

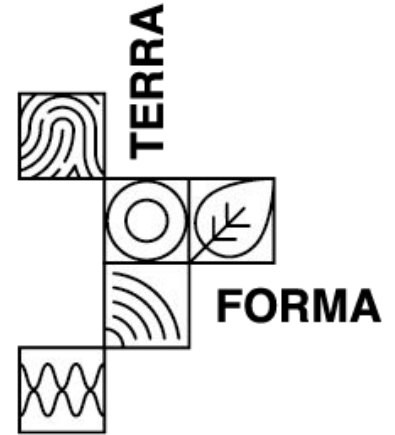


Parc national des Ecrins

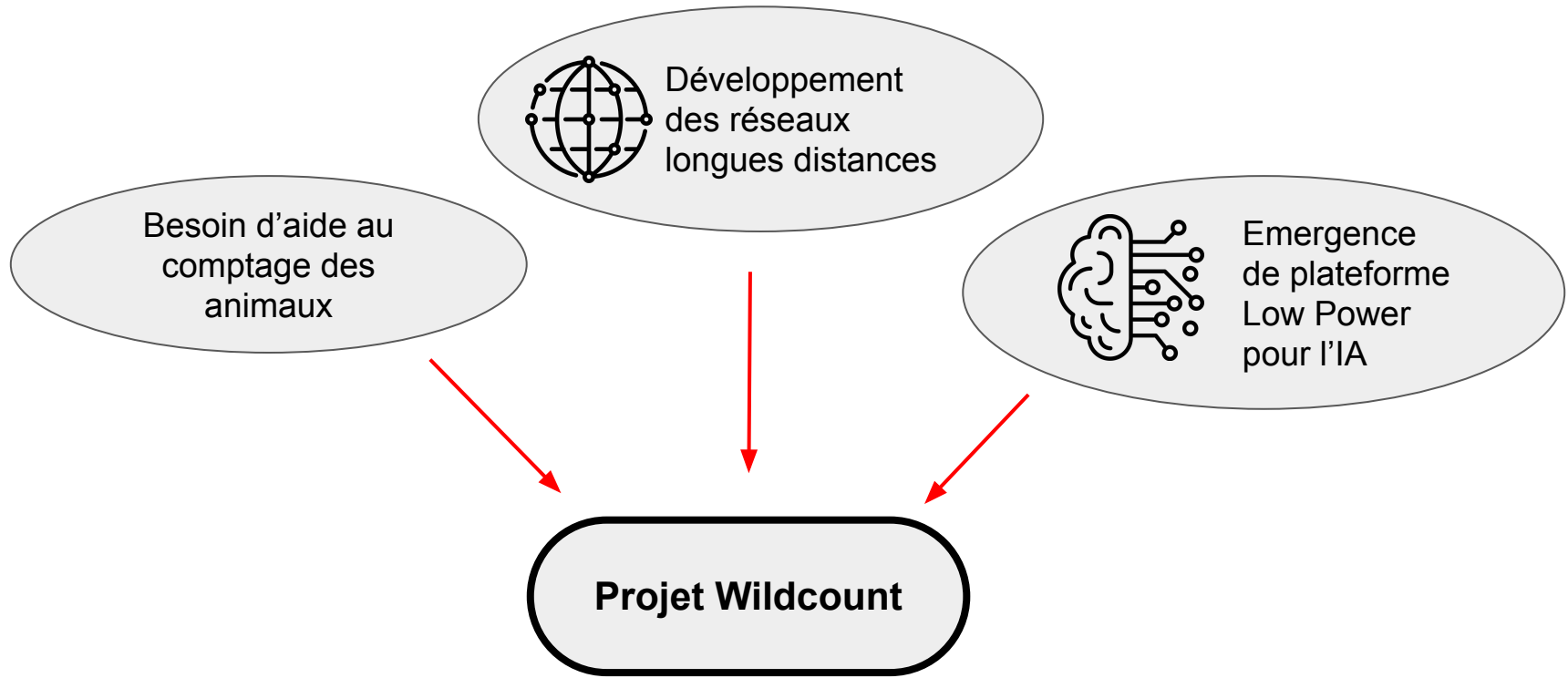
Terra Forma (Equipex 2021-2029)

<https://terra-forma.cnrs.fr/>

- Réseaux IoT pour observer les transitions dans les espaces naturels sensibles.
- Technologies : LPWAN, LPGAN (SatIoT), AI @ Edge, AI @ Extreme Edge
- Zones atelier
 - dont Col du Lautaret (SAJF) et Parc National des Ecrins (PNE)



Naissance du projet



Buts du projet

Collecte de données

- Réaliser du comptage d'animaux dans des zones naturelles protégées
- Transmission des données vers un serveur

Reconnaissance de la faune

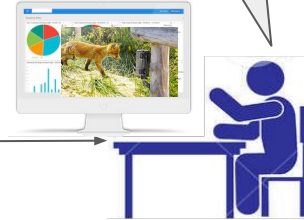
- Utilisation de réseaux de neurones
- Entraînement sur des images prises dans les alpes

Spécifications techniques

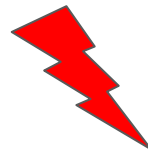
- Très faible consommation d'énergie, utilisation sur batterie sur de longues périodes
- Faible puissance de calcul
- Liaison longue portée sur des zones blanches (pas de 4G)

Comptage des animaux

piège photo (*hunting cam*)

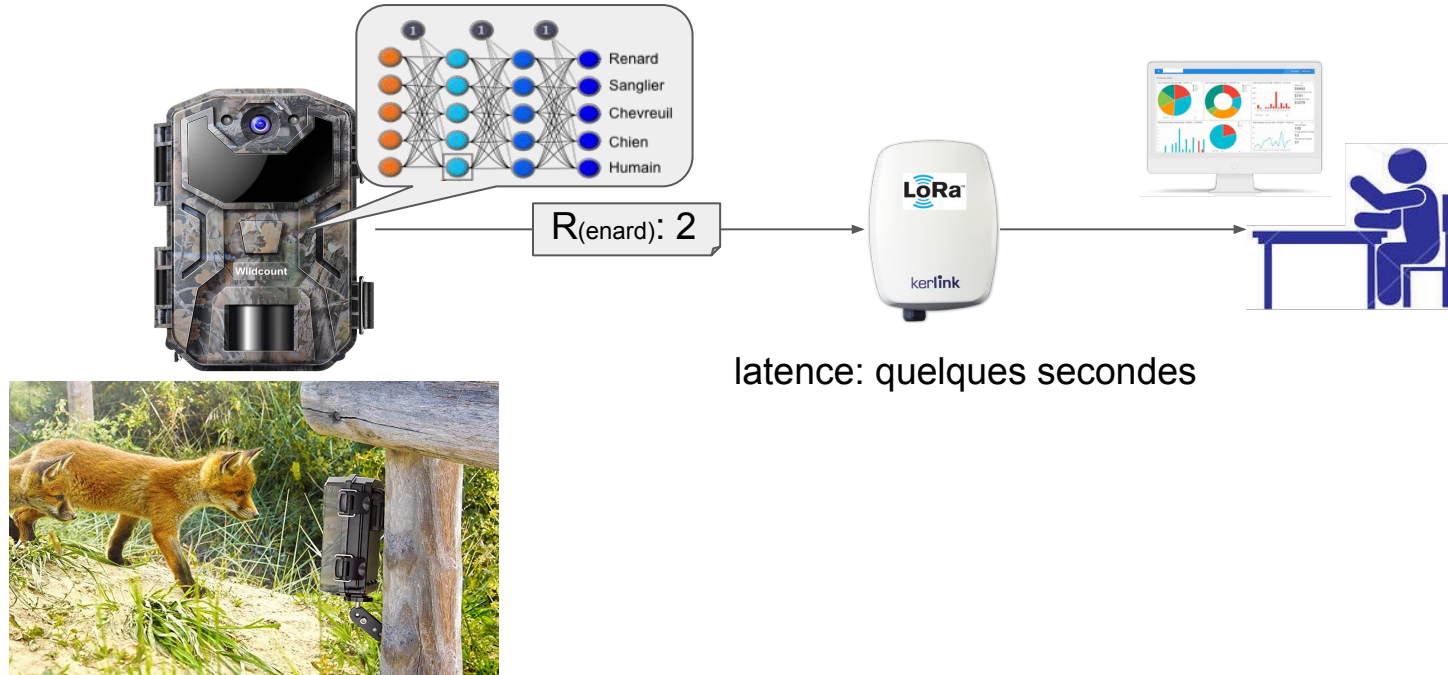


latence: ~1-2 mois



risque de vol/destruction de la caméra →
perte de 1-2 mois de comptage

Comptage des animaux



Historique du projet

- Projet porté par des étudiants
- But de recherche

2020

Début du projet Wildcount

- Utilisation de la carte Greenwave GAP8POC pour le premier prototype
- Prise en main de la carte

2021

Réalisation d'un prototype fonctionnel

- Création d'une banque d'image
- Définition de l'architecture software
- Création du premier prototype

2022

Début de mon stage

- Prototype fonctionnel (*Migration vers Spresense*)
- pas testé sur le terrain
- Besoin de rajouter des fonctionnalités

2

Etat de l'art

Réseau LoRa

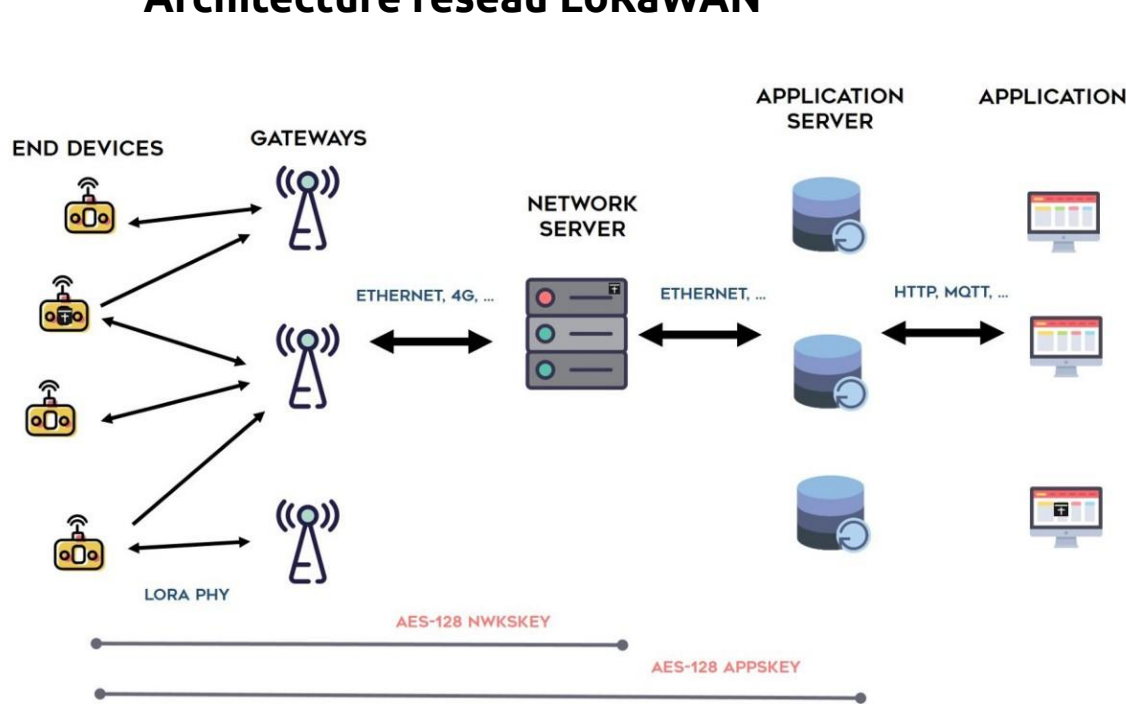
Caractéristique réseau LoRaWAN



- Modulation RF LoRa
 - Bande ISM (863-870 Mhz en Europe)
 - 2x1% de *duty-cycle*
 - 14 dBm max (25 mW)
 - portée théorique:
 - 5 km en environnement urbain
 - 16 km en zone rurale
 - Débit variable : 260 bit/s jusqu'à 27 Kbit/s
- Réseau LoRaWAN
 - identifie les terminaux
 - déduplique les messages recus par plusieurs stations
 - sécurise les communications
 - adapte le débit en fonction de la qualité du lien

Infrastructure LoRaWan

Architecture réseau LoRaWAN



- Possibilité de déployer des réseaux privés (comme CampusIoT)
- Peut fonctionner dans des zones non couvertes (ie blanches) par les réseaux 2G/3G/4G

Low-Power and Long Range WAN

Example: Elsys ELT2 (LoRa endpoint with temperature sensor)

- Battery lifetime from **6** to **18** years (1 temp. msg/hour)



Sample time	Sensor	Battery capacity	Battery performance
<input type="text" value="3600"/>	<input type="text" value="ELT2 HP"/>	<input type="text" value="2700"/>	<input type="text" value="80"/>
Seconds	Select Elsys sensor	Capacity(mAh)	Performance(%)

Spreading factor

SF7 SF8 SF9 SF10 SF11 SF12

Result:

The battery will last for **6.2** years*. The sensor will draw **40uA** and **351mAh** in one year.

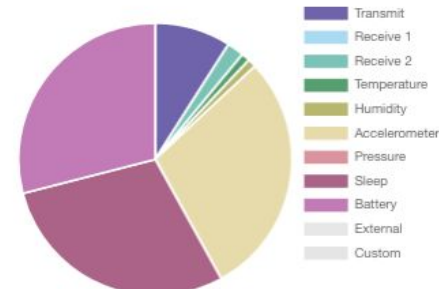
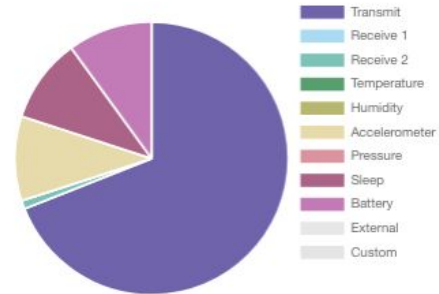
Sample time	Sensor	Battery capacity	Battery performance
<input type="text" value="3600"/>	<input type="text" value="ELT2 HP"/>	<input type="text" value="2700"/>	<input type="text" value="80"/>
Seconds	Select Elsys sensor	Capacity(mAh)	Performance(%)

Spreading factor

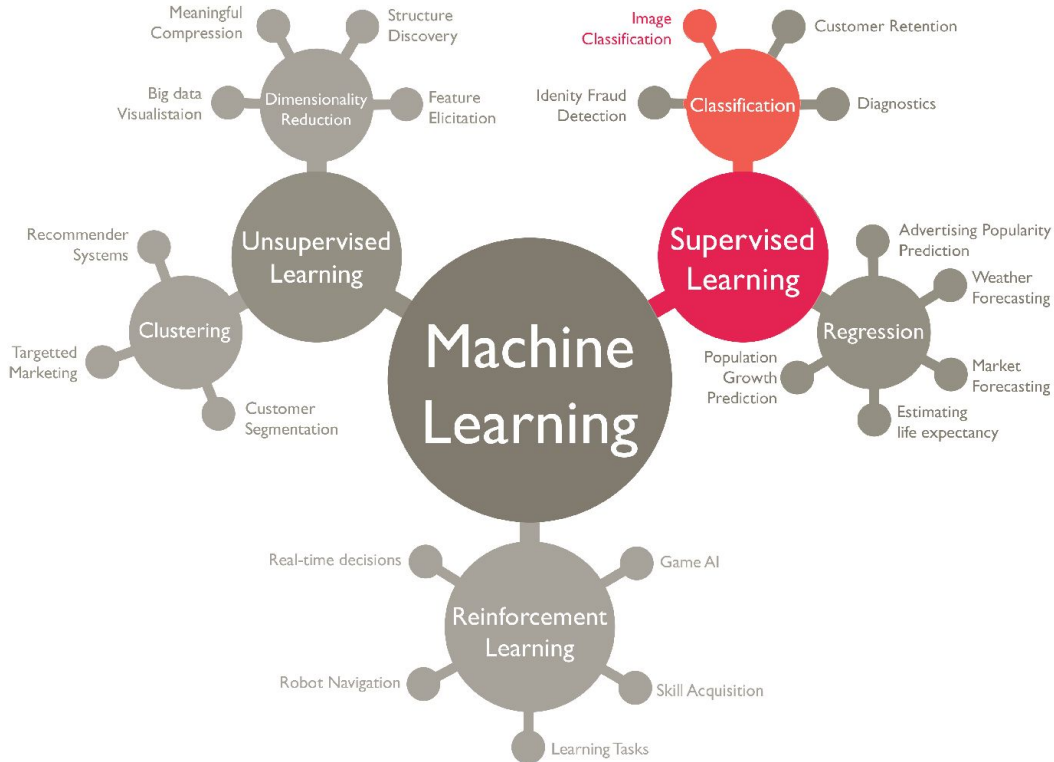
SF7 SF8 SF9 SF10 SF11 SF12

Result:

The battery will last for **18** years*. The sensor will draw **14uA** and **120mAh** in one year.



Réseaux de neurone



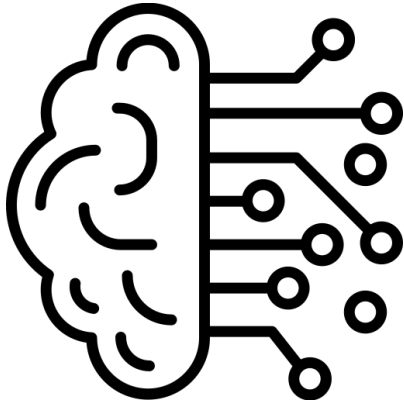
Différentes types de réseaux de neurones

⇒ **Classificateur d'image**

→ Set d'image fourni par le PNE Valbonnais

Contraintes du système embarqué

IA embarquée
(*Tiny ML*)



Contraintes importantes

- Capacité de calcul limité
- Espace de stockage restreint

≠

Machine learning coûteux en ressources

- Opérations sur des flottants
- Stockage des poids assez lourd

Architecture Serveur - IA

GPU Processors

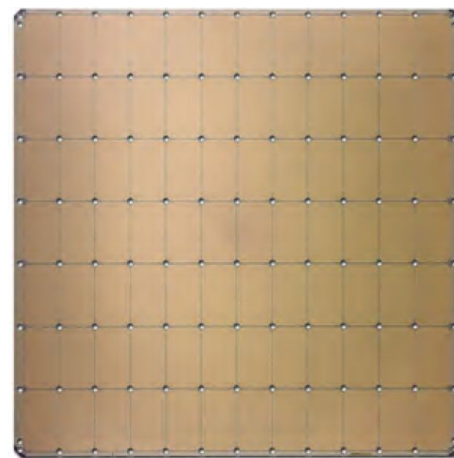
- NVidia GPU A100
- ...

AI Processors

- Google TPU v4 (275 TFLOPS FP16 @ 170 W)
- Huawei IA Ascend 910 (320 TFLOPS FP16, 640 TOPS INT8 @ 310 W)
- ...
- Celebras Wafer Scale Engine (@ **20kW**)

Others

- Processing-in-Memory (UpMem) → DRAM + DPU



Cerebras WSE-2
2.6 Trillion Transistors
46,225 mm² Silicon



Largest GPU
54.2 Billion Transistors
826 mm² Silicon



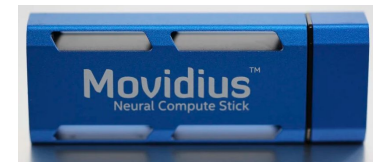
Architecture Desktop - IA

GPU Processors

- Jetson Nano : ? TOPS @ 12.5W

AI Processors

- Google Coral Edge TPU : 4 TOPS @ 2 W
 - MobileNet V2 model : 400 images per second
- Intel Movidius Myriad X VPU : 4 TOPS @ 1.5 W



Architecture Extreme Edge - IA

ML and DL on low-power (~ 1 mW) MPUs and DSPs
Challenges for inference and for On Device Learning (ODL).

- Fragmented MCU market (heterogeneity)
 - ISAs (ARM Cortex M, RISC-V, ESP32, x86 ...)
 - w/o extensions (ARM CMSIS-NN ...)
 - required specific optimizations
- SRAM (64KB to 1.5MB), FlashRAM (128KB to 8MB), w/o FPU, w/o File System
- Cost by unit (< 10 USD)
- standard tools and frameworks
- benchmarks for comparison

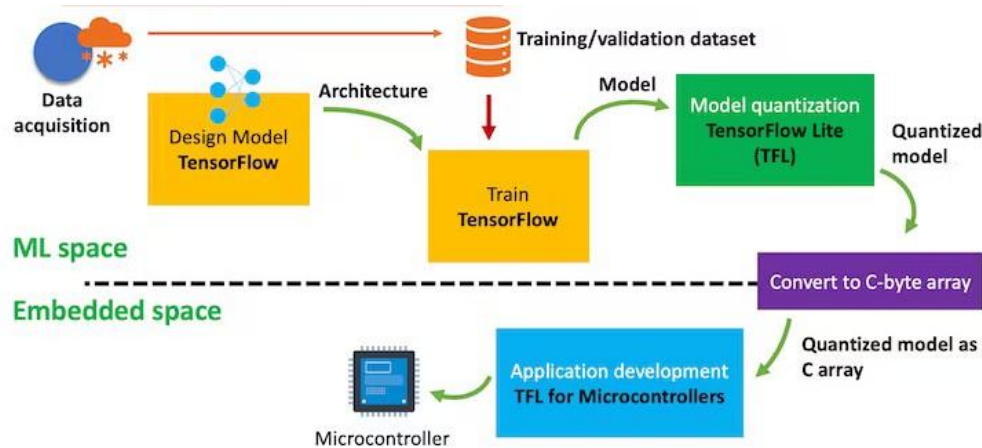


Conçu pour des architectures embarquées

- Quantité de RAM limitée
- CPU Low Power
- pas forcement de FPU
- Pas forcement de système de fichier

TensorFlow lite micro

Leviers d'optimisation TinyML



Quantisation

Conversion des poids *float 32* \Rightarrow *int 8*

- \rightarrow Réduction par 4 du poids du modèle
- \rightarrow *Inférence plus rapide*

Réduction du ensemble d'opérateurs

1200 opérateurs \Rightarrow *quelques dizaines*

- \rightarrow Réduction du poids de l'environnement TensorFlow

Gestion de la mémoire

- \rightarrow Allocation d'une tensor arena en RAM
- \rightarrow Stockage des modèles sous forme de tableau

Etat de l'art

Architecture Software

 **EDGE IMPULSE**



TensorFlow

Réseau de neurones profond

 **LoRaWAN**®



 **ChirpStack**

Coeur de réseau LoRaWAN



ARDUINO

Environnement de développement

3

Conception du prototype

Architecture Hardware



Prototype

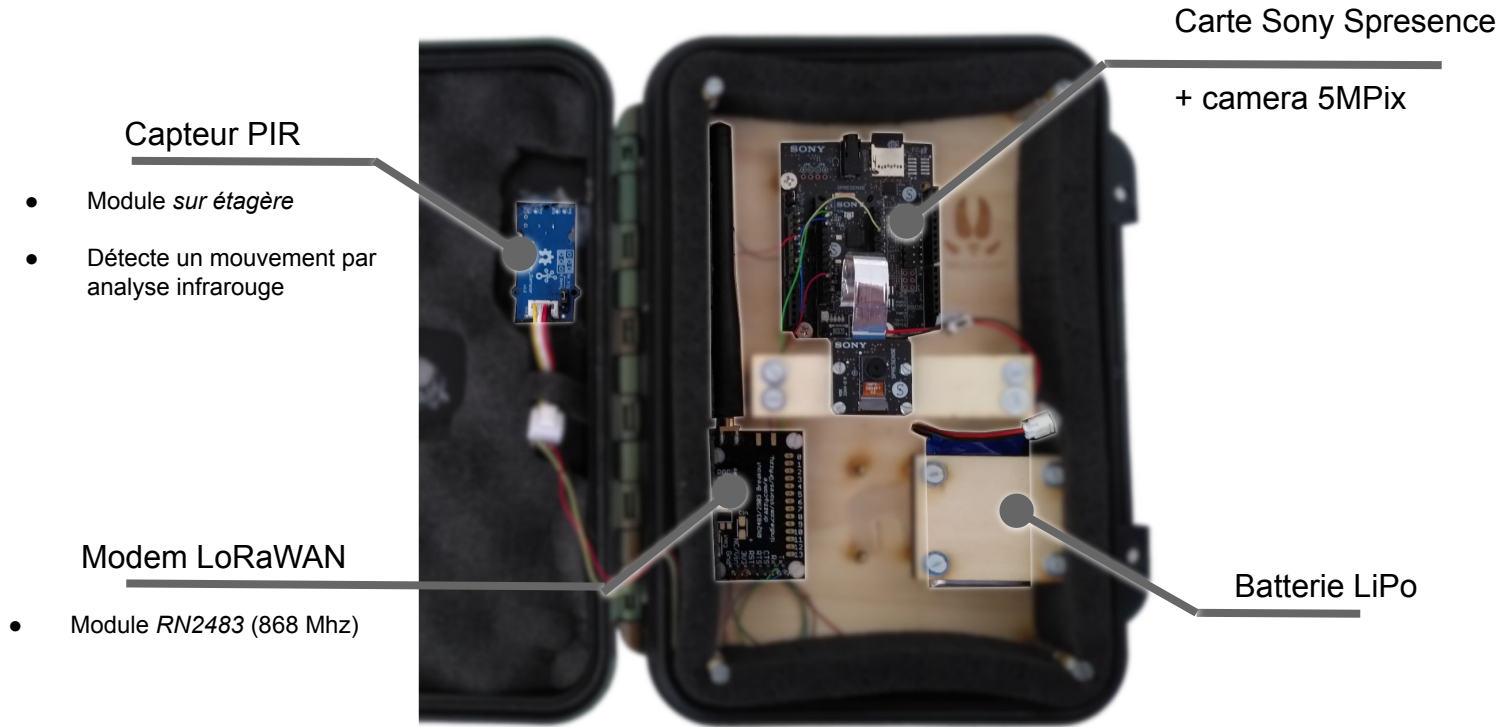
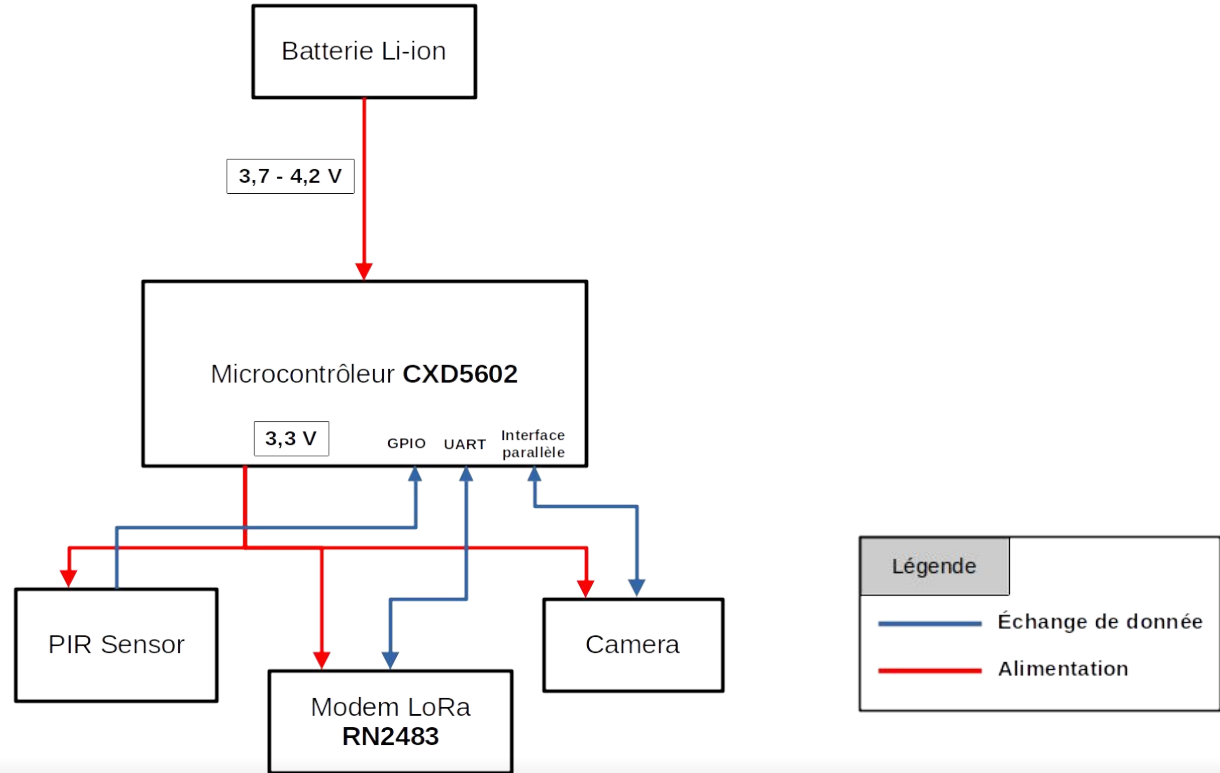
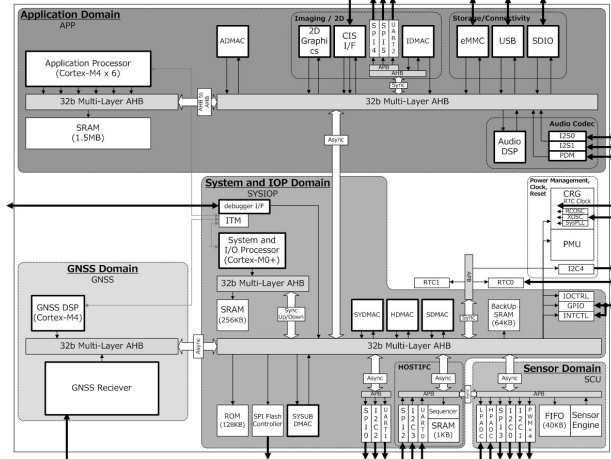
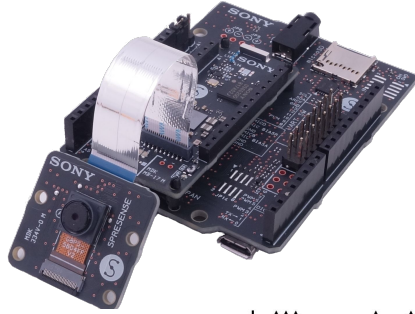


Diagramme hardware



Carte de developpement Spresense



- Développée par **Sony**
- 6 coeurs **Cortex-M4** (156 Mhz)
- **1536 Ko** de SRAM (Text, Bss, Heap)
- 8 Mo de Flash
- Environnement porté sur l'acquisition audio et **vidéo**
- Slot **carte SD**
- Plusieurs modes d'endormissements

Gestion de la veille

○ : Powered-On State
 × : Powered-Off State
 △ : Any State

CXD5247	CXD5602 Power Domain	Hardware Block(s)	Power Off	Deep Sleep	Cold Sleep	Hot Sleep	Run	
PMIC	PMU	RTC, DDC/LDO, Load Switch GPO, Charger				○		
		I/O control, RTC BackupSRAM(64KB)				○		
		SCU	SPI#3, I2C#0, I2C#1 PWM, LPADC, HPADC				△	
		CORE	SYS_RAM(256KB) GNSS_RAM(640KB)				○	
		SYSIOP	SYS_CPU, SYSIOP_BUS				○	
		SYSIOP_SUB	ROM, SPI-Flash Controller UART#1(DebugUART)	×	×		○	
		GNSS_ITP	GNSS				△	
		APP	APP_RAM(1.5MB), APP_BUS			×	○	
		APP_AUD	Audio SubSystem				△	
		APP_SUB	SPI#4, SPI#5, UART#2 USB, eMMC, SDIO				△	
	APP_DSP	APP_CPU x6				×	○	

- Middleware *Power Management*
- 3 modes d'endormissement
 - Hot sleep, **Cold sleep**, Deep sleep
 - ⇒ le mode Deep sleep ne permet pas le réveil par GPIO
- Pas de rétention de la RAM en Cold sleep
 - ⇒ stockage des variables nécessaires en Flash

Campus lot et gateway LoRa



The screenshot shows the ChirpStack web interface. The top navigation bar is blue with the ChirpStack logo and a search bar. The left sidebar contains a menu with the following items: Org. users, Service-profiles, Device-profiles, Gateways, Applications, and Multicast-groups. The main content area displays the breadcrumb path: Applications / WILDCOUNT / Devices / wildcount_proto_sdk. Below this, there are tabs for DETAILS, CONFIGURATION, KEYS (OTAA), ACTIVATION, DEVICE DATA, LORAWAN FRAMES, and FIRMWARE. The DETAILS tab is active, showing a table with the following information:

Details	
Name	wildcount_proto_sdk
Description	second prototype
Device-profile	OTAA_CLASS_A_WILDCOUNT

To the right of the details table is a Status section with the following information:

Status	
Last seen at	Mar 4, 2023 4:19 PM

A red DELETE button is visible in the top right corner of the main content area.

Architecture Software

OS Nuttx

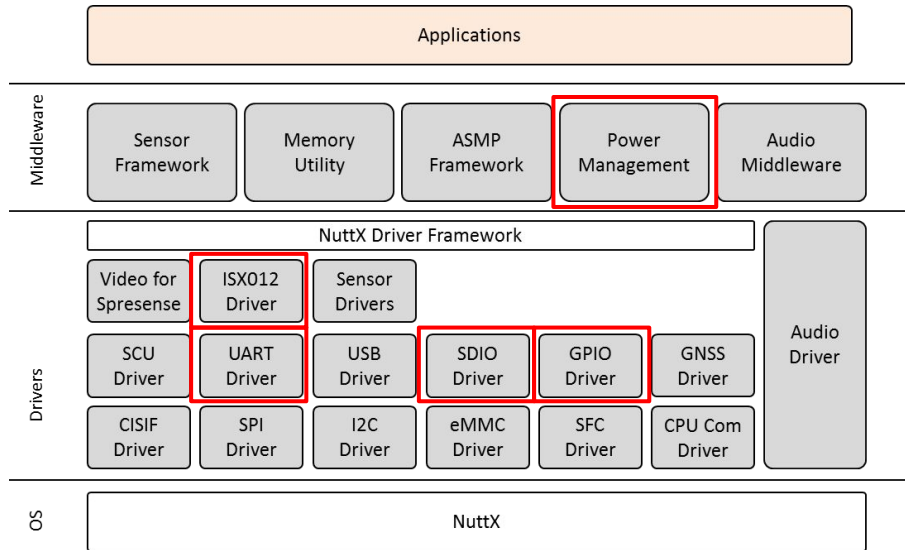
Présentation de Nuttx



- **OS temps réel**
 - système de fichiers
 - possibilité de monter des volumes externes
 - shell permettant de lancer des applications
 - multi-tasking
- **Léger et configurable**
- basé sur les standards de programmation **POSIX** et ANSI

SDK Sony

Présentation du SDK



- SDK basé sur Nuttx
- Middleware développés par Sony sur la couche des drivers Nuttx

Structuration du projet

Structure de l'application

Wildcount_main.c

Librairies

Gestion des périphériques

adc.c camera.c flash.c leds.c pir.c rn2483.c
rtc.c sd_card.c watchdog.c

Traitements des données

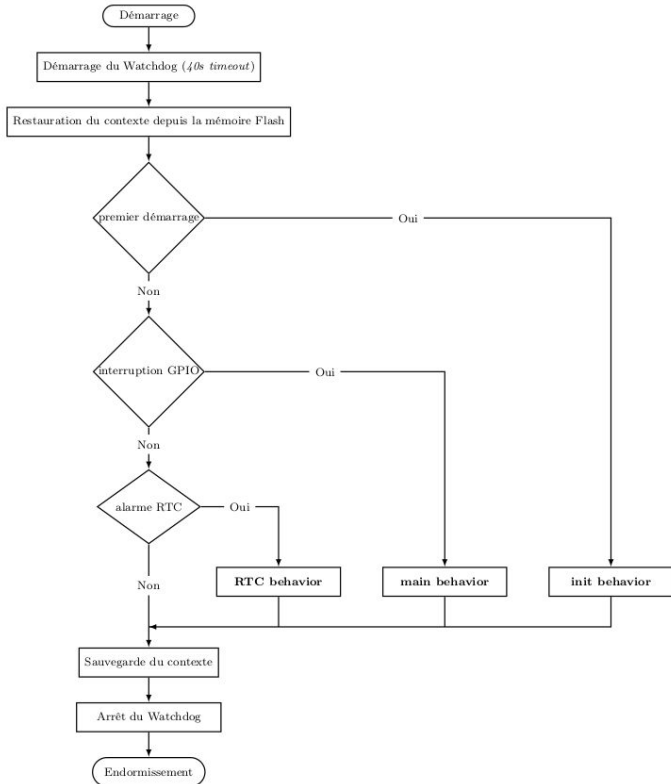
classifier.c datastruct.c graphic.c wildcount_dn_decode.c wildcount_up_encode.c

Configuration et Debug

config.c global.c debug.c

- Code écrit en **C**
- pas de multi-tasking
- programmation **séquentielle**

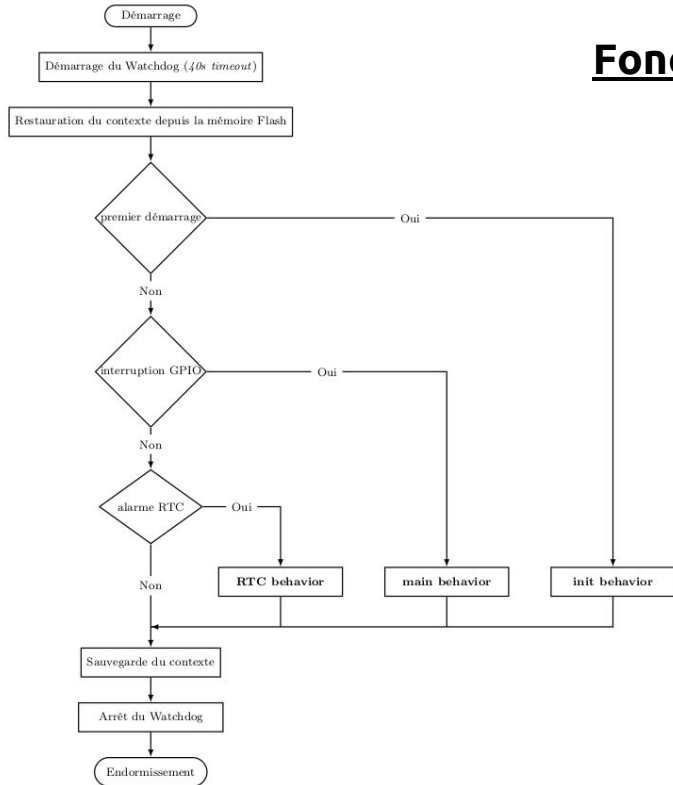
Algorithme du fonctionnement



3 réveils possibles

- Démarrage ⇒ **phase d'initialisation**
 - Appairage au réseau LoRa
 - Synchronisation de temps
- Déclenchement du PIR ⇒ **prise de photo et traitement de l'image**
 - Prise et stockage de la photo
 - Inférence sur le modèle chargé
 - Actualisation du compteur d'animaux
- Alarme RTC ⇒ **envoi de trames LoRa**
 - Envoi des compteurs d'animaux détectés
 - Envoi des informations de fonctionnement

Fonctionnement du programme



3 réveils possibles

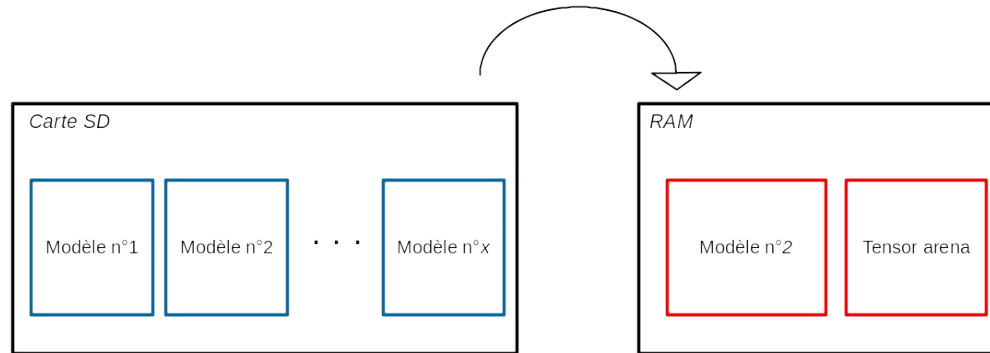
- Démarrage ⇒ **phase d'initialisation**
 - Appairage au réseau LoRa
 - Synchronisation de temps
- Déclenchement du PIR ⇒ **prise de photo et traitement de l'image**
 - Prise et stockage de la photo
 - **Inférence sur le modèle chargé**
 - Actualisation du compteur d'animaux
- Alarme RTC ⇒ **envoi de trames LoRa**
 - Envoi des compteurs d'animaux détectés
 - Envoi des informations de fonctionnement

Exécution des modèles (*classificateurs*)

multi-modèle

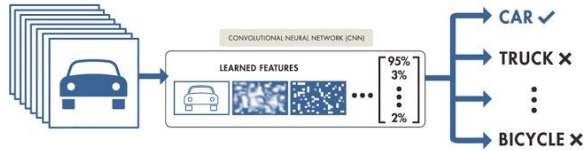
Programme

- Chargement du modèle en RAM
- Allocation de la tensor arena

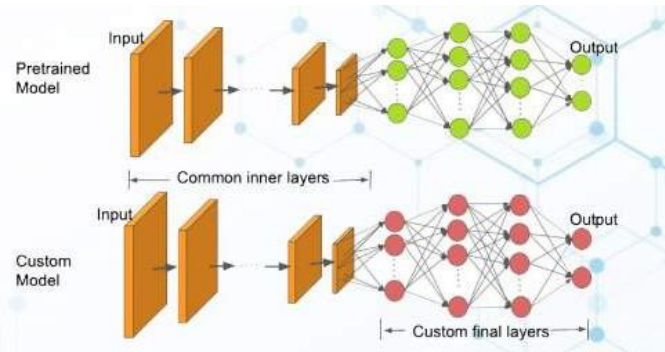
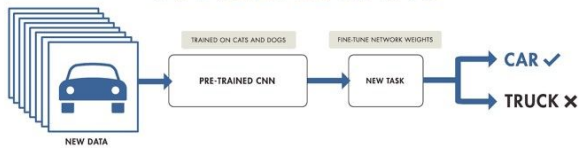


Transfer learning

TRAINING FROM SCRATCH



TRANSFER LEARNING



Principe de fonctionnement

- Couches profondes basées sur un modèle pré-entraîné
- Entraînement des couches finales seulement

Avantages

- Entraînement plus rapide
- Nombreux modèles adaptés à plusieurs cas de figure

Création d'un modèle classificateur d'image

Création d'un modèle avec 2 classes: **chamois** et **pas chamois**

Layer (type)	Output Shape	Param #
sequential (Sequential)	(None, 160, 160, 3)	0
rescaling (Rescaling)	(None, 160, 160, 3)	0
model (Functional)	(None, 1, 1, 1000)	475544
global_average_pooling2d_1 (GlobalAveragePooling2D)	(None, 1000)	0
flatten (Flatten)	(None, 1000)	0
dense (Dense)	(None, 16)	16016
dropout (Dropout)	(None, 16)	0
dense_1 (Dense)	(None, 1)	17

=====
Total params: 491,577
Trainable params: 16,033
Non-trainable params: 475,544

Couche de mise en forme des données

Modèle pré-entraîné : MobileNetv1

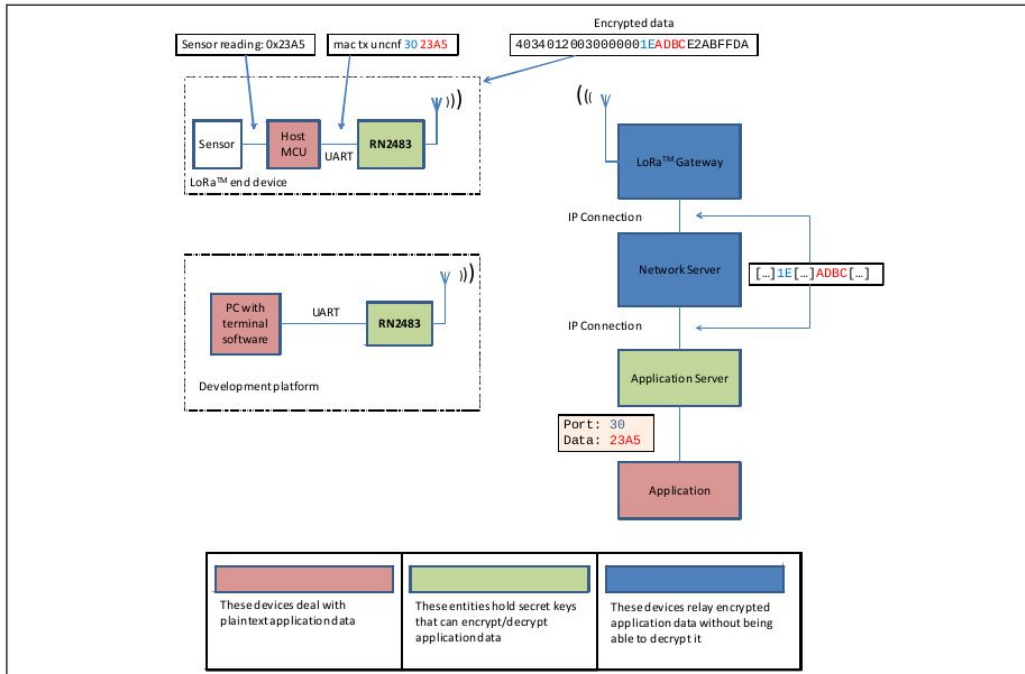
Couches de sorties



Spécifications du modèle

- Taille en Flash : **600 KBytes** (*2,7 Mbytes avant quantization*)
- Taille en RAM durant l'utilisation : **250 KBytes**
- Précision : **93%** (*A prendre avec des pincettes*)

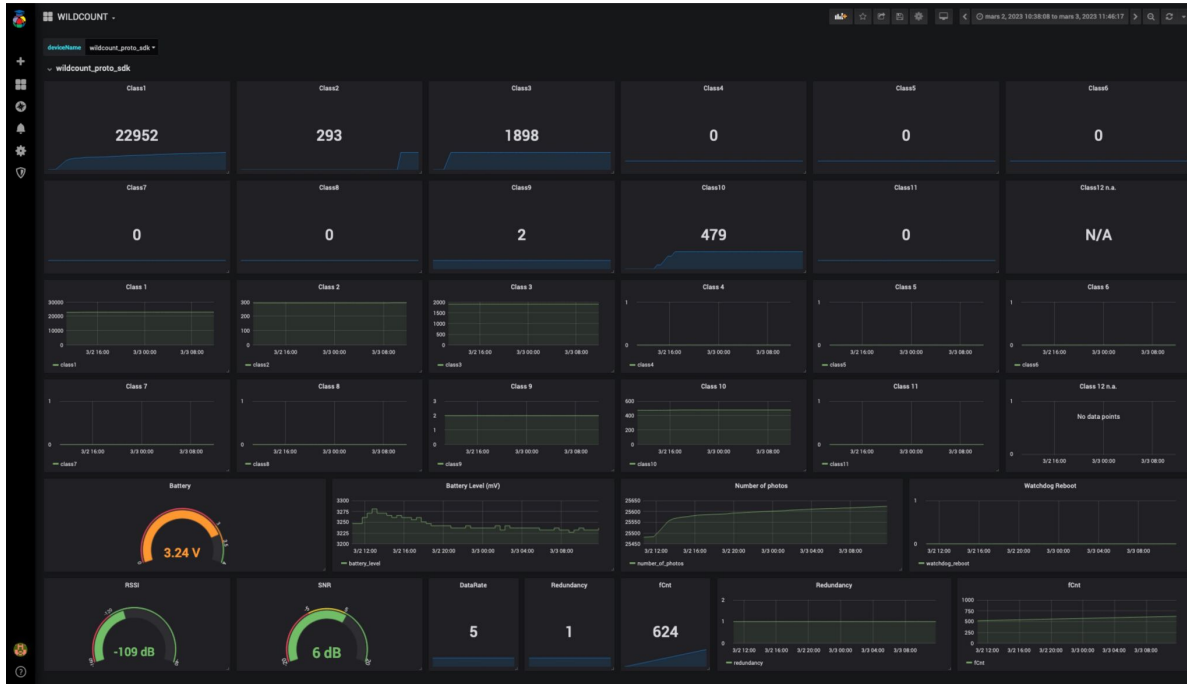
Envoi des trames (LoRaWAN)



- Commande `mac tx uncnf <port> <data>`
- Décodage des trames par une fonction JavaScript côté serveur

```
1 function Decode_UP_LAST_COUNTERS(bytes, variables) {  
2   var size = bytes.length;  
3  
4   var counters = [];  
5  
6   var bytes_pos = 0;  
7   var counter_pos = 0;  
8  
9   var t = {};  
10  t.detected_classes = bytes[bytes_pos++];  
11  counters[counter_pos] = t;  
12  counter_pos += 1;  
13  
14  while(bytes_pos < size){ // reading <class(8b);counter(8b)> pairs  
15  
16    var o = {};  
17    o.class = bytes[bytes_pos++];  
18    o.counter = bytes[bytes_pos++];  
19    counters[counter_pos] = o;  
20    counter_pos += 1;  
21  }  
22  
23  return counters;  
24 }
```

Interface données



Grafana

4

Résultats et suite du projet

Expérimentations

3 expérimentations sur le terrain à Entraigues

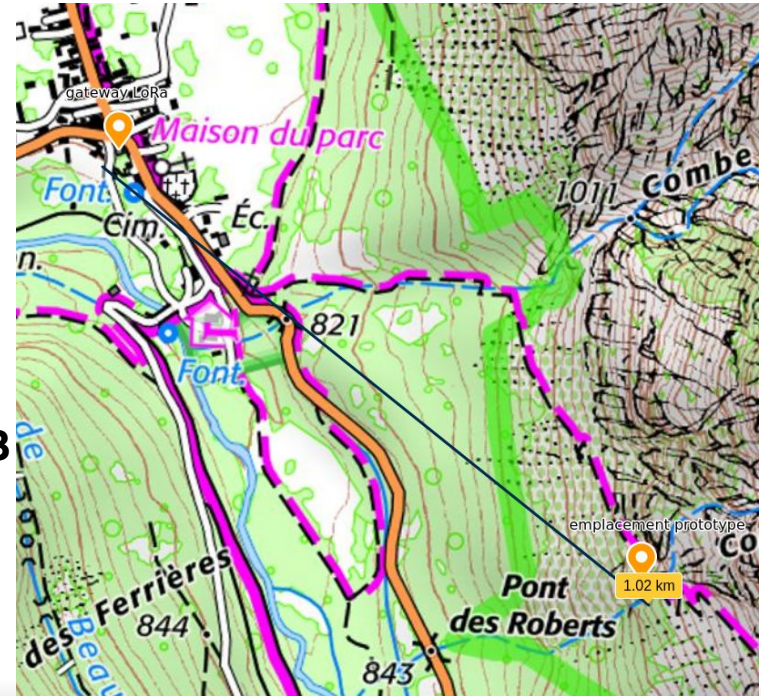
- Station LoRa installée sur la maison du parc
- Placement du piège dans une zone de passage



✗ problème batterie

✗ problème modem LoRa

✓ en production



Résultats

Résultats dernière expérimentation

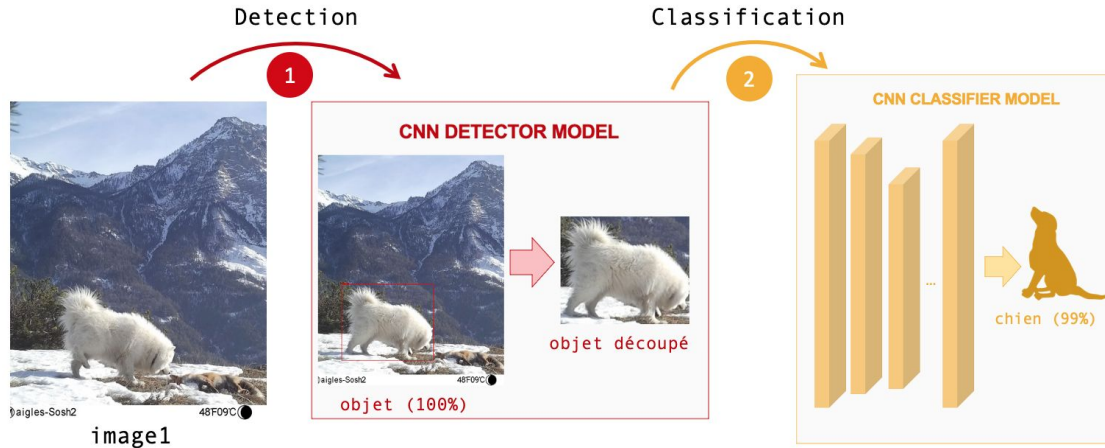
Dernière expérimentation en cours:

- Mis en place le 7 Février 2023 - récupéré le 14 Mars 2023
- Plus de deux mois de fonctionnement ininterrompu
- Résultats du modèle incohérents
- capteur PIR trop sensible

Avancement du projet

Ajout de fonctionnalités

Implémentation caméra thermique



Performances médiocres du modèle

Besoin de localiser l'animal sur la photo

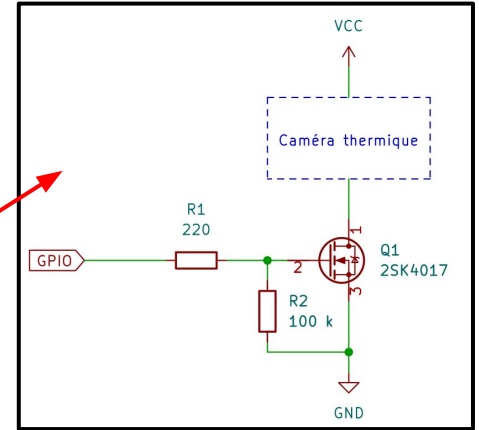
Avancement du projet

Ajout de fonctionnalités



Caméra thermique MLX90640

- Résolution de 32x24 pixels
- I2C
- Consommation : 20 mA



Shield Arduino

⇒ Implémentation en cours mais non terminée par manque de temps

Merci de votre écoute

Avez-vous des questions ?

WILDCOUNT

11:51:39 AM uplink

- Org. users
- Service-profiles
- Device-profiles
- Gateways
- Applications
- Multicast-groups

```

applicationID: "83"
applicationName: "WILDCOUNT"
deviceName: "wildcount_proto_sdk"
devEUI: "0004a30b00f31233"
▶ rxInfo: [] 1 item
▶ txInfo: {} 3 keys
  adr: true
  dr: 5
  fCnt: 198
  fPort: 3
  data: "Cw7AABwBAAAAAAAAAAAAAAAAABKAAA="
▼ objectJSON: [] 11 items
  ▼ 0: {} 2 keys
    class: 1
    counter: 3776
  ▼ 1: {} 2 keys
    class: 2
    counter: 28
  ▼ 2: {} 2 keys
    class: 3
    counter: 256
  ▼ 3: {} 2 keys
    class: 4
    counter: 0
  ▼ 4: {} 2 keys
    class: 5
    counter: 0
  ▼ 5: {} 2 keys
    class: 6
    counter: 0
  ▼ 6: {} 2 keys
    class: 7
    counter: 0
  ▼ 7: {} 2 keys
    class: 8
    counter: 0
  ▼ 8: {} 2 keys
    class: 9
  
```

Spreading factor (at 125 kHz)	Bitrate	Range (indicative value, depending on propagation conditions)	Time on Air (ms) For 10 Bytes app payload
SF7	5470 bps	2 km	56 ms
SF8	3125 bps	4 km	100 ms
SF9	1760 bps	6 km	200 ms
SF10	980 bps	8 km	370 ms
SF11	440 bps	11 km	740 ms
SF12	290 bps	14 km	1400 ms

(with coding rate 4/5 ; bandwidth 125Khz ; Packet Error Rate (PER): 1%)

Avancement du projet

Migration vers le SDK

Gestion du modem LoRa : Appairage au réseau

Spreading factor (at 125 kHz)	Bitrate	Range (indicative value, depending on propagation conditions)	Time on Air (ms) For 10 Bytes app payload
SF7	5470 bps	2 km	56 ms
SF8	3125 bps	4 km	100 ms
SF9	1760 bps	6 km	200 ms
SF10	980 bps	8 km	370 ms
SF11	440 bps	11 km	740 ms
SF12	290 bps	14 km	1400 ms

(with coding rate 4/5 ; bandwidth 125Khz ; Packet Error Rate (PER): 1%)

- Tentative de plusieurs *join* en commençant du débit le plus rapide (DR5) jusqu'au débit lent (DR0)
- Essais successifs avec diminution du débit
- Une fois l'appairage effectué, on active l'ajustement automatique du débit (ADR)