Comptage de moutons à partir d'une vidéo drone

Le comptage des moutons est une tâche à la fois fastidieuse et chronophage. Actuellement, il se fait manuellement ou à l'aide de boucles électroniques, ce qui reste lent et peu pratique lorsqu'il s'agit de centaines d'animaux. Dans le cadre du projet européen ICAERUS, nous explorons l'apport de la vision par ordinateur et de l'utilisation de drones pour automatiser ce processus. Pour en savoir plus sur le projet : https://icaerus.eu/hover-view-how-to-count-sheep-without-falling-asleep-using-drones-and-ai-to-monitor-flocks/

Ce protocole présente l'installation et l'utilisation d'un démonstrateur logiciel dédié au comptage de moutons filmés par drone. L'utilisateur importe une vidéo de son troupeau, prise depuis un drone volant à environ 10 mètres d'altitude alors que les animaux passent dans un couloir. Le démonstrateur permet d'optimiser la vidéo pour le comptage, puis de tracer une ligne virtuelle adaptée à la scène. Un algorithme basé sur le modèle d'intelligence artificielle YOLO, spécifiquement entraîné pour la détection aérienne de moutons et de chèvres, identifie ensuite les animaux franchissant cette ligne et les compte automatiquement. Un aperçu de ce qu'on peut obtenir en sorti de démonstrateur :

1. Préparation de la machine hôte

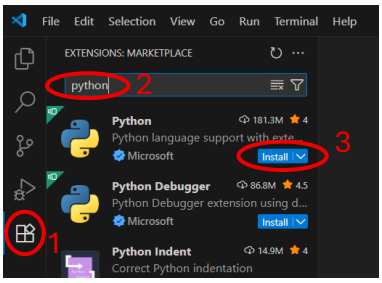
Le démonstrateur a été développé avec le langage de programmation Python. Pour pouvoir l'utiliser, il est donc nécessaire d'installer Python sur votre ordinateur. Ensuite, pour lancer et utiliser correctement le projet, on installe un environnement de développement. Dans notre cas, il s'agit de Visual Studio Code qui est comme une boîte à outils qui rassemble tout ce qu'il faut pour travailler avec Python.

- Installer python 3.11.0 :
  https://www.python.org/downloads/release/python-3110/

Version	Operating System	Description
Gzipped source tarball	Source release	
XZ compressed source tarball	Source release	
macOS 64-bit universal2 installer	macOS	for macOS 10.9 and later
Windows installer (64-bit)	Windows	Recommended
Windows installer (32-bit)	Windows	
Windows installer (ARM64)	Windows	Experimental
Windows embeddable package (64-bit)	Windows	
Windows embeddable package (32-bit)	Windows	
Windows embeddable package (ARM64)	Windows	

- Installer Visual Studio Code (VS Code) :
  https://code.visualstudio.com/download

- Installer l'extension Python dans VS Code
  - Ouvrir VS Code
  - Accéder à l'onglet des extensions (icone à gauche présenté ci-dessous)
  - Rechercher « python » et installer l'extension officielle de Microsoft

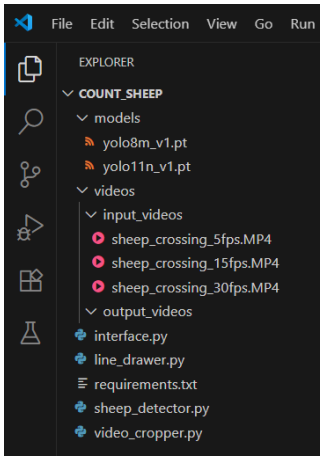


2. Récupération du projet GitHub

- Télécharger le dossier UC3_Livestock_Monitoring du projet ICAERUS : https://github.com/ICAERUS-EU/UC3_Livestock_Monitoring/tree/main
  - Cliquer sur <> Code
  - Download ZIP
  - Copier le ZIP vers le dossier souhaité
- Après extraction du dossier, vous obtenez tous les dossiers présents dans le github UC3_Livestock_Monitoring. L'outil de déploiement se trouve dans le dossier **application/deployment/count_sheep_user**
  Le répertoire contient :
  - models/ : modèles IA de détection
  - videos/ : vidéos d'entrées et de sorties
  - Scripts nécessaires à l'interface utilisateur

- requirements.txt : liste des bibliothèques python utilisées et leurs versions

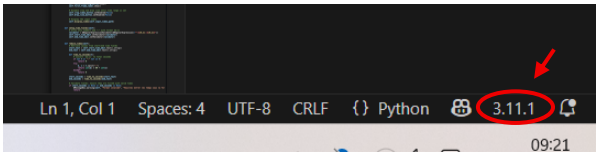
- Ajouter les vidéos prises aux drones de votre troupeau dans le dossier « videos/input_videos » (un exemple est présent dans le dossier data/Videos/ du répertoire UC3_Livestock_Monitoring).
- Ouvrir le projet dans VS Code :
  - Cliquer sur « File > Open Folder »
  - Sélectionner le dossier « **count_sheep_user** »
  - Un exemple de ce que vous devez obtenir ci-dessous :



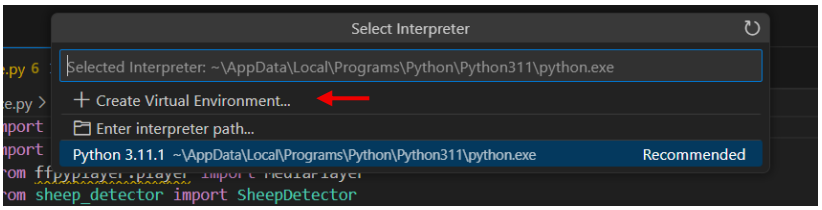
- Ouvrir le fichier « interface.py »

3. Création de l'environnement virtuel

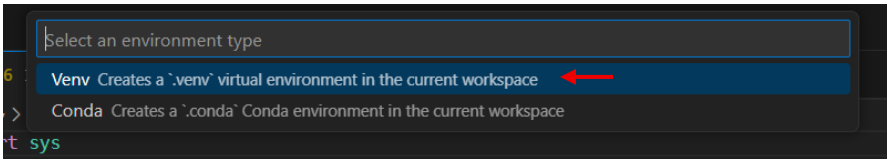
- Un environnement virtuel doit être installé pour installer sur la machine toutes les versions des libraires nécessaires au fonctionnement de l'interface.
- Dans VS Code, cliquer sur la barre en bas à droite, sur la version de python utilisé :
  Si la barre n'apparait pas, cliquer sur « View > Appearance > Status Bar »



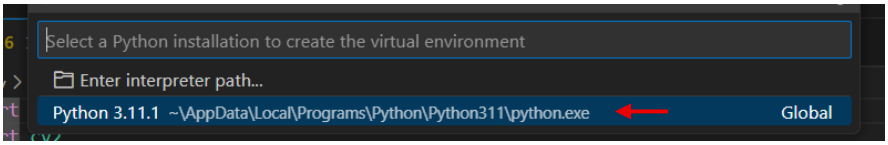
- Une fenêtre s'ouvre en haut de la page, cliquer sur « Create Virtual Environment » :



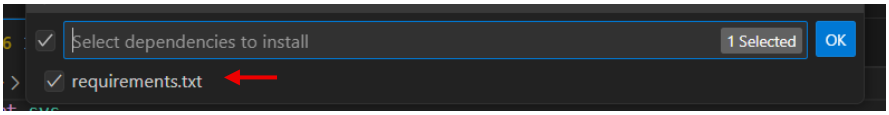
- Cliquer sur « venv », qui signifie qu'on créé un environnement python :



- Sélectionner l'interpréteur python qui convient (3.11.x) :



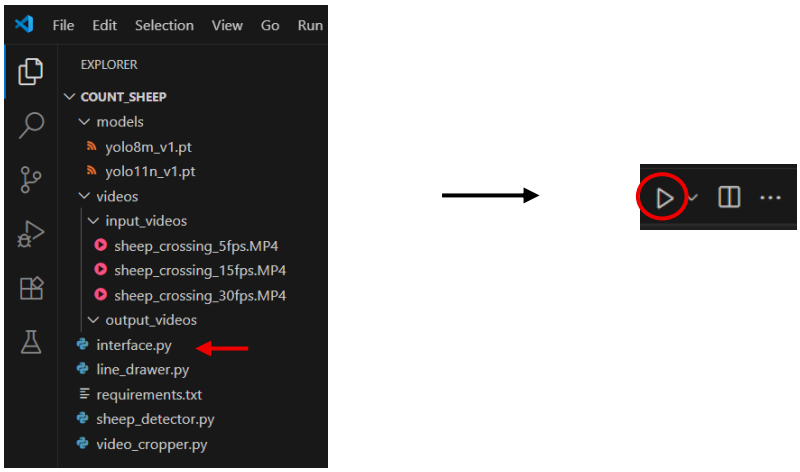
- Sélectionner le fichier « requirements.txt » , puis sur « ok » :



- L'installation des dépendances démarre automatiquement. Elle peut prendre entre 15 à 40 minutes selon votre ordinateur.
- Vérifier que le dossier « .venv » a bien été créé dans le répertoire du projet.

4. Tests fonctionnels

- Ouvrir le fichier « interface.py »
- Lancer le fichier en cliquant sur la flèche Run (en haut à droite). Des lignes rouges s'affichent mais l'interface va apparaitre dans quelques secondes.
- Une fenêtre s'ouvre, l'interface utilisateur est opérationnelle



<u>Sélection de la vidéo</u>

- Choisir une vidéo sur laquelle vous voulez réaliser le comptage d'ovins/caprins. Veiller à ce que ce soit une vidéo :
    a. Prise au drone, à environ 15 mètres d'altitude
    b. Pas d'obstacle visuelle masquant le troupeau
    c. Les animaux passent dans un couloir délimité
    d. L'image pas trop zoomée sur l'endroit de passage
- Choisir un FPS (Images par seconde) adapté : il correspond à la fluidité de la vidéo en sorti sur laquelle les détections seront réalisé, plus le FPS est petit, plus le traitement sera rapide.
- Choisir un modèle de détection pour parmi ceux proposés.
- Cliquer sur « commencer »

<u>Processing de la vidéo</u>

- Une phase de processing de la vidéo est à réaliser pour gagner en temps de calcul et être dans les meilleures conditions pour la détection :

a. Réduire la durée de la vidéo pour ne conserver que la séquence utile au format « minutes : secondes »
b. Rogner la vidéo pour cibler la zone d'intérêt. La vidéo sera convertie au format 640x640 pixels pour une détection optimale avec YOLO. Plus le dessin de cette zone est proche de 640x640px, moins la qualité de la vidéo sera dégradée.
c. Tracer une ligne de comptage, parallèle au sens de passage des animaux si possible et assez longue pour que tous les animaux la traversent
d. Réinitialiser la vidéo en cas de mauvais traçage ou rognage

Si le rognage ne permet pas de voir l'intégralité du troupeau, utiliser une autre vidéo.

- Cliquer sur « commencer le comptage » pour exporter la vidéo éditée dans le dossier « videos/crop_videos » et lancer l'algorithme de comptage. La vidéo avec les annotations de détection et de suivi ainsi que le compteur est ensuite affichée. La vidéo finale est également exportée dans « videos/output_videos ».

5. Modifications

- Les modèles d'intelligence artificielle peuvent évoluer rapidement. Il est donc possible d'ajouter des modèles supplémentaires dans l'interface. Pour cela, il faut modifier « interface.py » dans la section « #Model selection » (voir première capture d'écran ci-dessous) :
    a. Ajouter un nouveau « QRadioButton » avec le nom du modèle :
        self.radio_nomvariable = QRadioButton(« nomduyolo »)
    b. Ajouter ce bouton au « QButtonGroup » en lui attribuant une valeur unique :
        self.model_group.addButton(self. radio_nomvariable, 3)
    c. Ajouter le chemin du modèle au dictionnaire dédié ainsi que son suffixe (numéro de version et sa taille) :
        2 : (« models/yoloxx.pt », « xx »)

interface.py

```
class InterfaceApp(QWidget):
    def __init__(self):
        # Model selection
        self.model_label = QLabel("Choisissez le modèle de détection :")
        self.model_label.setStyleSheet("font-size: 16px;")
        self.model_label.setAlignment(Qt.AlignCenter)

        self.radio_yolo8m = QRadioButton("YOLO8m")
        self.radio_yolo11n = QRadioButton("YOLO11n")
        self.radio_yolo11l = QRadioButton("YOLO11l")
        self.radio_yolo11n.setChecked(True)  # Default value

        self.model_group = QButtonGroup(self)
        self.model_group.addButton(self.radio_yolo8m, 0)
        self.model_group.addButton(self.radio_yolo11n, 1)
        self.model_group.addButton(self.radio_yolo11l, 2)

        # Dictionary to link ID to model path
        self.model_paths = {
            0: ("models/yolo8m.pt", "8m"),
            1: ("models/yolo11n.pt", "11n"),
            2: ("models/yolo11l.pt", "11l")
        }

        self.model_group.buttonClicked[int].connect(self.update_model_selection)
```

Pour finir, modifier la section « #Layout for model sélection » pour ajouter le choix du modèle dans l'interface comme ci-dessous :

model_layout.addWidget(self.nomduyolo)

```
        # Layout for model selection
        model_layout = QHBoxLayout()
        model_layout.addStretch()
        model_layout.addWidget(self.model_label)
        model_layout.addWidget(self.radio_yolo8m)
        model_layout.addWidget(self.radio_yolo11n)
        model_layout.addWidget(self.radio_yolo11l)
        model_layout.addStretch()
```

- Pour modifier les valeurs prises par défaut, notamment dans la version utilisateur, le FPS (images par secondes) de la vidéo de sortie, le chemin du modèle et son suffixe sont à renseigner dans la section ci-dessous :

```
class InterfaceApp(QWidget):
    def __init__(self):
        super().__init__()

        # Window title
        self.setWindowTitle('ICAERUS')
        self.setGeometry(100, 100, 600, 400)


        # Default values
        self.selected_fps = 15
        self.model_path = "models/yolo11n.pt"
        self.model_suffix = "11n"
```