



CENTRE D'ECOLOGIE  
FONCTIONNELLE  
& EVOLUTIVE



## Rapport de Stage L3

Gilibert Rémy

06/05/2024 au 07/07/2024

# Table des Matières

## Contents

<b>Contents</b>	<b>2</b>
<b>Remerciements</b>	<b>3</b>
<b>Introduction</b>	<b>4</b>
<b>Présentation du travail réalisé</b>	<b>5</b>
Développement web . . . . .	5
Inscription / connexion . . . . .	5
Accueil . . . . .	6
Résultat de détection . . . . .	7
Annotation . . . . .	8
<b>Base de données</b>	<b>12</b>
Descriptif des Tables . . . . .	12
Table : clients . . . . .	12
Table: images . . . . .	12
Table: img_annot . . . . .	12
<b>Deep Learning</b>	<b>13</b>
Implémentation du modèle . . . . .	13
Active Learning . . . . .	14
Fine Tuning . . . . .	14
<b>Perspectives / Travaux à finir</b>	<b>18</b>
Travaux futurs . . . . .	18
Hébergement serveur/ Style général . . . . .	19
Limites . . . . .	19
<b>Conclusion</b>	<b>20</b>
<b>Bibliographie</b>	<b>21</b>

## Remerciements

Je tiens à remercier :

Jérôme Pasquet pour cette opportunité de stage ainsi que pour les connaissances acquises durant le semestre 5 grâce à son cours de développement web.

Théo Oriol pour m'avoir suivie tout au long de ce stage en corrigeant mes erreurs et m'orientant vers la bonne direction

Pierre Jay-Robert et toute l'équipe du CEFE de m'avoir accueillie au sein de leur structure et en particulier Ninon Delcourt qui a accepté de partager son bureau avec moi tout au long de mon stage.

Daniel Gilibert /languageTool/Chatgpt pour la correction orthographique de ce rapport.

Maximilien servajean pour le cours de programmation orientée objet qui m'ont permis d'avoir les bases de python.

## Introduction

Actuellement étudiant en 3ème année de MIASHS (mathématique et informatique appliquée aux sciences humaines et sociales) à l'université Paul Valéry Montpellier 3, j'ai eu la chance d'effectuer mon stage de fin de licence du 06/05/2024 au 07/05/2024 au sein du Centre d'écologie fonctionnelle et évolutive (sections du CEFE situé au campus Paul Valery) où j'ai pu découvrir le travail dans un laboratoire de recherche.

« Le CEFE est un des plus importants laboratoires de recherche en Ecologie en France. Le projet du CEFE vise à comprendre la dynamique, le fonctionnement et l'évolution du vivant de «la bactérie à l'éléphant» et «du génome à la planète». Il s'appuie sur trois ambitions: [1] comprendre le monde vivant pour anticiper ce que sera demain [2] conduire à des innovations et répondre aux attentes de la société; [3] pratiquer une science «rassembleuse» et diverse dans ses approches disciplinaires. Les questions de recherche sont posées dans un contexte marqué par la prégnance des changements planétaires, le développement de nouvelles technologies de manipulation du vivant et l'exigence croissante de la société pour la recherche. »

Source: CEFE

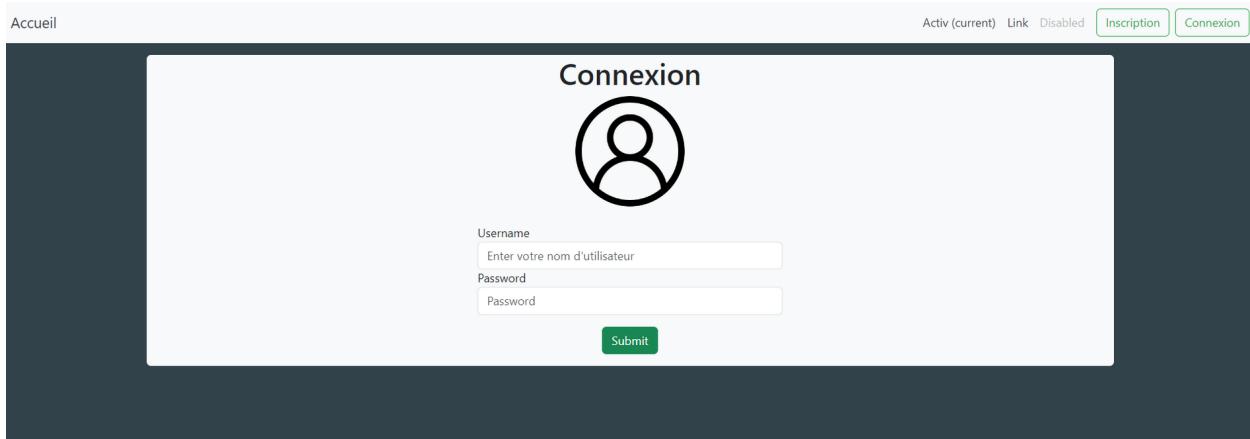
Durant ce stage, j'ai pu créer en collaboration avec Théo Oriol (Doctorant) une interface web dédiée à l'identification des collemboles grâce à la reconnaissance sur image (YOLO v5x6) ainsi qu'à l'amélioration du modèle via un système d'annotation. Ce sujet de stage m'a permis de manipuler plusieurs outils pour le développement web comme Flask et Bootstrap mais aussi plusieurs concepts du Deep Learning comme l'implémentation d'un modèle "l'Active Learning" lié à ce dernier ainsi que le "Fine Tuning". Cette interface web permettra à termes aux chercheurs et passionnés du monde entier d'avoir accès à une identification rapide et fiable, évitant les pertes de temps liées à cette partie, et permettra d'améliorer continuellement le modèle tout en collectant les données fournies par les utilisateurs de l'interface.

## Présentation du travail réalisé

### Développement web

#### Inscription / connexion

Les premières choses réalisées sur l'interface web furent la mise en place des pages d'inscription et de connexion ainsi que la création d'une base de données. La mise en place de ces pages nécessita premièrement des recherches poussées et précises sur le framework “Flask” pour apprendre à créer des pages web via des applications ainsi que les syntaxes et méthodes spécifiques à Flask. Une première interface web a donc été créée pour les pages inscription et connexion (Fig 1 et 2). La création de la base de données a été faite à la même période permettant le stockage des informations utiles comme le nom d'utilisateur, le mot de passe (haché au préalable) et pour finir, on stocke aussi à l'inscription le paramètre “expert” ou non expert pour des questions d'accessibilité que nous traiterons plus tard. Dans cette même semaine, une Nav-bar a été ajoutée, elle contiendra tous les liens utiles vers les autres pages du site web et permettra aussi la déconnexion qui remplace connexion et inscription si vous êtes connecté.



Accueil Activ (current) Link Disabled [Inscription](#) [Connexion](#)

### Connexion

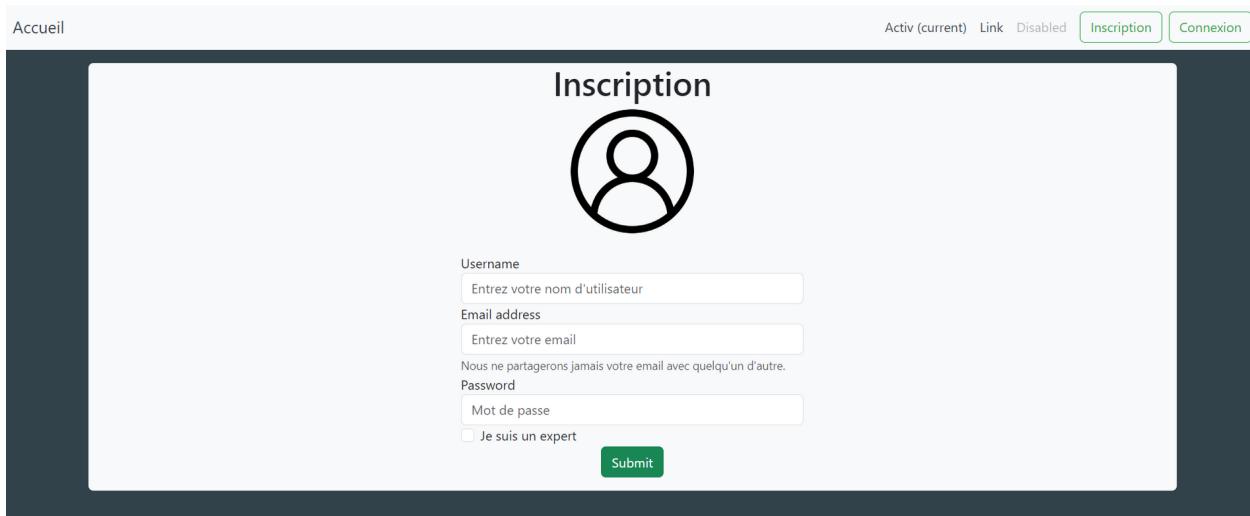


Username  
Enter votre nom d'utilisateur

Password  
Mot de passe

Submit

Figure 1: Page d'inscription



Accueil Activ (current) Link Disabled [Inscription](#) [Connexion](#)

### Inscription



Username  
Entrez votre nom d'utilisateur

Email address  
Entrez votre email

Nous ne partagerons jamais votre email avec quelqu'un d'autre.

Password  
Mot de passe

Je suis un expert

Submit

Figure 2: Page de connexion

## Accueil

L'accueil peut être défini comme l'élément central de cette interface web, en effet c'est elle qui contient l'input permettant de passer sa photo à l'intelligence artificielle en vue de la détection des espèces de collemboles. Durant les premières semaines, une première version a été réalisée (Fig 3). Cette version transmettait l'image importée à YOLOv5 étant plus généraliste en attendant d'avoir le modèle entraîné. Le choix de la version s'est orienté vers YOLOv5s car étant la plus rapide, elle permet de faire plus de tests. Après avoir validé l'image à analyser, le modèle charge et quand il fournit sa réponse l'utilisateur est automatiquement redirigé vers une nouvelle page.



Figure 3: Première version de la page d'accueil

Cette version de la page d'accueil manquait cruellement de vie et d'accessibilité, un travail sur ces deux points a donc été fait pour avoir une interface plus esthétique et permettant une meilleure compréhension des images attendues dans le but d'effectuer une détection précise grâce aux instructions écrites, mais aussi grâce au carrousel montrant des images adaptées à la détection (Fig 4).

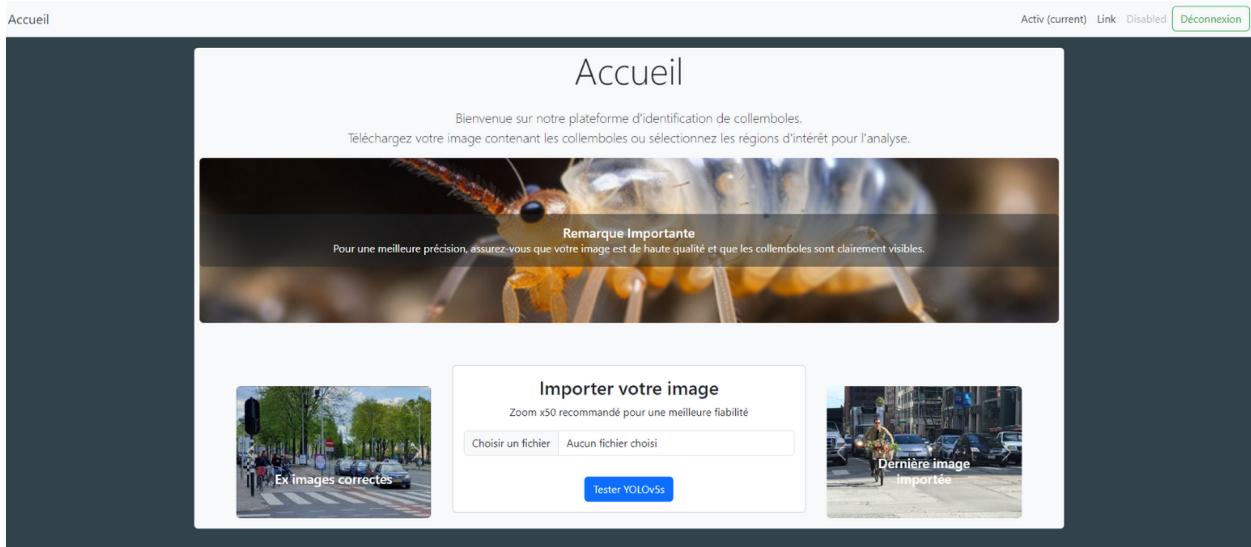


Figure 4: Version améliorée de la page d'accueil

Pour accéder à toutes ces fonctionnalités, l'inscription est nécessaire, ainsi s'il n'y a pas de session en cours, alors la page s'affichera avec un affichage différent suggérant à l'utilisateur de se connecter pour utiliser cette fonctionnalité (Fig 5).



Figure 5: Page d'accueil avec session inactive

## Résultat de détection

Après la redirection depuis la page d'accueil, l'utilisateur arrive sur cette page affichant les résultats de détection (Fig 6). On retrouve ici l'image donnée par l'utilisateur accompagnée de boxes ajoutées grâce au CSS qui récupère les coordonnées en pixels localisant les objets / collemboles détectés par YOLOv5. Ces objets sont accompagnés de leur nom ainsi qu'un chiffre unique à chacun. Une liste à gauche de l'image nous permet d'observer les niveaux de confiance associés à chaque objet nous donnant une indication précise quant à la précision de la classification de l'objet en question. Aussi, un espace commentaire permet de renseigner des informations sur la photo du type métadonnées (lieux / date/...) (Fig 7).



Figure 6: Page des résultats de détection

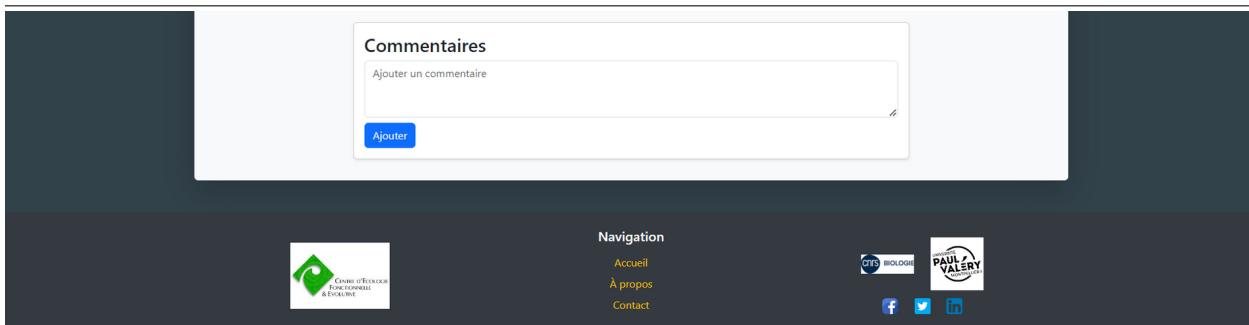


Figure 7: Page des résultats avec commentaires

## Annotation

La page Annotation a, elle aussi, un rôle prépondérant sur cette interface web, elle permet de récupérer dans la base de données les images sélectionnées par l'Active Learning pour les annoter avant d'entraîner à nouveau le modèle. Dans le cadre d'annotations nécessitant une réelle expertise du sujet, cette page est accessible uniquement au profil "Expert" pour éviter le maximum d'annotations erronées et donc indirectement éviter au modèle de fausses détections après entraînement. Si le profil n'est pas expert, mais essaie d'accéder à cette page, alors cette interface page apparaîtra.

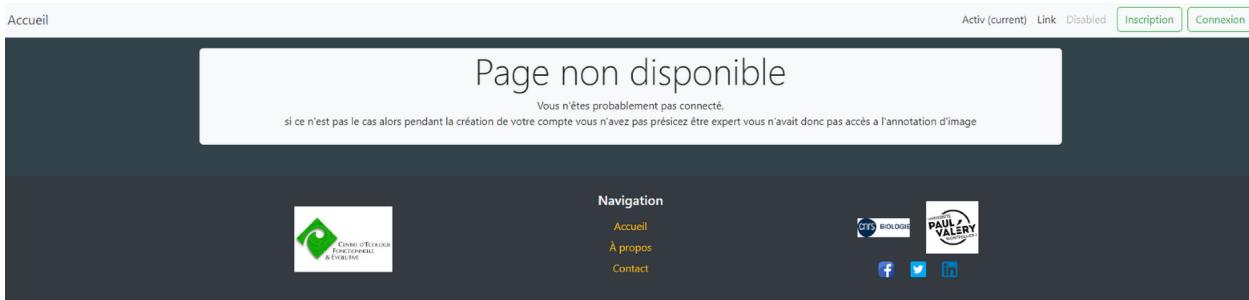


Figure 8: Page d'annotation non experts

Le design et les fonctionnalités de cette page ont évolué au cours du temps, mais le concept reste le même. L'utilisateur dispose d'une partie à gauche de la page où il peut sélectionner des zones grâce à la souris avant de leur donner un label. Ces zones seront répertoriées à droite de la page permettant une relecture du label accompagné de ses coordonnées. Pour envoyer ces annotations, il suffit à l'utilisateur d'appuyer sur le bouton en bas de page. La première version de cette page garde tous ces concepts, mais beaucoup de problèmes esthétiques et fonctionnels (annotation par système de pop-up, les zones saisies ne sont pas modifiables ou supprimables, problème de ratio pour les images, problème d'affichage dans la liste des zones) (Fig 9, Fig 10).

Accueil

Activ (current) Link Disabled Déconnexion

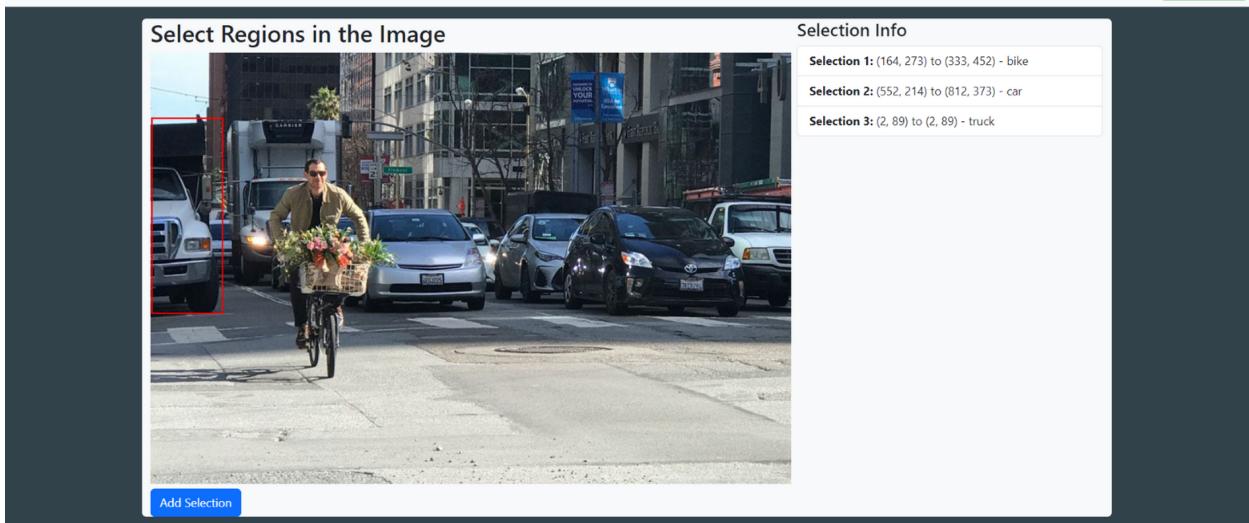


Figure 9: Première version de la page d'annotation

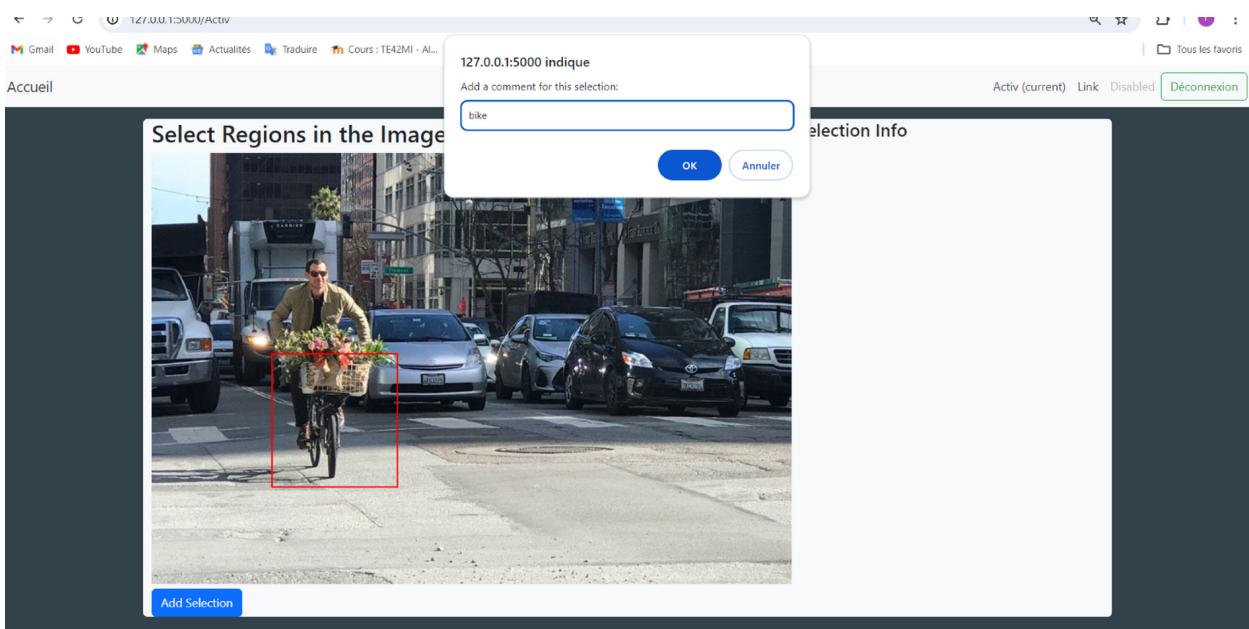


Figure 10: Première version avec zones saisies

Pour corriger tous les problèmes préalablement cités et améliorer l'accessibilité ainsi que l'esthétique de la page, une nouvelle version a été créée, étant beaucoup plus fonctionnelle et flexible pour l'utilisateur (Fig 11, Fig 12).

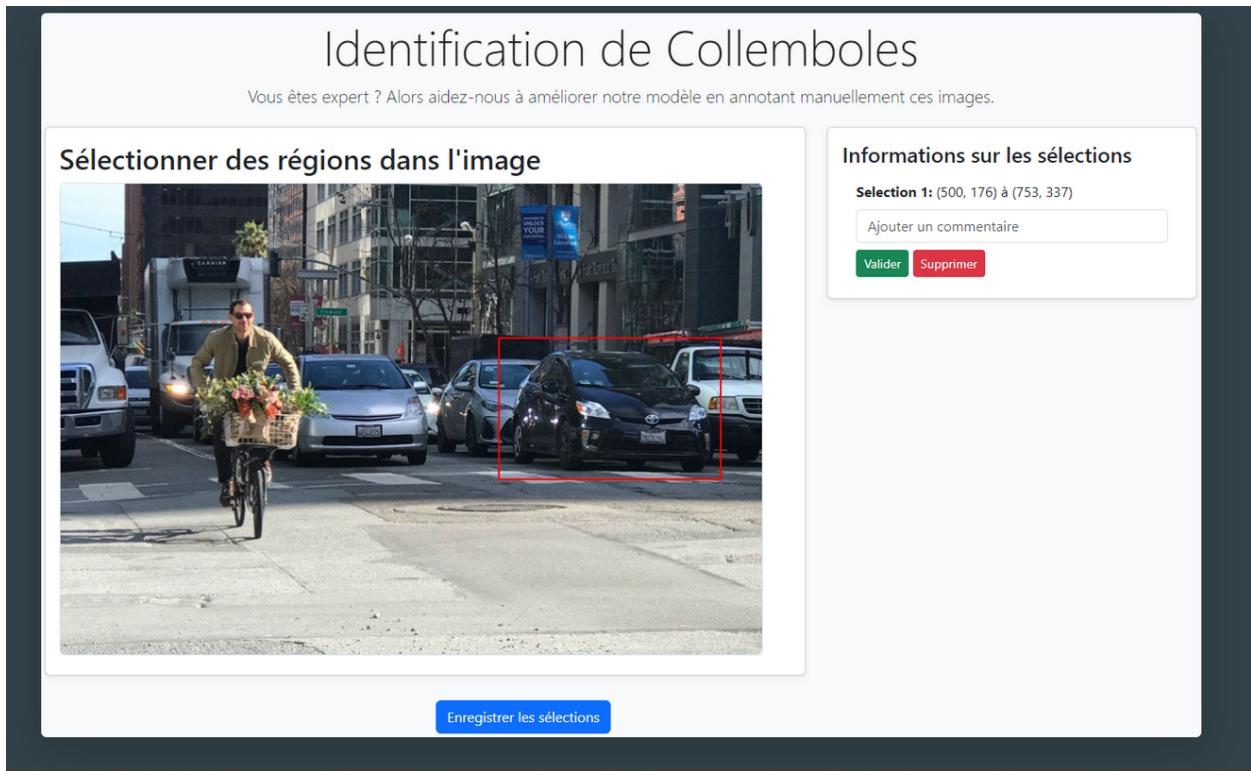


Figure 11: Nouvelle version de la page d'annotation

Un problème persistait quand même : les annotations semblaient peu précises, car sans zone précise, certains éléments sont difficiles à encadrer proprement. Le zoom semblait donc la meilleure option pour plus de précision, l'ajout de boutons le permettant semblait de ce fait utile ainsi qu'un zoom avec la souris (scroll) (Fig 13).

## Identification de Collemboles

Vous êtes expert ? Alors aidez-nous à améliorer notre modèle en annotant manuellement ces images.

### Sélectionner des régions dans l'image



Enregistrer les sélections

### Informations sur les sélections

**Selection 1:** (500, 176) à (753, 337) - car

[Modifier](#) [Supprimer](#)

---

**Selection 2:** (138, 131) à (306, 421) -  
ddddddddd

[Modifier](#) [Supprimer](#)

Figure 12: Nouvelle version avec label modifiable

## Identification de Collemboles

Vous êtes expert ? Alors aidez-nous à améliorer notre modèle en annotant manuellement ces images.

### Sélectionner des régions dans l'image

Zoom In
Zoom Out
Move Image
Reset Zoom
100%



### Informations sur les sélections

Figure 13: Page d'annotation avec zoom

## Base de données

### Descriptif des Tables

Table : clients

Nom de la colonne	Type de données	Signification	Caractéristiques
nom	Chaîne de caractère (varchar)	Nom du client	Clé primaire, non nul, unique
email	Chaîne de caractère (varchar)	Adresse email du client	-
mdp	Chaîne de caractère (varchar)	Mot de passe crypté du client	-
exp	Entier (tinyint)	expert ou non (0 ou 1)	Non nul

Table: images

Nom de la colonne	Type de données	Signification	Caractéristiques
id	Entier (integer)	Identifiant unique de l'image	Clé primaire, non nul, unique
username	Chaîne de caractère (varchar)	Nom d'utilisateur associé à l'image	Non nul
url	Chaîne de caractère (varchar)	URL de l'image	Non nul
commentaire	Texte (text)	Commentaire sur l'image(lieux/date)	-

Table: img\_annotation

Nom de la colonne	Type de données	Signification	Caractéristiques
img_id	Entier (integer)	Identifiant unique de l'image	Clé primaire, non nul, unique
img_url	Chaîne de caractère (varchar)	URL de l'image	Non nul
user	Chaîne de caractère (varchar)	Nom d'utilisateur associé à l'image	Non nul
label	Chaîne de caractère (varchar)	label/coordonnées associées à l'image	Peut être vide

# Deep Learning

## Implémentation du modèle

Avant de pouvoir procéder à toute détection, il est nécessaire d'implémenter un modèle dans le code. Comme mentionné précédemment, le modèle utilisé était “Yolov5s”, cependant il n'était pas encore entraîné. Pour utiliser ce modèle, plusieurs fonctions sont nécessaires pour charger et interpréter les images dans la fonction “detect”.

```
@app.route('/detect', methods=['POST'])
def detect_img():
    if request.method == 'POST':
        model = load_models() # Charger le modèle YOLOr5
        if 'image' in request.files:
            image_file = request.files['image']
            if image_file.filename == '':
                return "Aucun fichier sélectionné"

            image_path = os.path.join(app.config['UPLOAD_FOLDER'], image_file.filename)
            image_file.save(image_path)

            # Charger l'image avec PIL et la convertir en ndarray OpenCV
            img = Image.open(image_file.stream).convert("RGB")
            img_cv2 = cv2.cvtColor(np.array(img), cv2.COLOR_RGB2BGR)

            # Effectuer la détection avec YOLOr5s
            results = model(img_cv2)

            # Récupérer les résultats de détection au format JSON
            detections = results.pandas().xyxy[0].to_dict(orient='records')
            img_with_boxes = results.render()
            # Récupérer l'image avec les boîtes de détection

            # Convertir l'image en bytes
            img_byte_array = BytesIO()
            img.save(img_byte_array, format='JPEG')
            img_data = img_byte_array.getvalue()

            # Encoder les données d'image en base64
            img_base64 = base64.b64encode(img_data).decode('utf-8')
            image_url = url_for('static', filename='img_load/' + image_file.filename)

            # Vérifier si l'image doit être enregistrée
            if should_save_image(results, threshold_confidence=0.5):
                nom = session['username']
                if ajt_img(image_url, nom):
                    print("<script>alert('img!'); window.location.href='/';</script>")
                else:
                    print("<script>alert('not!'); window.location.href='/';</script>")

            # Retourner le template result.html avec les données de détection
            return render_template('result.html', detections=detections,
            image=img_base64, img_url=image_url)
```

```
else:
    return "Aucun fichier trouvé dans la requête"
```

## Active Learning

Comme discuté dans la partie précédente sur le développement web, l'Active Learning est essentiel pour améliorer le modèle. Les images pertinentes sont enregistrées dans la base de données pour entraîner et améliorer le modèle. Avant l'implémentation d'un Active Learning plus avancé, les images contenant plus de 3 objets avec un score de confiance inférieur à 0.40 sont automatiquement sauvegardées dans une nouvelle table de la base de données (img\_annot). (Fig 2)

```
def should_save_image(results, threshold_confidence=0.5):
    # Examinez les résultats de YOLO pour déterminer si l'image doit être enregistrée
    detections = results.pandas().xyxy[0].to_dict(orient='records')
    max_confidence = min(d['confidence'] for d in detections)
    return max_confidence < threshold_confidence
```

## Fine Tuning

En l'état actuel des choses, l'interface web fonctionne correctement et contient tous les éléments demandés et nécessaires à son bon fonctionnement. Les utilisateurs peuvent importer des images, les plus intéressantes sont sauvegardées dans la base de données ("Active Learning"), puis les images sont annotées par des experts et réintroduites dans la base de données avec les coordonnées et les étiquettes ajoutées. Cependant, pour l'instant, le modèle ne s'entraîne pas automatiquement. Pour atteindre cet objectif, il faut se concentrer sur la partie "Fine Tuning". Pour cela, beaucoup de recherches sont nécessaires car ces notions n'ont jamais été abordées en cours.

Pour créer le fine tuning, cela se découpe en plusieurs parties : les images sélectionnées doivent être déplacées dans les dossiers du modèle YOLO utilisé, c'est-à-dire dans "val" et "train" (fig 13). À chaque image est associé un fichier TXT contenant les annotations effectuées par les utilisateurs. Comme mentionné précédemment, les annotations faites sont associées à des coordonnées et à des labels. Cependant, YOLO étant un modèle de recherche sur images, il fonctionne grâce à un système d'ancrages. Ainsi, les coordonnées en pixels et les labels doivent être convertis pour répondre aux attentes de YOLO. Pour convertir ces coordonnées, il faut récupérer les informations dans la base de données, puis les transformer et créer les fichiers TXT au format YOLO.

```
def ensure_directory_exists(directory):
    if not os.path.exists(directory):
        os.makedirs(directory)

def split_dataset(source_dir, annotations_dir, train_dir, val_dir, test_dir,
                 , train_ratio=0.7, val_ratio=0.2, test_ratio=0.1):

    images = [f for f in os.listdir(source_dir) if f.endswith('.jpg')]
    random.shuffle(images)

    train_cutoff = int(len(images) * train_ratio)
    val_cutoff = train_cutoff + int(len(images) * val_ratio)

    train_images = images[:train_cutoff]
    val_images = images[train_cutoff:val_cutoff]
    test_images = images[val_cutoff:]
```

```

# Ensure directories exist
ensure_directory_exists(train_dir)
ensure_directory_exists(val_dir)
ensure_directory_exists(test_dir)

for img in train_images:
    shutil.copy(os.path.join(source_dir, img), os.path.join(train_dir, img))
    txt_file = img.replace('.jpg', '.txt')
    src_txt_path = os.path.join(annotations_dir, txt_file)
    dst_txt_path = os.path.join(train_dir, txt_file)
    if os.path.exists(src_txt_path):
        shutil.copy(src_txt_path, dst_txt_path)
        print(f"Copied {src_txt_path} to {dst_txt_path}")

for img in val_images:
    shutil.copy(os.path.join(source_dir, img), os.path.join(val_dir, img))
    txt_file = img.replace('.jpg', '.txt')
    src_txt_path = os.path.join(annotations_dir, txt_file)
    dst_txt_path = os.path.join(val_dir, txt_file)
    if os.path.exists(src_txt_path):
        shutil.copy(src_txt_path, dst_txt_path)
        print(f"Copied {src_txt_path} to {dst_txt_path}")

for img in test_images:
    shutil.copy(os.path.join(source_dir, img), os.path.join(test_dir, img))
    txt_file = img.replace('.jpg', '.txt')
    src_txt_path = os.path.join(annotations_dir, txt_file)
    dst_txt_path = os.path.join(test_dir, txt_file)
    if os.path.exists(src_txt_path):
        shutil.copy(src_txt_path, dst_txt_path)
        print(f"Copied {src_txt_path} to {dst_txt_path}")

```

Cette fonction nous permet de ranger les images sauvegarder et de les ranger dans les dossiers renseigner au préalable, néanmoins s'il n'existe pas, elle le créera.

La fonction “YoloBD” (Fig 4) contient toutes les fonctions liées au Fine Tuning et est activée après l'annotation par les experts. (Fig 4)

```

def save_annotations_to_file(annotations, img_name, image_width, image_height):
    def convert_to_yolo_format(x, y, width, height, img_width, img_height):
        x_center = (x + width / 2) / img_width
        y_center = (y + height / 2) / img_height
        normalized_width = width / img_width
        normalized_height = height / img_height
        return x_center, y_center, normalized_width, normalized_height

    def get_class_id(label):
        class_mapping = {
            'person': 0,
            'car': 1,
            'bicycle': 2,
            # Add other mappings here
        }

```

```

    return class_mapping.get(label, -1)

# Ouvrir le fichier en mode ajout
with open(f'yolov5/annotations/{img_name}.txt', 'a') as f:
    for annotation in annotations:
        # Vérifier que l'annotation contient les 5 valeurs attendues
        annotation_values = annotation.split('/')
        if len(annotation_values) != 5:
            print(f"Warning: Annotation '{annotation}' does not contain 5 values.")
            continue

        x, y, width, height, label = annotation_values
        x_center, y_center, normalized_width, normalized_height =
            convert_to_yolo_format(
                float(x), float(y), float(width), float(height), image_width, image_height
            )
        class_id = get_class_id(label)
        if class_id == -1:
            print(f"Warning: Label '{label}' not found in class mapping.")
            continue
        annotation_line =
            f'{class_id} {x_center} {y_center} {normalized_width} {normalized_height}\n'
        f.write(annotation_line)

```

On retrouve sur le code précédent la fonction “get \_class \_id” pour traduire les labels en chiffre présent en première position dans les fichiers txt, ici les classe sont : person/ car/ bicycle, ces catégories on était choisie, car ce sont des catégories simples à trouver sur des images. Le modèle n’étant pas à ce moment la version entraînée sur les collemboles, la création du fine tuning avait pour but d’entrainer un modèle très généraliste (Yolov5s) à reconnaître plus précisément ces catégories.

```

@app.route('/save_zones', methods=['POST'])
def save_zones():
    zones_data = request.json # Récupérer les données des zones depuis la requête JSON
    data = ""
    for zone in zones_data:
        x = zone['x']
        y = zone['y']
        width = zone['width']
        height = zone['height']
        label = zone['label']
        img_id = zone['id']
        if label != "":
            data += f'{x}/{y}/{width}/{height}/{label}' + "|||"
    ajt_label(img_id, data)
    get_annotations(img_id)

# Example usage
source_directory = 'C:/Users/remyg/Documents/STAGE2/static/img_load'
annotations_directory = 'C:/Users/remyg/Documents/STAGE2/yolov5/annotations'
train_directory = 'C:/Users/remyg/Documents/STAGE2/yolov5/data/image/train'
val_directory = 'C:/Users/remyg/Documents/STAGE2/yolov5/data/image/val'
test_directory = 'C:/Users/remyg/Documents/STAGE2/yolov5/data/image/test'

```

```
split_dataset(source_directory, annotations_directory,
train_directory, val_directory, test_directory)
return jsonify({'message': 'Données des zones enregistrées avec succès'})
```

Cette partie de code nous permet d'illustrer l'ordre des appels de fonction de YoloBD. On remarque donc qu'il ne nous manque plus que l'entraînement pour que cette partie soit finie, en effet maintenant que nous avons tous les éléments, il ne reste plus qu'à paramétriser l'entraînement, les scripts d'entraînement sont déjà dans le modèle, il suffit de lui indiquer les chemin et classe à suivre grâce à un fichier “.yaml” dont on donne la localisation dans “train.py”.

```
train: ../../datasets_collemboles_yolov5x6/cross/qftza/0.qftza/train/images
val: ../../datasets_collemboles_yolov5x6/cross/qftza/0.qftza/valid/images

nc: 8
names: ['AUTRE', 'CER', 'CRY_THE', 'HYP_MAN', 'ISO_MIN', 'LEP', 'MET_AFF', 'PAR_NOT']

roboflow:
  workspace: theo
  project: r7p3
  version: 4
  license: CC BY 4.0
  url: https://universe.roboflow.com/theo/r7p3/dataset/4
```

Cette fois, on observe le fichier “.yaml” servant au modèle entraîné sur les collemboles avec les labels contenu dans “names” et localisation des dossiers utiles à l'entraînement (train / val) ainsi que certain autre renseignement relatif à la propriété.

## Perspectives / Travaux à finir

### Travaux futurs

**Active Learning** L'Active Learning actuel doit être amélioré. Il sera basé sur le “least Confidence”, une probabilité calculée comme suit :  $1 - (P1 - P2)$ , la différence de probabilité entre les deux premières classes reconnues par le modèle pour un objet. Ensuite, l'Aggrégation de toutes les “least Confidences” des images sera effectuée, et seules les images ayant les Aggrégations les plus élevées seront conservées. Cela nécessitera des modifications directes dans le modèle Yolo pour inclure également la distribution de probabilité nécessaire à l'opération “last Confidence”.

L'Active Learning nécessite un nombre minimal d'images avant de pouvoir être activé, ce qui n'est pas encore possible avec la configuration actuelle de l'interface web. Pour résoudre ce problème, une nouvelle table dans la base de données sera créée pour stocker toutes les images en attente d'Active Learning, qui démarrera dès que 20 images seront enregistrées en base de données. De plus, chaque image sélectionnée aura un score et les images seront triées à chaque cycle pour que celles avec les scores les plus élevés soient annotées en premier.

Pour cette partie, j'ai pu avoir les outils pour commencer durant la semaine du rendu, après avoir examiné les solutions possibles l'importation du modèle devra probablement changer. Voici une première partie du futur Active learning.

```

filenames_val = []
valid = []
for (im, targets, path, shapes) in tqdm(dataloader):
    filenames_val.append(os.path.basename(path[0]))

    im = im.to(device, non_blocking=True)
    im = im.half() if False else im.float()
    im /= 255

    p = model(im)
    p = non_max_suppression(p)
    p = np.array([array[6:-2].cpu() for array in p[0]])
    if len(p) == 0:
        valid.append(0)
        continue
    #tri décroissant
    sorted_probs = np.sort(p, axis=1)[:, ::-1]

    #calcul de probabilités
    least_margins = 1 - (sorted_probs[:, 0] - sorted_probs[:, 1])

    #aggregation
    summ = np.sum(least_margins)
    valid.append(summ)

#fonction permettant de triée et sélectionné uniquement les x images choisies
def top_indices(lst, m):
    indexed_list = [(val, idx) for idx, val in enumerate(lst)]
    sorted_list = sorted(indexed_list, key=lambda x: x[0], reverse=True)
    top_indices = [idx for val, idx in sorted_list[:m]]

    top = []

```

```
for i in range(len(lst)):  
    top.append(-1 if i in top_indices else 1)  
  
return top  
  
results = top_indices(np.array(valid), topm)
```

Dans le cadre de la collecte de données, l'ajout de dossier contenant plusieurs images directement l'interface web nécessitera une modification de la page de détection pour afficher les résultats de chaque image téléchargée individuellement.

**Fine Tuning** Quelques éléments manquent encore pour finaliser le Fine Tuning, notamment le script d'entraînement qui devra automatiquement lancer l'entraînement du modèle après un certain nombre d'images annotées. Une comparaison devra être effectuée pour garantir que le modèle utilisé est toujours le meilleur après chaque cycle de Fine Tuning. De plus, l'Active Learning devra également être relancé après chaque cycle de Fine Tuning, car un changement de modèle entraînera également un changement dans la distribution des probabilités et donc des priorités des images à annoter en premier.

### Hébergement serveur/ Style général

La dernière partie de ce stage se fera sur serveur, l'interface développé tout au long de ce stage devra être hébergé sur serveur, impliquant probablement quelques adaptations en vue de correction de bug. Le modèle entrainé devra aussi être hébergé, mais sur un serveur différent de l'interface, il faudra donc créer une communication entre ces serveurs pour avoir un projet fonctionnel et abouti.

Aussi le style général de l'interface semble un trop neutre et sans âme, une re stylisation partielle semble ainsi être une bonne idée. La charte graphique va en conséquence sûrement bouger dans les prochaines semaines, Néanmoins l'accent sera mis sur la fonctionnalité du site et non son esthétique comme vu avec mon tuteur, cette partie passera donc en dernier, car jugée optionnel.

### Limites

Malgré les nombreuses notions acquises en informatique, mathématiques, développement web et recherche d'informations au cours de ma licence, j'ai rencontré des difficultés avec les concepts du Deep Learning. Les recherches furent souvent longues et peu fructueuses, le sujet n'ayant pas été abordé suffisamment en licence pour bien comprendre les enjeux et mettre en place cette interface.

De plus, des difficultés d'encadrement ont été ressenties pendant le stage. Mon tuteur étant légitimement très occupé par ses responsabilités, plusieurs parties du projet ont démarré avec du retard. L'attente de nouvelles missions et de retours sur le travail prenait généralement plusieurs jours, entraînant des erreurs et des pertes de temps évitables avec des points d'étape plus réguliers.

## Conclusion

En conclusion, cette opportunité de stage m'a permis de découvrir et de mettre en pratique divers concepts et outils liés au développement web et au Deep Learning. Cette expérience m'a non seulement enrichi sur le plan technique, mais m'a aussi permis de mieux comprendre le travail de recherche en laboratoire.

L'interface web créée sera un outil précieux pour les chercheurs, facilitant l'identification des espèces et améliorant continuellement le modèle de détection grâce aux annotations fournies par les utilisateurs experts. Il me reste donc encore trois semaines pour la rendre parfaitement fonctionnelle, je compte alors redoubler d'effort afin de mener ce projet à bout en créant un produit fonctionnel et utilisable par chaque passionné et expert.

## Bibliographie

Oriol, Théo, et al. "Automatic Identification of Collembola with Deep Learning Techniques." *Ecological Informatics*, vol. 81, 1 July 2024, p. 102606, [www.sciencedirect.com/science/article/pii/S1574954124001481?via%3Dihub](http://www.sciencedirect.com/science/article/pii/S1574954124001481?via%3Dihub), <https://doi.org/10.1016/j.ecoinf.2024.102606>. Accessed 15 June 2024. "Centre d'Ecologie Fonctionnelle et Evolutive - ACCUEIL." Cnrs.fr, 2019, [www.cefe.cnrs.fr/fr/](http://www.cefe.cnrs.fr/fr/). Accessed 15 June 2024.

"Comment Entraîner YOLOv5 Sur Un Dataset, Étape Par Étape — Picsellia." [Www.picsellia.fr, www.picsellia.fr/post/comment-entrainer-yolov5-dataset-personnalise](http://Www.picsellia.fr, www.picsellia.fr/post/comment-entrainer-yolov5-dataset-personnalise). Accessed 15 June 2024.

contributors, Mark Otto, Jacob Thornton, and Bootstrap. "Get Started with Bootstrap." [Getbootstrap.com, 2023, getbootstrap.com/docs/5.3/getting-started/introduction/](http://Getbootstrap.com, 2023, getbootstrap.com/docs/5.3/getting-started/introduction/).

Flask. "Welcome to Flask — Flask Documentation (3.0.x)." [Flask.palletsprojects.com, 2010, flask.palletsprojects.com/en/3.0.x/](http://Flask.palletsprojects.com, 2010, flask.palletsprojects.com/en/3.0.x/).

Jocher, Glenn. "Ultralytics/Yolov5." GitHub, 21 Aug. 2020, [github.com/ultralytics/yolov5](http://github.com/ultralytics/yolov5).

Kassel, Raphael. "Fine-Tuning : Qu'est-Ce Que C'est ? À Quoi Ça Sert En IA ?" Formation Data Science | DataScientest.com, 19 Feb. 2024, [datascientest.com/fine-tuning-tout-savoir](http://datascientest.com/fine-tuning-tout-savoir). Accessed 15 June 2024.

Numérique et Informatique à Mounier. "TUTORIEL FLASK #1 : Découverte Du Framework, Routes & Templates." YouTube, 16 Feb. 2022, [www.youtube.com/watch?v=Ihp\\_cG7c2Rk](http://www.youtube.com/watch?v=Ihp_cG7c2Rk). Accessed 15 June 2024.

—. "TUTORIEL FLASK #2 : Transmettre Des Paramètres Dans L'URL." YouTube, 16 Feb. 2022, [www.youtube.com/watch?v=lvxqvNXniVc&list=PLV1TsfPiCx8PXHsHeJKvSSC8zfi4Kvcfs&index=2](http://www.youtube.com/watch?v=lvxqvNXniVc&list=PLV1TsfPiCx8PXHsHeJKvSSC8zfi4Kvcfs&index=2). Accessed 15 June 2024.

—. "TUTORIEL FLASK #3 : Formulaires, Méthodes GET et POST." YouTube, 7 Mar. 2022, [www.youtube.com/watch?v=FdA1P7dY\\_18&list=PLV1TsfPiCx8PXHsHeJKvSSC8zfi4Kvcfs&index=3](http://www.youtube.com/watch?v=FdA1P7dY_18&list=PLV1TsfPiCx8PXHsHeJKvSSC8zfi4Kvcfs&index=3). Accessed 15 June 2024.

—. "TUTORIEL FLASK #4 : Fichiers Statiques & Héritage de Templates." YouTube, 25 Mar. 2022, [www.youtube.com/watch?v=urp\\_b3bWcfE&list=PLV1TsfPiCx8PXHsHeJKvSSC8zfi4Kvcfs&index=4](http://www.youtube.com/watch?v=urp_b3bWcfE&list=PLV1TsfPiCx8PXHsHeJKvSSC8zfi4Kvcfs&index=4). Accessed 15 June 2024.

—. "TUTORIEL FLASK #5 : Les Sessions Pour Mémoriser Des Informations." YouTube, 26 Mar. 2022, [www.youtube.com/watch?v=QAhZ8nmmYxw&list=PLV1TsfPiCx8PXHsHeJKvSSC8zfi4Kvcfs&index=5](http://www.youtube.com/watch?v=QAhZ8nmmYxw&list=PLV1TsfPiCx8PXHsHeJKvSSC8zfi4Kvcfs&index=5). Accessed 15 June 2024.

—. "TUTORIEL FLASK #6 : Le Jeu Du Nombre Mystère." YouTube, 1 Apr. 2022, [www.youtube.com/watch?v=TZGcVV6](http://www.youtube.com/watch?v=TZGcVV6). Accessed 15 June 2024.

PyTorch. "PyTorch." Pytorch.org, 2023, [pytorch.org/](http://pytorch.org/).

Sylvestre Apetcho. "Créez Votre Propre Application Web de Détection d'Objets Avec YOLOv5 et Streamlit /Detection APP." YouTube, 3 May 2023, [www.youtube.com/watch?v=dBbZ\\_jJaq7g&t=326s](http://www.youtube.com/watch?v=dBbZ_jJaq7g&t=326s). Accessed 15 June 2024.

—. "Détection d'Objets Avec YOLOv5 /Objects Detection with YOLOv5." YouTube, 25 Oct. 2022, [www.youtube.com/watch?v=NADYX1wAS\\_0](http://www.youtube.com/watch?v=NADYX1wAS_0). Accessed 15 June 2024.

—. "Entraîner Un Modèle de Détection d'Objet | YOLOv8 Des Données Personnelles YOLOv8 for Custom Data." YouTube, 22 Dec. 2023, [www.youtube.com/watch?v=Z8RfzpvMif4](http://www.youtube.com/watch?v=Z8RfzpvMif4). Accessed 15 June 2024.

—. "Image Annotation for Object Detection Using LabelImg." YouTube, 1 Nov. 2022, [www.youtube.com/watch?v=HTpfYMr](http://www.youtube.com/watch?v=HTpfYMr). Accessed 15 June 2024.

Ultralytics. "Guide Complet Pour Ultralytics YOLOv5." Docs.ultralytics.com, [docs.ultralytics.com/fr/yolov5/](http://docs.ultralytics.com/fr/yolov5/). Accessed 15 June 2024.