Guide D'utilisation

nereuscode@gmail.com
GitHub : NEREUS-code/YOTRACO

Manuel d'Instructions

Preparer par: **Yotraco Team**

Table des matières

1. Introduction

YOTRACO est un module Python avancé conçu pour le **suivi d'objets**, le **comptage automatisé** et l'**analyse statistique** dans des flux vidéo en temps réel. Basé sur l'algorithme **YOLO (You Only Look Once)**, ce logiciel offre une solution flexible pour des applications telles que :

- **Surveillance de trafic** : Comptage de véhicules entrants/sortants.
- **Analyse de foules** : Suivi de mouvements dans des espaces publics.
- **Suivi industriel** : Monitoring de pièces sur des chaînes de production.

Ce manuel s'adresse aux **utilisateurs techniques**, **développeurs** et **analystes de données** ayant une connaissance de base en programmation Python et en traitement d'images. Il couvre l'installation, la configuration, l'utilisation avancée et la maintenance du module.

II. Vue d'ensemble du manuel

Ce document est organisé pour guider l'utilisateur à travers toutes les fonctionnalités de YOTRACO :

- **Introduction** : Objectifs, cas d'utilisation et public cible.
- **Démarrage** : Procédures d'installation, dépendances et premier lancement.
- **Configuration requise** : Spécifications techniques minimales et recommandées.
- **Guide d'utilisation** : Fonctionnalités détaillées, paramétrage personnalisé et exemples pratiques.
- **Maintenance** : Mises à jour, gestion des dépendances et bonnes pratiques.
- **Dépannage** : Résolution des erreurs courantes et support technique.
- **Annexes** : Glossaire, ressources externes et contacts.

III. Démarrage

1. Installation du module

Installez YOTRACO via **pip** (gestionnaire de paquets Python) :

```
pip install yotraco
```

Mise à jour vers la dernière version :

```
pip install yotraco --upgrade
```

2. Installation des dépendances

YOTRACO nécessite les bibliothèques suivantes pour fonctionner :

```
pip install opencv-python ultralytics torch numpy pandas matplotlib tqdm
```

Remarque : Pour une accélération GPU, installez **CUDA Toolkit 11.7+** et **PyTorch avec support CUDA**.

3. Premier lancement

Créez un script Python (exemple.py) et exécutez le code suivant :

```
from YOTRACO import Yotraco

# Initialisation du modèle
model = Yotraco(
    model_path="yolov11l.pt",         # Modèle YOLO pré-entraîné
    video_path="chemin/vers/video.mp4", # Vidéo à analyser
    output_dir="resultats",           # Dossier de sauvegarde
    line_position="middle",           # Position de la ligne de comptage
    track_direction="BOTH",           # Direction : IN, OUT ou BOTH
    classes_to_track=[0, 2, 3],       # Classes YOLO (ex: 0=personne, 2=voiture)
    display=True                      # Affichage en temps réel
)

# Traitement de la vidéo
model.process_video()
```

IV. Configuration requise

Configuration minimale

- **Système d'exploitation** : Windows 10/11, Ubuntu 20.04 LTS (64 bits).
- **Processeur** : Intel i5 (génération 8+) ou équivalent AMD.

- **RAM** : 8 Go.
- **Espace disque** : 2 Go (pour les modèles et dépendances).

Configuration recommandée

- **Processeur** : Intel i7/AMD Ryzen 7.
- **RAM** : 16 Go ou plus.
- **GPU** : NVIDIA GTX 1660/RTX 3060 (avec CUDA 11.7+).
- **Python** : Version 3.10+.4. Configuration

V. Guide d'utilisation

Fonctionnalités principales

1. **Suivi en temps réel** :
   - Détection d'objets via YOLO avec une précision ajustable.
   - Attribution d'ID uniques pour le suivi continu des objets.
2. **Comptage automatisé** :
   - Définissez une **ligne virtuelle** (position personnalisable) pour déclencher le comptage.
   - Choix de la direction : **Entrant (IN)**, **Sortant (OUT)**, ou les **deux (BOTH)**.
3. **Personnalisation avancée** :
   - **Filtrage par classes** : Limitez le suivi à des catégories spécifiques (ex: véhicules, piétons).
   - **Paramètres visuels** : Ajustez la taille des boîtes de détection, la couleur des annotations, etc.
4. **Export des données** :
   - Formats supportés : JSON, CSV, Excel, TXT.
   - Structure des données : Timestamp, ID de l'objet, classe, direction, coordonnées.
5. **Visualisation des résultats** :
   - **Graphiques interactifs** : Histogrammes, camemberts, nuages de points.
   - **Vidéo annotée** : Exportez la vidéo traitée avec les détections et comptages superposés.

Exemple d'analyse statistique

```
from YOTRACO import YotracoStats

# Accéder aux statistiques après le traitement
stats = model.stats

# Générer un histogramme comparatif
stats.plot_bar(
    title="Répartition des objets entrants/sortants",
    xlabel="Classes",
    ylabel="Nombre d'objets",
    colors=["#4CAF50", "#F44336"]  # Couleurs personnalisées
)
```

VI. Procédures de maintenance

1. Mises à jour régulières

- **Module YOTRACO** :

```
pip install yotraco --upgrade
```

- **Dépendances** :

```
pip install --upgrade -r requirements.txt
```

2. Bonnes pratiques

- **Sauvegardes** : Archivez régulièrement les fichiers de configuration et les résultats.
- **Journalisation** : Activez les logs pour diagnostiquer les erreurs (`logging_level="DEBUG"`).

3. Optimisation des performances

- **GPU** : Utilisez `device="cuda:0"` pour accélérer les calculs.

- **Résolution vidéo** : Réduisez la taille des frames avec `frame_size=(640, 480)`.

VII. Dépannage

Problèmes fréquents et solutions

Problème	Solution
Erreur d'installation	Vérifiez la version de Python (`python --version`). Désinstallez/reinstallez les dépendances.
Vidéo non trouvée	Utilisez un chemin absolu. Vérifiez les permissions d'accès au fichier.
Performances lentes	Activez CUDA, réduisez la résolution ou utilisez un modèle YOLO plus léger.
Données non exportées	Vérifiez que le dossier de sortie existe et est accessible en écriture.

Support technique

Contactez en incluant :

- La version de YOTRACO (`pip show yotraco`).
- Le message d'erreur complet et les logs.
- Un exemple minimal de code reproduisant le problème.

VIII. Annexes

Glossaire

- **YOLO** : Algorithme de détection d'objets en temps réel.
- **Ligne de comptage** : Zone virtuelle déclenchant un compteur lors du franchissement.
- **ID de suivi** : Identifiant unique attribué à chaque objet détecté.

Ressources complémentaires

- Documentation Ultralytics YOLO :
- Dépôt GitHub :

IX. Conclusion

YOTRACO est un outil puissant et polyvalent pour le suivi et l'analyse quantitative d'objets dans des flux vidéo. Grâce à son intégration transparente avec YOLO et ses fonctionnalités personnalisables, il s'adapte à des besoins variés, de la recherche académique à l'automatisation industrielle. Pour maximiser son potentiel, consultez régulièrement les mises à jour et participez à la communauté via notre dépôt GitHub.

Pour toute question technique ou suggestion d'amélioration, contactez notre équipe à .

Merci de votre confiance dans YOTRACO ! 🚀