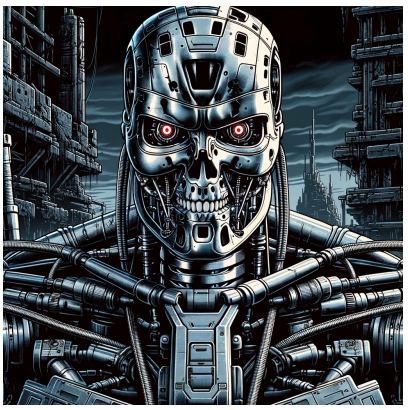
Étude de Cas : Développement d'un Système de Détection d'Objets en Temps Réel v4



Objectif de l'Étude

L'objectif de cette étude de cas est de développer un système de détection d'objets en temps réel similaire à la "vision de Terminator", utilisant Python avec des bibliothèques telles que PyTorch, OpenCV et NumPy. Les étudiants apprendront à appliquer des modèles de vision par ordinateur pré-entraînés, à traiter des vidéos en temps réel et à afficher des informations contextuelles sur les objets détectés.

Contexte

La détection d'objets est une tâche fondamentale en vision par ordinateur, qui permet de reconnaître et de localiser des objets dans des images ou des vidéos. Dans ce projet, les étudiants utiliseront le modèle Faster R-CNN, un des modèles de détection d'objets les plus performants, pré-entraîné sur le dataset COCO (Common Objects in Context).

Exemple :

Réalisation :
https://www.youtube.com/watch?v=FokZs5dp-™
Extrait du film :
https://www.youtube.com/watch?v=zzcdPA6qYAU

Outils et Technologies

- Python : Langage de programmation.
- PyTorch : Bibliothèque de machine learning pour la construction et l'entraînement de réseaux de neurones.

- Torchvision : Package de PyTorch pour les visionneurs, contenant des modèles pré-entraînés et des outils de prétraitement.
- OpenCV : Bibliothèque de vision par ordinateur et de machine learning.
- NumPy : Bibliothèque pour le calcul scientifique avec Python.
- Pillow (PIL) : Bibliothèque de traitement d'images pour Python.

Travail préparatoire / préable

Documentez les technologies citées ci-dessus et incorporez cette documentation à votre CR du jour.

Description du Projet

1. Chargement et Configuration du Modèle :
    - Utiliser le modèle Faster R-CNN de `torchvision.models.detection` pré-entraîné sur COCO.
    - Configurer le modèle en mode évaluation.
2. Traitement de la Vidéo en Temps Réel :
    - Ouvrir une vidéo à l'aide d'OpenCV.
    - Lire la vidéo frame par frame.
    - Convertir chaque frame de BGR à RGB pour le traitement.
3. Détection d'Objets sur les Frames :
    - Transformer les images en tenseurs à l'aide de transformations Torchvision.
    - Appliquer le modèle pour détecter les objets dans les images.
    - Dessiner des boîtes englobantes et étiqueter chaque objet détecté.
4. Amélioration Visuelle :
    - Appliquer un filtre rouge pour simuler la "vision de Terminator".
    - Afficher les labels de manière à éviter les superpositions en gérant l'espace vertical entre les étiquettes.
5. Sortie et Sauvegarde :
    - Convertir les images traitées de RGB à BGR pour les sauvegarder ou les afficher avec OpenCV.
    - Écrire les frames traitées dans un fichier de sortie vidéo.
6. Tests et Validation :
    - Tester le système avec différentes vidéos pour évaluer la performance et la précision de la détection.
    - Identifier et corriger les erreurs potentielles, telles que les indices de catégorie hors limites.

Livrables

- Code Source Complet : Un script Python implémentant le système de détection d'objets.
- Rapport de Projet : Un document expliquant les méthodes utilisées, les défis rencontrés et les solutions implémentées.
- Vidéo de Démonstration : Une vidéo traitée montrant les résultats de la détection d'objets.

Évaluation

Les étudiants seront évalués sur :

- La correcte mise en œuvre des fonctionnalités requises.
- La qualité et la robustesse de leur code.
- Leur capacité à résoudre les problèmes et à optimiser la performance du système.
- Rendu au format pdf illustré, commenté et professionnel.