

AI-Project: Entraînement et Test du Modèle Mask R-CNN pour la Détection de Montres

November 24, 2024

Introduction

Ce projet utilise **Mask R-CNN** pour entraîner un modèle de détection d'objets et de segmentation d'instances pour des images de montres.

Prérequis

Avant de commencer, assurez-vous de disposer des éléments suivants :

- Cloner le dépôt **Mask R-CNN**.
- Avoir le fichier `mask_rcnn_coco.h5` à la racine du répertoire **Mask R-CNN**.
- **Python 3.7** installé sur votre machine.
- Installer les dépendances nécessaires avec la commande :

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

- Décompresser l'archive du dataset :
 - Le dataset doit être organisé avec des dossiers `train` et `val` contenant les images et leurs annotations au format JSON.
 - Structure attendue :

```
dataset_watch/  
  train/  
    image1.jpg  
    image1.json  
    ...  
  val/  
    image2.jpg  
    image2.json  
    ...
```

- Construire le dataset : Assurez-vous que le dataset contient au moins 150 images (100 pour l'entraînement et 50 pour la validation) avec des annotations appropriées.

Commande d'entraînement

```
python train.py
```

Commande de test

Lancer Jupyter Notebook et exécuter le notebook `test.ipynb`.

Entraînement

Dans ce projet, nous avons entraîné deux modèles :

Modèle 1 : Hyperparamètres Utilisés

- Taux d'apprentissage (`LEARNING_RATE`) : 0.001
- Époques d'entraînement : 20
- Backbone : ResNet-50
- Nombre d'images par GPU : 2
- Mode de redimensionnement des images : `square`
- Dimensions minimales et maximales des images : 256 x 256
- `STEPS_PER_EPOCH` : 100

Modèle 2 : Hyperparamètres Utilisés

- Taux d'apprentissage (LEARNING_RATE) : 0.001
- Époques d'entraînement : 20
- Backbone : ResNet-50
- Nombre d'images par GPU : 2
- Mode de redimensionnement des images : `square`
- Dimensions minimales et maximales des images : 256 x 256
- STEPS_PER_EPOCH : 25

Résultats

Des logs sont disponibles dans le répertoire `logs` pour les deux modèles.

Entraînement avec 100 steps par epoch

100/100 [=====] - 1079s 11s/step - loss: 0.1594 - rpn_c

Les résultats sont enregistrés dans le répertoire `logs`.

Entraînement avec 25 steps par epoch

25/25 [=====] - 388s 16s/step - loss: 0.3681 - rpn_clas

Les résultats sont enregistrés dans le répertoire `logs`.

Matrices de Confusion

Les matrices de confusion pour les deux configurations d'entraînement (25 steps et 100 steps) sont disponibles dans les fichiers suivants :

- `matrice_de_confusion_25_steps.png`
- `matrice_de_confusion_100_steps.png`

Elles montrent la précision du modèle pour classer les objets en deux catégories : *"watch"* et *"background"*.

Graphiques de Loss

Les courbes de *loss* pour chaque modèle (25 et 100 steps par epoch) sont affichées dans un graphique montrant la diminution de la perte au fil des époques. Le graphique est disponible sous le nom `graph_nombre-d'epoch_steps.png`.