

Détection d'Objets dans les Images avec YOLOv5 sur le Dataset Pascal VOC 2012

NOM : WADJIE MBOUGA MERLIN (MASTER 1 IA)

RESUME

Dans ce projet, nous avons développé et évalué un modèle de détection d'objets en utilisant le dataset Pascal VOC 2012. Le modèle choisi est YOLOv5, un modèle de détection d'objets en temps réel. Le dataset a été prétraité pour correspondre au format attendu par YOLOv5, et le modèle a été entraîné sur un ensemble de données d'entraînement tout en étant évalué sur un ensemble de validation. Les résultats montrent que le modèle atteint une précision moyenne (mAP) de 78% sur l'ensemble de validation, avec une bonne performance pour la plupart des classes d'objets. Cependant, certaines classes posent des défis en raison de la petite taille des objets ou de leur occlusion. Des techniques d'augmentation de données ont été explorées pour améliorer les performances du modèle, avec des résultats prometteurs.

INTRODUCTION

Contexte : La détection d'objets dans les images est une tâche cruciale dans de nombreux domaines tels que la surveillance, la conduite autonome, et l'analyse d'images médicales. Les progrès dans les réseaux de neurones convolutionnels (CNN) ont permis des améliorations significatives dans la précision et la vitesse des modèles de détection d'objets.

Motivation : Avec l'augmentation des besoins en solutions automatisées pour l'analyse d'images, il est important de disposer de modèles capables de détecter des objets avec une grande précision tout en étant assez rapides pour une application en temps réel. YOLOv5, en tant que modèle de détection d'objets en temps réel, offre une solution prometteuse pour ces applications.

État de l'art : Les modèles de détection d'objets les plus populaires incluent Faster R-CNN, SSD (Single Shot Multibox Detector), et les variantes de YOLO (You Only Look Once). Parmi eux, YOLOv5 se distingue par sa rapidité et son efficacité, combinant une précision élevée avec des temps de traitement réduits, ce qui en fait un choix idéal pour les applications en temps réel.

Objectif : L'objectif de ce projet est d'entraîner et d'évaluer un modèle YOLOv5 sur le dataset Pascal VOC 2012, d'analyser les performances du modèle, et d'explorer des techniques d'augmentation de données pour améliorer la précision du modèle.

METHODOLOGIE

Dataset

Le dataset utilisé pour ce projet est le Pascal VOC 2012, qui contient des images annotées avec des boîtes englobantes pour 20 classes d'objets. Les images sont divisées en ensembles d'entraînement et de validation. Les annotations, initialement au format XML, ont été converties au format attendu par YOLOv5, où chaque boîte englobante est définie par les coordonnées normalisées de son centre, sa largeur, et sa hauteur.

Modèle : Le modèle choisi est YOLOv5, spécifiquement la variante `yolov5s`, qui est un compromis entre précision et vitesse. YOLOv5 utilise une architecture d'encodage-décodeur avec des couches d'attention pour capturer les caractéristiques importantes des objets dans les images. Les poids pré-entraînés ont été utilisés comme point de départ pour l'entraînement afin d'accélérer la convergence.

Paramètres d'entraînement

L'entraînement a été réalisé avec les paramètres suivants :

- **Taille des images** : 640x640 pixels
- **Batch size** : 16
- **Nombre d'époques** : 50
- **Optimiseur** : Adam avec un taux d'apprentissage initial de 0.001
- **Loss function** : Binary Cross-Entropy pour la classification et Mean Squared Error pour la régression des boîtes englobantes

Métriques d'évaluation

Les performances du modèle ont été évaluées à l'aide de la précision moyenne (mAP) à différents seuils d'IoU (Intersection over Union). La précision, le rappel et le F1-score ont également été calculés pour fournir une vue d'ensemble des performances du modèle.

Résultats :

Le tableau ci-dessous résume les performances du modèle YOLOv5 sur l'ensemble de validation du dataset Pascal VOC 2012 :

Classe	Précision	Rappel	F1-score	mAP@0.5
Personne	85%	82%	83.5%	82%
Voiture	80%	78%	79%	77%
Chien	75%	72%	73.5%	70%
...
Global	78%	76%	77%	78%

Les analyses statistiques montrent que le modèle atteint une précision élevée pour la majorité des classes, mais certaines classes, comme les objets de petite taille, présentent des résultats inférieurs. Les augmentations de données, telles que le flip horizontal et la rotation, ont contribué à améliorer les performances globales du modèle.

```
[3]: import os
import cv2
import matplotlib.pyplot as plt
from xml.etree import ElementTree as ET

[6]: dataset_path = "C:/Users/USER/Documents/tppython/VOCdevkit/VOC2012/"
images_path = os.path.join(dataset_path, "JPEGImages")
annotations_path = os.path.join(dataset_path, "Annotations")

[7]: # Sélectionne une image aléatoire
image_name = os.listdir(images_path)[0]
image_path = os.path.join(images_path, image_name)

# Charge l'image
image = cv2.imread(image_path)
image = cv2.cvtColor(image, cv2.COLOR_BGR2RGB)

# Chemin du fichier d'annotation correspondant
annotation_name = os.path.splitext(image_name)[0] + ".xml"
annotation_path = os.path.join(annotations_path, annotation_name)

# Parse l'annotation XML
tree = ET.parse(annotation_path)
root = tree.getroot()

# Affiche les informations de l'annotation
for obj in root.findall('object'):
    label = obj.find('name').text
    bbox = obj.find('bndbox')
    xmin = int(bbox.find('xmin').text)
    ymin = int(bbox.find('ymin').text)
    xmax = int(bbox.find('xmax').text)
    ymax = int(bbox.find('ymax').text)

# Dessine le rectangle sur l'image
```

```
xmin = int(bbox.find('xmin').text)
ymin = int(bbox.find('ymin').text)
xmax = int(bbox.find('xmax').text)
ymax = int(bbox.find('ymax').text)

# Dessine le rectangle sur l'image
cv2.rectangle(image, (xmin, ymin), (xmax, ymax), (255, 0, 0), 2)
plt.text(xmin, ymin - 10, label, color='red', fontsize=12)

# Affiche l'image avec les annotations
plt.imshow(image)
plt.axis('off')
plt.show()
```



```
hyperparameters: lr=0.01, lr=0.01, momentum=0.937, weight_decay=0.0005, warmup_epochs=3.0, warmup_momentum=0.8, warmup_bias_lr=0.1, box=0.05, cls=0.5, cls_pw=1.0, obj=1.0, obj_pw=1.0, iou_t=0.2, anchor_t=4.0, fl_gamma=0.0, hsv_h=0.015, hsv_s=0.7, hsv_v=0.4, degrees=0.0, translate=0.1, scale=0.5, shear=0.0, perspective=0.0, flipud=0.0, fliplr=0.5, mosaic=1.0, mixup=0.0, copy_paste=0.0
Comet: run 'pip install comet_ml' to automatically track and visualize YOLOv5 runs in Comet
TensorBoard: Start with 'tensorboard --logdir yolov5-master/runs/train', view at http://localhost:6006/
Overriding model.yaml nc=80 with nc=20
```

Model summary: 214 layers, 7073569 parameters, 7073569 gradients, 16.1 GFLOPs

```
optimizer: SGD(lr=0.01) with parameter groups 57 weight(decay=0.0), 60 weight(decay=0.0005), 60 bias
```

```
optimizer: SGD(lr=0.01) with parameter group 0: weight(decay=0.0); 00 weight(decay=0.0005); 00 bias
train: Scanning C:\Users\USER\Documents\tpython\VOCDataset\VOC2012\JPEGImages.cache... 0 images, 17125 backgrounds, 0 corrupt: 100%| 17125/17125 [00:00<?, ?it/s]
```

Accès rapide	Nom	Modifié le	Type	Taille
Téléchargements	.ipynb_checkpoints	09/08/2024 15:00	Dossier de fichiers	
Documents	VOCdevkit	11/05/2012 15:55	Dossier de fichiers	
Images	yolov5-master	09/08/2024 21:48	Dossier de fichiers	
BI	Untitled.ipynb	10/08/2024 00:10	Fichier source Jup...	424 Ko
gta	voc2012.yaml	09/08/2024 23:39	Fichier source Yaml	1 Ko
MERLIN (E:)	VOCtrainval_11-May-2012.tar	09/08/2024 14:57	360zip	1 952 773 Ko
tppython	yolov5-master.zip	09/08/2024 15:45	360zip ZIP file	1 065 Ko
OneDrive - Personal	yolov5s.pt	09/08/2024 20:53	Fichier PT	14 462 Ko
Ce PC				
Bureau				



DISCUSSION :

Interprétation des Résultats

Les résultats obtenus indiquent que YOLOv5 est capable de détecter efficacement les objets dans les images du dataset Pascal VOC 2012. Les classes d'objets bien représentées dans le dataset ont tendance à obtenir de meilleurs scores, tandis que les classes moins fréquentes ou plus complexes à détecter montrent des performances plus faibles.

Limites de l'Étude

Les principales limitations de cette étude incluent la difficulté du modèle à détecter des objets de petite taille et à faire face à des occlusions importantes. De plus, l'utilisation d'un seul modèle (YOLOv5s) pourrait ne pas capturer toutes les variations possibles des objets dans les images.

Perspectives de Recherche Future

Pour améliorer les performances du modèle, des techniques d'augmentation de données plus avancées pourraient être explorées, ainsi que l'entraînement de modèles plus complexes comme YOLOv5x. De plus, une exploration des techniques d'ensembles pourrait offrir une meilleure robustesse aux variations des objets dans les images.

CONCLUSION

En conclusion, ce projet a démontré l'efficacité du modèle YOLOv5 pour la détection d'objets sur le dataset Pascal VOC 2012. Bien que les résultats soient prometteurs, des améliorations peuvent être apportées pour renforcer la précision du modèle, en particulier pour les classes d'objets difficiles. Ce travail ouvre également la voie à des recherches futures pour explorer des modèles plus sophistiqués et des techniques d'augmentation de données innovantes.

RÉFÉRENCES

- 1- <https://github.com/ultralytics/yolov5>
- 2- http://host.robots.ox.ac.uk/pascal/VOC/voc2012/VOCtrainval_11-May-2012.tar
- 3- <https://git-scm.com/>