

Object detection using R-CNN

pre-trained on COCO dataset

SADADOU Mohamed

15/10/2023

Abstract

L'objectif de ce mini projet est de Tirer parti de l'architecture Faster R-CNN avec un backbone ResNet-50, pré-entraîné sur l'ensemble de données COCO et de l'adapter afin de réussir à détecter les masques et réussir à les classifier en : avec masque: classe 1, 'sans masque':classe 2, 'masque incorrect':3 le fond étant mis classe 0.

Dans le but d'évaluer le modèle plusieurs métriques ont été implémentés ainsi que l'implémentation de Grad-CAM dans le but de vérifier les zones intervenantes dans la décision.

Dataset

Le dataset se compose de deux parties, les images stocker dans un dossier sur drive ainsi qu'une deuxième partis étant des fichiers xml comportant les labels et bounding boxes des détections,

le dataset est composé de 853 imges au format "RGBA" accompagné de 853 fichiers XML chacun représentatif de la présence ou non des masque sur chaque images avec les positions des boîtes englobantes ainsi que de la classe d'appartenance de cette dernière, les images et fichiers XML sont numéroté de 0 à 852, chaque XML correspondant à l'image du même ID.

Réponse question 1 : Bounding box format

Coordinates of the top-left and bottom-right corners:

(x_min, y_min, x_max, y_max)

Répartition des données

Total	A_mask	S_mask	Class 1	Class 2	Class 3	Class1_2	Class1_3	Class2_3
853	768	85	516	60	21	222	30	4
	90.03%	9.96%	60.49%	7.03%	2.46%	26.02%	3.51%	0.46%

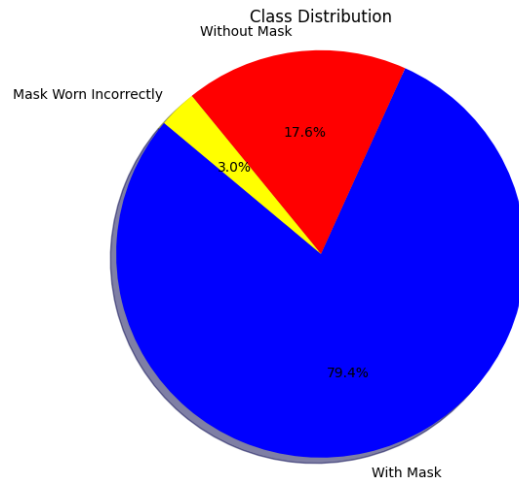
A_mask : au moins un masque sur l'image.

S_mask : Sans aucun masque sur l'image.

Classe i : image contenant uniquement la classe i.

Classe i_j : Images comportant la classe i et la classe j en même temps.

Ces données ont été extraites du data set, la fonction est présente dans le code.



Réponse question 2 :

Déséquilibre de classe :

La classe 1 est significativement représentée avec 79,37 % du total des instances. Ce qui crée automatiquement un biais, le modèle aura dès lors plus de facilité à classer en classe 1 et sera également plus précis pour cette classe.

La classe 2 est modérément représentée avec 17,61%. toutefois ça reste relativement faible comparé à l'écrasante majorité de la classe 1 ce qui garde le biais vers la classe 1.

La classe 3 est sous-représentée avec seulement 3,02 %. ce qui complique grandement la tâche que ce soit pour l'apprentissage (trop peu d'exemple de train, de validation) ou pour l'évaluation, le modèle aura du mal à détecter correctement cette classe n'ayant pas suffisamment appris.

Ce problème peut être résolu en utilisant des techniques d'augmentation de données en ciblant les classes sous représenté, ou en utilisant un poids de classes lors de l'entraînement

Biais au niveau de l'image :

Il y a 768 images avec au moins un masque et 85 images sans masque. Cela signifie que l'ensemble de données est orienté vers les images avec des masques. Ceci implique dès lors que le modèle aura tendance à partir du principe qu'il y a un masque sur l'image.

Il y a 516 images avec uniquement la classe 1 , 60 images avec uniquement la classe 2 et 21 images avec uniquement la classe 3. Cela montre un biais en faveur des images qui n'ont que la classe 1. et donc que le modèle aura tendance à prédire excessivement la classe 1.

Réponse question 3 :

Nous avons comme dataset 853 images labellisées.

Cela peut s'avérer suffisant dans le cas d'utilisation d'un modèle pre-entraîner et déjà performant, toutefois dans le cas présent les classes sont déséquilibré c'est pourquoi pour avoir des résultats concluant nous devons mettre en places certaines stratégies :

- Data Augmentation :

Appliquez des transformations aléatoires aux images d'entraînement, telles que des rotations, effet miroir vertical ou horizontal. Cela augmente la diversité des données d'entraînement et permet d'éviter le surapprentissage.

- Utilisé un modèle pré-entraîné performant :

étant notre cas pour cet exercice, toutefois une augmentation de données est fortement recommandé pour cette exercice

réponse question 4 :

La même méthode que pour visualiser la distribution du data set a été implémentée, avec une répartition de 80% / 10% / 10% respectivement pour l'entraînement, la validation et le test. la répartition étant random il y aura des déséquilibre possible. l'ensemble de train a été favorisé dans la répartition car au vu du petit jeu de données il est important d'avoir suffisamment pour l'entraînement .

Training_data

Total	A_mask	S_mask	Class 1	Class 2	Class 3	Class1_2	Class1_3	Class2_3
682	605	77	402	53	20	182	21	4
	88.70%	9.96%	58.94%	7.77%	2.93%	26.68%	3.07%	0.58%

Validation_data

Total	A_mask	S_mask	Class 1	Class 2	Class 3	Class1_2	Class1_3	Class2_3
85	82	3	57	2	1	23	2	0
	88.70%	9.96%	67.05%	2.35%	1.17%	27.05%	2.35%	0%

Test_data

Total	A_mask	S_mask	Class 1	Class 2	Class 3	Class1_2	Class1_3	Class2_3
86	81	5	57	5	0	17	7	0
	94.18%	5.81%	66.27%	5.81%	0%	19.76%	8.13%	0%

Réponse question 5 :

Mon approche de la sélection des hyperparamètres (epochs, batches, learning_rate, momentum) était itérative, garantissant que le modèle n'était ni sous-ajusté ni surajouté. Le changement des hyperparamètres se faisait en surveillant les loss fonctions, ensuite du fine-tuning en modifiant légèrement toujours en surveillant les loss, la validation servait également pour choisir les meilleurs hyperparamètres.

Une méthode Grade CAM a été implémentée pour surveiller les zones rentrant en jeu dans la décision afin d'éviter l'exemple de cours de "l'ours polaire" a partir de [plusieurs sources](#).

Résultats

Metric	Accuracy	Précision	Recall	F1-Score	IoU
Score	65%	93%	68%	78%	0.81

Pour les "TP" ce dernier n'est pris en compte que si il dépasse le seuil fixé ce qui est considéré comme "correctement détecté **ET** classé "

Un seuil de confiance a été fixé à 70% pour les prédictions et un IoU a un minimum de 0.2.