

INFO942 : Apprentissage automatique et fouille de données :

Détection d'objets avec le deep learning :

Introduction :

Ce rapport a pour but l'exploration de la détection d'objets en utilisant des techniques avancées du Deep Learning. Ce sujet qui a une importance essentielle dans le domaine de l'apprentissage automatique est caractérisé aussi par la complexité de son implémentation. L'objectif est de découvrir cet outil et s'approfondir dans ses applications pratiques.

À travers ce rapport, nous étudierons les différentes composantes de la détection d'objets, et nous analyserons les diverses étapes d'implémentation de notre modèle de détection d'objets. En plus, ce rapport expliquera les problèmes techniques rencontrés, les solutions adoptées et les résultats obtenus, tout en offrant un aperçu sur les améliorations possibles sur ce projet.

Présentation du sujet :

La détection d'objets est un processus important dans le domaine du Deep Learning, permettant de reconnaître et classer les objets dans des images. Cette technologie a des nombreuses applications pratiques dans divers domaines, comme la sécurité, la surveillance, la santé et l'industrie.

Nous allons étudier la manière dont les modèles du Deep Learning, en particulier les réseaux de neurones convolutionnels (CNN), peuvent être utilisés pour effectuer cette tâche, et la démarche utilisée.

Le modèle le plus utilisé dans ce contexte est le modèle YOLO (You Only Look Once), mais les modèles YOLO peuvent être assez exigeants en termes de ressources de calcul, surtout pendant l'entraînement, car ils requièrent une grande quantité de mémoire GPU et une puissance de calcul importante. Ce qui fait que l'objectif de ce projet est d'établir une version simplifiée de YOLO effectuant la même tâche de détection d'objets.

Objectifs :

Les objectifs de ce projet s'articulent autour de plusieurs points clés, reflétant les divers aspects de la détection d'objets, qu'on peut classer comme suit :

Détection des objets : Identifier la présence d'objets dans les images, avec une bonne précision et une bonne fiabilité de la détection.

Classification des objets : Classer correctement les objets détectés en catégories prédéfinies, une étape cruciale pour l'interprétation correcte des données visuelles.

Localisation des objets : Évaluer les techniques permettant de déterminer la position exacte des objets détectés dans l'espace de l'image.

Classification Multiclasses : Etablir une classification d'images contenant plusieurs objets de différentes catégories.

Reconnaissance de forme : Reconnaître l'objet présent dans une image indépendamment de sa taille.

En abordant ces objectifs, ce projet vise à construire un modèle permettant de détecter des objets multiples dans une image donnée, et de les classer dans des classes prédéfinies.

Etapas de réalisation :

Pour réaliser ce projet, nous avons dû relever la tâche complexe de la détection d'objets sur les images. Pour y parvenir, nous avons choisi d'aborder le problème par étapes. Dans un premier temps, nous avons développé une version initiale de notre programme axée sur la classification Monoclasse des objets. Et après avoir des bons résultats lors de cette première version, nous envisageons de passer à la version suivante de notre programme qui permet la détection des différents objets présents de notre image indépendamment de leurs formes et leurs positions dans l'image.

Version 1 : Classification des images :

Objectifs :

Les objectifs de cette première partie consistent à établir une bonne classification des images données, et s'adapter aux dimensions des objets présents dans l'image, et bien adapter le modèle pour qu'on obtient une précision meilleure pour pouvoir entamer l'étape suivante de notre projet, on peut détailler les objectifs de cette partie comme suit :

Classification Précise : L'objectif premier est d'assurer une classification précise des objets détectés. Cela signifie que le modèle devrait correctement identifier et catégoriser l'objet dans une image avec le moins d'erreurs possible.

Résilience aux Variations d'Images : notre modèle doit être résistant aux variations dans les images, telles que les changements d'éclairage, les déformations, et les arrière-plans complexes.

Précision importante : Partant de principe que cet algorithme sera la base de notre programme et sera développé après dans la deuxième version. La précision, en tant que mesure de performance, était un objectif primordial lors de cette partie, pour pouvoir partir sur de bonnes bases et aller plus loin avec notre algorithme dans la deuxième partie.

Jeu de données :

Pour réaliser cette première version, nous utilisons un jeu de données composés de 9 classes différentes composés d'une centaine de photos représentant plusieurs objets : Personnes, motos, voitures..., nous utiliserons 70% de nos données pour l'entraînement et 30% pour la validation.

Architecture du Modèle :

Notre modèle utilisé lors de cette partie est constitué de couches convolutionnelles qui sont essentielles pour l'extraction de caractéristiques à partir des images. Ces couches utilisent la normalisation par lots et le MaxPooling pour améliorer l'efficacité et réduire la dimensionnalité.

La fonction d'activation ReLU est appliquée pour introduire la non-linéarité, ce qui permet au modèle de capturer des motifs complexes.

La normalisation par lots est effectuée après chaque couche convolutive pour accélérer la convergence du modèle lors de l'apprentissage.

Le MaxPooling est utilisé pour réduire la dimensionnalité et donc, les exigences en calcul, tout en préservant les caractéristiques pertinentes.

Des couches denses sont implémentées pour la classification. Une couche dense de 256 unités conduit à une couche de sortie avec activation Softmax, qui gère la classification multi-classes.

Hyperparamètres :

Comme hyperparamètres de notre modèle, on trouve :

Dimensions des Images : Les images sont traitées à une résolution de 256×256 pixels.

Taille du Lot (Batch Size) : Fixée à 32, cette taille de lot est choisie pour optimiser l'utilisation de la mémoire et l'efficacité de l'apprentissage.

Nombre de Classes : Il y a 9 classes d'objets que le modèle est entraîné à reconnaître.

Taux d'Apprentissage : Fixé à 0.0001, ce taux est assez faible pour permettre un ajustement fin et éviter de passer à côté de minima locaux pendant l'entraînement.

Époques d'Entraînement : Le modèle est entraîné sur 10 époques, où chaque epoch représente une passe intégrale sur la totalité du jeu de données.

Résultats :

L'algorithme 1 est donc une structure efficace pour la classification d'images, offrant une base solide sur laquelle des améliorations peuvent être apportées pour affiner davantage la précision et la robustesse du modèle.

Notre algorithme a abouti à un bon résultat, et converge en précision à partir de 10 epochs, et donne une précision de 0,8. Cette performance indique que le modèle est capable de classer correctement les objets avec une fiabilité de 80%. Cette précision reste acceptable compte tenu de la complexité à la tâche de la détection et de la classification d'objets dans des images. Les résultats actuels servent de base solide pour les utilisations futures du modèle qui vise la détection d'objets.

Version 2 :

Objectifs :

En plus des objectifs de la première version, l'objectif de cette deuxième version est d'établir un algorithme permettant de détecter les objets sur une image, et classer ces objets sur des classes prédéfinies, ainsi la localisation de ces objets en gardant une précision optimale pour cette version. Contrairement au script précédent, cette version aura une couche de sortie conçue pour prédire plusieurs boîtes englobantes par image, ce qui est un pas vers les fonctionnalités des algorithmes de détection d'objets tels que YOLO.

Jeu de données :

En plus des images png utilisées lors de la version précédente de notre algorithme, on ajoutera l'utilisation des annotations sur cette partie.

En effet, Les annotations sont des métadonnées associées aux images. Elles fournissent des informations étiquetées détaillant l'emplacement des objets dans une image et les catégories auxquelles ils appartiennent. Ils servent à marquer les images avec des informations précises sur l'emplacement et la classification des objets. En délimitant les objets avec des boîtes englobantes, elles permettent aux modèles d'apprentissage automatique de s'entraîner à reconnaître et localiser des objets dans différentes images.

Nous avons aussi comme outil les masques, mais nous n'avons pas pu l'intégrer par manque de temps.

Architecture du Modèle :

Notre deuxième version est composée en plus de la première version de :

Couches de convolution avec 32, 64 et 128 filtres : Pour capturer des caractéristiques à divers niveaux de granularité.

Utilisation de la fonction d'activation ReLU Pour introduire de la non-linéarité.

Application de MaxPooling : Pour réduire la dimensionnalité, ce qui aide à réduire le surajustement et le temps de calcul.

Une couche dense de 512 unités avec activation ReLU : Pour l'interprétation des caractéristiques extraites.

Couche de sortie dense avec activation linéaire Pour prédire les coordonnées des boîtes englobantes et les classes d'objets, permettant au modèle de localiser et identifier les objets dans les images.

Hyperparamètres :

Comme hyperparamètres, on a utilisé :

Dimension des images traitées : 256x256 pixels pour une uniformité et une gestion efficace de la mémoire.

Taille de lot (Batch Size) : Fixée à 32, cette taille de lot est choisie pour optimiser l'utilisation de la mémoire et l'efficacité de l'apprentissage.

Nombre de classes : 9, représentant les différentes catégories d'objets à détecter.

Nombre de boîtes par image : 2, indiquant que chaque image contiendra des annotations pour deux objets.

Époques d'entraînement : 10, déterminant le nombre de fois que l'ensemble de données d'entraînement sera utilisé pour ajuster les poids du modèle.

Résultats actuels :

Variabilité de la Précision :

La précision moyenne du modèle varie entre 0.2 et 0.6, indiquant une performance inégale à travers différents tests ou catégories. Le résultat est acceptable dans certains cas et pas dans d'autres. Il y a une variation notable de la précision à travers les différentes classes d'objets, ce qui pourrait indiquer une possible baisse de performance pour certaines catégories spécifiques.

Performance Temporelle :

Le temps de traitement est long, suggérant que l'efficacité du modèle pourrait être un problème, particulièrement dans des applications en temps réel.

Améliorations possibles :

Si on avait plus de temps sur ce projet, plusieurs améliorations seraient possibles sur les différentes axes du projet, notamment une amélioration de la précision surtout pour les classes qui présentent une précision faible, et une optimisation de la vitesse de calcul, de plus l'extension du jeu de données et l'intégration d'autres classes à notre modèle.

Conclusion :

En somme, le projet illustre la capacité du Deep Learning dans la détection d'objets, malgré une variabilité dans la précision et des contraintes de temps de traitement. Les résultats actuels sont satisfaisants vu le temps consacré au projet. Cette étude établit une base solide pour la compréhension et l'application des CNN en détection d'objets, ouvrant la voie à des applications pratiques dans une variété de domaines tout en relevant les défis associés à cette tâche complexe.