ENSA
ÉCOLE NATIONALE DES SCIENCES APPLIQUÉES
KHOURIBGA

RAPPORT

Master Big Data et Aide à la Décision

Présenté par :

Bella Abedlouahab

Boulanouar Mohamed Amine

Détection des panneaux de signalisation routière marocains

Remerciement

Avant de plonger dans le détail de ce travail académique, nous souhaitons prendre un moment pour exprimer notre gratitude envers ceux qui ont rendu ce parcours à la fois possible et fructueux.

Tout d'abord, nous tenons à adresser nos remerciements les plus sincères à l'ENSA Khouribga, pour nous avoir offert un environnement d'apprentissage stimulant et propice à l'épanouissement intellectuel. Les ressources mises à disposition et le cadre académique exceptionnel ont été des piliers fondamentaux dans la réalisation de ce travail.

Nous sommes particulièrement reconnaissants envers Pr. Youssef El hadfi notre superviseur, pour son accompagnement précieux, ses conseils éclairés et son soutien sans faille tout au long de ce parcours. Sa passion pour le sujet, son expertise et sa capacité à motiver ses étudiants ont été une source d'inspiration constante pour nous.

Nous souhaitons aussi exprimer notre gratitude à notre famille et à nos amis pour leur amour, leur soutien inconditionnel et leurs encouragements. Leur foi en nos capacités et leur présence rassurante ont été des atouts indispensables pour surmonter les défis rencontrés et atteindre nos objectifs.

Résumé

La conduite autonome est l'un des domaines de recherche les plus intéressants de notre époque, et la détection des panneaux de signalisation routière constitue un problème très important et crucial dans ce domaine. Dans ce projet, nous proposons d'explorer l'architecture YOLO et sa compatibilité pour résoudre ce problème. L'objectif est de localiser et de classer les panneaux de signalisation dans des scènes urbaines naturelles. Le principal défi à relever dans ce problème est la reconnaissance de petites cibles dans un arrière-plan complexe et étendu.

Pour aborder cette tâche, nous avons décidé de mettre en œuvre un modèle de réseau neuronal convolutionnel (CNN), combinant des couches de convolution et de pooling pour extraire les caractéristiques pertinentes des images.

Par ailleurs, nous avons également intégré l'architecture YOLO (You Only Look Once) qui est particulièrement adapté à ce genre de tâche car il permet de détecter et de localiser des objets dans une image en une seule passe, ce qui le rend extrêmement rapide et efficace, même dans des environnements complexes comme ceux rencontrés pour la détection de panneaux de signalisation.

Mots clés :

Conduite autonome, détection d'objets, panneaux de signalisation, YOLO, réseau neuronal convolutionnel, vision par ordinateur, apprentissage profond.

Abstract

Autonomous driving is one of the most fascinating research fields today, and traffic sign detection plays a critical role in ensuring the safety and effectiveness of autonomous systems. This project explores the use of the YOLO (You Only Look Once) architecture for solving this problem. The goal is to localize and classify traffic signs in natural urban scenes. A key challenge is the recognition of small targets in complex and extensive backgrounds.

To address this task, we implemented a convolutional neural network (CNN) model that combines convolutional and pooling layers to extract relevant features from images. Additionally, we integrated the YOLO architecture, which is particularly well-suited for such tasks as it enables fast and efficient object detection and localization in a single pass, even in complex environments typically encountered for traffic sign detection.

Keywords :

Autonomous driving, object detection, traffic sign detection, YOLO, convolutional neural networks, computer vision, deep learning.

Table des matières

Table des figures

No table of figures entries found.

I. Introduction générale

Au cours des dernières années, les voitures autonomes ont gagné en popularité au sein de la communauté de recherche. Une voiture autonome est capable de percevoir son environnement et de naviguer sans intervention humaine. L'un des aspects les plus importants pour une voiture autonome est la « vision », c'est-à-dire la capacité de reconnaître et de détecter les panneaux de signalisation routière. La détection des panneaux de signalisation est une tâche complexe en raison d'obstacles tels que l'occlusion, les conditions d'éclairage changeantes, la perspective de la caméra et d'autres facteurs qui surviennent dans des scènes naturelles. La présence de multiples panneaux dans un même champ de vision constitue également un défi supplémentaire à surmonter.

Jusqu'à ces dernières années, les approches conventionnelles pour la détection des panneaux de signalisation étaient basées sur certains algorithmes traditionnels. Les techniques de détection des panneaux de signalisation utilisaient généralement des caractéristiques sélectionnées manuellement pour obtenir les propositions de régions, puis des classificateurs étaient formés pour éliminer les négatifs. Les réseaux de neurones convolutionnels profonds (CNN) sont désormais appliqués à la reconnaissance d'images et à la détection d'objets, car ils offrent des performances optimales en termes de vitesse et de précision. Les CNN n'ont pas besoin de caractéristiques prédéfinies, car ils apprennent naturellement les caractéristiques généralisées. Les CNN, qui ont gagné une immense popularité dans diverses tâches de classification, sont maintenant étendus à des tâches plus complexes, telles que la détection d'objets, qui combine à la fois la classification et la localisation.

L'un des facteurs les plus importants dans la détection en temps réel des panneaux de signalisation est la latence au moment du test. Les CNN n'étaient pas considérés comme réalisables pour la détection en temps réel des panneaux de signalisation en raison de leur complexité computationnelle. Cependant, l'évolution des GPU (Graphics Processing Unit) a ouvert la voie à l'utilisation des CNN pour cette tâche, grâce à leurs hautes performances de calcul. Nous avons besoin d'un modèle capable de détecter et de classer les panneaux en temps réel. Dans ce travail, nous explorons l'architecture YOLO, qui offre une détection et une classification d'objets en temps réel à environ 45 images par seconde. Cela s'avère particulièrement adapté à notre problème, qui nécessite à la fois rapidité et précision. Nous avons développé une architecture basée sur YOLO pour la détection des panneaux de signalisation s dans notre jeu de données de panneaux marocains.

II. Les panneaux routiers

La signalisation routière constitue un élément fondamental de tout système de circulation. Les panneaux de signalisation routière sont généralement implantés de part et d'autre sur nos routes, indiquent les règles de la circulation établies pour permettre aux véhicules et aux piétons de se déplacer en toute sécurité sur les routes.

Les usagers de la route doivent d'être bien sensés de la signalisation routière qui demeure aujourd'hui un des éléments phares de la prévention routière. Grâce aux signaux routière, les conducteurs sont informés des règles d'avertissements sur les dangers pouvant apparaître sur la route ou tout type d'informations intéressant le conducteur, parmi lesquelles on peut citer ces exemples :

Attention animaux


Attention Croisement


Attention sortie de ferme


Passer à droite


Rambouillet


Interdit moins de 7,5 t


Interdit Camion

Céder le passage

1. Les types des panneaux de signalisation routière

Il y a beaucoup de panneaux de signalisation sur les routes, nous avons présenté les différents types les plus importants qui existent actuellement :

Panneaux d'interdiction ou de restriction

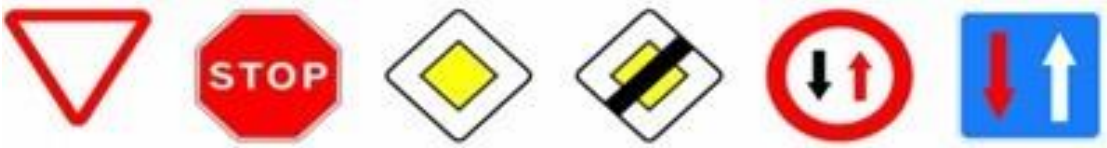
Ces types de signaux interdisent ou limitent certaines actions à ceux qui les trouvent devant dans la direction de leur marche et de l'endroit où ils se trouvent. Ces signes sont circulaires et avec un bord rouge. Comme montre la figure suivante.

Panneaux d'avertissement de danger

Les signes de danger ont pour mission d'indiquer la nature d'un danger, leur objectif se conformer aux règles de comportement et d'éviter les chocs éventuels lors de la conduite. Sa forme est triangulaire avec un bord rouge, comme indique la figure suivante.

Panneaux de priorité

Ce sont destinées à informer les usagers de la route aux règles de priorité spéciales aux intersections ou aux passages étroits. À l'intérieur de cette classe, nous pouvant trouver deux des signes les plus importants qui existent « stop » et « Cédez le passage ». Comme vous pouvez le voir ils n'ont pas de formulaire ou une couleur spécifique.



III. Le réseau de neurones convolutionnel (CNN)

Il existe plusieurs types d'apprentissage profond, parmi eux le réseau de neurones convolutionnel (CNN). CNN est une amélioration des réseaux de neurones artificiels (ANN) traditionnel, qui comprend généralement des couches convolutions, des couches de pool et des couches entièrement connectées. Le CNN peut être divisé en deux parties :

Une partie d'extraction d'entités (couches de convolution et couches de regroupement)

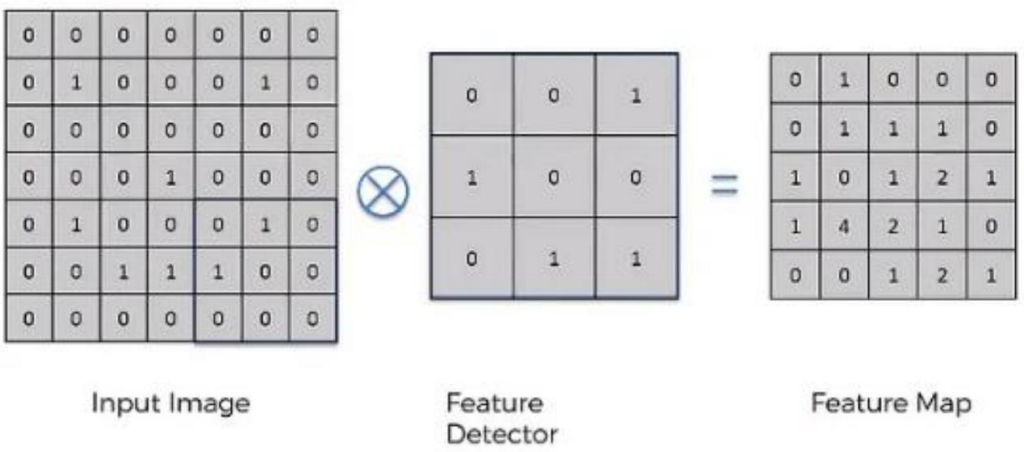
Une partie de classification (couches entièrement connectées).

L'image est d'abord passée à travers une série de convolution, regroupant des couches pour l'extraction d'entités, puis passer à travers des couches entièrement connectées pour la classification.

Couche convolution (Convolution layer)

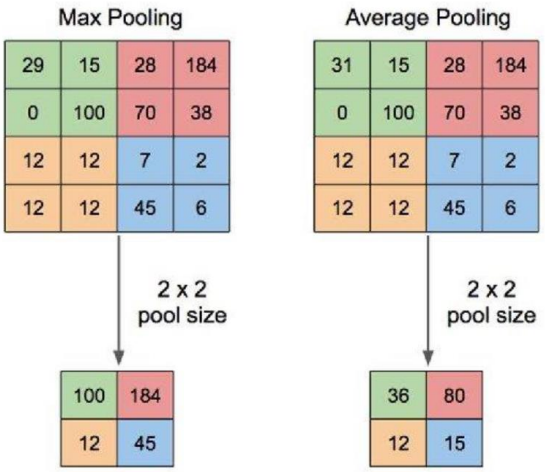
La couche convolution est un composant fondamental des réseaux CNN, jouant un rôle crucial dans l'extraction de caractéristiques à partir des images d'entrée. Son objectif principal est d'identifier et de capturer les caractéristiques pertinentes dans une image. Ceci est réalisé grâce au processus de filtrage convolutif. La couche convolutive fonctionne en faisant glisser une fenêtre de filtre sur l'image, balayant systématiquement chaque partie de l'image. A chaque position, le filtre et la région correspondante de l'image sont multipliés élément par élément, et les valeurs résultantes sont additionnées pour obtenir une seule valeur de sortie. Cette opération de convolution est effectuée pour chaque position dans l'image, résultant en une carte de caractéristiques de sortie. Les filtres utilisés dans la couche convolutive sont conçus pour capturer des caractéristiques ou des motifs d'intérêt spécifiques dans l'image.

Chaque filtre met en évidence un aspect différent, comme les bords, les textures ou les formes. En appliquant divers filtres, la couche convolutive peut détecter efficacement un large éventail de caractéristiques dans l'image d'entrée. Pour chaque paire d'image d'entrée et de filtre, la couche convolutive produit une carte d'activation de caractéristiques. Cette carte indique les emplacements dans l'image où la caractéristique correspondante est détectée. Les valeurs d'activation dans la carte représentent le degré de similitude entre le filtre et la région de l'image. Des valeurs d'activation plus élevées indiquent des correspondances plus fortes entre les positions du filtre et de l'image qui possèdent la caractéristique souhaitée.



La couche de pooling (pooling layer)

Elle reçoit plusieurs « feature map » en entrée et applique une opération de pooling (subsampling) à chaque feature map. Elle permet la réduction de la taille de l'image en tenir en compte ses caractéristiques importantes. Le principe est de couper l'image en cellules régulières, puis nous conservons la valeur maximale dans chaque cellule par rapport le filtre utilisé (2 x 2 pixels, 3x3…). Donc des petites cellules carrées sont souvent utilisées afin de ne pas perdre trop d'informations. Nous obtenons le même nombre de « features map » que l'entrée, mais elles sont beaucoup plus petites. Cela permet d'accélérer non seulement les calculs, mais d'éviter également le problème du sur-ajustement.



• Le pooling maximal : est basé sur la détection de la valeur maximale dans la région sélectionnée

• Le pooling minimal : sur la détection de la valeur minimale dans la région sélectionnée

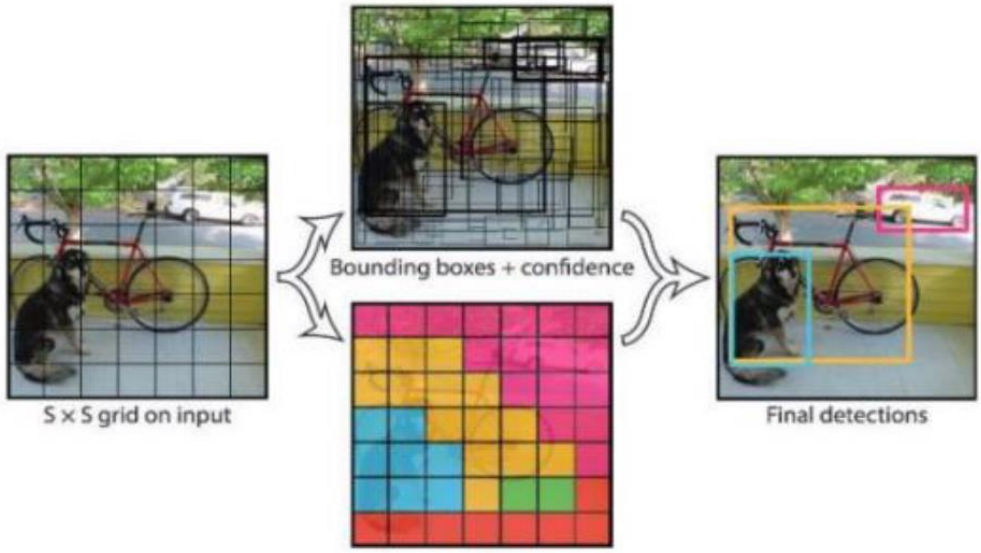
• Le pooling moyenne (average pooling) : basé sur la détection de la valeur moyenne dans la région sélectionnée.

You Only Look Once (YOLO)

YOLO (You Only Look Once) est un algorithme populaire de détection d'objets en temps réel qui a révolutionné le domaine de la vision par ordinateur. Il a introduit une approche innovante de la détection d'objets en combinant la localisation et la classification d'objets dans un seul modèle unifié. L'idée principale derrière YOLO est de diviser l'image d'entrée en une grille et de prédire les boîtes englobantes et les probabilités de classe directement dans chaque cellule de la grille. Cette approche de

cellule de grille permet à YOLO de faire des prédictions pour plusieurs objets simultanément, ce qui se traduit par une vitesse d'inférence impressionnante.

YOLO utilise une architecture de réseau neuronal à convolution profonde (CNN), généralement basée sur l'architecture Darknet, pour extraire les caractéristiques de l'image d'entrée. Le réseau divise l'image en une grille S × S, où chaque cellule de grille prédit B boîtes englobantes et leurs probabilités de classe correspondantes. Ces boîtes englobantes sont paramétrées par leurs coordonnées par rapport à la cellule de la grille, ainsi que les prédictions de largeur et de hauteur. Les probabilités de classe représentent les scores de confiance pour chaque classe d'objets.



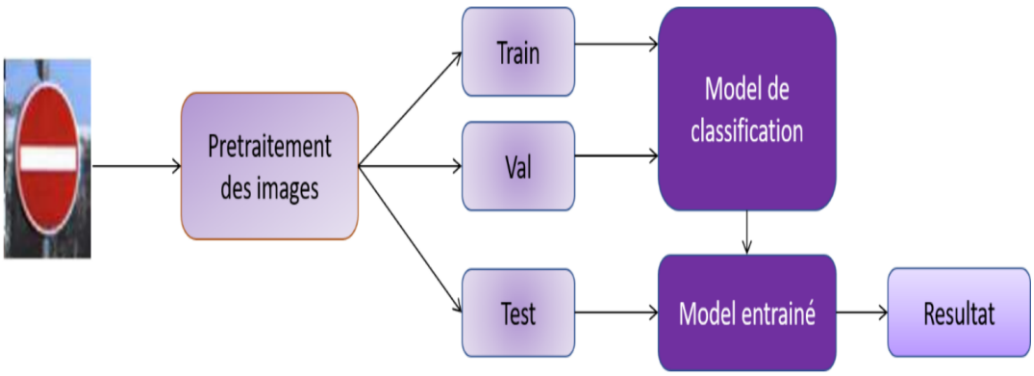
Pendant la formation, YOLO utilise un ensemble prédéfini de boîtes d'ancrage avec différents rapports d'aspect pour correspondre aux objets de vérité terrain. Le modèle est entraîné à l'aide d'une combinaison de perte de localisation (mesure de la précision des prédictions de la boîte englobante) et de perte de classification (mesure de la précision des prédictions de classe). La fonction de perte est optimisée à l'aide de techniques telles que la rétropropagation et la descente de gradient. L'un des principaux avantages de YOLO est sa vitesse remarquable, permettant la détection d'objets en temps réel sur des appareils à ressources limitées. Cependant, l'approche en un seul passage de YOLO peut entraîner des difficultés à détecter avec précision les petits objets, et elle peut avoir des difficultés avec des objets d'échelles et de rapports d'aspect variables dans la même cellule de grille.

Au fil des ans, plusieurs versions de YOLO ont été développées, notamment YOLOv2, YOLOv4 et YOLOv8, chacune introduisant des améliorations en termes de précision

et de performances. Ces versions intègrent des techniques telles que le regroupement de boîtes d'ancrage, des réseaux pyramidaux et des modifications architecturales avancées pour améliorer la précision de la détection et répondre aux limites du YOLO d'origine.

IV. L'implémentation des models de classification

Dans ce travail, et afin de classifier les images en 43 classes, nous avons suivi deux approches : la première est un CNN conçu from scratch, et la deuxième est basée sur le transfert de connaissance en utilisant YOLOv8.

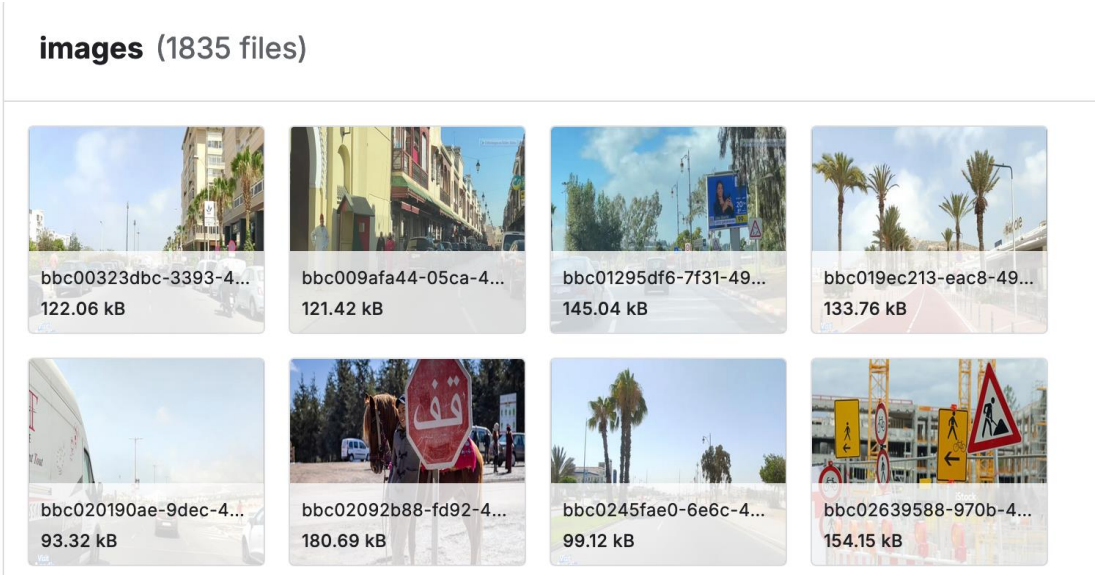


Dataset

L'ensemble de données utilisé dans cette étude est un ensemble personnalisé que nous avons collecté et annoté à partir de différentes sources, telles que des vidéos YouTube et des prises de photos.

Plus de 1900 photos de panneaux de signalisation marocain sont incluses dans l'ensemble de données de référence qui représentent ensemble 43 classes de signes différentes

Un total de 1,955 images des panneaux compose l'ensemble de données, qui est ensuite divisé en deux sous-ensembles : un ensemble d'apprentissage de 1,835 photos et un ensemble de test distinct de 120 images.



Approche 1 : model CNN from scratch

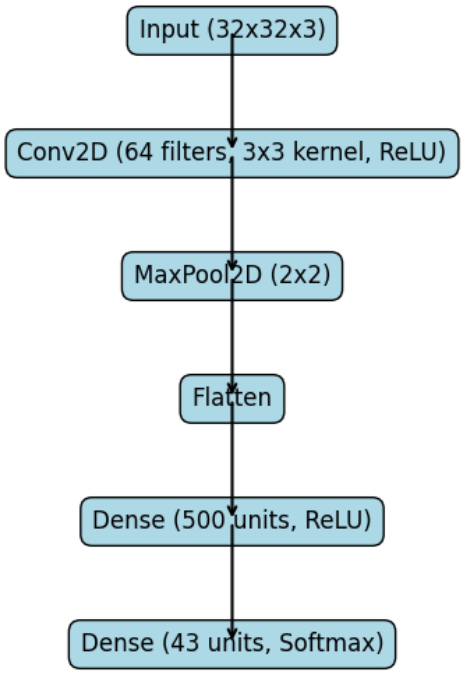
Les réseaux profonds de type CNN est largement utilisé et recommande pour le problème de classification, pour cette raison nous avons proposé un réseau convolutionnel (CNN) comporte 7 couches : Une couche d'entrée, une couches de convolution, une couches de MaxPooling, une couche Flatten et 2 couches Dense et une couches de sortie.

La première couche (conv2D) : est destiné à l'extraction d'identité (feature extraction). Il effectue une convolution avec 64 filtres de taille 3x3 pour filtrer une image d'entrée de taille 32*32*3. Cette couche est effectuée avec un padding same, la fonction d'activation utilisée est RELU.

La couche pooling : Notre architecture se compose d'une couche de pooling (maxPooling) pour réduire la dimension spatiale (feature map) des couches convolutionnelles avec un filtre de taille 2x2.

La couche fully connected (dense) : Notre modèle se termine par une couche flatten et une couches entièrement connectées (FC) avec respectivement 500 et 43 unités. En choisissant ce nombre des unités pour minimiser le nombre des paramètres et accélérer le calcule (le taux de calcule). La première couche fully connected utilise une fonction d'activation RELU suivi par une couche dropout pour réduire le problème d'overfitting.

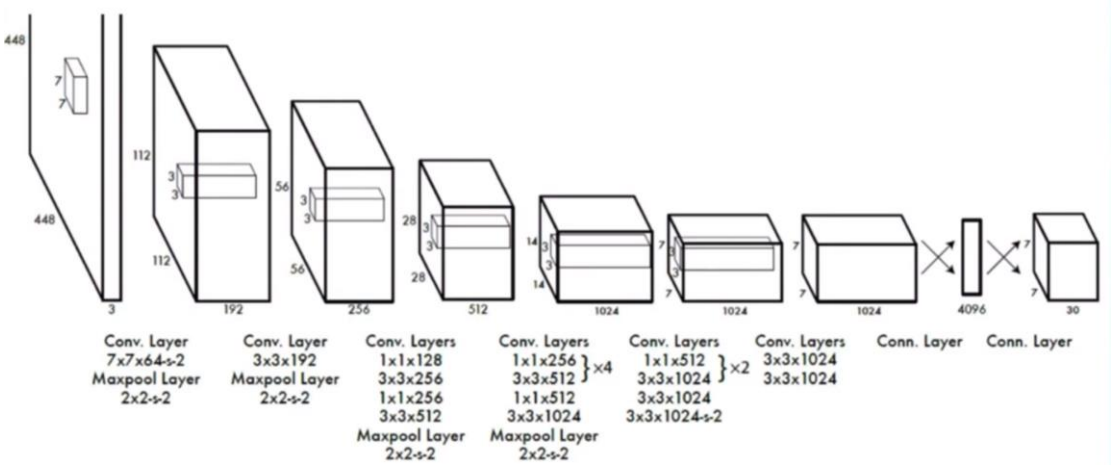
La couche de sortie constitue de 43 unités qui ressemble à nos classes de sortie (catégories des panneaux) avec une fonction Softmax.



Approche 2 : model Yolov8 (transfer learning)

Dans cette méthode, nous avons utilisé une approche d'entrainement basé sur le Transfer learning, où on utilise un model existant pré-entrainé sur une large dataset.

Pour cela nous avons choisi le modèle YOLOv8 pré-entrainé sur les données COCO.



Expérimentation et résultats

Pour entraîner notre modèle nous avons joué les paramètres suivants :

• Batch size : nombre d'échantillons ou de points de données qui sont traités ensemble en un seul passage avant et arrière pendant la formation. Au lieu de mettre à jour les paramètres du modèle après chaque échantillon individuel, il est plus efficace de les mettre à jour après le traitement d'un lot d'échantillons.

• Epoch : détermine le nombre de fois que le modèle itérera sur l'ensemble du jeu de données.

Pour évaluer notre modèle nous avons basé sur la metrique d'accuracy

• Accuracy : quantifie la justesse globale des prédictions du modèle, Acc = (Nombre d'instances correctement classées) / (Nombre total d'instances).

Alors nous avons lancé l'apprentissage du model par plusieurs valeurs du batch-size et différent nombre d'epoch. Voici les meilleur résultats obtenus après la modification des valeurs de ces options pour chaque méthode utilisée :

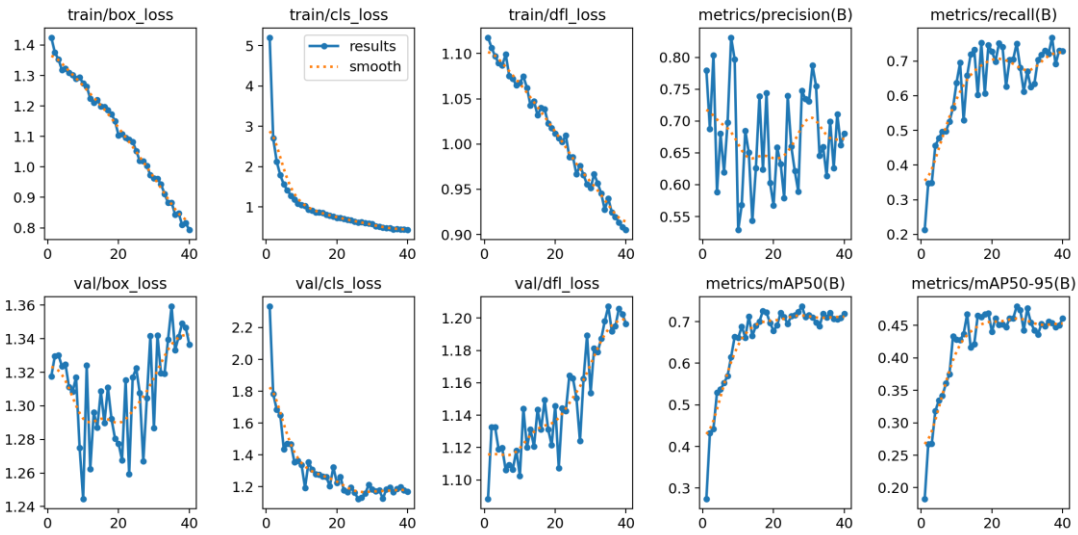
Méthode 1: CNN from scratch

Le meilleur résultat est obtenu apres 10 epochs . Où nous avons obtenus un taux d'accuracy égale à 99%.

```
Epoch 1/10
2719/2719 [==============================] - 16s 6ms/step - loss: 0.0968 - accuracy: 0.9805
Epoch 2/10
2719/2719 [==============================] - 15s 6ms/step - loss: 0.0591 - accuracy: 0.9887
Epoch 3/10
2719/2719 [==============================] - 17s 6ms/step - loss: 0.0342 - accuracy: 0.9930
Epoch 4/10
2719/2719 [==============================] - 15s 6ms/step - loss: 0.0234 - accuracy: 0.9952
Epoch 5/10
2719/2719 [==============================] - 16s 6ms/step - loss: 0.0205 - accuracy: 0.9961
Epoch 6/10
2719/2719 [==============================] - 15s 6ms/step - loss: 0.0173 - accuracy: 0.9967
Epoch 7/10
2719/2719 [==============================] - 15s 6ms/step - loss: 0.0130 - accuracy: 0.9973
Epoch 8/10
2719/2719 [==============================] - 15s 6ms/step - loss: 0.0138 - accuracy: 0.9974
Epoch 9/10
2719/2719 [==============================] - 15s 6ms/step - loss: 0.0083 - accuracy: 0.9984
Epoch 10/10
2719/2719 [==============================] - 15s 6ms/step - loss: 0.0094 - accuracy: 0.9982
```

Méthode 2: Yolov8

Nous avons fixé les poids de toutes les couches du modèle pré-entrainé, et entrainer juste la partie de classifieur qui comporte les couches entièrement connectées. Nous avons lancé l'entrainement du modèle avec un batch size de 16 Voici l'evolution du model



Le meilleur résultat est obtenu apres 40 epochs . Où nous avons obtenus un taux d’accuracy égale à 99.8%.

Finalement, bien que les deux approches offrent des résultats très compétitifs, l'option basée sur le transfert de connaissances avec YOLOv8 semble offrir un léger avantage en termes de précision, tout en étant potentiellement plus robuste et adaptable à des variations dans les données.

Conclusion

En somme, nous avons exploré deux approches de classification d'images de panneaux de signalisation marocains en utilisant des modèles de réseaux neuronaux profonds. D'une part, nous avons conçu un modèle de CNN (Convolutional Neural Network) from scratch, et d'autre part, nous avons utilisé une approche de transfert de connaissances avec le modèle pré-entrainé YOLOv8.

En conclusion, bien que les deux méthodes aient montré des performances très satisfaisantes, le modèle basé sur le transfert de connaissances avec YOLOv8 présente des avantages notables, notamment en termes de précision et d'efficacité. Toutefois, il est important de noter que le choix de l'approche dépend du contexte spécifique de l'application, des ressources disponibles pour l'entraînement et de la complexité des données. Ces résultats ouvrent également la voie à des améliorations futures, comme l'optimisation des architectures ou l'augmentation de la diversité des données pour améliorer encore la performance des modèles dans des scénarios réels.

Ainsi, ce travail contribue à la compréhension et à l'application de techniques modernes de deep learning dans la classification d'images, tout en offrant des perspectives pour d'autres applications dans des domaines similaires.