

Note méthodologique : preuve de concept

Dataset retenu

Le dataset utilisé est celui du projet 6 du parcours de data scientist intitulé "classifier automatiquement des biens de consommation".

Il s'agit de photos d'articles proposés par les vendeurs de l'entreprise anglophone de e-commerce "Place de marché".

Le but était d'automatiser la tâche d'attribution de la catégorie des articles proposés par ses vendeurs en faisant une classification supervisée des images des articles avec l'algorithme VGG16.

Le dataset est composé de 1050 images labellisées en 7 catégories:

- Home Furnishing: 150 images
- Baby Care: 150 images
- Watches : 150 images
- Home Decor & Festive Needs: 150 images
- Kitchen & Dining: 150 images
- Beauty and Personal Care: 150 images
- Computers: 150 images

Les images sont en couleur et de taille variable.

Concepts de l'algorithme récent: YOLOv8

YOLOv8 (You Only Look Once version 8) est la dernière avancée de la série de détecteurs d'objets en temps réel YOLO.

YOLO est une famille de modèles de vision par ordinateur (YOLOv1: 2015, ... YOLOv8: 2023)

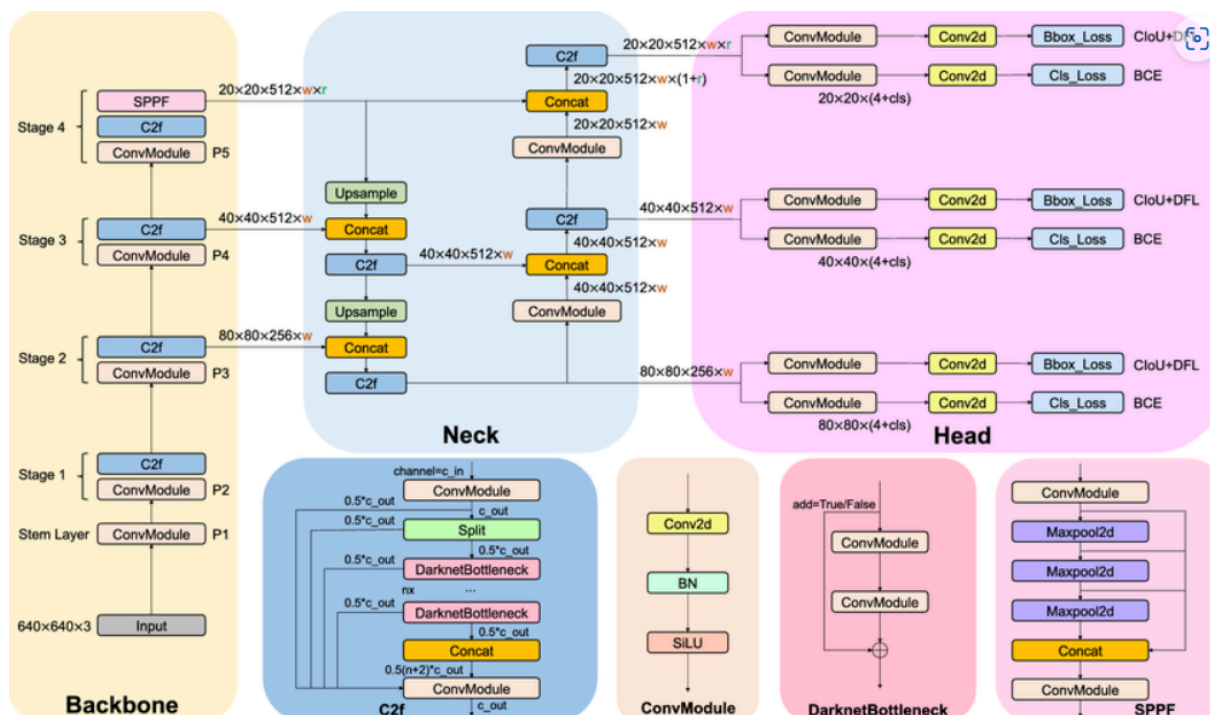
Il offre des performances de pointe en termes de précision et de vitesse.

YOLOv8 est référencé sur le site de recherche ([Arxiv](#)) et fait l'objet de nombreux articles.

Principes de fonctionnement de YOLOv8 :

1. Détection d'Objets en Temps Réel : YOLOv8 détecte les objets directement sur l'image complète, sans découpage en régions.
2. Réseau de Neurones Convolutifs (CNN) : Utilisation d'un réseau de neurones convolutifs profond pour extraire des caractéristiques. Il exploite des couches de convolution, de pooling et fully connected pour extraire des features et effectuer la classification.
3. Ancres et grilles : Prédiction des boîtes englobantes pour chaque grille de l'image.
4. Multi-Échelle : Utilisation de couches de détection à différentes échelles.

Architecture de YOLOv8:



- **Backbone:**
 - a. CNN pour extraction de features à partir de l'image en entrée
 - b. Customisé à partir de CSPDarknet53
 - c. Utilisation de connexions partielles entre les stages pour augmenter le flux d'informations entre les couches

- **Neck:**
 - a. Également appelé extracteur de features, le neck assemble les “feature maps” qui proviennent des différents “stages” du backbone pour capturer l’information à différentes échelles.
 - b. YOLOv8 utilise un nouveau module appelé C2f à la place du traditionnel FPN (Feature Pyramid Network). Ce module combine des caractéristiques sémantiques de haut niveau avec des informations spatiales de bas niveau, ce qui améliore la précision de la détection, notamment pour les petits objets”.
- **Head:** YOLOv8 utilise plusieurs modules de détection qui prédisent les boîtes englobantes, les scores d’objectivité et les probabilités de classe pour chaque cellule de grille dans la features map. Ces prédictions sont ensuite agrégées pour obtenir les détections finales

Modélisation

1. Méthodologie de Modélisation :



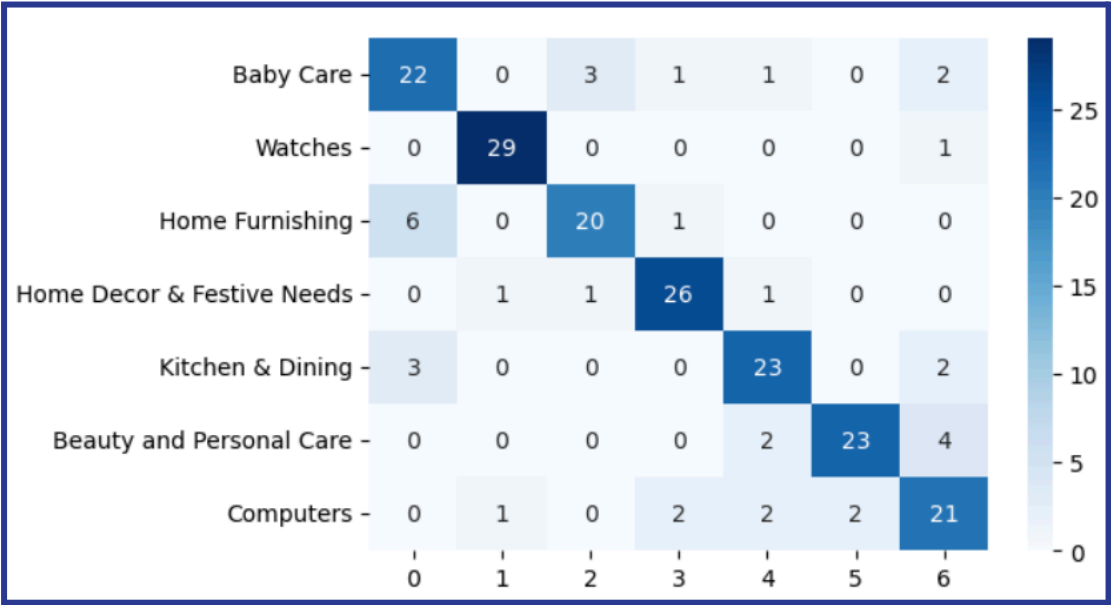
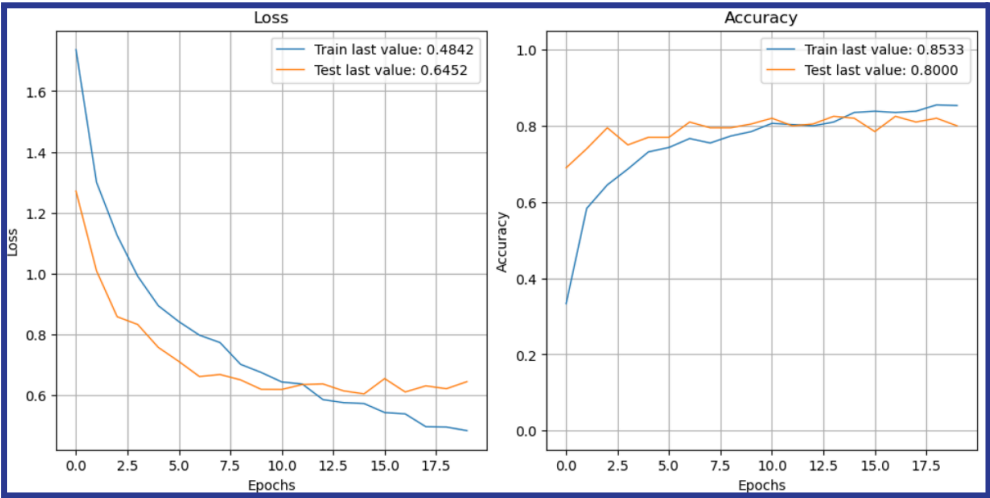
- Utilisation d’un modèle Yolo V8 **pré-entraîné en classification**
- Ré-entraînement sur notre dataset (1050 images et 7 classes) sur **50 epochs**
- Minimisation de l’erreur de classification (loss)

2. Métrique d’Évaluation :

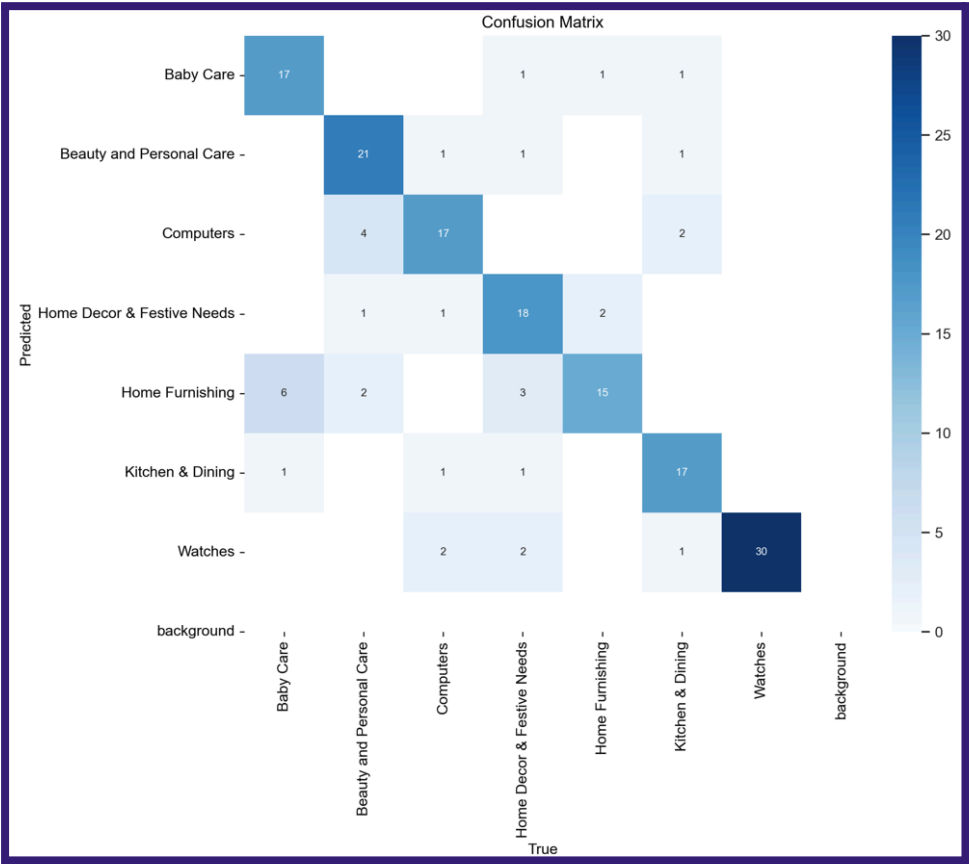
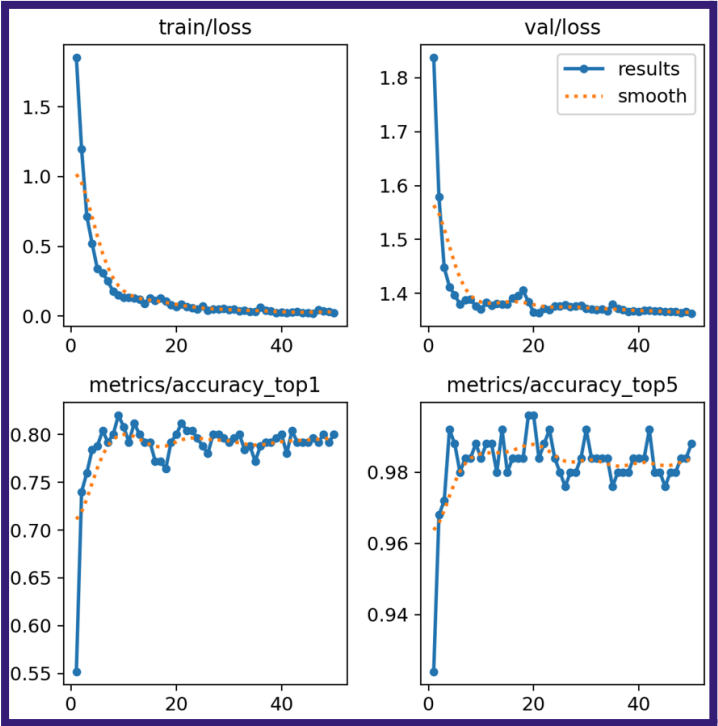
- **Exactitude (accuracy)** comme métrique principale
- Matrice de confusion

Synthèse des résultats

Approche classique: VGG16



Approche nouvelle: YOLOv8



Modèle	Hyperparamètres principaux	Nb d'epochs	Taille des images	Validation Accuracy	Test Accuracy	Temps de train
VGG16	optimizer='rmsprop' RandomFlip=H RandomRot=0.1 RandomZoom=0.1 Dropout=0.5	20	224x224	0.82	0.77	17min
Yolo (YOLOv8s-cls)	lr0: 0.01, lrf: 0.01 ... translate: 0.1, scale: 0.5 ... flipud: 0.0, fliplr: 0.5	50	longueur:224 largeur: ajustée	0.82	0.794	25min

- Les hyperparamètres ont été optimisés pour VGG16 mais pas pour YOLOv8 (bug de la fonction tune, hyper paramètres par défaut utilisés)
- Malgré cela, les scores d'accuracy en validation sont les mêmes (0.82)
- L'accuracy de YOLOv8 est meilleure sur le jeu de test (0.794 contre 0.77).
Meilleure capacité de YOLOv8 à généraliser.

L'analyse de la feature importance globale et locale du nouveau modèle

Pas applicable à la classification d'images.

Les limites et les améliorations possibles

- Limites :
 - Les hyperparamètres n'ont pas été optimisés (bug de la fonction tune, hyper paramètres par défaut utilisés)
 - Entraînement en local, ressources limitées, nous avons dû utiliser le modèle pré-entraîné le plus simple de la famille YOLOv8-cls
- Améliorations Envisageables :

- Augmenter la taille des images. 224 -> 690 (valeur par défaut). Le temps de train sera plus élevé mais l'accuracy meilleure.
 - Entraîner [YOLOv8x-cls](#) (demande plus de ressources) sur le cloud (ex: google collab)
 - Optimisation des hyperparamètres pour améliorer la performance.
-

Cette preuve de concept démontre l'efficacité de l'algorithme YOLOv8 dans la classification d'images.