

Réaliser un dashboard et assurer une veille technique

Introduction

- Partie 1: Réaliser un dashboard
 - Suite du Projet 7: “Prêt à dépenser” propose des prêts à des clients n’ayant pas ou peu d’historique de prêt
 - Outil de scoring (probabilité de défaut et classification demande de prêt)
 - Dashboard interactif destiné aux chargés de clientèle
 - Souci de transparence conseiller clientèle -> client
- Partie 2: Réaliser une veille technique
 - Technique récente de classification d’images (YOLOv8)
 - Modélisation, dataset projet 6 “classifier des biens de conso”
 - Comparaison performances VS VGG16



Partie 1: Présentation du dashboard

Partie 2: Veille - classification d'images avec YOLOv8

YOLOv8: vue d'ensemble



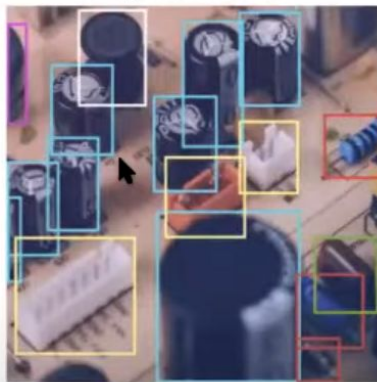
- **Points importants:**
 - YOLO: famille de modèles de vision par ordinateur
 - YOLOv1 (2015), ... YOLOv8 (2023), YOLOv9 (2024)
 - Rapidité, précision, capacité à généraliser
 - Référencé sur Arxiv

Classification



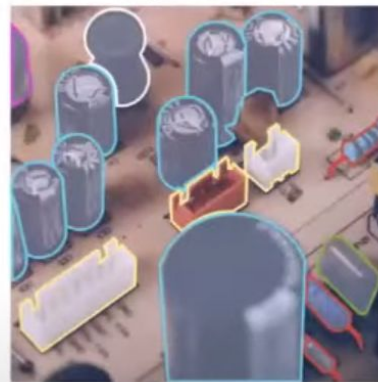
Capacitor

Object Detection



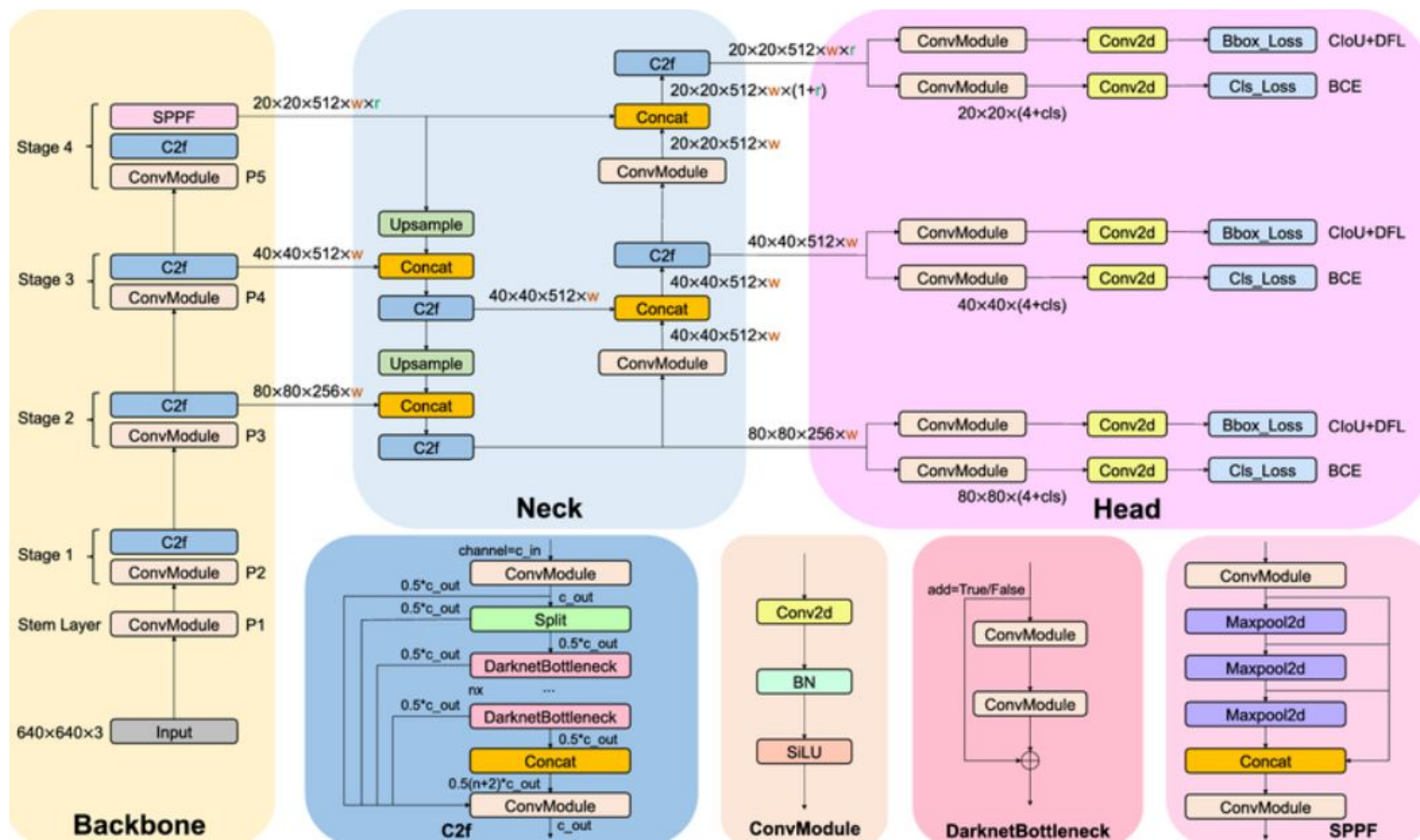
Capacitor, Resistor, Transformer, Connector,
Inductor, Polyester Capacitor

Segmentation



Capacitor, Resistor, Transformer, Connector,
Inductor, Polyester Capacitor

YOLOv8: architecture





Démarche de modélisation




YOLOv8 - choix du modèle pré-entraîné

← → ↻ 🔒 https://docs.ultralytics.com/tasks/classify/#models

 **Classify**



Home Quickstart Modes Tasks Models Datasets NEW  Explorer Guides Integrations HUB Reference Help

Tasks

Detect

Segment

Classify

Pose

OBB

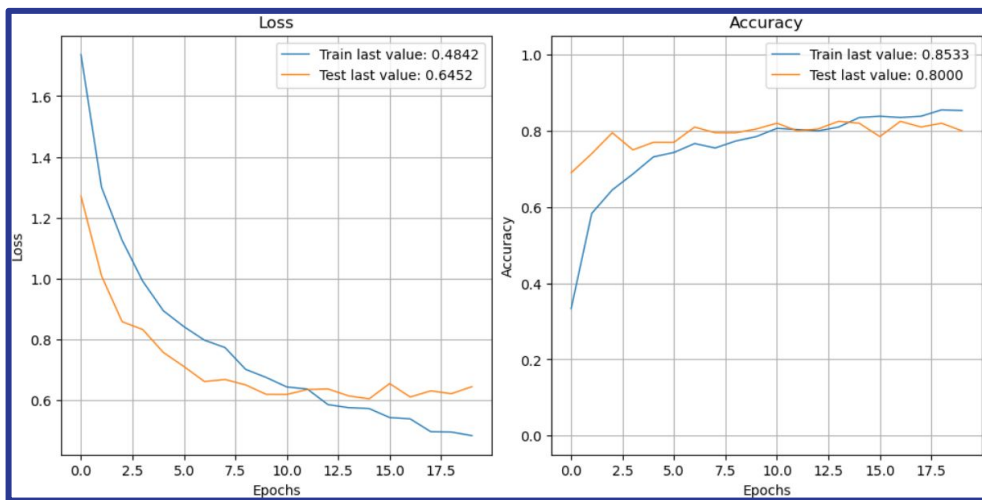
Models

YOLOv8 pretrained Classify models are shown here. Detect, Segment and Pose models are pretrained on the [COCO](#) dataset, while Classify models are pretrained on the [ImageNet](#) dataset.

[Models](#) download automatically from the latest Ultralytics [release](#) on first use.

Model	size (pixels)	acc top1	acc top5	Speed CPU ONNX (ms)	Speed A100 TensorRT (ms)	params (M)	FLOPs (B) at 640
YOLOv8n-cls	224	69.0	88.3	12.9	0.31	2.7	4.3
YOLOv8s-cls	224	73.8	91.7	23.4	0.35	6.4	13.5
YOLOv8m-cls	224	76.8	93.5	85.4	0.62	17.0	42.7
YOLOv8l-cls	224	76.8	93.5	163.0	0.87	37.5	99.7
YOLOv8x-cls	224	79.0	94.6	232.0	1.01	57.4	154.8

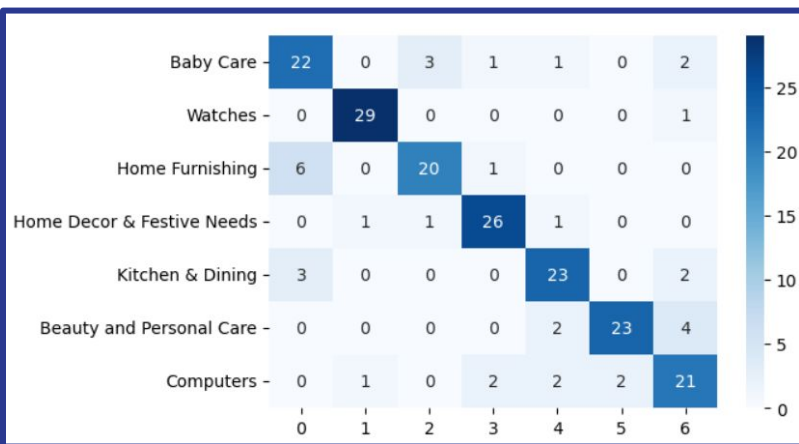
Approche classique: VGG16



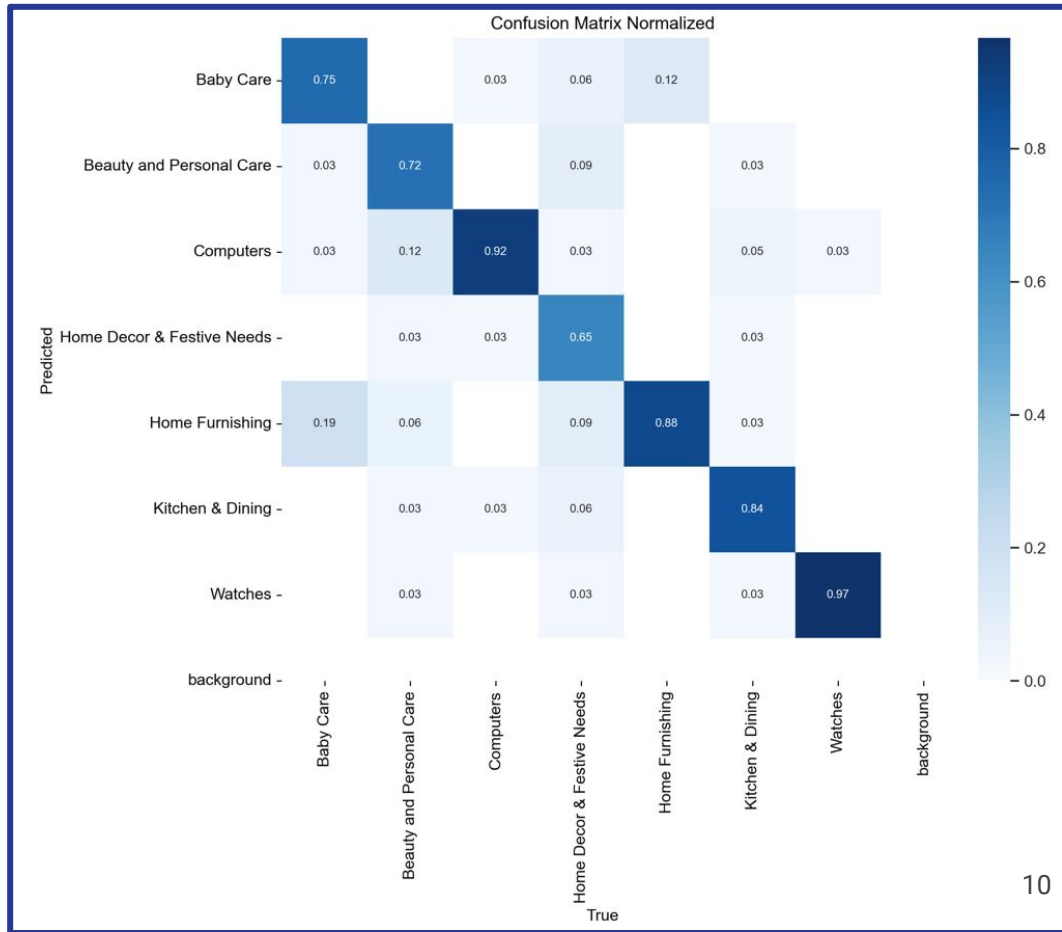
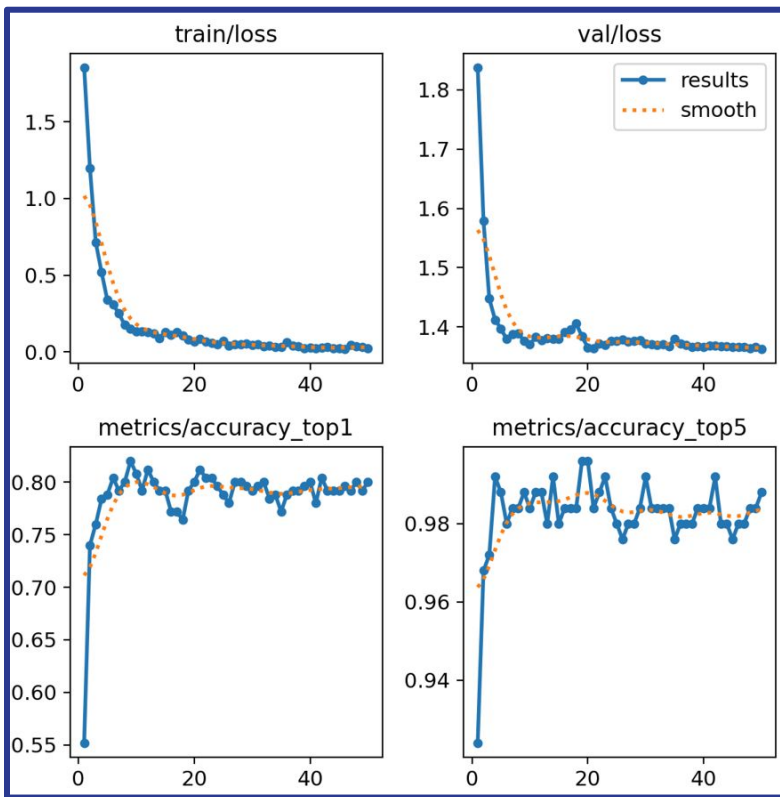
- Epoch optimal (min loss):

Validation Accuracy : 0.82

Test Accuracy : 0.77



Approche nouvelle - YOLOv8



Comparaison des performances

Modèle	Hyperparamètres principaux	Nb d'épochs	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