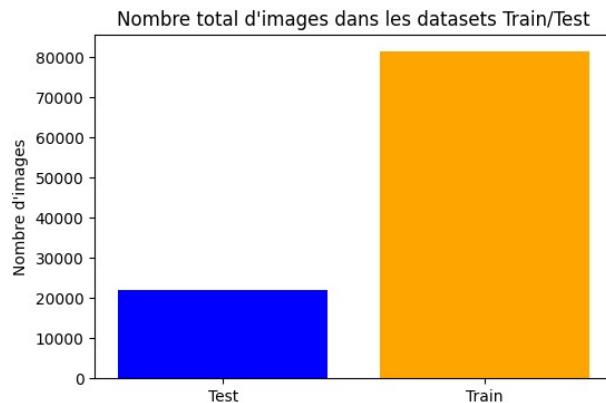


Packed Fruits and Vegetables Recognition

<https://www.kaggle.com/datasets/sergeynesteruk/packed-fruits-and-vegetables-recognition-benchmark/code>



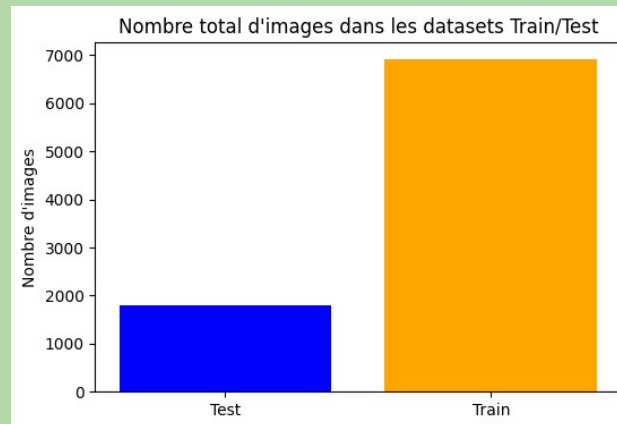
Non-Segmented Dataset



73% Train
27% Test

- 65 Catégories de Fruits et Légumes labellisées
- Train = 81.489 images
- Test = 21.923 images
- Total = 103.412 images
- Un Fichier CSV regroupant 17 variables pour chaque image (Species; Variety; Packed; Weight; Amount; Shop; Subset.....)

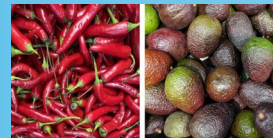
Segmented Dataset



74% Train
26% Test

- 71 Class de 0 à 70 - Non-Labellisées
- Train = 6916 images.jpg et images.txt (annotations de segmentation)
- Test = 1803 images.jpg et images.txt (annotations de segmentation)
- Total = 8719 images
- Pas de CSV pour labellisations des classes

Segmented Dataset - EDA



1 - Labelliser les classes 0 à 70 en Fruits et Légumes :

Attribution des labels en comparant les images avec les images labellisées du Dataset Non-Segmented.
Des classes en doubles ou en triple: Apple-Spartan / Banana_Yellow / Pepper_Sweet_Green / Tomato_Cherry_Yellow etc...

Une fois bien-labellisé on Passe bien de 71 classes au 65 classes du Dataset Non-Segmented

2 - Nettoyage et Réorganisation des images



En observant en subplot toutes les images pour les 65 classes, On constate des erreurs de classifications dont voici la plus parlante. La Classe 10 labellisée - Cucumber_Long comportant 50% d'images de grenades.

Cette observation nous a permis de Ré-affecté correctement toutes les images à leur véritable classe

Changer la classe d'une image.jpg consiste à modifier l'Id du fichier.txt correspondant

Segmented Dataset Final – EDA - 1



Une fois le dataset bien organisé et clean, Observons la répartition des images par classe

Class	Label	Train_Origin	Test_Origin	%	Total_Origin
3	garlic_garlic	80	12	15%	92
4	apple_red_delicious	153	46	30%	199
5	tomato_cherry_yellow	107	123	115%	230

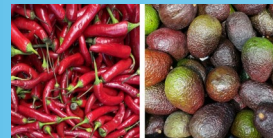
On constate que la répartition des images en Train/Test ainsi que le nombre d'images par Classe n'est pas homogène

On a donc regroupé toutes les images Train/Test pour pouvoir ensuite splitté équitablement par classe

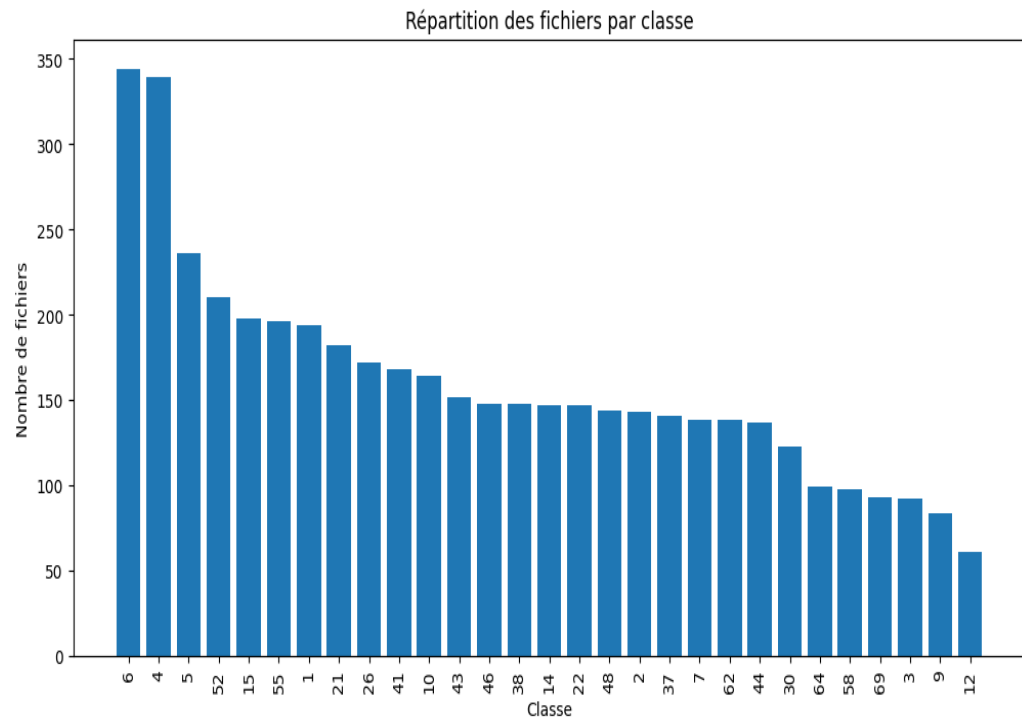
Dans la continuité d'homogénéisation du dataset on a :

- Supprimer toutes les classes sous-représentées comme Tangerine_Tangerine 39 images
- Supprimer toutes les classes peu communes comme le Daikon (radis chinois)
- Supprimer les classes trop répétitives (Tomates_cherry_Yellow, Tomato_Yellow.. Cucumber_Long, Cucumber_Moyen... Etc...)

Segmented Dataset Final – EDA - 2



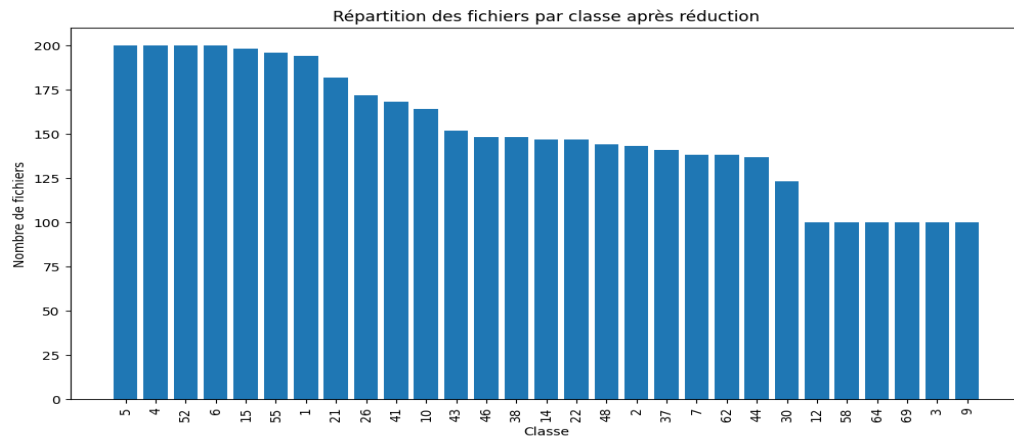
**On a pu conservé 29 Classes
mal réparties en nombres d'images**



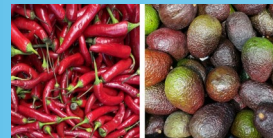
Data-Diminution des classes sur-représentées
Suppression random des classes > à 200 images

Data-Augmentation des classes sous-représentées
Ajout soft pour les classes < 100 images pour = 100 images

**Enfin - Toutes les classes
sont dans un range de 100 et 200 images
Avec une moyenne de 144 images/classes**

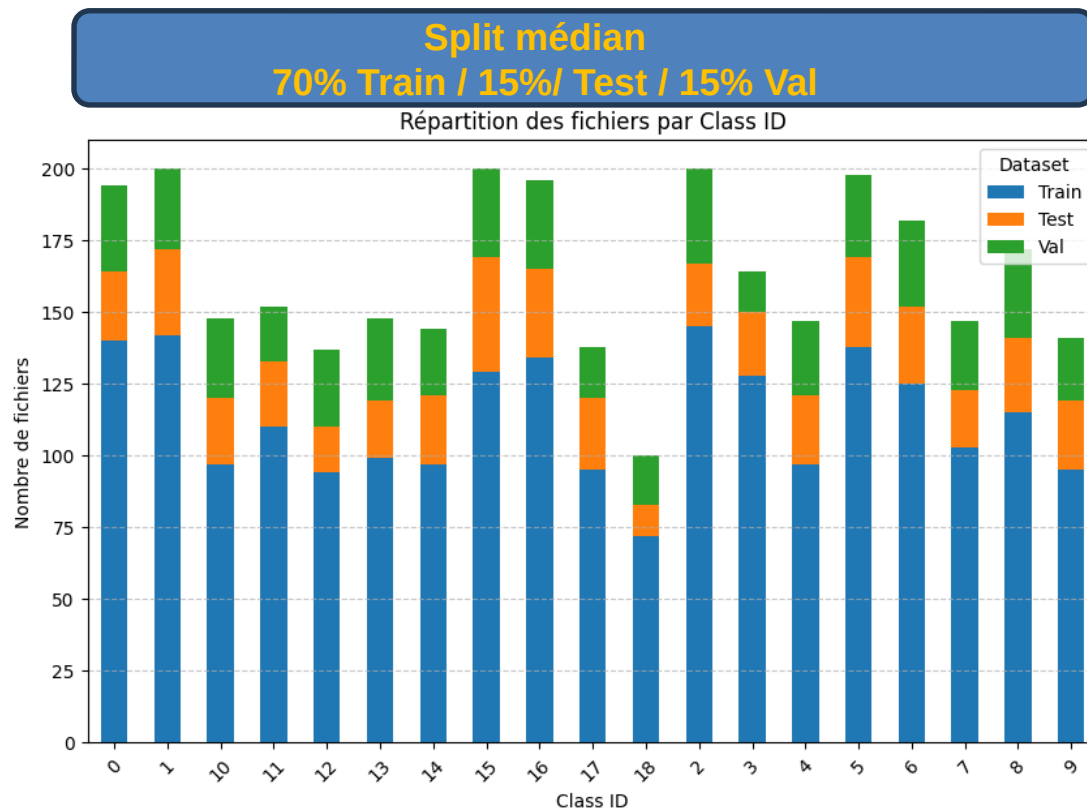


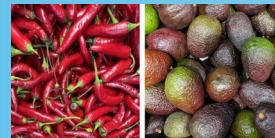
Segmented Dataset Final – EDA - 3



Parmi les 29 classes pré-sélectionnées nous en avons conservées 19 – les plus représentatives de ce que l'on trouve dans les grands commerces.

Class Pack	Label
0	Carotte
1	Pomme_Pink_Lady
2	Tomate
3	Concombre
4	Banane
5	Pomme_Granny
6	Poire_Williams
7	Poivron_Rouge
8	Orange
9	Oignon
10	Pomme_Gala
11	Citron_Vert
12	Avocat
13	Pomme_Golden
14	Kiwi
15	Tomate_Cerise
16	Poivron_Vert
17	Poivron_Jaune
18	Citron_Jaune





Dans l'optique d'avoir un modèle de prédiction rapide et efficace en précision pour notre dataset final de 3160 images réparties en:

- 19 classes d'environ 150 images/classes
- Environ 70% Train / 15% Test / 15% Val

Nous avons choisi d'utiliser YoloV8:

Notre choix c'est porté principalement sur :
Yolov8~~m~~-seg et YoloV8~~l~~-seg



Modèle 1

YoloV8_{m_seg} = 78% d'accuracy

On a stoppé l'entraînement à la 25e epoch dès qu'il manifestait de la stabilité (sans signes d'overfitting – et juste avant qu'il passe en surapprentissage)

Il s'avère avoir été le plus performant en conditions réelles.



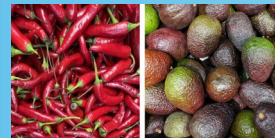
Modèle 2

YoloV8I_seg = 95% d'accuracy

On a poussé au maximum l'entraînement jusqu'à la 63^e epoch (sans signes d'overfitting)

Ne fonctionnait pas du tout en conditions réelles.

Mais en testant les Epochs 20 et 30 ils performaient bien mieux sans dépasser les prédictions du Modèle 1

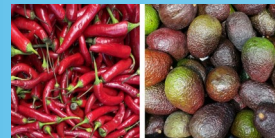


En comparant les prédictions sur des images réelles des Modèle 1 / epoch 30 / epoch 20. On observait une complémentarité sur les prédictions.

On a donc créé une prédiction Globale des 3 modèles :

- Classe la plus prédite par les 3 modèles
- Moyenne d'éléments détectés pour cette Classe
- Moyenne d'accuracy de cette Classe

Résultat : Prédiction plus juste et solide



On a décidé de tester de **Fusionner ces 3 modèles par une variante dite de la fusion pondérée des boîtes (Weighted Box Fusion - WBF)** combinée avec un **Non-Maximum Suppression (NMS)**.

1.Chargement des 3 modèles YOLOv8

1. Trois modèles (epoch30.pt, epoch20.pt, YoloV8m-seg.best.pt) sont chargés.
2. Chaque modèle fait des prédictions indépendantes sur les 95 images extérieures au Train prises a Monoprix avec Iphone. (induisant un filtre plus réelles que celles du dataset original)

2.Fusion des prédictions des modèles

1. On collecte toutes les prédictions des modèles pour une image donnée.
2. On applique **Weighted Box Fusion (WBF)** avec suppression des doublons via **NMS**.
3. Cette fusion permet de **combiner les prédictions** et de **réduire les faux positifs**.

3.Enregistrement des nouvelles annotations

1. Après la fusion, les boîtes sont converties au format YOLO.
2. Les fichiers .txt des nouvelles annotations sont sauvegardés.

4.Réentraînement d'un modèle YOLOv8m-seg fusionné

1. Un nouveau modèle YOLOv8m-seg est **réentraîné** avec les labels fusionnés.
2. Des **hyperparamètres optimisés** sont utilisés pour un meilleur apprentissage.

Modélisation – 7



Ce nouveau modèle fusionné nous a donné satisfaction en dépassant les prédictions de Epoch20

En re testant les 3 modèles : Modèle 1 / Epoch30 / Modèle_Fusionné:

- On constate à moyen terme que le modèle Epoch30 n'est plus significatif
- Mais que son arbitrage à la Prédiction globale reste utile.

On a donc recréé un Nouveau Modèle fusionné de : Modèle_Fusionné et Modèle 1 puis entraîné toujours sur Yolov8~~m~~-seg.

Ce nouveau modèle entraîné sur à peine 10 epochs puis Fine-Tunné jusqu'à la 14^e epoch nous a donné un modèle plus performant que le Modèle 1 jusqu'alors Modèle indétrônable.

Ce qui nous a permis d'évincer le Epoch30

Finalement nous utilisons pour notre Projet :

La Prédiction Globale des 3 modèles : Modèle 1 / Modèle_Fusionné1 / Modèle_Fusionné2

Cette méthode de Prédiction Globale sur 3 modèles nous a démontré une Prédiction plus solide qu'un seul modèle. Que cette méthode agit comme Régulateur pour une meilleure Généralisation.