Université Lumière Lyon 2

Master 2 SISE

Deep learning - Machine learning - Computer Vision

Étudiants :
- KOCAB Cyril
- COLLIN Hugo

Enseignant:
FERNANDO CRISPIM JUNIOR Carlos

17 février 2025

I. Introduction

La thématique de ce projet repose sur l'application des techniques de deep learning à la reconnaissance de produits, en mettant en œuvre une architecture avancée, YOLOv11, qui intègre notamment un backbone performant, un neck optimisé et un head multi-échelle renforcé par un mécanisme d'attention (C2PSA).

Notre jeu de données se compose d'images issues de supermarchés internationaux, incluant des contextes variés – d'un côté, des images provenant de supermarchés étrangers et, de l'autre, des images prises dans des supermarchés français. Cette diversité apporte son lot de défis, notamment en raison des variations de packaging et de l'hétérogénéité des produits, ce qui nécessite une approche robuste et adaptable pour garantir une détection précise.

Dans ce rapport, nous décrivons en détail la conception et la mise en œuvre de notre modèle de détection. Nous présentons d'abord l'architecture de YOLOv11 ainsi que les différentes étapes du traitement des images, puis nous abordons les paramètres d'entraînement et les techniques d'augmentation des données utilisées pour optimiser les performances. Enfin, nous analysons les résultats obtenus, tant sur le plan quantitatif que qualitatif, afin d'évaluer l'efficacité de notre approche et d'identifier les axes d'amélioration pour de futurs travaux.

II. Architecture du modèle : YoloV11

Elle se subdivise en trois parties et traite séquentiellement l'information dans l'ordre suivant : *Backbone*, *Neck* puis *Head*.

A. Backbone

- Extraction des dimensions c, h et w à partir des tenseurs caractéristiques des images.
- Envoie de ces dimensions dans un bloc de convolution constitué d'une couche de convolution 2D, puis de normalisation par batch 2D et enfin d'une fonction d'activation SiLU (Sigmoid Linear Unit).
- Envoie des informations dans un bottleneck qui correspond à une séquence de blocs de convolution avec la possibilité d'emprunter un raccourci afin de ne pas calculer les résidus
- Transmission des informations dans le C3K2. Il s'agit d'un bloc de convolution (de petite taille 3x3), suivi de *n* blocs C3K[1], dont les résultats globaux sont concaténés avant de passer dans un dernier bloc de convolution (1x1).
- Cette étape représente une amélioration notable par rapport aux anciennes versions car elle permet de préserver plus d'informations avec moins de paramètres.

[1] Chaque bloc C3K est constitué d'une couche de convolution, puis de *n* bottleneck dont les résultats sont concaténés et envoyés dans une dernière couche de convolution.

B. Neck

- Transfert des informations à l'étape SPPF (Spatial Pyramid Pooling Fast). Il s'agit d'un bloc de convolution précédent une série d'opérations de max-pooling 2D à différentes échelles afin d'extraire la même qualité d'information pour les petits et grands objets (sinon la détection des petits objets est rendue plus difficile). Les résultats sont ensuite concaténés et passés dans un dernier bloc de convolution (1x1).
- L'information est ensuite soumise à une opération d'upsampling afin de retrouver une dimensionnalité supérieure.

C. Head

- L'information est enfin envoyée à la "tête" (multi-scale head detection) capable de détecter plusieurs objets à différentes échelles.

D. Mécanisme d'attention

En supplément de l'architecture initialement présentée, un mécanisme d'attention est sollicité dans le réseau. Ce mécanisme, C2PSA (Cross Stage Partial with Spatial Attention) est un mécanisme augmenté car il permet au réseau d'augmenter l'influence des features nécessaires au traitement des régions d'une image plus difficiles à reconnaître (petits objets, occlusion, etc.).

III. Paramètres du modèle

A. Paramètres liés à l'entraînement

Ces paramètres influencent directement l'apprentissage du modèle.

- **epochs = 30 :** Nombre de passes complètes sur l'ensemble des données d'entraînement. Un nombre insuffisant d'époques peut entraîner un sous-apprentissage, tandis qu'un nombre excessif risque de provoquer un sur-apprentissage. Nous avons expérimenté avec 50, 100 et même 200 époques, sans obtenir de résultats significativement meilleurs. 30 époques offrent donc un bon compromis entre convergence et généralisation.
- **batch = 8 :** Nombre d'images utilisées à chaque itération avant la mise à jour des poids du modèle. Un batch trop petit peut rendre l'optimisation instable, tandis qu'un batch trop important exige une plus grande capacité mémoire GPU. Une valeur de 8 représente ainsi un équilibre optimal entre efficacité de l'apprentissage et utilisation des ressources disponibles.
- **device = 0 :** Indique que l'entraînement se déroule sur le premier GPU disponible. L'utilisation du GPU accélère considérablement les calculs par rapport à un CPU.
- **half = True :** Active l'entraînement en demi-précision (float16). Cette approche réduit l'utilisation de la mémoire et accélère les calculs sur GPU, tout en conservant des performances comparables à la précision standard (float32). La mémoire économisée peut ainsi être allouée à d'autres paramètres gourmands en ressources (tels que batch ou imgsz).

B. Paramètres liés aux images

Ces paramètres influencent la taille et la transformation des images utilisées pendant l'entraînement.

- **imgsz = 736 :** Définit la taille des images d'entrée après redimensionnement. Une résolution plus élevée améliore la détection des petits objets, mais augmente également le coût computationnel. La valeur de 736 pixels constitue un compromis judicieux entre qualité d'image et efficacité, d'autant plus que des résolutions supérieures n'ont pas permis d'obtenir de meilleurs résultats, voire ont même légèrement dégradé les performances.

C. Paramètres d'augmentation des données

Ces paramètres modifient les images d'entraînement pour améliorer la robustesse du modèle.

- **mosaic = 1.0 :** Définit l'intensité de l'augmentation "mosaic", qui combine plusieurs images en une seule afin d'accroître la diversité des données. Cette technique favorise la généralisation du modèle en lui présentant des compositions variées des objets à détecter.
- **hsv_h = 0.005 :** Spécifie la variation maximale appliquée à la teinte (Hue) des images. Une légère variation permet d'améliorer la robustesse du modèle aux différences d'éclairage sans altérer excessivement les couleurs.
- **hsv_s = 0.7 :** Spécifie la variation maximale appliquée à la saturation (Saturation) des images. Une variation plus prononcée aide à gérer les différences de couleurs et de contrastes, renforçant ainsi la capacité du modèle à s'adapter à divers environnements lumineux.
- **hsv_v = 0.4 :** Spécifie la variation maximale appliquée à la luminosité (Value) des images. Cette augmentation permet au modèle de mieux s'adapter à des conditions de luminosité variées, ce qui est crucial pour des applications en conditions réelles.

Concernant ces trois derniers paramètres (**hsv_h, hsv_s, hsv_v**), leur optimisation permet d'améliorer la capacité du modèle à généraliser sur des images présentant des variations de couleurs et de luminosité. En ajustant ces valeurs, on cherche à maximiser la performance du modèle. Pour cela, une recherche par grille (*grid search*) est effectuée sur différentes valeurs, et la combinaison offrant les meilleures performances avec le modèle est sélectionnée.

IV. Résultats obtenus

A. Résultats quantitatifs

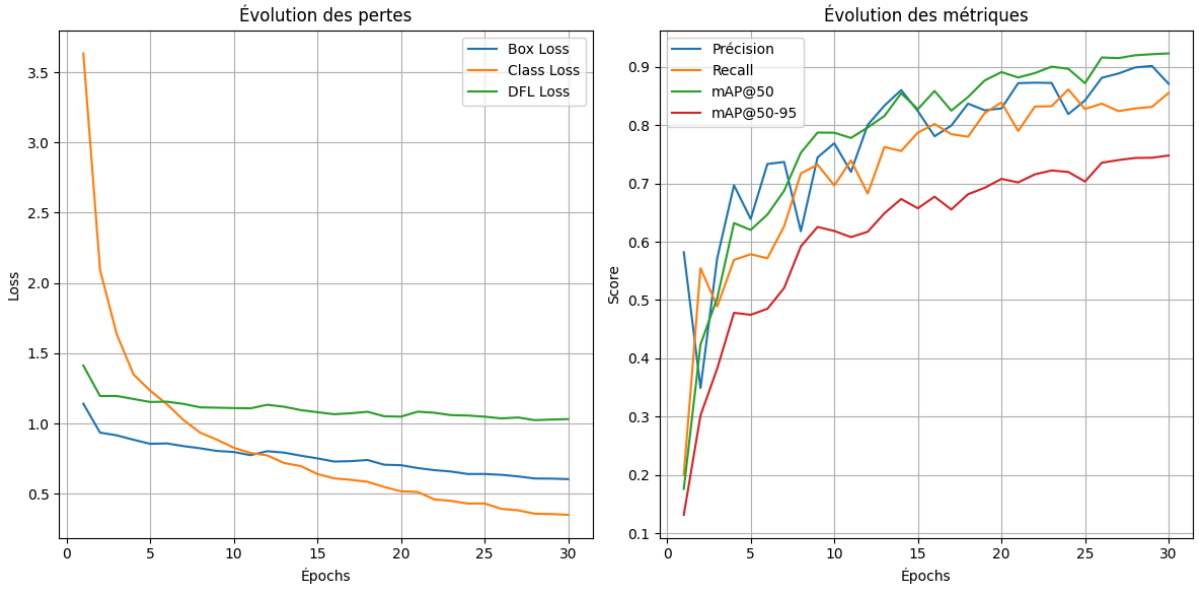
Dans cette section, nous présentons les performances finales de notre modèle ainsi que l'évolution des différentes métriques durant l'entraînement. Pour évaluer la qualité des prédictions, nous avons retenu plusieurs indicateurs :

- **Box Loss :** Mesure l'erreur de régression des boîtes englobantes (bounding boxes). Plus cette valeur est faible, plus les prédictions de position et de taille de l'objet sont précises.
- **Class Loss :** Évalue l'erreur de classification, c'est-à-dire la capacité du modèle à associer correctement chaque boîte englobante à la bonne classe d'objet.
- **DFL Loss (Distribution Focal Loss) :** Reflète la qualité de la répartition spatiale des prédictions. C'est une forme de loss qui améliore la précision de la localisation en affinant la prédiction des contours de la boîte englobante.

En plus de ces pertes (losses), nous suivons également les métriques suivantes :

- **Précision (Precision) :** Proportion de prédictions correctes parmi l'ensemble des prédictions faites par le modèle. Une haute précision indique que le modèle commet peu de faux positifs.
- **Rappel (Recall) :** Proportion d'objets correctement détectés parmi tous les objets présents dans l'image. Un haut rappel indique que le modèle commet peu de faux négatifs.
- **mAP@50 (Mean Average Precision à IoU=0.5) :** Moyenne de l'Average Precision pour chaque classe, avec un seuil d'Intersection over Union (IoU) fixé à 0.5. Elle reflète la capacité du modèle à détecter correctement les objets pour un seuil d'IoU donné.
- **mAP@50-95 :** Moyenne de l'Average Precision pour chaque classe, calculée sur plusieurs seuils d'IoU (de 0.5 à 0.95 par pas de 0.05). C'est une métrique plus stricte qui évalue la robustesse du modèle à différents niveaux de recouvrement.

Graphiques de l'évolution des métriques au cours de l'apprentissage



Le graphique de gauche "Évolution des pertes" illustre la diminution progressive des Box Loss, Class Loss et DFL Loss sur 30 époques. On observe :

- Une forte baisse du Class Loss lors des premières époques, indiquant que le modèle apprend rapidement à distinguer les classes.
- Une diminution régulière du Box Loss et du DFL Loss, démontrant que la localisation des objets s'améliore au fil des itérations.

Sur le graphique de droite "Évolution des métriques", on constate l'évolution des principales métriques de performance (Précision, Rappel, mAP@50 et mAP@50-95) :

- Précision et Rappel augmentent sensiblement au cours des premières époques, puis se stabilisent autour d'une valeur élevée.

- mAP@50 et mAP@50-95 suivent une progression régulière, avec une nette amélioration au cours des 10 à 15 premières époques. Par la suite, la hausse se poursuit de manière plus modérée, signe d'un raffinement progressif des prédictions.

De manière globale on remarque que :

- Les pertes (Box, Class et DFL) diminuent régulièrement, ce qui témoigne de l'apprentissage progressif du modèle.
- Les métriques de détection (Précision, Rappel, mAP@50 et mAP@50-95) augmentent de manière significative au fil des époques, pour atteindre des valeurs satisfaisantes en fin d'entraînement.

À l'issue des 30 époques d'entraînement, les performances mesurées sur l'ensemble de validation sont :

- **Box Loss :** 0.6045
- **Class Loss :** 0.3505
- **DFL Loss :** 1.032
- **Précision :** 0.871
- **Rappel :** 0.855
- **mAP@50 :** 0.923
- **mAP@50-95 :** 0.748

Ces résultats montrent que le modèle parvient à détecter correctement la majorité des objets (haut rappel), tout en produisant relativement peu de fausses détections (haute précision). Par ailleurs, le mAP@50-95 de 0.748 témoigne d'une bonne robustesse sur différents seuils d'IoU, confirmant que la localisation et la classification des objets restent performantes dans des conditions variées de recouvrement. L'entraînement par transfert (finetuning) réalisé ici a permis d'obtenir un modèle de détection d'objets précis et fiables, comme l'indiquent les différentes métriques de performance. L'évolution des pertes et des scores au cours des époques témoigne d'une convergence progressive et d'un bon compromis entre sous-apprentissage et sur-apprentissage.

En plus des performances globales, il est intéressant d'analyser les résultats par catégorie de produits sur le jeu de validation. Le tableau ci-dessus récapitule, pour chaque classe, le nombre d'images et d'instances testées, ainsi que la précision (**P**), le rappel (**R**), le **mAP@50** et le **mAP@50-95** obtenus.

Class	Images	Instances	P	R	mAP50	mAP50-95
all	102	1338	0.871	0.855	0.923	0.748
Bakery	8	64	0.834	0.944	0.971	0.671
Biscuits	8	68	0.838	0.971	0.988	0.823
Bombons	4	26	0.796	0.9	0.946	0.775
Canned	4	89	0.728	0.482	0.668	0.447
Cereals	7	63	0.834	0.683	0.891	0.78
Cheese	5	68	0.915	0.779	0.909	0.694
Chips	2	24	0.671	0.958	0.929	0.616
Choco	3	31	0.968	0.992	0.992	0.912
Coffee	7	91	0.989	0.997	0.994	0.871
DriedFruitsAndNuts	1	3	0.62	1	0.995	0.995
IceTea	1	16	1	0.149	0.88	0.656
Juices	2	54	1	0.99	0.995	0.836

Class	Images	Instances	P	R	mAP50	mAP50-95
Milk	6	70	0.917	0.814	0.862	0.771
Pasta	5	54	0.959	0.926	0.978	0.818
Rice	5	58	0.979	0.931	0.99	0.919
Sauce	11	170	0.923	0.981	0.978	0.841
Snacks	2	20	0.902	0.85	0.932	0.753
SoftDrinks	1	11	0.491	0.909	0.683	0.392
Soups	1	22	0.704	0.455	0.603	0.408
Spices	3	37	0.947	1	0.994	0.784
Spreads	3	60	1	0.888	0.987	0.81
Tea	9	122	0.952	0.967	0.992	0.841
Water	3	33	1	0.846	0.97	0.827
Yoghurt	3	41	0.832	1	0.965	0.697
Chocolate	3	43	0.976	0.966	0.988	0.759

- Catégories avec d'excellentes performances globales :

    - *Coffee* et *Juices* affichent des valeurs de précision et de rappel très élevées, indiquant que la quasi-totalité des objets de ces classes sont correctement détectés et peu de faux positifs sont produits.
    - *Choco*, *Chocolate* et *Rice* présentent également une excellente détection, avec un équilibre solide entre précision et rappel.
    - Certaines classes obtiennent une précision parfaite, comme *IceTea*, *Juices*, *Spreads* et *Water*, même si leur rappel peut varier.

- Catégories à haut rappel :

    - *DriedFruitsAndNuts* et *Spices* atteignent un rappel parfait, ce qui signifie que tous les objets de ces classes ont été détectés. Les précisions de 0.620 et 0.947 respectivement restent toutefois dépendantes du nombre total d'exemples testés et du contexte visuel de ces produits.
    - *Yoghurt* affiche également un rappel maximal, associé à une précision de 0.832.

- Catégories présentant un déséquilibre entre précision et rappel :

    - *IceTea* : La précision parfaite indique que lorsque le modèle détecte un produit de cette classe, il ne se trompe pas. Cependant, le faible rappel révèle qu'il en détecte très peu, possiblement à cause d'un nombre d'exemples limité ou de variations visuelles importantes.
    - *SoftDrinks* : Le rappel élevé montre que le modèle parvient à détecter la plupart des boissons gazeuses, mais la précision plus faible signale un taux non négligeable de fausses détections.

- Catégories plus difficiles à détecter :

    - *Canned* et *Soups* se distinguent par un rappel relativement bas. Cela peut s'expliquer par la grande diversité visuelle des emballages ou par la similarité avec d'autres catégories, rendant la détection plus complexe.
    - *Chips* montre un rappel très élevé mais une précision plus modeste, suggérant que le modèle confond parfois les chips avec d'autres emballages similaires.

- Exemples de classes équilibrées :

    - *Bakery* et *Biscuits* présentent un bon compromis entre précision et rappel, avec une capacité de détection très satisfaisante (mAP@50 élevé).
    - *Sauce* illustre également cette cohérence, ce qui se traduit par une identification fiable des sauces, malgré leur potentiel grand nombre de variantes.

Dans l'ensemble, ces résultats par produit confirment les performances solides du modèle : la plupart des classes obtiennent une mAP@50 supérieure à 0.90 et une mAP@50-95 dépassant 0.70. Les écarts observés pour certaines catégories peuvent être dus à un

nombre d'images réduit, à des emballages particulièrement variables ou à une similarité visuelle avec d'autres produits. Néanmoins, le modèle démontre une bonne capacité de généralisation et une robustesse appréciable, notamment grâce au finetuning effectué et aux techniques d'augmentation de données utilisées. Nous allons donc maintenant regarder les résultats du modèle sur les données de test afin de confirmer les hypothèses émises grâce aux résultats quantitatifs et la capacité du modèle à généraliser.

- **Catégories à performance stables :**

Class	Images	Instances	P	R	mAP50	mAP50-95
all	102	1341	0.807	0.834	0.848	0.694
Bakery	13	96	0.821	0.812	0.831	0.592
Biscuits	5	59	0.905	0.972	0.98	0.748
Bombons	4	26	0.527	0.962	0.642	0.496
Canned	7	127	0.903	0.89	0.934	0.805
Cereals	3	54	0.923	0.668	0.835	0.684
Cheese	9	100	0.877	0.923	0.942	0.75
Chips	5	78	0.79	0.923	0.929	0.593
Choco	1	6	0.494	1	0.872	0.78
Coffee	10	127	0.935	0.911	0.956	0.839
Creme	1	21	0.887	0.905	0.962	0.767
DriedFruitsAndNuts	5	34	0.794	0.906	0.897	0.726
IceTea	1	11	0.346	0.727	0.614	0.539

Class	Images	Instances	P	R	mAP50	mAP50-95
Juices	3	57	0.882	0.982	0.969	0.85
Milk	6	34	0.785	0.858	0.863	0.725
Oil-Vinegar	4	63	0.846	0.714	0.871	0.64
Pasta	6	68	0.726	0.857	0.811	0.633
Rice	2	25	0.965	0.48	0.804	0.72
Sauce	3	53	0.882	0.986	0.98	0.791
Snacks	6	61	0.799	0.656	0.636	0.52
Spices	5	65	0.726	1	0.916	0.837
Spreads	2	39	0.855	0.872	0.968	0.871
Tea	2	26	0.796	1	0.995	0.871
Water	2	28	0.893	0.893	0.945	0.715
Yoghurt	5	56	0.815	0.945	0.961	0.796
Chocolate	3	27	1	0	0.0831	0.0593

Que ce soit en termes de précision, rappel, mAP50 ou mAP50-95, le modèle parvient à obtenir des performances stables en prédiction des articles suivants : *Coffee*, *Tea*, *Juices*, *Spreads*, *Water*, *DriedFruitsandNuts*, *Yoghurt*, *Chips*, *Bakery*, *Biscuits*, *Sauce*, *Cheese* et *Milk*.

- **Catégorie à précision dégradée :**
  Pour un niveau de rappel stable, les articles suivants font l'objet d'une baisse en précision : *Choco*, *Spices*, *Bombons* et *Pasta* à l'exception de *IceTea* qui montre même une hausse en précision.
- **Catégorie à rappel dégradé :**
  Pour un niveau de précision stable, les produits suivants voient leur rappel baisser par rapport aux performances sur les données de validation : *Rice*, *Snacks* et *Chocolate* ayant son rappel réduit zéro (signifiant beaucoup de faux négatifs).
- **Catégorie à performances améliorées :**
  Enfin, les produits suivants font l'objet de performances améliorées pour le modèle, que ce soit en termes de précision ou de rappel : *Cereals* (bien que le rappel soit resté stable) et *Canned*.

En somme, cela montre que le modèle est capable de généraliser sur toute une panoplie d'articles, mais voient ses performances réduites pour certains produits plus complexes à détecter (articles à faible rappel) ou à bien classer (articles à faible précision). Ces derniers pourraient faire l'objet d'une étude approfondie afin d'identifier les raisons d'une performance dégradée (grande variété de packaging, orientation, ombre, etc.)

B. Résultats qualitatifs

Dans cette section, nous présentons une analyse qualitative des capacités de généralisation du modèle. Pour cela, nous étudions les prédictions effectuées sur un autre jeu de données issu d'une distribution différente des données de base, et sur lesquelles nous n'avons pas effectué de labellisation. En effet, les images constituant le dernier ont été prises dans des supermarchés français, tandis que celles des données de base viennent d'un autre pays (peut-être américain).

- **Catégories d'articles détectables :**
  La plupart des articles semblent pouvoir être détectés par le modèle, mais une bonne détection ne signifie pas nécessairement une bonne classification.
  - Catégories d'articles bien classés :
    - La majorité des boîtes de céréales est bien détectée et classée.
    - Les snacks (hormis chips et fruits secs) sont bien classés mais parfois confondus avec des boîtes de céréales voire des paquets de bonbons, à la différence des paquets de chips (cf. "articles mal classés").
    - Les jus de fruits, les paquets de pâtes ainsi que les bouteilles d'huile et de vinaigre sont quant à eux pratiquement parfaitement détectés et classés. C'est également le cas des boîtes de conserve et des sachets de bonbons, bien qu'une partie ne soit pas détectée du tout, probablement en raison d'un phénomène d'ombre sur les images.

  - Catégories d'articles mal classés :
    - Le thé et le café sont systématiquement détectés mais confondus avec d'autres articles (céréales, conserves (*canned*), pâtes, biscuits).
    - De même pour le chocolat, confondus avec du riz ou des céréales, à l'exception d'une unique classification correcte.
    - Les chips sont systématiquement détectés mais confondus avec des snacks principalement (céréales et pâtes également). Même si ce n'est pas totalement faux, la tâche de classification donnée au modèle distinguait pourtant les snacks des chips.
    - De même pour les bouteilles d'IceTea. Systématiquement détectées, elles sont néanmoins inexorablement classées comme sodas. Même si ça n'est pas faux, le modèle avait des informations pour distinguer l'IceTea des autres sodas.
    - Les bouteilles de lait quant à la moitié d'entre-elles qui ont pu être détectées, sont, avec certains emballages bien classés, mais pour les autres, une confusion fréquente avec de nombreux articles a lieu (pâtes, jus de fruit, pâtisseries, eau, bonbons).
    - Les paquets de riz, selon la forme de leur emballage, sont systématiquement, à l'exception d'une classification, confondus avec des pâtes, des céréales et des snacks..
    - Les épices, quant à elles, sont largement confondues avec principalement de l'eau, sinon, elles ne sont en grande partie pas détectées du tout. Cela peut s'expliquer par un mauvais éclairage de certaines étagères.

- Du côté des yaourts, en grande partie détectés, un nombre non négligeable d'articles sont confondus avec du fromage.
- Tandis que le fromage en portion emballé dans un plastique transparent, également non systématiquement détecté, est uniquement classifié par le modèle comme de la pâtisserie (*bakery*).
- Les produits de crèmerie (crème fraîche, beurre, etc.), sont difficilement détectés et lorsque c'est le cas, il n'y a qu'une unique classification qui soit correcte, les autres classifications, alors erronées, touchent un large spectre d'autres produits (yaourts, chocolats, etc.).
- Les produits de type condiments et sauces sont bien détectés, mais systématiquement mal classés et confondus avec de nombreux produits : conserves, huile, vinaigre, soda et IceTea.

- **Catégories d'articles indétectables :**
  Parmi ces articles, le fromage camembert est notamment pratiquement impossible à détecter pour notre modèle. Ce qui peut s'expliquer par le fait que la forme circulaire de l'article n'ait jamais été vue dans les données d'entraînement.

  En résumé, il faut comprendre que la différence de distribution suivie par les images de base (venant d'un supermarché étranger à la France) et celles des images prises dans un supermarché français, se manifeste par la différence de packaging. Ainsi cela induit, en raison du fait que le type d'emballage contient des *features* utiles au modèle pour la détection et la classification, des difficultés pour le modèle à classer correctement un certain nombre d'articles, voire à leur détection. Néanmoins, d'autres facteurs entrent en jeu comme la qualité photos de l'appareil, la luminosité, le flou cinétique ou encore la présence d'ombre.

V. Conclusion

Notre étude sur l'architecture YOLOv11 a permis de mettre en lumière l'efficacité des approches modernes de détection d'objets. Les résultats obtenus, illustrés par des métriques robustes telles qu'une précision élevée, un rappel significatif et des scores mAP remarquables, témoignent de la capacité du modèle à s'adapter à des contextes visuels variés et à relever des défis complexes en matière de classification et de localisation. Bien que certaines catégories aient révélé des difficultés particulières, notamment en termes de confusion entre classes, l'utilisation d'un mécanisme d'attention et de techniques d'augmentation des données a largement contribué à la robustesse globale du système. Ce travail ouvre la voie à de nouvelles perspectives d'optimisation et à l'application concrète des systèmes de vision par ordinateur dans divers domaines, soulignant ainsi l'importance d'une innovation continue pour améliorer la performance et la généralisation des modèles de détection d'objets.