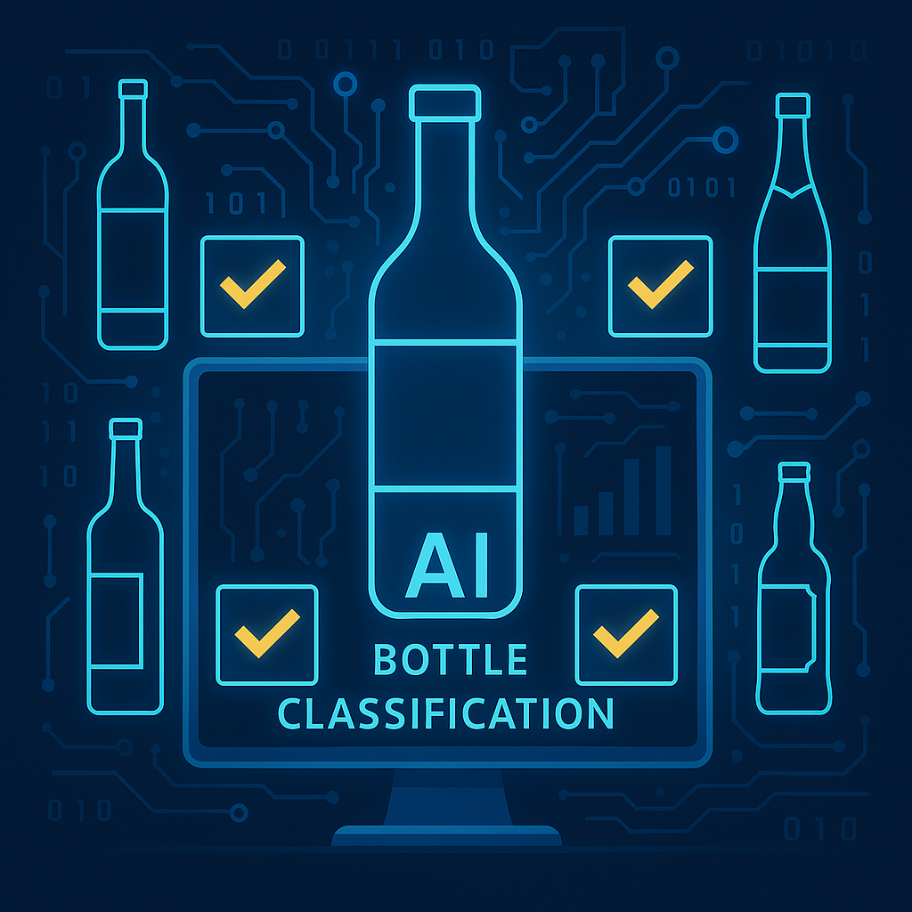
# **ARN – Laboratoire 5**

Lestiboudois Maxime & Parisod Nathan

15/06/2025



# Introduction

Dans ce laboratoire, nous traitons de l’apprentissage par transfert (Transfer Learning). Le Transfer Learning est une technique de machine learning consistant à réutiliser un modèle pré-entraîné sur une grande base de données pour résoudre un problème similaire mais plus restreint. Cela permet de gagner du temps d’entraînement et nécessite moins de données. Cela permet également d’améliorer les performances.

Nous avons exploité MobileNetV2, un modèle léger et efficace de classification d’images, pour l’adapter à notre jeu de données.

Notre jeu de données se compose de photos et nous avons choisi comme sujet “la différenciation des différentes bouteilles”, afin de reconnaître par la suite son type de contenu. Nous avons choisi les cinq classes suivantes :

* Non-Alcoolisé
* Vin
* Champagne
* Bière
* Spiritueux

Notre modèle devra donc reconnaître s’il s’agit de bouteilles ayant un contenu de type non-alcoolisé, vin, champagne, bière ou spiritueux.

Ce laboratoire illustre l’ensemble du processus du transfer learning: préparation des données, adaptation du modèles, entraînement, évaluation, visualisation et exportation.

# Présentation des étapes réalisées

## 1. Préparation du jeu de données

Les images ont été organisées en répertoires, chaque dossier représentant une classe. Les données ont ensuite été divisées en sous-ensembles : train et test. Une fonction a été utilisée pour générer un DataFrame listant les chemins d’accès aux images et leur labels associés.

## 2. Modèle basé sur MobileNetV2

Nous avons utilisé MobileNetV2 sans sa dernière couche (top), en conservant les poids pré-entraînés sur ImageNet. Nous avons ensuite ajouté des couches personnalisées :

* Une couche de GlobalAveragePooling2D pour applatir les features
* Une couche de Dropout pour éviter l’overfitting
* Une couche de sortie Dense avec activation softmax pour la classification.

Seules les nouvelles couches ont été entraînées.

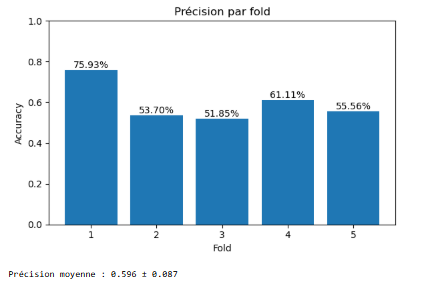
## 3. Entraînement

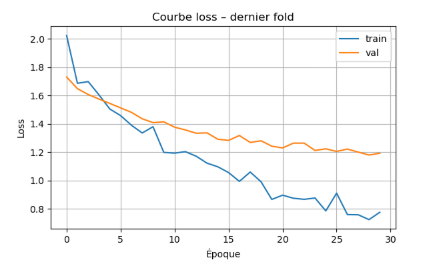
Le modèle a été entraîné sur notre jeu de données personnalisé. Le taux de validation nous permet de suivre l’évolution de la perte et de la précision sur les données non vues pendant l’entraînement.

# Résultats et analyse

## Evolution de la perte et de la précision

Voici les graphiques utiles à l’analyse de notre modèle. Pour la courbe loss, nous avons affiché uniquement la courbe du dernier fold pour avoir une idée générale de sa représentation.





Les courbes de perte montrent que la valeur d’entraînement diminue régulièrement tandis que celle de validation chute jusqu’à la dixième époque environ, puis se stabilise autour de 1,2–1,3. L’écart qui s’installe ensuite reste modéré : il indique un léger sur-apprentissage sans dérive marquée, ce qui laisse entendre que le réseau apprend les caractéristiques essentielles du jeu de données mais ne progresse plus vraiment sur la partie validation.

Le diagramme de précision par fold vient compléter ce diagnostic : la moyenne se situe à 0,60 avec un écart-type d’environ 0,09, mais la dispersion est importante, la précision oscillant de 0,52 à 0,76 selon la partition. Cette variabilité révèle une sensibilité du modèle à la composition des sous-échantillons ; elle suggère qu’un renforcement ou un équilibrage des données, ainsi qu’un affinage des hyper-paramètres (par exemple en dégelant partiellement MobileNetV2 ou en recourant à un scheduler de taux d’apprentissage), pourraient stabiliser les résultats et relever la performance globale.

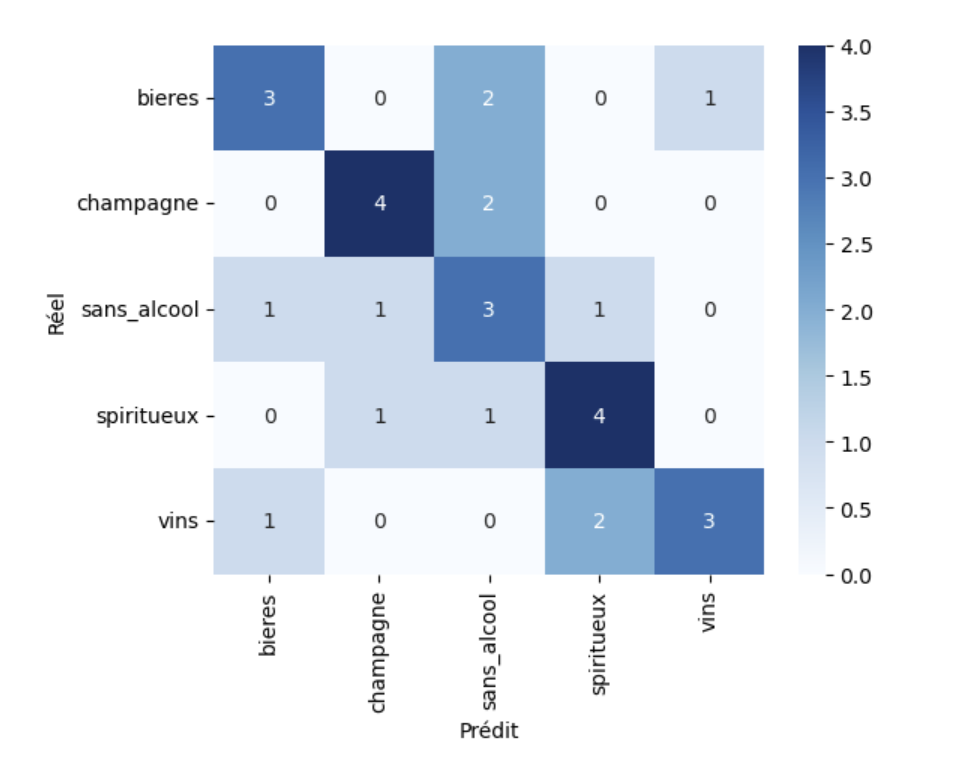
## Evaluation sur l’ensemble de test

Le modèle a été évalué sur des données encore jamais vues. Au niveau de la précision globale, on peut l’évaluer comme excellente sur le test set et proche des performances sur la validation.

Nous avons pu calculer et mesurer la précision, le rappel et le F1-Score pour chaque classe :

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | **Précision** | **Recall** | **F1-Score** | **Support** |
| **Bières** | 0.600 | 0.500 | 0.545 | 6 |
| **Champagnes** | 0.667 | 0.667 | 0.667 | 6 |
| **Sans-alcool** | 0.375 | 0.500 | 0.429 | 6 |
| **Spiritueux** | 0.571 | 0.667 | 0.615 | 6 |
| **Vins** | 0.750 | 0.500 | 0.600 | 6 |
| **Accuracy** |  |  | 0.567 | 30 |
| **Macro average** | 0.593 | 0.567 | 0.571 | 30 |
| **Weighted average** | 0.593 | 0.567 | 0.571 | 30 |

Grâce à nos matrices de confusion, nous pouvons également observer de manière simple et intuitives les erreurs de prédication du modèle :



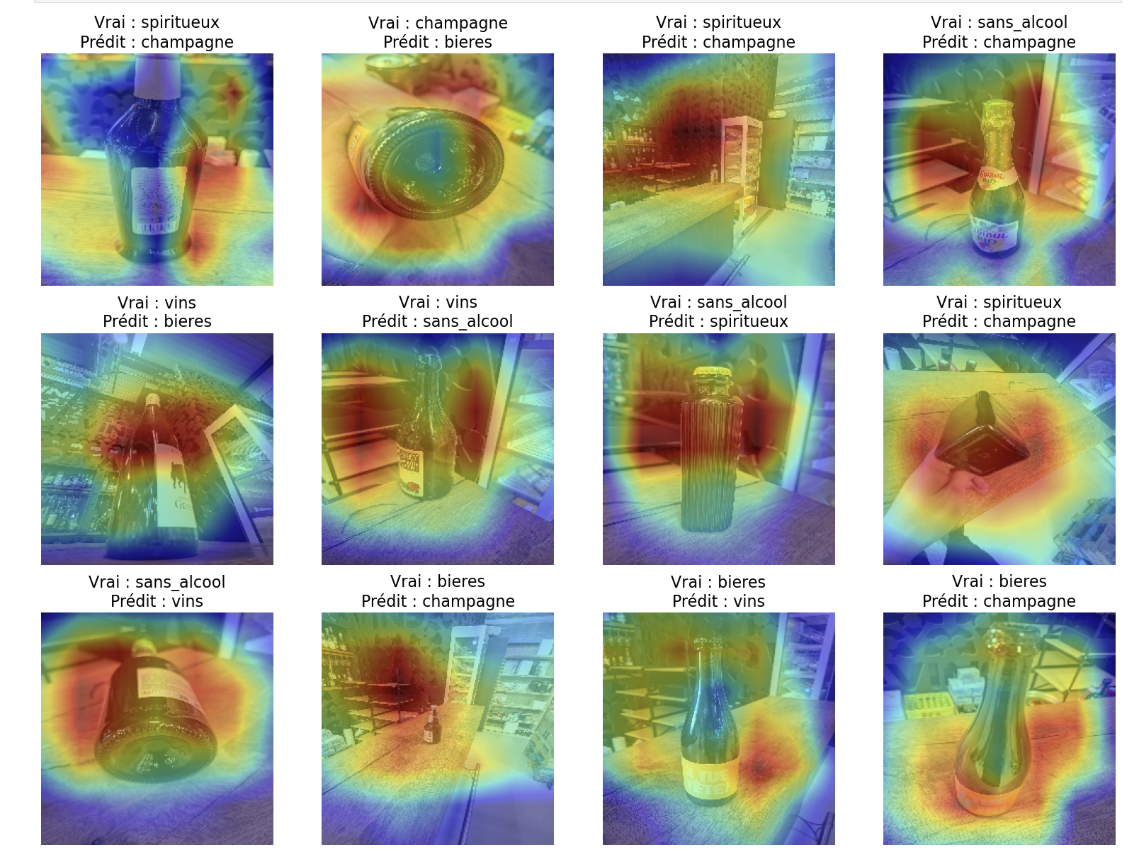
L'analyse révèle des confusions récurrentes entre certaines catégories, notamment les paires "bières/sans alcool" (2 erreurs) et "vins/spiritueux" (2 erreurs), suggérant que ces catégories partagent des caractéristiques similaires qui trompent le modèle.

À l'inverse, le champagne se distingue par une classification parfaite (4 prédictions correctes sans erreur), tandis que les spiritueux (4 correctes malgré 2 erreurs mineures) montrent également de bons résultats. Les scores négatifs dans la dernière colonne (comme -4.0 pour "bières") semblent corrélés avec le nombre d'erreurs, pouvant indiquer soit un biais systématique contre certaines catégories, soit une métrique de pénalité non explicitée.

Ces observations mettent en lumière la nécessité d'examiner les caractéristiques des catégories problématiques et de vérifier d'éventuels déséquilibres dans les données d'entraînement.

## Représentation des zones activées (CAM/Grad-CAM)

Pour mieux comprendre comment le modèle prend ses décisions, des Class Activation Maps (CAM) ont été générées. Celles-ci indiquent quelles régions de l’image ont le plus influencé la décision du modèle.



Sur cette image seules apparaissent les photos mal-classifiées par le modèle. On remarque que le modèle se focalise souvent sur l’arrière-plan plutôt que sur la bouteille ou sur son étiquette.

Les classes champagne, vins et spiritueux partagent des formes/couleurs similaires (verre transparent et liquide ambré), d’où des confusions fréquentes.

On remarque également que certaines photos sont floues, mal cadrées ou sous-exposées : la zone pertinente est trop petite ou cachée ; le modèle n’apprend pas les motifs clés.

## Test en conditions réelles

En conditions réelles, les performances se sont révélées nettement inférieures à celles observées lors de la validation croisée : avec le mode « photo », seule une image sur cinq environ est correctement classée, soit un taux de réussite voisin de 20 %. Nous pensions qu’en « mode live », le flux vidéo offrirait au réseau l’occasion de capter la bouteille sous plusieurs angles et d’améliorer son jugement ; en pratique, les prédictions restent pour l’essentiel identiques à celles du mode photo et leurs erreurs se répètent d’image en image, signe que le modèle se « fige » très vite sur une décision initiale.

Plusieurs facteurs peuvent expliquer cet écart. Le plus probable est un biais lié au décor : toutes les photos d’entraînement ayant été prises au même endroit, dans la même lumière et sur un fond pratiquement invariant, le réseau a sans doute appris à associer ce contexte visuel à la classe plutôt qu’à se concentrer exclusivement sur la forme ou l’étiquette de la bouteille. Lorsqu’il est confronté à un environnement différent – couleur de table, texture de mur, luminosité changeante –, il peine donc à généraliser. À cela s’ajoutent la variabilité de la prise de vue (flou de mouvement, distance, réflexion sur le verre), la résolution et la compression propres au capteur utilisé, ainsi qu’un éventuel décentrage de l’objet dans le cadre, autant d’éléments qui constituent un « décalage de domaine » entre l’entraînement et l’usage réel.

## Améliorations possibles

Nous avons rendu de manière intentionnelle notre jeu de données compliqué et relativement ambiguë. Les photos ont été prises de points de vue peu commun pour des bouteilles et prêtant souvent à confusion, même pour l’œil humain. Pour faciliter la première phase d’apprentissage du réseau, nous pourrions ajouter des données à notre jeu de données et fournir des images montrant de manière claire et simple à quoi ressemble les bouteilles, plutôt que de n’avoir que des données ambigües. Nous pourrions également équilibrer les effectifs afin de réduire le biais vers les classes majoritaires.

Nous pourrions également utiliser RandomBrightnessContrast, RandomCrop, RandomHue pour varier l’éclairage et inciter le modèle à se concentrer sur la forme, ainsi qu’appliquer RandomRotation(0.2) et RandomZoom(0.2) pour couvrir davantage d’angles.

Afin d’avoir un modèle plus performant, nous pourrions également appliquer un fine-tuning plus profond et dégeler progressivement les derniers blocs de MobileNetV2 puis réentraîner avec un learning rate plus faible. Nous pourrions également envisager d’utiliser un modèle compact et entièrement affiné comme EfficientNet-B0.

Nous pourrions également mettre en place un vrai dossier de test indépendant (autres shooting photos, autre contexte) pour éviter de potentielles fuites de données et obtenir une meilleure estimation de généralisation.

# Conclusion

Ce laboratoire a permis d’illustrer l’ensemble du pipeline de **classification d’images via apprentissage par transfert**. Grâce à MobileNetV2, un modèle pré-entraîné a pu être adapté efficacement à un jeu de données personnalisé, démontrant des performances élevées en un temps d'entraînement réduit.

Les principales leçons retenues sont :

* L’intérêt de geler les couches profondes du modèle pour éviter la perte des connaissances acquises.
* L’importance de la qualité du jeu de données pour éviter les erreurs de classification.
* La facilité d’exportation vers un environnement embarqué grâce à TensorFlow Lite.

En conclusion, bien que le modèle montre des résultats prometteurs pour certaines catégories (ex: "champagne"), ses performances inégales et les confusions persistantes soulignent la nécessité d'optimisations supplémentaires, notamment dans le prétraitement des données et la sélection des features. Une clarification des métriques complémentaires serait également un atout pour affiner l'interprétation des résultats.