

Projet annuel - 2024-2025

Identification de véhicules

CODEN Erwann (22001714)
DELILLE Matthieu (22102252)
MAFILLE Léo (22102705)
PREUX Marilou (22108693)



UNIVERSITÉ
CAEN
NORMANDIE

Table des matières

1	Introduction	3
1.1	But	3
1.2	Répartition de la charge du travail	3
2	Datasets	5
2.1	Analyse de l'existant	5
2.1.1	Compcars (étudié par Léo MAFILLE)	5
2.1.2	60,000+ Images of Cars (étudié par Erwann CODEN)	5
2.1.3	Stanford Cars Dataset (étudié par Marilou PREUX)	5
2.1.4	VMMR Cars Dataset (étudié par Matthieu DELILLE)	6
2.1.5	Récapitulatif des datasets analysés	7
2.2	Analyse du marché	7
2.2.1	Comparaison avec le marché français	9
2.2.2	Espace de recherche	9
2.3	Scraping	10
2.3.1	Google Images	10
2.3.2	Cars.com	10
2.3.3	Leboncoin	11
2.4	Filtrage des résultats	11
2.4.1	Anti-doublons	12
2.4.2	Second filtre	13
2.4.3	Méthode sans ré-apprentissage	14
2.4.4	Tri manuel	14
2.5	Présentation de notre dataset	14
3	Modèles	17
3.1	Méthode d'entraînement	17
3.2	Analyse de résultats et comparaison	18
4	Travail effectué en plus	22
4.1	Plaque d'immatriculation	22
4.1.1	Traitement de l'image	22
4.1.2	Reconnaissance du texte et post-traitement	22
4.1.3	Méthode dite "du bourrin"	22
4.2	Détection de texte avec EasyOCR	23
4.2.1	Résultats	24
5	Conclusion	25
6	Sources	25

Remerciements

Nous tenons à remercier messieurs Tanguy GERNOT et Emmanuel GIGUET pour le suivi du projet, les informations et l'aide qu'ils nous ont apportée tout au long de ce projet annuel. Le projet n'aurait pas été réalisé avec autant de qualité sans leurs conseils avisés.

1 Introduction

Dans un monde où la circulation routière devient de plus en plus dense, la reconnaissance et l'identification automatique des véhicules représentent des enjeux cruciaux en matière de sécurité, de surveillance et de gestion du trafic.

Ce projet vise à concevoir un système d'identification de véhicules en utilisant des techniques d'intelligence artificielle, capable de reconnaître la marque et le modèle d'un véhicule, focalisé sur le marché automobile français. Ce travail allie traitement d'image, classification, et entraînement de modèles d'apprentissage profond.

A aussi été requise la construction d'une base de données vaste, nécessaire aux tâches d'apprentissage automatique. Une grande partie de ce rapport est consacrée à cette tâche. De ces remarques là, plusieurs problématiques se sont imposées à nous durant tout le long du projet :

1. Comment s'assurer d'avoir une base de données d'images suffisante et adaptée dans le cadre de notre pays ?
2. Comment adapter un modèle au marché automobile français ?
3. Comment concevoir un modèle d'intelligence artificielle capable d'identifier automatiquement un véhicule à partir d'une image, tout en garantissant précision, robustesse et adaptabilité aux conditions réelles ?

1.1 But

En France, de plus en plus de modèles de voitures apparaissent sur le marché au fur et à mesure des années, et chacun des modèles peut se décliner de plusieurs façons. Pour suivre cette évolution, la création d'un système permettant l'identification d'un véhicule avec précision devient intéressante et importante.

Notre projet consiste donc en la création d'un système d'identification de véhicule à partir d'une photo. Notre système devra donc être capable de détecter des informations du dit véhicule, comme sa marque, le nom de son modèle, ou encore la déclinaison en question puisqu'un même modèle peut avoir plusieurs versions, chacune sortie à des périodes différentes.

Dans un premier temps, le travail se concentre sur l'analyse du marché français dans l'optique de définir ce que nous chercherons à collecter comme données. Un second temps est consacré à la collecte et au tri de ces données pour les rendre exploitables. Pour finir, nous exploiterons lesdites données via un apprentissage d'un réseau de neurones pour répondre à l'objectif du projet cité ci-dessus.

1.2 Répartition de la charge du travail

- Matthieu s'est occupé de la création du modèle, d'un filtre de suppression des doubles et d'un filtre de pré-tri.
- Erwann s'est occupé de la partie concernant les plaques d'immatriculation et du script pour récupérer les images provenant du site leboncoin.
- Léo s'est occupé de la création de script pour récupérer les images de cars.com, d'un autre filtre de suppression des doublons et d'un filtre de pré-tri d'images. Il s'est également occupé des scripts d'analyse du marché français. Il s'est également occupé de l'intégralité du scraping des images sur le site cars.com, et s'est spécialisé dans les marques Audi et BMW.

- Marilou s’est occupée de régler des bugs dans le script de récupération et de la rédaction d’une grande partie du rapport en latex.

Au début du projet, chaque membre a également analysé un dataset existant.

Chaque membre du groupe a participé à sa façon à la création du dataset en se répartissant les marques à rassembler. Certaines marques prennent plus de temps dû à moins d’images présentes pour chaque classe.

Tr Marque	Tr Personne	Nombre de classes
Audi	Léo Erwann	21
BMW	Léo Matthieu	16
Citroën	Matthieu	12
Dacia	marilou	8
Fiat	Léo Erwann	9
Ford	Léo Marilou	10
Hyundai	Léo Matthieu	9
Kia	Léo Erwann	12
Mercedes-Benz	Léo marilou Erwann	18
Mini	Erwann	8
Nissan	Erwann	13
Opel	Erwann	21
Peugeot	Matthieu	16
Renault	Erwann	30
Tesla	Erwann	4
Toyota	Erwann	14
Volkswagen	Erwann	24
Total		245

FIGURE 1 – Répartition des marques scrapées entre chaque membre du groupe

2 Datasets

Pour que notre modèle d'apprentissage puisse correctement attribuer un modèle et une marque à une image qui lui est donnée, il faut un dataset. Un dataset constitue la base des connaissances des modèles d'apprentissage. Dans notre cas, il s'agit d'une grande quantité d'images de voitures labellisées, c'est-à-dire qu'à chaque image du dataset est également attribuée sa marque et son modèle.

Cette récolte de données constitue une partie essentielle de notre projet. Elle s'est faite en 2 parties : une première sur l'analyse des datasets existant, une deuxième sur nos techniques et méthodes pour créer notre propre dataset adapté au projet.

2.1 Analyse de l'existant

Tout d'abord, nous avons analysé différents datasets déjà existants, afin de comprendre comment un dataset est formé, et ce qui différencie plusieurs datasets. Voici ici la liste des datasets analysés par le groupe :

2.1.1 Compcars (étudié par Léo MAFILLE)

Ce dataset semble intéressant au premier abord, notamment par le fait qu'il aborde une grande variété de modèles différents (plus de 1.600 modèles), et que l'organisation des fichiers le rend labellisé, en plus de proposer une hiérarchisation des informations.

Cependant, plusieurs problèmes ont été repérés sur ce dataset, le problème principal étant que les voitures présentées sont exclusivement de marques chinoises. En effet, dans notre projet, nous ne nous intéresserons qu'au marché français, ainsi un modèle entraîné sur des données basées sur un marché différent ne peut qu'être désavantageux.

Compcars Dataset : <https://www.kaggle.com/datasets/renancostaalencar/compcars>

2.1.2 60,000+ Images of Cars (étudié par Erwann CODEN)

Le dataset présente plusieurs avantages notables dans le cadre de la reconnaissance de véhicules. Il contient plus de 300 modèles de véhicules, chacun étant entièrement labellisé avec des informations détaillées, ce qui facilite son exploitation. De plus, il inclut des marques largement présentes sur les routes françaises, comme Toyota, Audi et Nissan, le rendant plus pertinent que le dataset précédent.

Cependant, ce dataset présente certaines limites. Il ne contient pas les marques les plus populaires en France, telles que Renault, Peugeot, Citroën et Dacia, ce qui réduit son utilité pour notre projet. De plus, les images étant issues des sites officiels des constructeurs, elles ne comportent aucune plaque d'immatriculation française, empêchant alors de pouvoir en extraire le modèle avec une plus grande précision via la reconnaissance de celle-ci.

60,000+ Images of Cars : <https://www.kaggle.com/datasets/prondeau/the-car-connection-picture-dataset>

2.1.3 Stanford Cars Dataset (étudié par Marilou PREUX)

L'ensemble Stanford Cars Dataset comprend 16 185 images réparties en 196 classes de voitures, avec une répartition équilibrée entre 8 144 images pour l'entraînement et 8 041 pour les tests. Cette organisation structurée en fait un jeu de données adapté aux applications d'apprentissage automatique. On y trouve également des sous-ensembles

notables, tels que le BMW-10, qui se concentre sur dix modèles spécifiques de la marque BMW, offrant une granularité fine, et le Car-197, couvrant une large gamme de 97 types de voitures.

Chaque image du dataset est annotée de manière détaillée, incluant la marque, le modèle et l'année du véhicule. Les photos proviennent de sources variées, ce qui contribue à une certaine diversité en termes d'arrière-plans et de contextes visuels. Toutefois, malgré ces atouts, le dataset reste relativement limité en taille pour des tâches complexes de deep learning. De plus, certaines classes de véhicules sont sous-représentées, ce qui peut entraîner un déséquilibre dans l'apprentissage et potentiellement biaiser les performances des modèles développés.

Stanford Cars Dataset : <https://www.kaggle.com/datasets/jessicali9530/stanford-cars-dataset>

2.1.4 VMMR Cars Dataset (étudié par Matthieu DELILLE)

Le VMMR est un dataset paru en 2017 dans l'optique d'apporter une solution pour l'entraînement de modèles dans une situation de la vie réelle. Il contient 291 752 images de voitures réparties dans 9 170 classes différentes. La granularité choisie pour ce dataset est "make", "model" et "year" (correspondant à l'année de fabrication du véhicule). Publié par des chercheurs américains, il a été construit à partir du scraping de deux sites de vente de voiture entre particuliers (amazon.com et craigslist.com).

L'ensemble permet de tester les différentes granularités possibles. Il se trouve que le principal problème dans l'identification de véhicule vient notamment de l'année de fabrication d'une voiture. Deux générations différentes d'un même modèle peuvent se ressembler très fortement, là où d'autres sont radicalement différentes. Se pose alors la question de quelle granularité nous cherchons. Les travaux déjà effectués tendent à dire que l'ajout de l'année de fabrication dans la classification réduit de manière drastique l'efficacité des modèles. L'idée du VMMR est de proposer une grande diversité de temporalités d'un même modèle mais ayant des années de fabrication différentes, pouvant s'étaler jusqu'à une dizaine d'années d'écart, afin de tester des modèles dessus.

Enfin, le fait de mélanger des datasets ayant des prises de vues différentes semblent améliorer grandement les performances, même sur des véhicules sur lequel un modèle n'est pas entraîné. Ainsi, une prise de vue depuis une caméra de surveillance, avec des véhicules en mouvement, et une prise de vue depuis une photo pour une annonce en ligne sont complémentaires l'une de l'autre.

VMMR Cars Dataset : <https://www.kaggle.com/datasets/abhishektyagi001/vehicle-make-model-recognition-dataset-vmmrdb>

2.1.5 Récapitulatif des datasets analysés

Nom du dataset	Avantage(s)	Inconvénient(s)
CompCars	<ul style="list-style-type: none">— Dataset labellisé— Grande variété de modèles— Hiérarchisation des informations	<ul style="list-style-type: none">— Uniquement des marques chinoises— Points de vue qui ne correspondent pas à nos besoins
60.000+ Images of Cars	<ul style="list-style-type: none">— Plus de 300 modèles de véhicules— Labellisé— Inclut des marques présentes en France	<ul style="list-style-type: none">— Ne contient pas les marques les plus populaires en France
Stanford Cars	<ul style="list-style-type: none">— Annotations détaillées— Variété de classes (196 classes)— Origine variée des images	<ul style="list-style-type: none">— Taille limitée du nombre d'images— Peu de voitures représentatives du parc français— Déséquilibre entre les classes
VMMR Cars	<ul style="list-style-type: none">— Beaucoup d'images— Beaucoup de classes	<ul style="list-style-type: none">— Point de vue intérieur— Beaucoup de classes avec très peu d'images

TABLE 1 – Comparaison des datasets automobiles

Ces ensembles de données offrent une base solide pour la classification et l'identification des véhicules. Cependant, chacun présente des limitations spécifiques, nécessitant parfois un mélange avec d'autres datasets. Le problème principal étant l'absence de données concernant notre marché cible.

2.2 Analyse du marché

Afin de préciser l'apprentissage du modèle que nous créerons dans la version finale du projet, il est important de savoir quelles sont les marques et les modèles les plus utilisés et vendus en France. Cela nous permettra notamment de décider pourquoi ces datasets ne seront pas intéressants, dans le cas du marché français. En récupérant les données

Dataset	Nombre d'images total	Nombre de classes	Images par classe
CompCar	136,726	1,716	79
VMMR	291,752	9,170	31
60.000+ Images of Cars	64,500	2,069	31
Stanford Car	16,185	196	82

TABLE 2 – Comparaison des datasets de voitures

du site fiches-auto grâce à la bibliothèque Python Selenium, nous avons pu obtenir les informations suivantes :

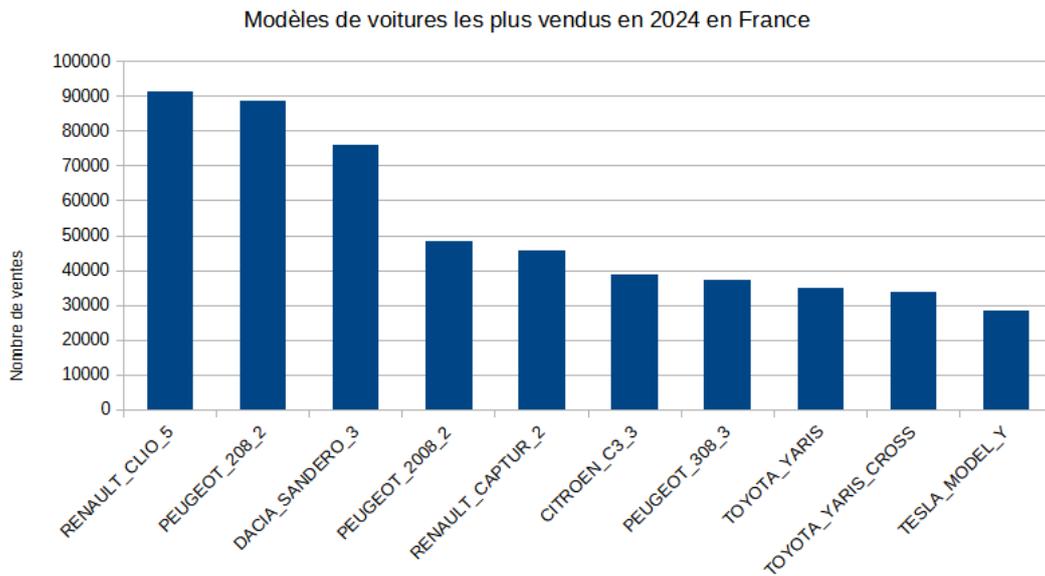


FIGURE 2 –

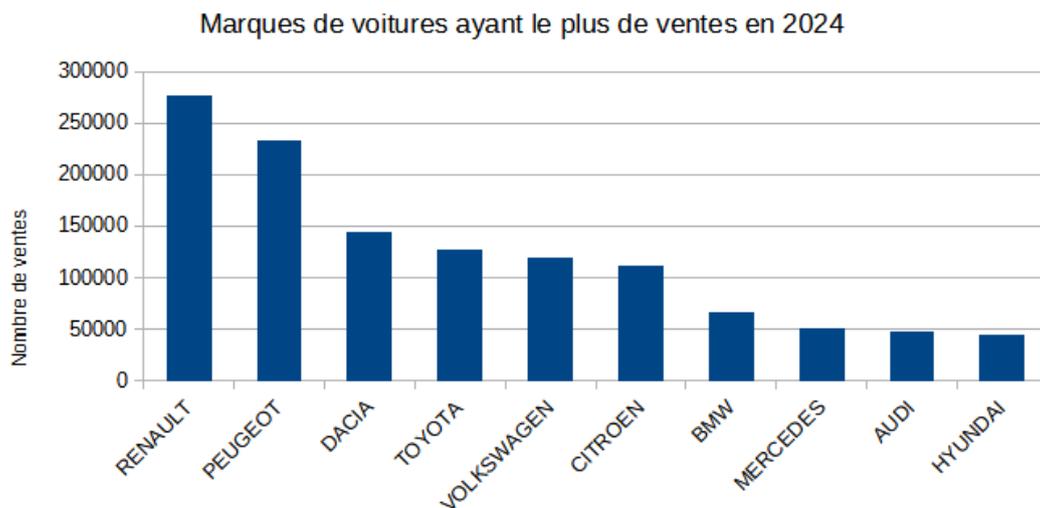


FIGURE 3 –

Grâce à ces données, nous savons quelles marques doivent être prises en compte dans le dataset que nous choisirons.

2.2.1 Comparaison avec le marché français

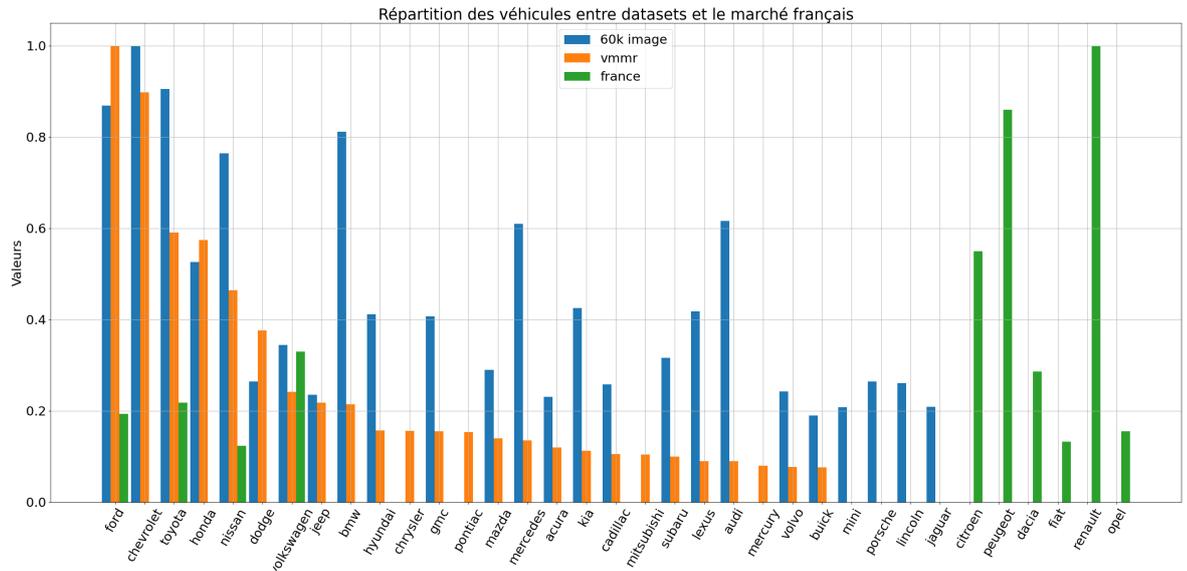


FIGURE 4 –

Nous voyons lors de l'observation du graphique ci-dessus la part des marques sur les différents datasets, ici le 60K image et le VMMR, ainsi que les données que nous avons récupérées et qui figure dans la section précédente. Nous observons que les véhicules présents sur le marché français et ceux présents dans les datasets analysés sont assez différents. Plusieurs grandes marques de l'automobile française sont absentes de ces datasets.

2.2.2 Espace de recherche

Dans l'idéal, on aimerait avoir un modèle d'apprentissage qui soit capable de reconnaître toute les voitures du monde en les classifiant correctement. Il est évidemment extrêmement compliqué d'arriver à ce résultat, tant le nombre de voitures sorties des usines depuis l'aube de l'automobile est important, en plus d'augmenter avec le temps. Chaque dataset couvre alors une partie de l'ensemble des classes qui pourrait exister dans le cadre d'un dataset ayant accès à tous les modèles de voitures sorties dans le monde, en tout temps.

On nomme espace de recherche les classes du dataset provenant du meilleur dataset "parfait". La taille de l'espace de recherche peut grandement varier en fonction de si l'on cible une région du monde ou non. On cherche alors le nombre de véhicules qui ont existé dans l'histoire (en prenant comme granularité finale le modèle de voiture). On peut alors trouver 12 423 véhicules différents dans l'histoire. Cependant, il est difficile de trouver une source fiable et complète comportant l'ensemble théorique total. Sur d'autres sources(<https://www.allcarindex.com/>, <https://www.carfolio.com/car-makes/>), on trouve qu'il a existé au cours du temps 14 559 marques, toutes n'ayant pas sorti de modèle, seulement des prototypes.

Nous pouvons alors identifier 2 métriques différentes pour évaluer nos modèles d'apprentissage. La première correspond au nombre de véhicules différents que l'on peut reconnaître. Néanmoins, il est discutable d'utiliser cette métrique car elle ne correspond pas

à la réalité. Si nous nous posons dans la rue en France et que nous observons les véhicules passant, il est plus intéressant de reconnaître une Peugeot 208, tant le modèle de voiture est répandu, là où la reconnaissance de Ferrari s’effectue sur un tout petit échantillon, quoi qu’en fonction de l’endroit du territoire français.

À l’issue de notre recherche sur les jeux de données existants, nous avons constaté qu’ils ne répondent pas pleinement à nos besoins. En effet, la majorité des datasets accessibles en ligne contiennent principalement des véhicules étrangers, souvent peu représentatifs du parc automobile que l’on trouve en France. De plus, certaines marques ou modèles de voitures courants en France y sont absents ou sous-représentés. Ainsi, afin de garantir la pertinence et la performance de notre modèle d’apprentissage dans un contexte local, nous avons décidé de constituer notre propre dataset, en collectant des images de véhicules typiquement présents sur le territoire français.

2.3 Scraping

Dans un premier temps, nous avons envisagé de constituer notre dataset en prenant nous-mêmes des photos de véhicules dans la rue. Bien que cette approche garantisse des images réalistes et adaptées à notre objectif, elle s’est rapidement révélée trop contraignante en termes de temps, de logistique et de volume d’images à collecter. Nous nous sommes donc tournés vers une solution plus efficace : le scraping d’images à partir de sources en ligne. Des sites comme leboncoin.fr, cars.com ou encore Google Images offrent un large éventail de photos de véhicules, ce qui nous permet d’automatiser partiellement la collecte et de cibler des modèles courants en France. Cette méthode nous a permis de construire un dataset riche, varié et mieux adapté aux besoins de notre projet, puisque nous avons pu diriger le scraping selon nos besoins.

Cependant, l’une des principales difficultés rencontrées lors de l’utilisation du scraping concernait les protocoles de sécurité mis en place par les sites web. De nombreux sites consultés disposaient de mesures de protection contre l’extraction automatisée de données, rendant la collecte parfois complexe. Conscients à la fois de l’existence de ces barrières techniques et des considérations légales associées, nous avons fait le choix, dans le cadre strictement pédagogique de ce projet d’étude, de récupérer un nombre limité d’annonces. Cette collecte a été réalisée de manière raisonnée, en veillant à ne pas surcharger les serveurs : nous avons ainsi contrôlé la quantité de données téléchargées ainsi que la fréquence des requêtes, dans le respect de l’usage normal des plateformes consultées.

2.3.1 Google Images

Google Images est une des premières possibilités auxquelles nous avons pensé. Théoriquement, on pourrait y trouver une quantité quasi infinie d’images, et ce pour n’importe quel modèle de n’importe quelle marque.

Cependant, ce site n’était pas réellement utilisable, puisque la précision des images que nous trouvions en faisant des recherches était très variable. Au bout d’une vingtaine d’images, elles n’étaient plus suffisamment cohérentes pour être utilisées, en donnant par exemple d’autres modèles et marques.

Nous n’avons finalement pas utilisé Google Images.

2.3.2 Cars.com

Cars.com est un site de revente de véhicules pour particuliers et professionnels. Ce site est avantageux par la quantité d’offres disponibles, ainsi que la diversité des modèles et

marques présentés. Seules quelques exceptions n'étaient pas disponibles en comparaison avec nos besoins, comme par exemple le modèle A1 de la marque Audi.

Un autre avantage de ce site pour notre projet est l'absence de vérification de type CAPTCHA. Ainsi, nous n'avons pas besoin de faire en sorte que notre code contourne les protocoles de sécurité anti-robots, puisque le site n'en possède pas.

2.3.3 Leboncoin

Leboncoin est un site de petites annonces françaises qui comportent une section voiture où l'on peut trouver des annonces de particuliers comme de professionnels. Il est donc devenu primordial de faire une méthode de récupération d'images sur ces annonces. Le principal problème qui nous est apparu rapidement a été le fait que le site comporte des programmes anti-scrap assez efficaces qui détectent les programmes et bannissent immédiatement les adresses IP.

La solution que nous avons trouvée à ce problème a été de créer un programme semi automatique. Lorsque l'on exécute le script, il va ouvrir une page leboncoin sur un navigateur Google Chrome qui sera écouté. Au lieu de récupérer automatiquement les liens de la page d'annonces et de tous les ouvrir d'un coup ou avec un délai, ce qui aurait attiré l'attention de l'anti-scrap, nous avons décidé d'ouvrir nous même les onglets pour avoir le côté humain. Une fois que l'on avait ouvert tous les onglets que l'on souhaitait, il nous a suffi d'appuyer sur le touche Q pour lancer la seconde partie du script, qui, grâce à l'écouteur, va regarder le code de la page et récupérer toutes les images qui sont intéressantes, soit celles concernant la voiture. À chaque image, le programme va l'enregistrer dans un dossier qui sera "Marque_Modele_Année" et fermer la fenêtre une fois tout sauvegarder.

2.4 Filtrage des résultats

Après le scraping sur différents sites, nous avons rencontré d'autres problèmes. Tous ces sites proposaient des photos avec des points de vue qui ne nous intéressaient pas. Nous avons beaucoup de photos de l'intérieur des véhicules, mais aussi des photos de roues, feux et autres parties précises de la voiture. Un autre problème, notamment sur Cars.com, était que la majorité des photos étaient téléchargées deux fois, créant une grande quantité de doublons. Le dernier problème était que, quelques images étaient complètement inadaptées à notre utilisation, certaines représentant par exemple des fiches techniques des véhicules.

Nous avons donc dû créer des scripts filtrants pour tenter de résoudre une majorité des problèmes, et de finir les derniers détails en supprimant manuellement les photos incorrectes que les scripts n'avaient pas détectées.

Prenons l'exemple de la classe Audi A3, juste après la récupération des images sur Cars.com.

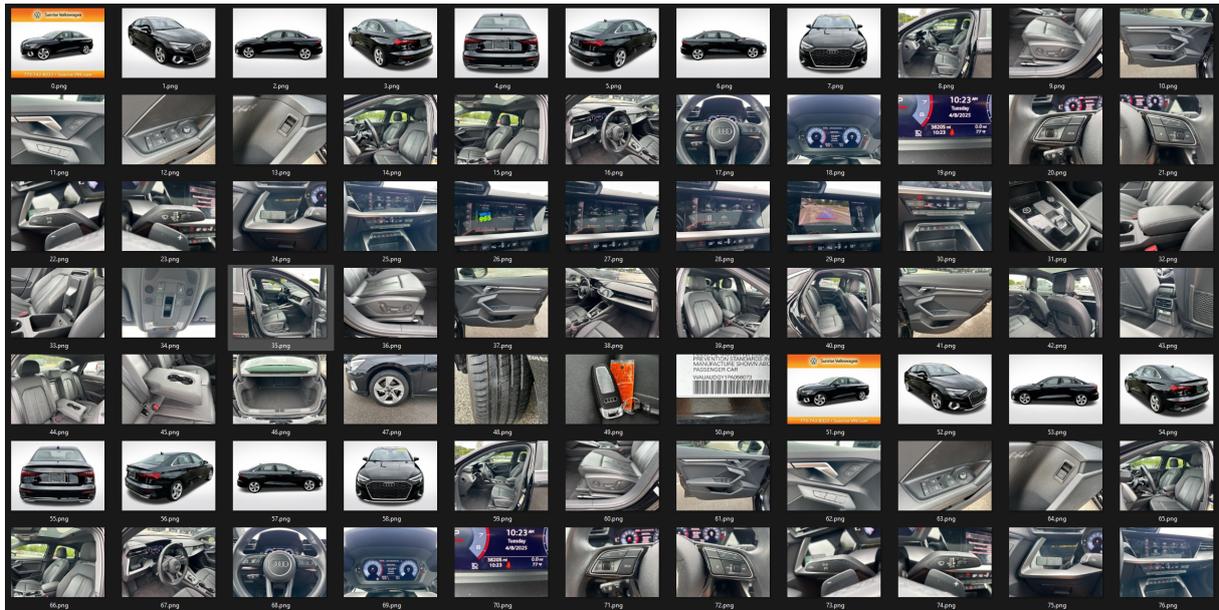


FIGURE 5 – Dossier d’images récupéré avant filtres

Nous pouvons y voir que les images sont doublées, par exemple la première image est retrouvée à la 5e ligne. Avec toutes ces images, nous arrivons à un total de 3028 images.

2.4.1 Anti-doublons

Afin de pallier ce premier problème, nous avons rédigé un script qui compare les images avec pour objectif de détecter les doublons parfaits.

Après ce premier algorithme de tri automatique, nous arrivons à un total de 1514 images, soit une division par deux du nombre d’images.

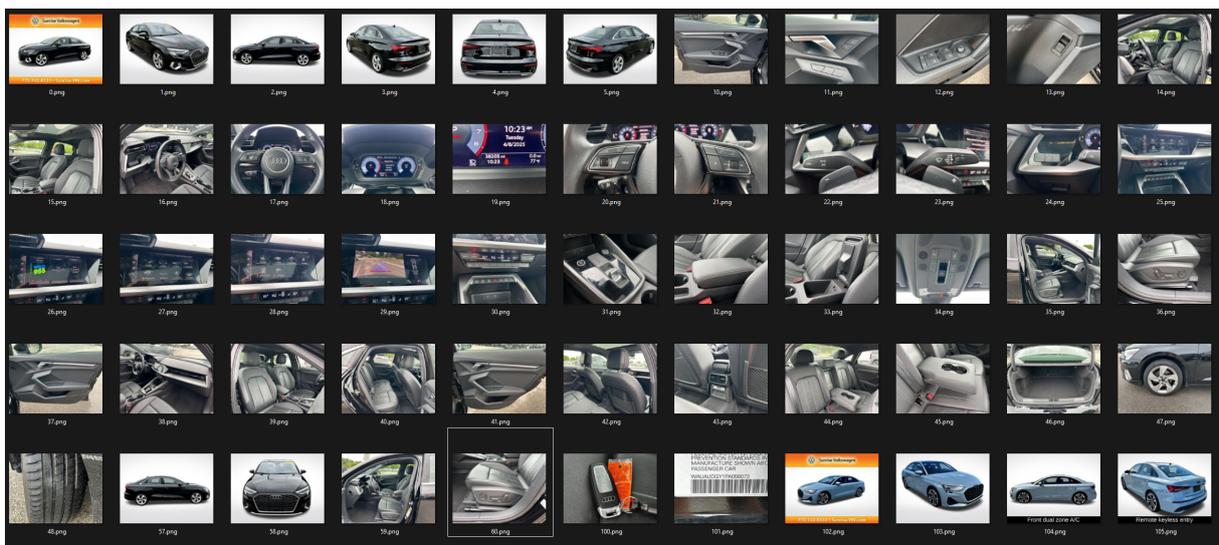


FIGURE 6 – Dossier après épuration par le premier filtre

Maintenant, le problème principal avec les images restantes est la très grande quantité d’images que l’on considère comme incorrectes. Cela inclut des images de portières, de tableaux de bord, d’autocollants...

2.4.2 Second filtre

Pour faire le second filtre automatique de ces images, nous avons utilisé le modèle CLIP d'OpenAI. Ce modèle nous a permis de séparer les images détectées comme d'intérieur de celles de l'extérieur des véhicules. Pour éviter de supprimer des images que nous aurions pu utiliser, nous avons vérifié que ce traitement n'avait pas détecté des images d'extérieur comme étant d'intérieur, situation que nous appellerons "faux négatif". Voici un extrait des images ayant été détectées comme d'intérieur.

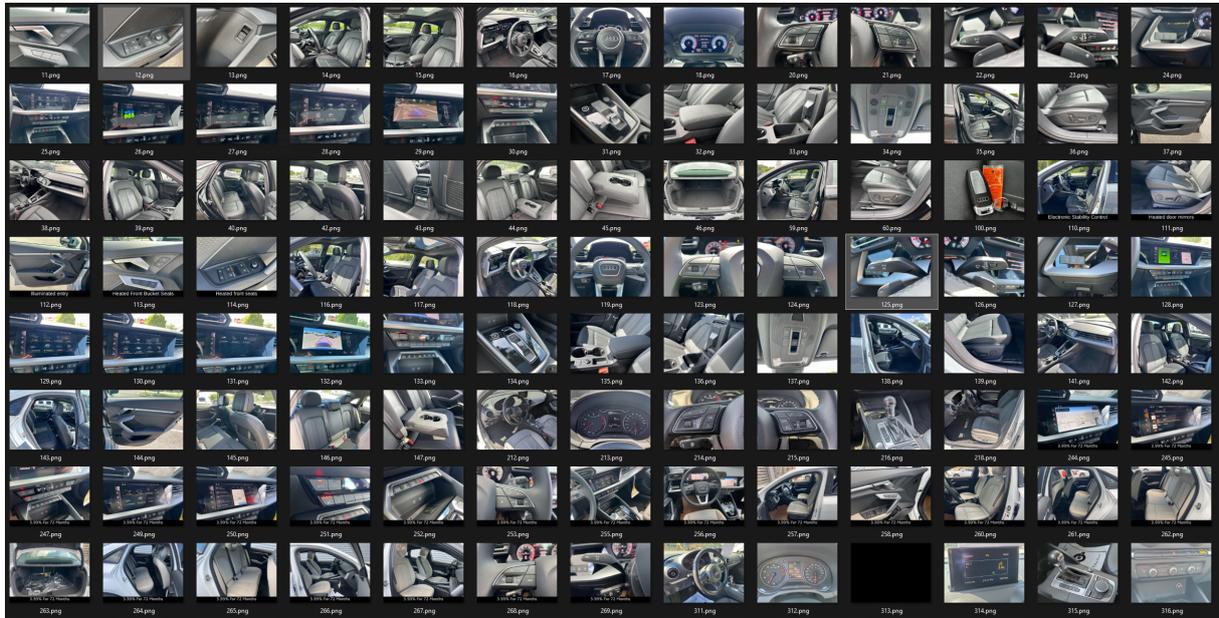


FIGURE 7 – Image d'intérieur de voiture

Aucune image extérieure de voiture n'est présente, la quantité de faux négatifs est nulle, ce qui est parfait dans notre situation. Cependant, afin de garantir que le nombre de faux négatifs soit nulle, des "faux positifs" (images d'intérieur détectées comme d'extérieur) sont encore présentes, comme visible ci-dessous.

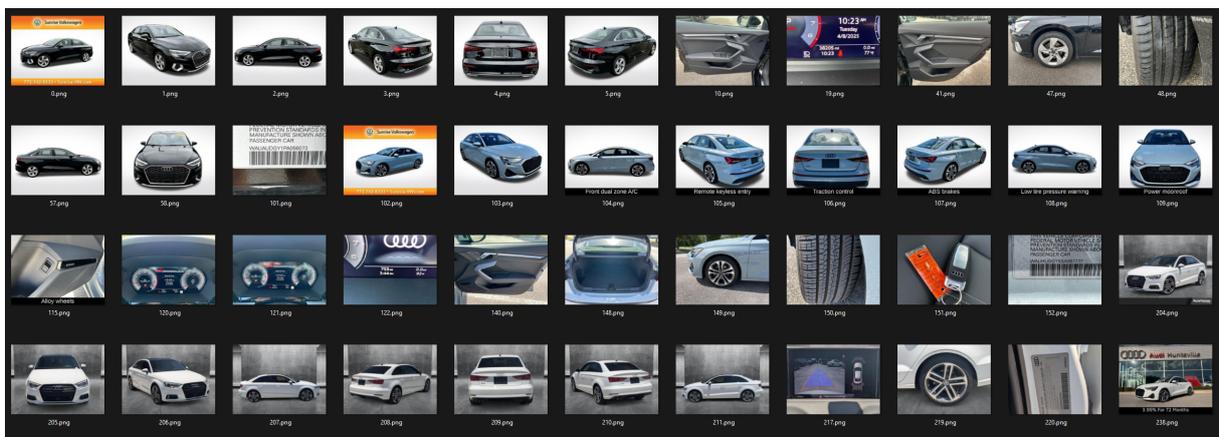


FIGURE 8 – Images détectées comme "extérieures" par le tri automatique

2.4.3 Méthode sans ré-apprentissage

Pour filtrer les résultats obtenus via le scraping, nous pouvons combiner plusieurs outils dans l'optique de faire diminuer autant que possible le nombre d'images à trier à la main, en supprimant les images qui ne permettent pas de reconnaître directement la voiture (images des roues, du tableau de bord...).

La première méthode consiste en l'utilisation d'un modèle YOLO pré-entraîné pour la détection d'objets notamment les voitures. Elle détecte si une voiture est présente ou pas sur une image donnée.

La seconde méthode utilise un modèle clip-vit-large-patch14, développé par OpenAI. Il s'agit d'un transformer qui cherche à maximiser la similarité entre un texte, photo d'intérieur et photo d'extérieur dans notre cas, avec des photos via un contrastive loss. Ce transformer a été développé dans l'optique de ne pas avoir à le ré-entraîner dans le cadre de tâches de vision par ordinateur.

Concrètement, un tri manuel a été effectué sur un sous ensemble de notre dataset pour avoir, ce que l'on considère comme étant la vérité. Nous avons ensuite appliqué trois filtres, un avec YOLO, un avec clip-vit-large-patch14 et la dernière avec la validation des deux, ce qui nous donne les résultats suivants :

- prediction_yolo : 87.79%
- prediction_vit : 78.16%
- prediction_both : 93.85%

Une double validation, certes plus coûteuse en temps, permet d'augmenter la précision de ce tri. Il est annoté que les photos de face et de l'arrière du véhicule sont souvent les plus mal classés en terme de faux positifs.

2.4.4 Tri manuel

Pour terminer, nous nous sommes répartis les images restantes afin de supprimer manuellement les images qui ne correspondaient pas à nos attentes après ces deux tris automatiques. Nous avons chacun supprimé entre 60.000 et 200.000 images.

2.5 Présentation de notre dataset

Notre dataset, que l'on a nommé "170K images of cars in France", contient précisément 170 277 images réparties entre 245 classes sur 17 marques au total. Notre objectif était d'atteindre 500 images au minimum par classe, en comparaison aux autres dataset qui contiennent en moyenne assez peu d'images par classe. Un élément de comparaison peut être le VMMR, qui possède en moyenne 31 images par modèle. Ainsi, le manque de données de certaines classes n'est pas un problème dans notre dataset. De plus, la granularité est différente. Nous nous sommes appuyés sur une granularité constructeur (make), modèle (model) et génération de voiture. Par exemple, les Citroën C4 sont séparés par les différentes générations sorties (I, II, III). Cela permet de ne pas avoir d'images qui se ressemblent entre différentes années du fait qu'il n'y ait pas eu de changement entre deux années consécutives, problème que l'on va retrouver plus tard dans la partie comparaison du modèle. Nous avons essayé de prendre un nombre d'années assez large pour les modèles mais nous nous sommes heurtés au fait que les voitures datant d'avant les années 2000 sont très peu présentes sur les sites que l'on a scrapés.

Notre Dataset <https://polifra.fr/dataset.php>

Dataset	Nombre d'images total	Nombre de classes	Images par classe
CompCar	136,726	1,716	79
VMMR	291,752	9,170	31
170k images of cars in France	170,277	245	659

TABLE 3 – Comparaison des datasets de voitures

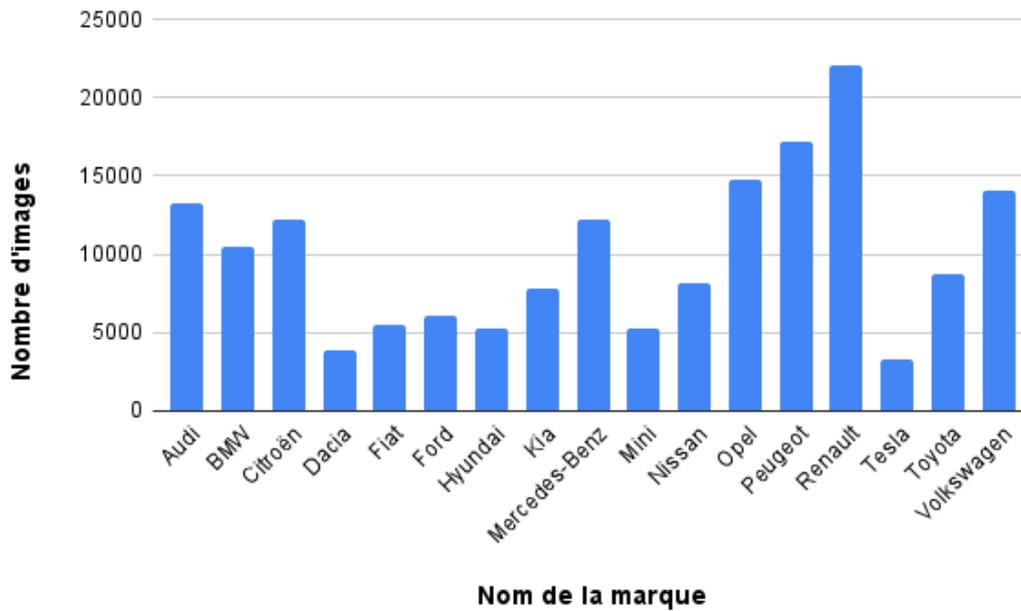


FIGURE 9 – Nombre d'images par marque

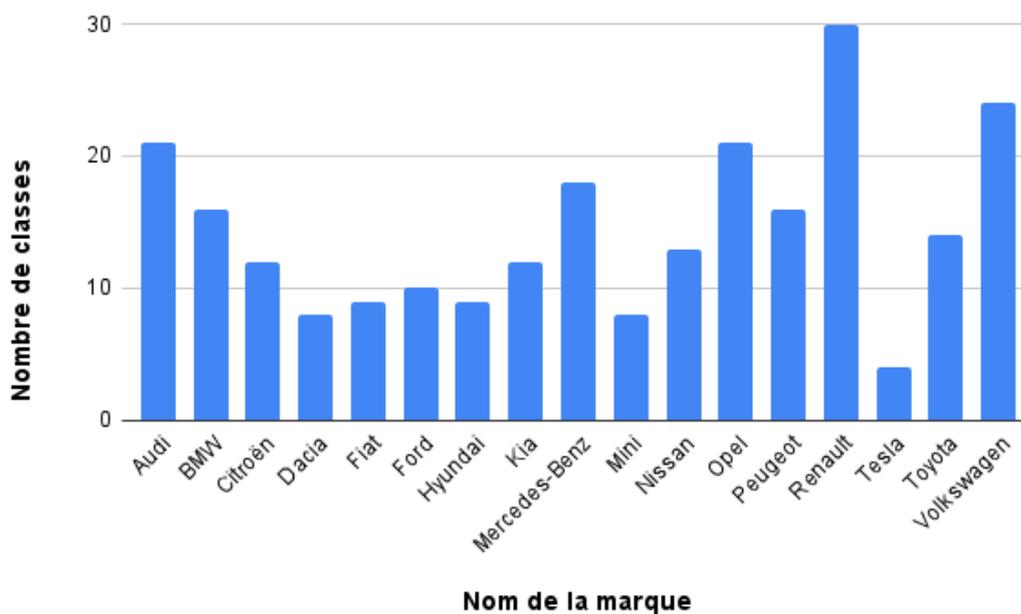


FIGURE 10 – Nombre de classes pour chaque marque

Grâce à notre dataset, les marques présentes en France telles que Renault, Peugeot ou Citroën sont représentées tout en ne discriminant pas les autres marques.

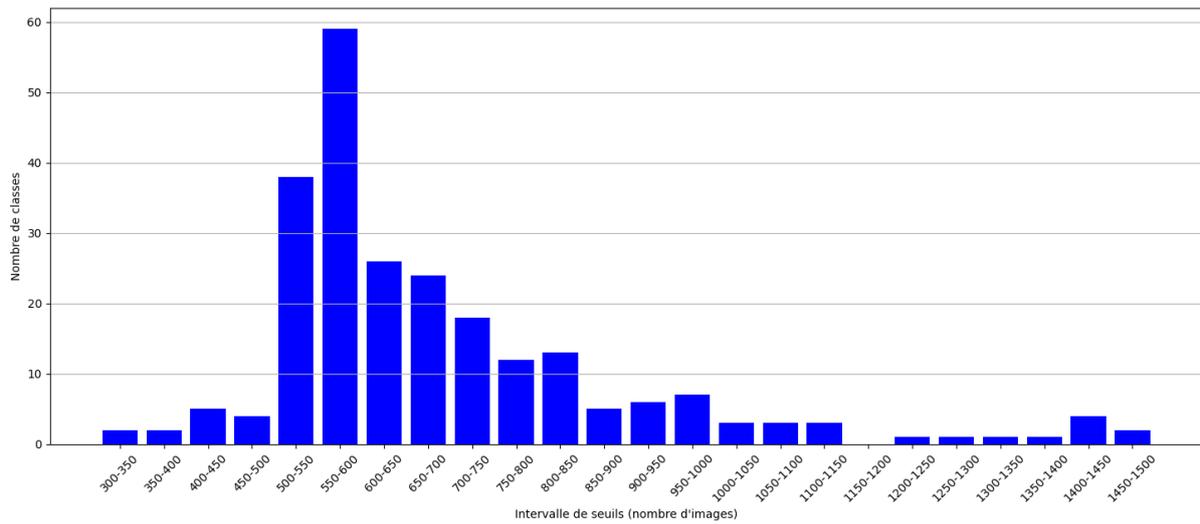


FIGURE 11 – Nombre de classes par intervalle seuils d'image

La grande majorité des classes contient entre 500 et 700 images, réduisant le risque de sur apprentissage de certaines classes par rapport à d'autres.

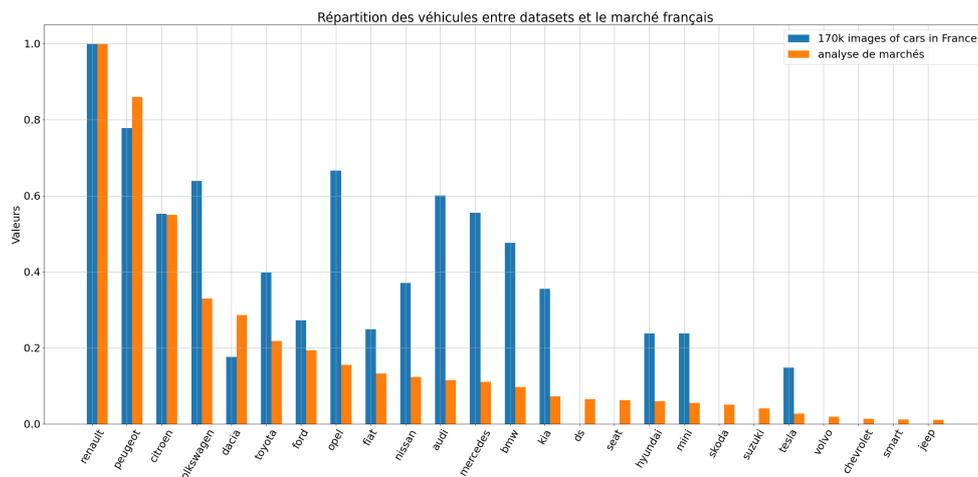


FIGURE 12 – Comparaison notre dataset et du marché français

La constitution de notre propre dataset a représenté un travail long, minutieux et parfois contraignant, notamment en raison des limites des jeux de données existants et des obstacles techniques liés à la collecte d'images en ligne. Néanmoins, cette étape s'est révélée essentielle pour répondre à notre objectif : concevoir un modèle d'identification de véhicules réellement adapté au contexte français. Nous sommes satisfaits d'avoir pu rassembler un dataset à la fois conséquent, diversifié et représentatif du parc automobile français, là où les bases de données classiques n'y répondait pas. Afin de valoriser ce travail, nous avons choisi de rendre ce dataset accessible, dans une logique de partage et de réutilisation. Il pourra ainsi servir à d'autres projets pédagogiques ou de recherche traitant de problématiques similaires.

3 Modèles

La mise en place d'un réseau de neurones pour reconnaître nos véhicules est la suite logique de la création et de l'évaluation d'un dataset.

3.1 Méthode d'entraînement

Après avoir réuni nos données, nous avons mis en place un réseau de neurones pour apprendre sur ces dernières. Parmi les choix d'architecture disponibles, nos yeux se sont tournés vers le ResNet, dans notre cas un ResNet50. Cette décision vient des nombreuses utilisations de ce type de modèle dans les différents projets dont nous nous sommes inspirés pendant notre veille au début du projet. Certes, l'utilisation d'architectures plus modernes, comme EfficientNet, peut accroître la performance du réseau, mais dans notre cas, il n'est pas nécessaire de le faire, tant les résultats nous semblent convaincants. Le choix du nombre de couches dans notre réseau est lié à des tests fait en amont, qui montre que le ResNet50 offre une meilleure précision que ses compères les ResNet18 et ResNet34 pour notre problème.

Le ResNet50 est un modèle de 50 layers qui sont répartie en 5 stage, la différence entre les 5 stages étant la taille des convolutions (longueur, largeur et nombre de channels de nos entrées). Nous avons seulement dégelé les stage 4, stage5 et évidemment la dernière couche fc, avec des learning rates de plus en plus grand en avançant vers la fin.

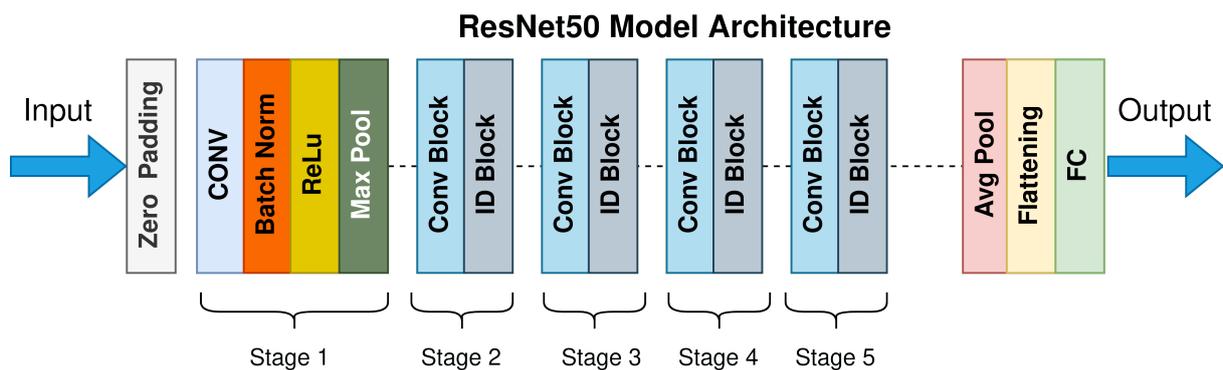


FIGURE 13 – Architecture d'un resnet 50

Le choix du backbone étant fait, passons à notre méthode d'entraînement. Nous cherchons à régler finement (fine-tuning) notre modèle pour lui faire apprendre à reconnaître des voitures. Pour l'entraînement, nous avons choisi de ne pas rétropropager les gradients dans les premières couches de notre réseau. En effet, ces premières couches sont responsables de l'extraction de features de bas niveau (bords, contours, ...). Comme les poids proviennent d'un entraînement sur ImageNet, il n'est pas nécessaire de les modifier. Nous avons seulement dégelé les Stage 4, Stage 5 et évidemment la dernière couche FC, avec des learning rates de plus en plus grands en avançant vers la fin. La séparation des données est la suivante : 80% pour l'entraînement, 10% pour la validation et 10% pour le test.

Le modèle a été entraîné sur l'ordinateur d'Erwann, avec une carte graphique NVIDIA GeForce RTX 4070 Ti Super, permettant d'entraîner totalement le modèle en 16h, sur 136 130 images d'entraînement.

3.2 Analyse de résultats et comparaison

Les résultats de l'entraînement sont présentés ci-dessous.

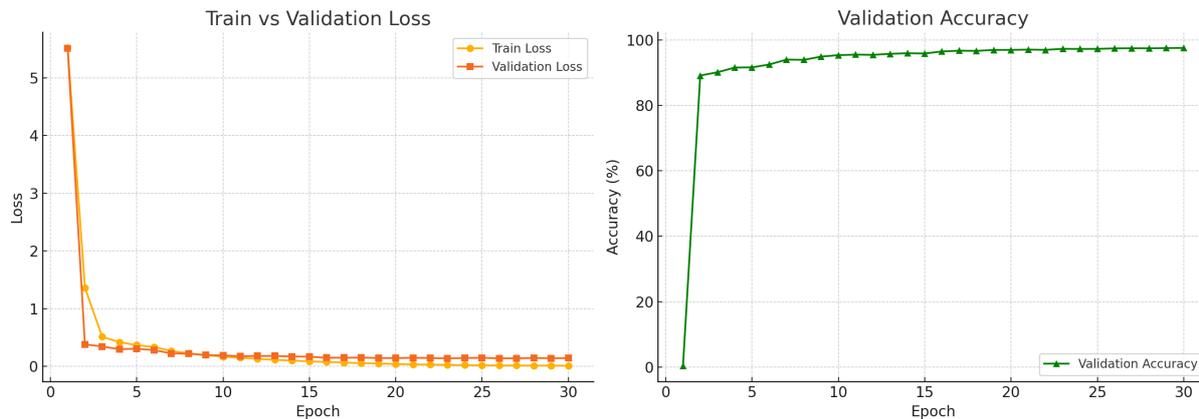


FIGURE 14 – Précision et perte sur la validation

Avec cette méthode d'entraînement, nous avons atteint une accuracy Top-1 de 97.46% et une accuracy Top-3 de 99.27%, avec un F1-score de 97%. Nous nous apercevons donc qu'il y a très peu d'erreur de la part du modèle. En regardant la matrice de confusion (15) qui présente les classes avec le plus d'erreurs de prédiction, nous pouvons remarquer que les erreurs viennent du même modèle de voiture mais dont la génération n'est pas la bonne, notamment sur les Renault Mégane I et les Renault Mégane II, Hyundai Kona I et Hyundai Kona II. Il existe très peu de cas où le réseau se trompe de modèle de voiture.

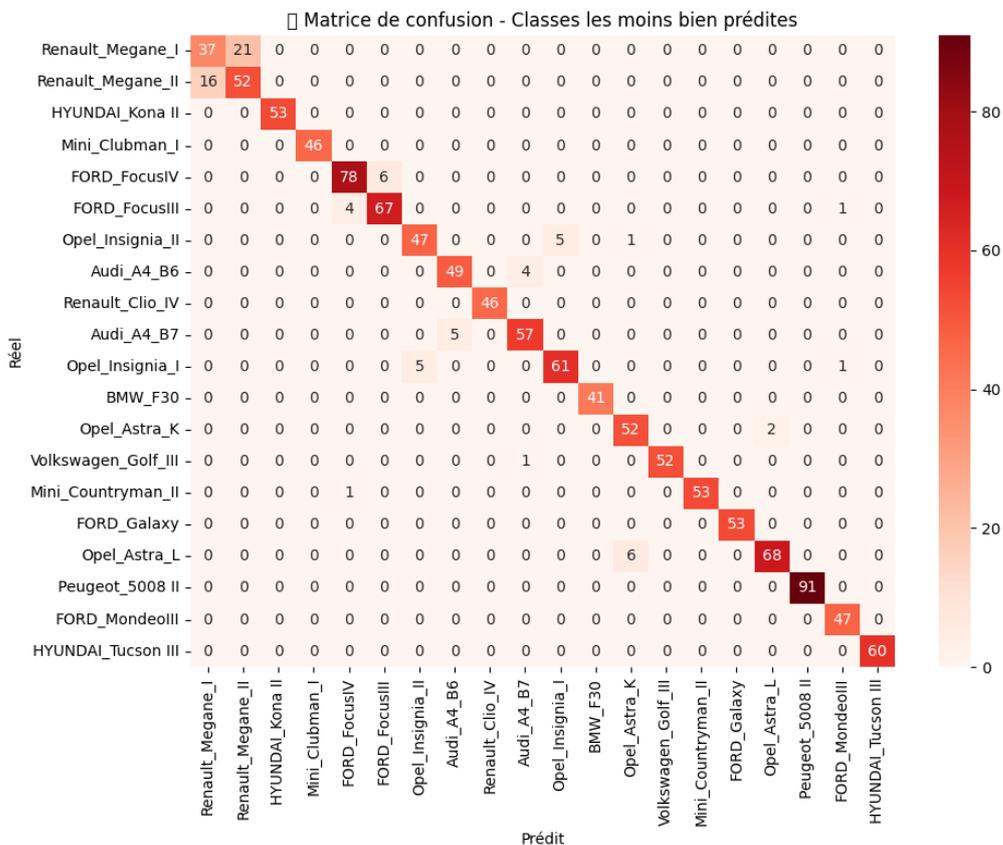


FIGURE 15 – Matrice des mauvaises prédictions

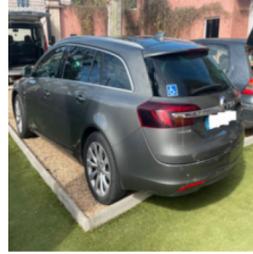
Vrai: HYUNDAI_Kona II | Prédit: HYUNDAI_Kona I



HYUNDAI_Kona I: 83.03%
HYUNDAI_Kona II: 16.94%
Volkswagen_ID.4: 0.01%
DACIA_Sandero: 0.00%
Volkswagen_Polo_VI: 0.00%

FIGURE 16 – HYUNDAI Kona II vs I

Vrai: Opel_Insignia II | Prédit: Opel_Insignia I



Opel_Insignia_I: 85.45%
Opel_Insignia_II: 14.55%
Kia_Ceed: 0.00%
Audi_A4_B9: 0.00%
Peugeot_508 II: 0.00%

FIGURE 17 – Opel Insignia II vs I

Vrai: Peugeot_308 II | Prédit: Peugeot_5008 II



Peugeot_5008 II: 87.59%
Peugeot_308 II: 6.65%
Peugeot_5008 I: 5.76%
Peugeot_2008 I: 0.00%
Peugeot_5008 I: 0.00%

FIGURE 18 – Peugeot 308 II vs 5008 II

Vrai: Renault_Megane I | Prédit: Renault_Megane II



Renault_Megane_II: 63.98%
Renault_Megane_I: 36.02%
Opel_Corsa_C: 0.00%
Opel_Insignia_I: 0.00%
FORD_MondeoII: 0.00%

FIGURE 19 – Renault Mégane I vs II

En comparaison, nous avons effectué un entraînement avec les mêmes conditions que sur notre précédent réseau, sur un sous ensemble de VMRR de 500 classes qui contiennent plus de 100 images par classes, afin d'avoir un peu de matière. Dans ce cas, nous avons atteint une accuracy Top-1 de 64.45% et une accuracy de Top-3 de 92.50%. On peut voir sur la matrice de confusion que les classes confondues sont souvent des modèles de voitures dont il n'existe pas de différence visuel entre 2 années de sorties d'usine, d'où la confusion du modèle. Par exemple, ils nous semblent compliqué d'apprendre à un modèle à faire la différence entre 2 classes lorsque ces dernières ne sont pas différenciables via un coup d'œil, même d'un humain expérimenté.

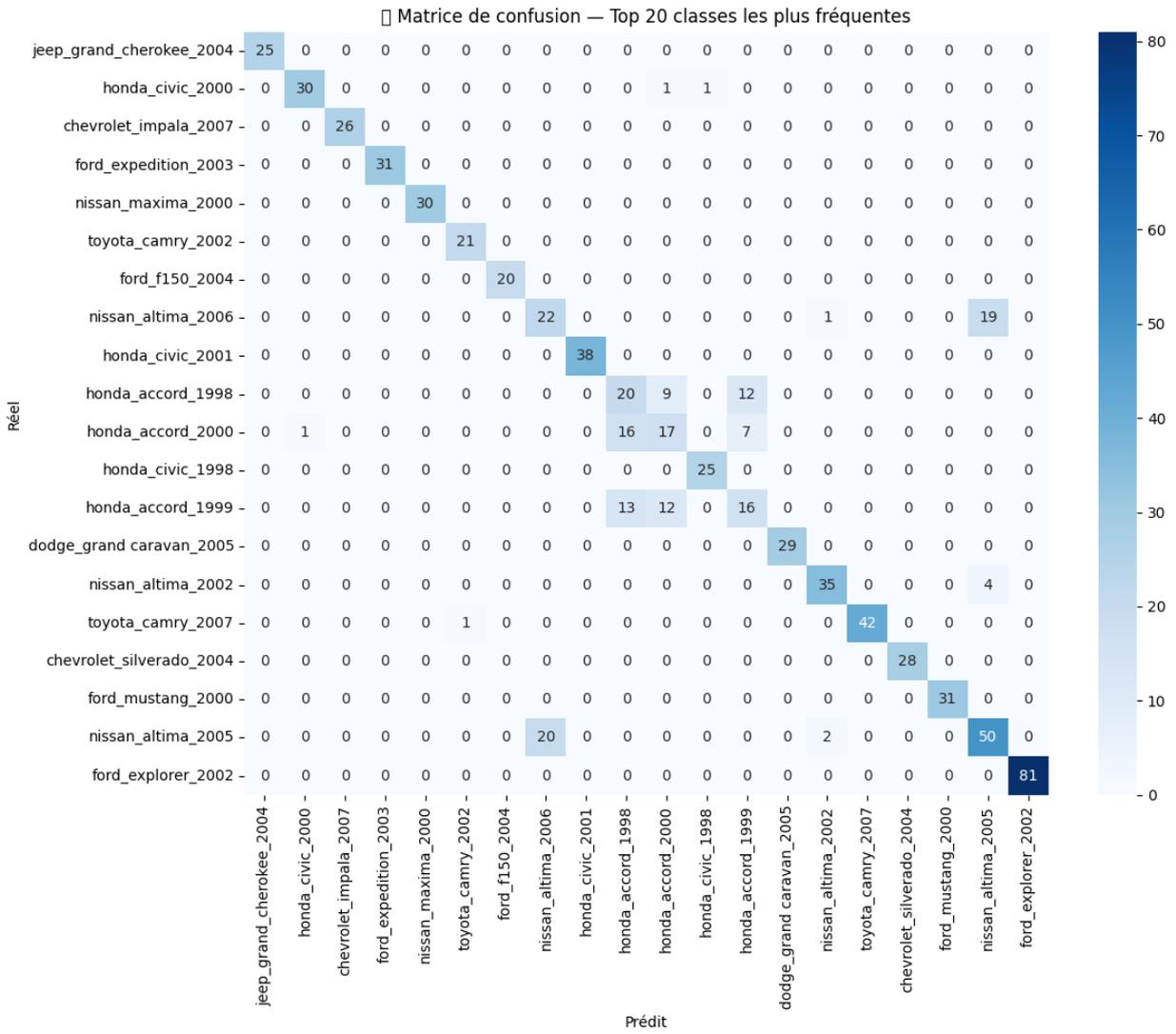


FIGURE 20 – Matrice de confusion du dataset VMMR

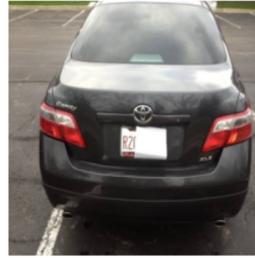
Vrai: nissan_altima_2004 | Prédit: nissan_altima_2002



nissan_altima_2002: 71.15%
nissan_altima_2004: 28.71%
nissan_altima_2005: 0.12%
nissan_altima_2006: 0.01%
toyota_corolla_2007: 0.00%

FIGURE 21 – Nissan Altima 2004 vs 2002

Vrai: toyota_camry_2007 | Prédit: toyota_camry_2008



toyota_camry_2008: 99.66%
toyota_camry_2007: 0.34%
nissan_murano_2005: 0.00%
toyota_camry_1997: 0.00%
toyota_camry_2011: 0.00%

FIGURE 22 – Toyota Camry 2007 vs 2008

Vrai: honda_accord_2000 | Prédit: honda_accord_1998



honda_accord_1998: 87.61%
honda_accord_2000: 6.98%
honda_accord_1999: 5.26%
honda_accord_2001: 0.14%
honda_civic_1994: 0.00%

FIGURE 23 – Honda Accord 2000 vs 1998

Vrai: infiniti_g35_2005 | Prédit: infiniti_g35_2006



infiniti_g35_2006: 96.06%
infiniti_g35_2005: 3.94%
infiniti_g35_coupe_2004: 0.00%
mini_cooper_2005: 0.00%
chevrolet_cobalt_2009: 0.00%

FIGURE 24 – Infiniti G35 2005 vs 2006

Nous remarquons ici que la granularité choisie par le dataset a un très fort impact sur l'accuracy du modèle. Cependant, il existe entre le VMMR et 170k images of cars in France une relation possible. Une partie des images de notre dataset contient toujours l'année de sortie du véhicule, ce qui permet de redéfinir la granularité au sein même de notre dataset.

L'apprentissage du modèle s'est très bien passé avec des résultats plus que concluants, les réseaux de neurones récents n'ont aucun mal à différencier les différentes générations de modèle de voiture entre elles. Le fait d'avoir atteint une accuracy en Top-1 à 97,46%, qui monte à plus de 99% en Top-3, nous rassure quand à la fiabilité du modèle dans son application de la vie de tous les jours.

4 Travail effectué en plus

Entre la création de dataset et l'entraînement de modèle, une partie sur la reconnaissance de véhicules via la lecture de plaque d'immatriculation a été explorée. La plaque permet d'identifier le type de véhicule et le modèle par l'intermédiaire de fichier comme le SIV (Système d'Immatriculation des Véhicules), qui est un dictionnaire comprenant toutes les informations sur les véhicules.

4.1 Plaque d'immatriculation

Une autre méthode d'identification du modèle de véhicule consiste à reconnaître la plaque d'immatriculation sur un véhicule et en extraire les numéros présents dessus. Comme nous travaillons principalement sur le marché français, nous avons décidé de n'extraire que les plaques qui correspondent aux plaques françaises, soit le format XX-111-XX pour les plaques après 2009 ou 111 XX 11 pour celle avant 2009.

4.1.1 Traitement de l'image

Afin de faire ressortir la plaque d'immatriculation sur une image, les premières fonctions que nous avons appliquées à l'image consiste à la passer en niveau de gris, puis à appliquer un flou gaussien pour réduire le bruit. Cela permet de pouvoir différencier, sur les véhicules de couleurs, la voiture de la plaque. Ensuite, nous détectons les bords et les contours dans l'image. Les contours sont ensuite triés par aire décroissante et les 10 plus grandes aires sont gardées. Pour chaque contour, nous l'approximons sous forme de rectangle, puis nous extrayons le texte trouvé grâce à la librairie Tesseract de Python, afin de repérer le texte de la plaque d'immatriculation.

4.1.2 Reconnaissance du texte et post-traitement

Les plaques d'immatriculation potentielles sont ensuite passées dans une fonction de normalisation afin de remplacer les U par des V, les I par des L ou des 1, de mettre des tirets à la place des espaces et de retirer tout caractère étranger à la plaque. Ensuite, nous allons tester les possibles plaques afin de trouver une correspondance. Étant donné que la lecture peut oublier les tirets, il a aussi été pris en compte les plaques qui sont juste séparées par des espaces ou qui n'ont pas d'espace.



FIGURE 25 – Plaque détectée seulement par reconnaissance de forme

4.1.3 Méthode dite "du bourrin"

Dans le cas où la plaque ne peut pas être lue, une méthode dite "du bourrin" est utilisée afin d'essayer de lire entièrement l'image et, à partir du texte qui en ressort, d'essayer

d'en extraire un motif correspondant à une plaque. Tesseract va alors essayer de lire du texte sur l'image entière et plus seulement sur une zone définie.



FIGURE 26 – Exemple de plaque non reconnue par la méthode des contours

Il est évidemment possible de combiner les deux méthodes afin d'améliorer la précision de l'algorithme comme testé ci dessous.



FIGURE 27 – Détection de forme incorrecte, utilisation de la méthode dite "du bourrin"

4.2 Détection de texte avec EasyOCR

En complément des méthodes précédentes basées sur Tesseract, nous avons expérimenté l'utilisation de la bibliothèque EasyOCR, conçue pour offrir une reconnaissance sur des images naturelles contenant du texte. Cette méthode permet de détecter automatiquement les zones textuelles dans une image sans étape préalable de segmentation ou de traitement morphologique.

Pour cela, un modèle pré-entraîné pour la langue française est utilisé afin d'optimiser la reconnaissance des caractères présents sur les plaques d'immatriculation françaises. Contrairement à la méthode par contours, cette approche ne repose pas sur une détection préalable de la plaque : l'intégralité de l'image est analysée, et les zones où un texte est détecté avec une probabilité suffisante (supérieure à 50 %) sont conservées.

Quand un texte est détecté comme pouvant être la plaque d'immatriculation, on rajoute sa probabilité de confiance, permettant de pouvoir estimer si on garde ou non le résultat.



Texte	Confiance
114	0.77
FF-913-SD	1.00

FIGURE 28 – Utilisation de l’EasyOCR pour détecter la plaque d’immatriculation

Il se peut que le modèle reconnaissance plusieurs types de textes et dans ce cas là, il va chercher à prendre celui qui a le plus haut taux de confiance comme étant la plaque d’immatriculation.

4.2.1 Résultats

Les premiers résultats sont plutôt encourageants sur les véhicules. Grâce à ces méthodes, certaines plaques arrivent à être lues et nous pouvons en tirer quelque chose à exploiter derrière.

En multipliant les méthodes, on peut arriver à augmenter le pourcentage de chance de récupérer une plaque, mais il est très difficile d’atteindre un très haut taux de réussite, tant les angles ou la distance ne permettent pas d’avoir une image correcte à traiter. Il se peut aussi que la détection retourne un mauvais rectangle blanc (Figure 27) et donc que ça ne prenne pas en compte la plaque.

D’autres méthodes pour détecter les plaques d’immatriculations sur une voiture existent évidemment, nous n’avons exploré qu’une partie des possibilité. Des alternatives à Tesseract et EasyOCR comme notamment Mistral OCR peuvent permettre d’offrir d’autres moyens de récupérer le texte.

5 Conclusion

Ce projet nous a permis d'explorer de manière concrète l'application de l'intelligence artificielle sur l'identification automatique de véhicules et toutes les problématiques qui en découlent. Celle des données est la plus importante, car sans une base de données fiable, solide et adaptée à nos besoins, le modèle entraîné sera au mieux médiocre. Nous proposons donc un nouveau dataset, concentré sur le marché français, qui comble les inconvénients des autres datasets analysés. Le manque d'images par classe, notamment lié à l'espace de recherche ample et à une granularité plus fine, pose la problématique de la pertinence du dataset par rapport à sa taille. Nous avons donc préféré un dataset concis en classe mais fourni en images, nous permettant d'assurer une fiabilité du modèle très importante. Nous laissons à de futures recherches, la création d'un dataset encore plus ample avec une meilleure couverture de l'espace de recherche.

6 Sources

- Fiches-auto : <https://www.fiches-auto.fr/>
- Stanford Cars Dataset : <https://www.kaggle.com/datasets/jessicali9530/stanford-cars-dataset>
- Compcars Dataset : <https://www.kaggle.com/datasets/renancostaalencar/compcars>
- VMMR Cars Dataset : <https://www.kaggle.com/datasets/abhishektyagi001/vehicle-make-model-recognition-dataset-vmmrdb>
- 60,000 Images of Cars : <https://www.kaggle.com/datasets/prondeau/the-car-connection-picture-dataset>
- Cars makes database : <https://www.allcarindex.com/>
- Cars model database : <https://www.carfolio.com/car-makes/>
- Notre Dataset : <https://polifra.fr/dataset.php>