

UNIVERSITÉ DU QUÉBEC

MÉMOIRE PRÉSENTÉ À
L'UNIVERSITÉ DU QUÉBEC À TROIS-RIVIÈRES

COMME EXIGENCE PARTIELLE
DE LA MAÎTRISE EN MATHÉMATIQUES ET INFORMATIQUE
APPLIQUÉES

PAR
MAMADOU MOCTAR NDIAYE

IDENTIFICATION ET ÉVALUATION AUTOMATIQUES DES DOMMAGES SUR
VÉHICULES ACCIDENTÉS VIA LES
DES ALGORITHMES D'APPRENTISSAGE PROFOND.

Janvier 2024

Université du Québec à Trois-Rivières

Service de la bibliothèque

Avertissement

L'auteur de ce mémoire, de cette thèse ou de cet essai a autorisé l'Université du Québec à Trois-Rivières à diffuser, à des fins non lucratives, une copie de son mémoire, de sa thèse ou de son essai.

Cette diffusion n'entraîne pas une renonciation de la part de l'auteur à ses droits de propriété intellectuelle, incluant le droit d'auteur, sur ce mémoire, cette thèse ou cet essai. Notamment, la reproduction ou la publication de la totalité ou d'une partie importante de ce mémoire, de cette thèse et de son essai requiert son autorisation.

CE MÉMOIRE A ÉTÉ ÉVALUÉ PAR :

M. Usef Faghihi, directeur de recherche, Professeur au département de Mathématiques et Informatique

Université du Québec à Trois-Rivières

M. Fadel Touré, codirecteur de recherche, Professeur au département de Mathématiques et Informatique

Université du Québec à Trois-Rivières

M. François Meunier, Professeur au département de Mathématiques et Informatique.

Université du Québec à Trois-Rivières

RÉSUMÉ

La croissance significative du nombre de véhicules a entraîné une expansion majeure du marché de l'assurance automobile. Toutefois, cette expansion s'accompagne de défis liés à la gestion des réclamations, aux évaluations des sinistres et aux coûts associés. Dans ce contexte, l'automatisation joue un rôle de plus en plus important dans de nombreux domaines connexes, offrant des avantages en termes de gain de temps et de fiabilité. Dans l'optique de pousser plus loin cette automatisation, la société PROGI, spécialisée dans les technologies de l'information et fournissant des solutions informatiques aux compagnies d'assurances, souhaite développer une technologie permettant d'estimer automatiquement les dommages subis par les véhicules après un accident. Actuellement, le processus d'évaluation consiste entre autres, à prendre des photos ou des vidéos du véhicule accidenté, puis, de les faire évaluer par agent (expert sinistre) qui estime les dommages en termes de coût de réparation afin de le classer en déterminant s'il s'agit de véhicule réparable ou d'une perte totale. À notre connaissance, il n'existe à ce jour, aucune application capable d'automatiser tout le processus à partir de la prise de photo.

Pour répondre à ce besoin, nous nous sommes procurés un vaste jeu de données composé de nombreuses images de véhicules accidentés et non-accidentés. Nous avons utilisé le modèle de réseaux de neurones profonds ResNet pour classer les voitures par marques dans un premier temps. Par la suite, nous avons utilisé les modèles de réseau de neurones de segmentations et de détections d'images tels que U-Net ou YOLO, dans le but d'identifier les différentes parties des véhicules et de circonscrire celles qui sont endommagées, grâce à une détection plus précise dans les images. L'utilisation combinée de ces techniques permettra d'adopter une approche méthodologique dans cette étude, avec pour objectif de développer une technologie automatisée et précise pour classer les marques de véhicules, segmenter les différentes pièces externes de la voiture et détecter les dommages de véhicules après une collision.

Mots clés : Réseaux de neurones, Apprentissage profond, Détection de dommages sur véhicule, Estimation de dommages, Réseaux convolutifs.

ABSTRACT

The significant growth in the number of vehicles in recent years has led to a major expansion of the motor insurance market. However, this expansion is accompanied by challenges relating to claims management, claims assessment and associated costs. Against this backdrop, automation is playing an increasingly important role in many levels, offering time- and cost-saving benefits. To bring automatization process further, PROGI, an IT company providing solutions to insurance companies, wants to rely on AI to develop a new approach that's detect and estimates vehicle damages after an accident. Currently, the auto damage assessment process involves picking photos or videos showing vehicle damage. Then, an insurance employee expert attempts to determine the severity of the impact to make appropriate decisions regarding total vehicle loss, necessary repairs or a more accurate cost estimate. However, to date, there is no application or system capable of automating the whole process.

To address this need, we have built a large dataset consisting of numerous vehicle images. Initially, we used the ResNet deep neural network model to classify cars by brand. Then, we used neural network models for image segmentation and detection such as U-Net or YOLO, with the aim of identifying the different parts of the vehicles and circumscribing those that are damaged, thanks to precise image detection. The combined use of these techniques will enable a methodological approach to be adopted in this study, with the aim of developing an automated and accurate technology for classifying vehicle makes, segmenting different external car parts and detecting vehicle damage after a collision.

REMERCIEMENTS

Chers membres de ma famille, amis chers, encadreurs les professeurs Usef FAGIHI et Fadel TOURE,

Tout d'abord, je remercie DIEU de m'avoir accordé la santé et le bien-être nécessaires pour réaliser ce travail.

Aujourd'hui, alors que j'achève ce chapitre important de ma vie, je tiens à exprimer ma profonde gratitude envers chacun d'entre vous. Mon parcours pour réaliser cette mémoire a été parsemé de défis, de découvertes et d'apprentissages, et vous avez tous joué un rôle essentiel dans cette aventure.

Je souhaite exprimer mes plus sincères remerciements à toute ma famille : mon père Massamba NDIAYE, ma mère Sokhna Royakha FALL, mes frères, mes sœurs ainsi qu'à mes oncles et tantes. Votre soutien inconditionnel, votre encouragement constant et vos mots de réconfort ont été ma source d'inspiration tout au long de ce voyage. Votre patience et votre croyance en moi m'ont aidé à surmonter les moments de doute et à persévérer jusqu'au bout. Chaque étape franchie a été rendue plus significative grâce à votre présence à mes côtés.

Mes chers amis, vous êtes la lumière qui a illuminé mes jours les plus sombres. Votre enthousiasme, vos rires partagés et vos encouragements joyeux ont apporté une touche d'optimisme à ce processus parfois ardu. Votre amitié m'a rappelé l'importance de l'équilibre entre le travail et les moments de détente, et je suis reconnaissant(e) d'avoir pu compter sur vous pour me soutenir.

Fadel et Usef, mes encadreurs, je ne saurais exprimer assez à quel point votre guidance a été précieuse. Votre expertise, vos conseils éclairés et votre patience ont été les fondations sur lesquelles j'ai construit cette mémoire. Vos critiques constructives m'ont poussé(e) à

repousser mes limites et à viser l'excellence. Votre mentorat a été un véritable privilège, et je suis honoré(e) d'avoir eu la chance de travailler sous votre direction.

Enfin, à tous, je veux dire que cette réalisation n'est pas seulement la mienne, mais aussi la vôtre. Votre présence dans ma vie a apporté une richesse et une profondeur inestimables à ce projet. Que vous ayez été là pour célébrer mes succès ou pour m'épauler dans les moments difficiles, vous avez fait une différence significative.

Avec une profonde reconnaissance,

TABLE DES MATIÈRES

RÉSUMÉ	iii
ABSTRACT	iv
REMERCIEMENTS	v
TABLE DES MATIÈRES	vii
LISTE DES FIGURES	x
LISTE DES TABLEAUX	xii
LISTES DES ABRÉVIATIONS	xiii
Chapitre 1 : Introduction	1
1.1 Introduction.....	1
1.2 Contexte et problématique	2
1.3 Objectifs	2
1.4 Organisation du mémoire.....	3
Chapitre 2 : Revue de littérature	2
2.1 Introduction.....	2
2.2 La reconnaissance d’image	2
2.3 L’identification et la localisation d’objets	6
2.3.1 L’identification des objets.....	6
2.3.2 Localisation des objets	7
2.5 Conclusion	12
Chapitre 3 : Méthodologie et analyses des données	5
3.1 Méthodologie	5
3.1.1 Approche par la classification.....	5

3.1.2	Approche par la segmentation	14
3.1.3	Approche par la détection	15
3.2	Présentation des modèles d'apprentissages automatiques utilisés.....	16
3.2.1	Les réseaux de neurones convolutifs	18
3.2.2	Le Modèle ResNet	20
3.2.2.1	Architecture.....	22
3.2.3	Le Modèle Unet	26
3.2.4	Yolo version 8.....	28
3.3	Analyse de données.....	32
3.3.1	Traitement des données pour la reconnaissance des marques de voitures.....	33
3.3.2	Préparation des données pour la segmentation des pièces externes de la voiture	38
3.3.3	Distribution des données pour la détection des dommages	42
3.4	Autres Outils.	42
3.4.1	Google Colaboratory.....	43
3.4.2	Python	43
3.4.3	Google drive.....	44
3.4.4	Apeer.....	44
3.5	Conclusion	44
Chapitre 4 : Résultats et discussions		14
4.1	Introduction.....	14
4.2	Reconnaissance des marques de véhicules.	14
4.2.1	Performance des modèles	14
4.2.2	Résultats des prédictions.....	48
4.3	Résultats de la segmentation d'image pour les pièces externes du véhicule.	54
4.3.1	Segmentation avec le modèle U-net	54
4.3.2	Segmentation avec le modèle Unet-ResNet.....	57
4.4	Détection des pièces externes avec le modèle Yolov8	59
4.4.1.	Performance du modèle	59

4.4.2. Résultats de la détection.....	61
4.5 Détection des dommages avec le modèle Yolov8	62
4.5.1 Performances du modèle.....	63
4.5.2. Résultats de la détection.....	64
4.6 Discussion des résultats	69
Chapitre 5 : Conclusion finale et perspectives	46
Bibliographies.....	72

LISTE DES FIGURES

Figure 1 : Détection d'objet.....	7
Figure 2 : Degmentation d'image avec.....	8
Figure 3 : Différence entre segmentation par contour et segmentation par région.....	14
Figure 4 : Segmentation sémantique.....	15
Figure 5 : Organigramme du processus de la détection des dommages	16
Figure 6 : Exemple de réseau de neurones convolutifs.....	19
Figure 7 : Exemple de " padding"	19
Figure 8 : Exemple de technique de max pooling.....	20
Figure 9 : Connexion de sauts de ResNet.	23
Figure 10 : Vue détaillée du modèle ResNet couche par couche. Source	24
Figure 11 : Architecture de ResNet 34 à partir du papier original. Source.....	25
Figure 12 : Architecture du modèle U-net	28
Figure 13: Implémentation de YOLOv8 sur Google Colab.....	31
Figure 14 : Graphique du nombre d'images en fonction de la classe.....	36
Figure 15 : Pourcentage des différentes couleurs dans notre base de données.....	37
Figure 16 : Pourcentage de représentation de chaque couleur dans nos données de segmentation	40
Figure 17 : Processus de segmentation d'images	41
Figure 18 : Évolution de fonction perte	46
Figure 19 : Évolution de l'exactitude, apprentissage.....	47
Figure 20 : Matrice de confusion	47
Figure 21 : Prédiction sur une Honda côté avant.....	48
Figure 22 : Prédiction sur une Honda sur le côté arrière	49
Figure 23 : Prédiction sur la BMW	50
Figure 24 : Mauvaise prédiction sur une Honda.....	51
Figure 25 : Prédiction sur une Hyundai Elantra accidentée.....	51
Figure 26 : Ressemblance entre deux véhicules de marques différentes.....	52
Figure 27 : Prédiction sur quatre angles du véhicule.....	53
Figure 28 : Courbe de perte du modèle Unet.....	55

Figure 29 : Courbe de précision du modèle Unet	55
Figure 30 : première prédiction avec le modèle Unet.....	56
Figure 31 : Deuxième prédiction avec le modèle Unet.....	56
Figure 32 : Courbe de perte du modèle Unet-ResNet.....	57
Figure 33 : Courbe de précision du modèle Unet-ResNet	58
Figure 34 : Prédiction avec le modèle Unet-ResNet.....	58
Figure 35 : Courbes de précision et de perte.....	59
Figure 36 : Matrice de confusion	60
Figure 37 : Première prédiction du modèle.....	61
Figure 38 : Deuxième prédiction du modèle.....	62
Figure 39 : Résultats des entraînements pour la détection des dommages	64
Figure 40 : courbe de précision mAP du modèle.....	64
Figure 41 : Prédiction sur des véhicules endommagés	65
Figure 42 : Détection de dommage sur une voiture à partir de plusieurs angles	66
Figure 43 : Résultat du modèle après prédiction	67
Figure 44 : Détection des pièces externes de la voiture.....	68

LISTE DES TABLEAUX

Tableau 1 : Les modèles gagnant lors du concours ImageNet Large Scale Visual Recognition Challenge source	20
Tableau 2 : Tâche prise en charge par yoloV8.....	29
Tableau 3 : Performance des modèles.	30
Tableau 4 : Tableau comparatif des voitures les plus vendues ces trois dernières années au Canada.....	34
Tableau 5: Nombre d'images et couleurs dominant sur chaque classe	35
Tableau 6 : Statistiques descriptives pour la détection des marques	37
Tableau 7 : Nombre d'images de chaque couleur pour la segmentation.....	39
Tableau 8 : Classes pour la segmentation	39
Tableau 9 : Statistique descriptive pour la segmentation.....	40
Tableau 10 : Performance de la classification des marques de véhicules.....	46

LISTES DES ABRÉVIATIONS

VGG: **V**isual **G**eometry **G**roup

YOLO: **Y**ou **O**nly **L**ook **O**nce

ResNet: **R**esidual **N**etwork

DL: **D**eep **L**earning

TCAC : **T**aux de **C**roissance **A**nnuelle **C**omposée

CNN: **C**onvolutional **N**eural **N**etwork

FCN: **F**ully **C**onvolutional **N**etwork

CE: **C**ross **E**ntropie

CVAT: **C**omputer **V**ision **A**notation **T**ools

R-CNN: **R**egion-based **C**onvolutional **N**eural **N**etwork

FL: **F**ocal **L**oss

RELU: **R**ectified **L**inear **U**nit

ICCV: **I**nternational **C**onference of **C**omputer **V**ision

Chapitre 1 : Introduction

1.1 Introduction

Les développements récents de l'intelligence artificielle dopés par l'accroissement des capacités de calcul et leur accessibilité ont permis des avancées majeures dans le domaine de la vision par ordinateur, le traitement du langage naturel ainsi que d'autres avancées des technologies (Lu, Li et al. 2018). Les algorithmes d'apprentissages profonds ont permis d'automatiser des tâches jusque-là exclusivement réservées aux humains (Tontiwachwuthikul, Chan et al. 2020). Les outils d'automatisation des tâches constituent une forte valeur ajoutée pour une entreprise et permettent aux différentes équipes de se focaliser sur des tâches beaucoup plus stratégiques. Par exemple, l'automatisation dans la gestion des stocks utilisée par les entreprises permet de suivre les stocks, gérer les expéditions ainsi que le traitement des commandes (Praveen, Kumar et al. 2020).

Une tâche particulièrement ardue dans le domaine de l'assurance automobile consiste à estimer les dégâts à l'issue d'un accident, à partir de photos fournies par les clients et/ou les agents sur le terrain. Pour automatiser ce processus, Progi, une entreprise en TI basée à Trois-Rivières et offrant des solutions informatiques aux compagnies d'assurances (entre autres) souhaite développer un outil d'aide à la décision qui s'appuie sur les dernières avancées de l'IA en matière de traitement d'images, pour l'analyse des photos des véhicules accidentés. À notre connaissance, il n'existe encore aucune application ou système capable d'effectuer tout le processus automatiquement, même si des solutions très partielles ont été développées. Dans le processus actuel, l'inspection des dommages après un accident se fait par un estimateur, expert en dommages automobiles. « Ce dernier est un professionnel employé par la société d'assurance et certifié par les consortiums des assureurs automobiles (GAA) » (infoassurance 2019)

1.2 Contexte et problématique

Avec l'augmentation du nombre de véhicules, le marché de l'assurance automobile connaît une véritable expansion au cours de ces dernières années (Intelligence 2022). Un TCAC (Taux de Croissance Annuelle Composé) de plus de 4,4% sur la période de 2022-2027 est enregistré par le marché canadien de l'assurance (Intelligence 2022). En général, après un accident de véhicule, les victimes font appel à leur assureur. Ce dernier envoie un estimateur en dommage automobile chargé de constater les dégâts. Son rôle sera d'inspecter entièrement le véhicule, d'analyser les dommages, de déterminer les types de réparations à effectuer ainsi que leurs coûts pour enfin faire une estimation globale. Avec la fréquence des accidents, le processus d'estimation peut prendre du temps avant que l'estimateur ne rende les rapports finaux à l'assureur. En outre, dans l'industrie, de nombreuses pertes financières sont engendrées en raison des fuites de sinistres (Castonguay 2023). Les fuites de sinistres correspondent à l'écart entre l'indemnisation idéale et celle effectivement accordée. Pour réduire les fuites de réclamations, des inspections visuelles et des validations sont souvent effectuées. Toutefois, ces inspections peuvent augmenter le temps de traitement des réclamations. Un système automatisé d'inspection et d'estimation serait d'une grande utilité pour améliorer le processus.

1.3 Objectifs

L'objectif général de ce projet est l'automatisation entière du processus d'estimation des dommages sur un véhicule après un accident à partir de photos prises de différents angles vus. Cet objectif se divise en plusieurs sous objectifs parmi lesquels la détection des types des véhicules à partir de photos de différents angles de vue ainsi que l'identification des différentes parties des véhicules en utilisant des algorithmes d'apprentissage profond. Pour atteindre les sous-objectifs cités plus haut, nous procéderons de manière suivante :

- Préparations des données : collecte des données, prétraitement, dimensionnement des images et filtrages, balancement, création d'un ensemble de données robustes et dûment classifié.

- Détection des marques de véhicules : Exploration des différentes architectures existantes et nouvelles, détection à plusieurs étapes, inclusion de plusieurs de vues, affinement à plusieurs niveaux.
- Détection des différentes parties du véhicule : Entraînement sur des données étiquetées avec APEER, détection et localisation des différentes pièces externes des véhicules à l'aide de la segmentation et de la détection utilisant les boîtes (bounding-box).
- Détection des parties endommagées du véhicule : Détection des différentes parties du véhicule qui ont subi un dommage.

1.4 Organisation du mémoire

Notre mémoire est structuré comme suit. Dans le premier chapitre, l'introduction aborde le contexte, pose la problématique et décrit les objectifs de notre mémoire. Le deuxième chapitre fait l'état de l'art des articles qui traitent des modèles de reconnaissances d'images selon le type d'apprentissage, des modèles d'identifications et de localisations d'objets récemment utilisés. On aborde aussi dans ce chapitre les travaux qui ont été effectués dans le domaine de l'estimation des dommages sur un véhicule. Le chapitre 3 est consacré à la méthodologie de travail utilisée. Nous avons commencé par la collecte et la préparation des données. Nous avons effectué trois traitements de données différentes. Le premier traitement des données est consacré à la classification des marques de véhicules. La deuxième quant à elle est destinée à la segmentation des pièces externes du véhicule et le troisième pour la détection des parties endommagées du véhicules. Par la suite, nous avons exploré en détail, le modèle de classification que nous avons utilisée pour la reconnaissance des marques de véhicule (ResNet), modèle de segmentation d'images Unet pour la segmentation des différentes parties externes du véhicule et le modèle Yolo8 pour détecter les dommages. Dans le chapitre 4, nous présentons et discutons les résultats des différentes prédictions effectués sur les différents modèles utilisés dans ce projet. Enfin, le chapitre 5 conclut par une synthèse de notre travail, en brosse quelques perspectives de recherches et identifie les opportunités d'amélioration comme futures pistes de recherches.

Chapitre 2 : Revue de littérature

2.1 Introduction

Les dix dernières années se sont distinguées par un certain nombre d'avancées significatives dans la recherche en intelligence artificielle, plus particulièrement dans le domaine de l'apprentissage profond (Lu, Li et al. 2018). Les réseaux de neurones profonds ont longtemps été immobilisés par le problème de la singularité des calculs ou de la disparition du gradient (Grosse 2017) lors de la phase d'entraînement. Avec l'amélioration des outils de collecte de données, des capacités de stockage et de calcul, tout l'intérêt et le potentiel des réseaux de neurones profonds ont été dévoilés. Les réseaux neuronaux profonds ont rendu possible l'analyse de grandes masses d'informations telles que les photographies (milliers de pixels), et les résultats ont permis des avancées significatives dans le domaine de la vision par ordinateur (Rovelloti 2016). Selon certaines recherches (Kaur, Krishan et al. 2020), l'utilisation d'algorithmes de reconnaissance faciale a considérablement augmenté le niveau de sécurité dans différents domaines. L'analyse de grande masses de données en temps réel permet à des machines telles que les automobiles et les robots de progresser de manière autonome dans des environnements complexes qui étaient jusque-là considérés comme réservés uniquement aux humains.

Dans toutes les situations citées plus haut, il est primordial de pouvoir identifier et localiser des objets dans une image ou une vidéo. C'est un domaine de recherche qui est aujourd'hui exploré en profondeur (Haradji 2020) et qui est important si l'on veut voir émerger un jour un degré de compréhension et d'interprétation (des scènes) des images par les machines qui est proche de celui de l'homme.

2.2 La reconnaissance d'image

La reconnaissance d'images a débuté avec la reconnaissance des caractères de texte (Wu et Chen 2015). Vu son importance et le bon qualitatif qu'un modèle de reconnaissance d'images peut apporter dans plusieurs secteurs, les chercheurs ont adressé ce problème en

profondeur. L'arrivée de l'apprentissage automatique en général et de la vision par ordinateur en particulier a permis aux chercheurs de développer différents modèles de reconnaissance d'images. Pour évaluer et comparer ces modèles, des bases de données standardisées comme ImageNet (Deng, Dong et al. 2009) ont été construites. Le VGG (Simonyan et Zisserman 2014) figure parmi les modèles de reconnaissances d'images les plus connus. Il est basé sur une architecture de réseau de neurones convolutifs.

Auparavant plusieurs techniques avaient permis d'améliorer notablement l'état de l'art de la reconnaissance d'images en se focalisant sur les paramètres des réseaux autres que leurs profondeurs. Ainsi, AlexNet (Krizhevsky, Sutskever et al. 2012) qui atteignit une précision remarquable utilisait une taille de fenêtre réceptive plus petite avec des petits pas dans la première couche convolutive. Le VGG va par la suite améliorer les résultats du AlexNet. L'approche du modèle VGG utilisé par Simonyan et Zisserman (Simonyan et Zisserman 2014), fixe tous les modèles de l'architecture dans un premier temps pour ensuite, augmenter sa profondeur en ajoutant plusieurs couches de convolution munis de petits filtres. Ainsi après le modèle VGG, le modèle ResNet voit le jour.

He, K., et al. (He, Zhang et al. 2016) ont effectué l'apprentissage résiduel profond pour la reconnaissance d'image. Ils ont utilisé un cadre d'apprentissage résiduel pour faciliter l'entraînement des réseaux très profonds. Ils ont évalué les réseaux résiduels en augmentant leur profondeur avec un nombre de couches allant jusqu'à 152. Le succès de ce modèle réside dans sa capacité à prévenir la disparition/explosion du gradient et l'augmentation de l'erreur d'apprentissage. Bien que coûteux en calcul, il obtient de très bonnes performances en classification à tous les niveaux de détails de l'image.

Le modèle ResNet a fourni aux chercheurs une base de travail solide qui va nettement améliorer l'état de l'art dans le domaine. C'est dans ce contexte d'exploration de réseaux de plus en plus profonds, qu'a vu le jour EfficientNet (Tan et Le 2019). Un nouveau modèle qui surpasse ResNet en termes de précision et d'efficacité des modèles. EfficientNet peut être construit à partir de VGG, ResNet, etc. La mise à l'échelle d'un réseau de neurones convolutif concerne ses trois dimensions à savoir sa profondeur, sa largeur et sa résolution d'images. La profondeur d'un réseau de neurones est déterminée par son nombre de couches, sa largeur par le nombre de canaux dans les couches et enfin sa résolution, par la taille de sa couche d'entrée. Sachant que cette mise à échelle peut considérablement

affecter l'espace mémoire utilisé, le modèle est construit sous la contrainte de la disponibilité d'espace mémoire.

2.3 L'identification et la localisation d'objets

La reconnaissance et la localisation d'objets sont deux aspects essentiels de l'apprentissage automatique. Leur objectif est de donner aux ordinateurs la capacité de repérer, d'identifier et de cibler des éléments spécifiques dans des images fixes et des vidéos en mouvement (Zhao, Zheng et al. 2019, Waheed, Suaib et al. 2021). Ces algorithmes examinent les propriétés visuelles distinctes de chaque objet et les associent à des étiquettes définies par l'utilisateur, tout en calculant la position de chaque objet dans l'image. Les résultats ont une grande variété d'applications potentielles telles que la vidéosurveillance, l'identification de formes et de mouvements, la détection d'objets pour la conduite autonome, et d'autres encore (Wu, Sahoo et al. 2020).

2.3.1 L'identification des objets

L'identification et la localisation d'objets ou des formes particulières dans une image, ou une vidéo sont nécessaires dans notre contexte. Nous adresserons ce problème à l'aide de réseaux de neurones spécifiques/hybrides issus des deux catégories de modèles (Cao, Wu et al. 2019) d'apprentissage profond utilisé pour détecter des objets dans une image.

Les modèles de la première catégorie reconnaissent les objets en délimitant les zones (boîtes rectangulaires circonscrites) qui les contiennent alors que dans la deuxième catégorie, les modèles classifient des pixels obtenus à la suite de la résolution de l'image en question. La forme des objets que contiennent les images ou les vidéos considérées peut alors considérablement influencer sur le choix de la technique à utiliser. Les modèles YOLO (Redmon et Farhadi 2018) et RetinaNet (Lin, Goyal et al. 2017) font partie de la première catégorie annoncée ci-dessus. L'architecture YOLO, prend en entrée des images étiquetées avec des vecteurs de caractéristiques nécessaires pour définir des rectangles circonscrivant les objets détectés. Dans un problème de classification avec deux classes d'objets, ces vecteurs prennent la forme suivante :

$V [P_c, B_x, B_y, B_w, B_h, C_1, C_2]$

Avec P_c la probabilité de classe autrement dit, la confiance qu'a le modèle d'avoir détecté un objet d'intérêt pour notre classification, B_x et B_y sont les coordonnées du centre du rectangle circonscrit autour de l'objet détecté. B_h et B_w sont la hauteur et la largeur de ce rectangle, et C_1 et C_2 sont les probabilités que l'objet détecté soit respectivement de classe 1 ou de classe 2.

Chaque image est divisée en plusieurs cellules qui représentent les régions dans lesquelles est effectuée la recherche. Chaque cellule de l'image a son vecteur correspondant. En cas de chevauchement des boîtes englobantes pour la même classe, l'algorithme utilise une technique appelée intersection sur union pour éliminer les boîtes superflues et garder celle ayant la probabilité de classe la plus élevée. L'ensemble du processus se fait en un seul passage, d'où le nom YOLO (You Only Look Once).

En 2020, la publication de YOLOv4 (Bochkovskiy, Wang et al. 2020) a apporté des innovations comme l'utilisation de l'augmentation de données Mosaic, une nouvelle tête de détection non ancrée et une nouvelle fonction de perte.

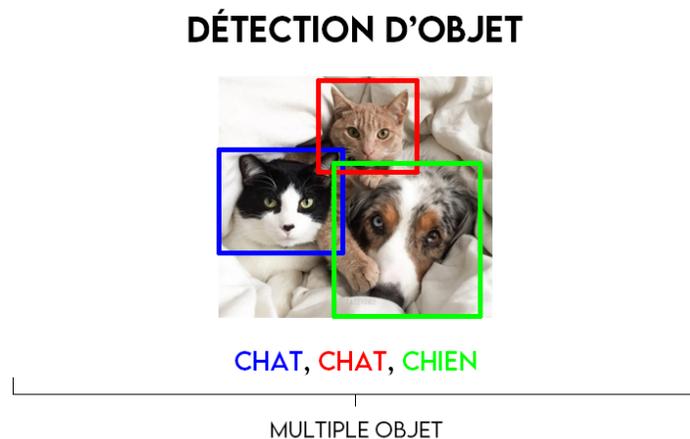


Figure 1 : Détection d'objet. Source (Learning 2020)

2.3.2 Localisation des objets

La segmentation d'images en apprentissage automatique est une technique permettant de diviser une image en plusieurs régions ou segments distincts (Hesamian, Jia et al. 2019).

Cette technique est utilisée dans de nombreux domaines tels que la reconnaissance d'objets, la surveillance, la médecine, l'analyse de données, etc.

L'objectif de la segmentation d'image est de détecter et d'isoler les régions d'intérêt de l'image en fonction de caractéristiques spécifiques telles que la couleur, la texture, la forme ou la luminosité. Cette tâche est souvent réalisée à l'aide de réseaux de neurones convolutifs qui permettent d'extraire automatiquement les caractéristiques des images et de les utiliser pour effectuer la segmentation.

Il existe plusieurs modèles d'apprentissage profond qui peuvent effectuer la tâche de segmentation d'images parmi lesquels nous avons :

- Le modèle U-net : basé sur des réseaux entièrement convolutifs, le modèle U-net (Ronneberger, Fischer et al. 2015) est une architecture conçue pour la segmentation sémantique. Comparé à d'autres réseaux profonds, U-net a été conçu pour fonctionner avec peu d'images d'entraînement. Son architecture se compose de deux parties principales: la partie contractante qui est généralement constituée de réseaux convolutifs, et la partie expansive qui est un suréchantillonnage de la carte des caractéristiques, suivie d'une convolution 2x2. En plus de la classification des images, U-net localise les caractéristiques de chaque classe générée en effectuant une classification sur chaque pixel (Cao, Wu et al. 2019).

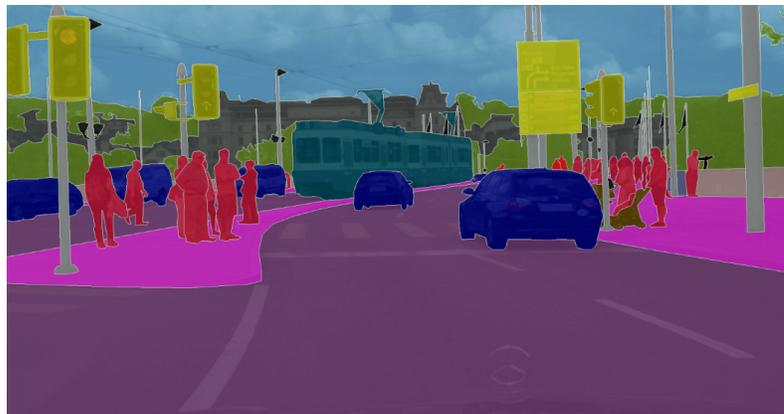


Figure 2 : segmentation d'image avec U-net source (Vault 2020)

- Le modèle RetinaNet : RetinaNet (Lin, Goyal et al. 2017) est une architecture fréquemment utilisée pour la détection des textes ou des objets dans des images.

Les auteurs ont développé une nouvelle fonction de perte appelée « Focal Loss » en se basant sur l'entropie croisée (CE) pour améliorer les résultats de détection. Contrairement aux algorithmes en deux étapes tels que R-CNN (Girshick, Donahue et al. 2014) et Fast R-CNN (Girshick 2015), qui nécessitent la proposition de régions contenant potentiellement des objets d'intérêt, la plupart des régions proposées sont en fait des exemples négatifs. Cela crée un déséquilibre dans les classes d'objets lors de l'entraînement, ce qui peut affecter les performances de la détection. La fonction Focal Loss a été conçue pour corriger ce problème de déséquilibre des classes. Cette technique permet aux régions (proposées) positives d'avoir plus de poids dans la fonction de perte totale que les régions négatives.

La fonction d'entropie croisée est définie comme suit :

$$CE(y, pt) = -[y * \log(pt) + (1 - y) * \log(1 - pt)]$$

Avec $y=1$ on obtient

$$CE(1, pt) = -\log(pt)$$

La fonction perte Focal Loss est donnée par:

$$FL(pt) = -(1 - p)^{\gamma} \log(pt)$$

Pour améliorer les performances de RetinaNet, des chercheurs ont développé des architectures hybrides comme RétinaNet+ avec des réseaux entièrement convolutifs (FCN) utilisés lors de la phase de classification des images (Lin, Goyal et al. 2017).

Cependant, les architectures actuelles telles que RetinaNet et Unet ne sont pas en mesure de raisonner. En effet, une fois que l'arrière-plan des objets est modifié, ces architectures ne sont plus en mesure de détecter le même objet sur lequel elles ont été entraînées, comme l'a souligné (Lu, Li et al. 2018). Il est donc nécessaire de trouver des solutions pour combler cette lacune dans les architectures actuelles en termes de raisonnement. Les chercheurs dans (Faghihi, Robert et al. 2020) ont proposé une architecture hybride basée sur les auto-encodeurs variationnelles qui permettent de raisonner, comme l'ont noté (Lu, Li et al. 2018) dans leur étude.

Dans notre projet, nous aurons à détecter (localiser et d'identifier) différentes parties externes d'une voiture à partir de photos entières. Pour se faire, nous allons étudier les performances des modèles issus des deux catégories de classificateurs (cités plus haut) et les comparer. Les formes irrégulières des parties d'un véhicule et leur variabilité suivant les modèles (marques et années), contribuent à complexifier le processus de détection. En effet, les outils de détection classiques adressent plus souvent des formes rectangulaires, régulières. Nous explorerons des modèles hybrides comme ceux utilisés par (Chen, Yao et al. 2020) combinant le modèle Unet et ResNet afin d'améliorer le processus de détection.

2.4 Détection des dommages sur les véhicules

Les chercheurs ont exploré différentes technologies utilisant la vision par ordinateur et le traitement d'images pour effectuer la reconnaissance, la segmentation et la détection des dommages sur les véhicules. Quelques-unes de ces études sont présentées ci-dessous.

(Kyu et Woraratpanya 2020) ont effectué une étude sur la détection et la classification des dommages automobiles en utilisant des algorithmes d'apprentissage profond pour détecter les dommages sur un véhicule et évaluer leur localisation et leur gravité. Les algorithmes d'apprentissage profond utilisés dans leur étude incluent les réseaux de neurones convolutifs (CNN) tels que VGG16 et VGG19, ainsi que l'apprentissage par transfert. Ils appliquent également l'apprentissage par transfert et la régularisation L2 pour améliorer la précision du système. Les résultats montrent que le modèle VGG19 est plus performant que le modèle VGG16 dans la détection des dommages avec une précision de 94%, une précision de 71% et 59% pour la localisation et la gravité des dommages. Cependant les problèmes qu'ils ont rencontrés incluent le temps de calcul élevé nécessaire pour entraîner les modèles, ainsi que la nécessité de collecter manuellement des images de voitures endommagées et non endommagées pour l'ensemble de données.

(Dwivedi, Malik et al. 2021) ont présenté un modèle de classification et de détection des dommages automobiles en classant les dommages en sept catégories à savoir : bosse de pare-chocs, rayures, bosse de porte, éclat de verre, phare cassé, feu arrière cassé et brisé. L'ensemble de données comprenait des images d'entraînement et de test pour chaque type de dommage, avec des tailles spécifiques pour chaque classe. Ils ont utilisé un modèle CNN

de base, puis ont effectué un pré-entraînement avec un encodeur automatique, suivi d'un réglage fin des modèles de classification d'images pré-entraînés sur l'ensemble de données ImageNet. Ils ont constaté que l'apprentissage par transfert était le plus performant pour la classification des dommages automobiles. De plus, ils ont utilisé le détecteur d'objets YOLOv3 pour la détection des dommages et ont obtenu un score cartographique maximum de 77,78 %. Les auteurs suggèrent également dans le futur une combinaison des tâches de classification et de détection. Ces résultats ouvrent la voie à des recherches plus approfondies dans ce domaine problématique. Cependant avec l'utilisation de différentes classes pour les dommages de véhicules, la nature des dommages à la suite d'un accident pourrait différer de celle identifiées dans cet article. Le nombre limité de types de dommage proposés par ces auteurs ne convient pas à notre problématique d'évaluation globale des dommages.

Les chercheurs (van Ruitenbeek et Bhulai 2022) ont amélioré les travaux précédents en utilisant des réseaux de neurones convolutifs pour détecter et classifier les dommages subis par le véhicule. Ils étendent leur ensemble de données avec plus de 10000 images de véhicules endommagés grâce à l'augmentation des données. Ils ont développé un modèle de détection qui localise les dommages sur les véhicules et les classe dans douze catégories. Ils ont utilisé des techniques d'apprentissage par transfert pour améliorer les performances du modèle, ainsi que des méthodes de prétraitement pour nettoyer l'ensemble de données. Ils ont rencontré des difficultés dans des environnements lumineux avec de fortes réflexions.

(Qaddour et Siddiq 2023) ont présenté un nouveau modèle-cadre basé sur l'apprentissage profond pour détecter et estimer les dommages causés aux véhicules de manière rapide et efficace. La méthode qu'ils ont utilisé combine l'apprentissage profond, l'apprentissage par transfert, Mask R-CNN et la segmentation d'instance pour identifier et catégoriser les dommages sur une image de véhicule. Trois modèles pré-entraînés différents sont utilisés pour raccourcir les périodes de test et de formation à savoir VGG-16, VGG-19 et Inception ResNetV2. Et Chacun de ces modèles a été utilisé pour effectuer des tâches spécifiques dans le processus d'évaluation des dommages, notamment pour déterminer si un véhicule est endommagé ou non, localiser la partie endommagée et estimer la gravité des dommages. De plus, une application web de détection automatique et d'estimation de classification est

développée et déployée pour faciliter le processus. Les résultats expérimentaux indiquent une précision de 89,13% avec le réseau Inception ResNetV2, 86,09% avec le réseau VGG19 et 83,5% avec le réseau VGG16 pour la détection des dommages. En ce qui concerne la localisation des dommages, les précisions sont de 74,33% avec le réseau VGG16, 70,66% avec VGG19 et 80,4% avec le réseau Inception ResNetV2. Pour évaluer la sévérité des dommages, des précisions de 65,66% avec VGG16, 66,33% avec VGG19 et 67,12% avec le réseau Inception ResNetV2 ont été obtenues. Toutefois, la limitation est l'absence d'informations spécifique sur la marque/modèle du véhicule.

2.5 Conclusion

Ces dernières années, les performances des réseaux de neurones ont considérablement augmenté dans les domaines de la reconnaissance d'images, de la détection et de la localisation d'objets. Cette amélioration peut être attribuée au développement de réseaux neuronaux profonds ainsi qu'à la disponibilité d'énormes quantités de données d'entraînement. Néanmoins, malgré ces progrès, de nombreuses difficultés subsistent dans ce domaine. Il s'agit notamment de la reconnaissance d'objets dans des scénarios complexes dans lesquels un certain nombre d'objets peuvent être présents dans un environnement encombré, ainsi que de la détection et de la localisation d'objets même lorsque l'arrière-plan change ou que l'éclairage change. Dans ces conditions, les modèles d'apprentissage profond auront plus de difficultés à remplir leur mission.

D'un autre côté, des recherches prometteuses sont menées dans le but d'accroître le pouvoir de raisonnement des cadres de reconnaissance d'images. Cette recherche se concentre sur la compréhension des scènes et des objets dans leurs contextes environnants afin de fournir une perspective plus complète et une meilleure interprétation de l'image.

Dans la section suivante de ce mémoire, nous fournirons les solutions proposées au problème de l'identification des marques de véhicules, ainsi que la détection de pièces distinctes, en faisant usage de modèles d'apprentissage automatique et de modèles hybrides qui combinent plusieurs modèles. Ces solutions seront présentées dans le cadre du chapitre suivant de ce mémoire

Chapitre 3 : Méthodologie et analyses des données

Dans ce chapitre, nous présentons la méthodologie que nous avons élaborée pour résoudre la problématique abordée dans ce mémoire. Nous commençons par expliquer les algorithmes d'apprentissage profond utilisés pour la classification des marques de voitures, la détection des pièces externes et des dommages sur le véhicule. Ensuite, nous détaillons les approches choisies pour le traitement des données afin d'entraîner ces différents modèles. La solution proposée pour résoudre ce problème consiste à automatiser le processus d'inspection et de validation en utilisant des technologies telles que la vision par ordinateur et l'intelligence artificielle. Cela permettrait d'accélérer considérablement le traitement des réclamations et de réduire les délais pour les victimes tout en minimisant les pertes financières pour les assureurs.

3.1 Méthodologie

Nous nous basons sur trois approches pour résoudre la problématique abordée dans ce mémoire. Il s'agit de la classification, la segmentation et la détection.

3.1.1 Approche par la classification

La classification est une tâche où un modèle d'apprentissage automatique qui attribue des étiquettes ou des catégories à des données en fonction de certaines caractéristiques. L'approche par classification consiste à prédire la classe qui va correspondre à la marque de l'automobile. Nous distinguons la classification binaire, où le modèle doit choisir entre deux classes distinctes de la classification multi-classes, où le modèle doit choisir parmi plus de deux classes. Dans notre étude, nous utilisons la classification multi-classes afin d'identifier les diverses marques de voitures.

L'objectif de cette identification de la marque du véhicule est d'estimer par la suite les pièces détectées de ce véhicule en fonction de la marque, du modèle et de l'année. En effet, les coûts des pièces automobiles varient en fonction de ces paramètres

3.1.2 Approche par la segmentation

L'approche par la segmentation consiste à faire le découpage d'une image en un certain nombre de parties ou de segments distincts. L'objectif de cette segmentation est de permettre l'analyse et l'interprétation de nos images afin de localiser les différentes pièces externes de la voiture. La segmentation d'une image peut être réalisée à l'aide de diverses méthodes, notamment la segmentation par seuil, la segmentation par région, la segmentation par contour et la segmentation sémantique, pour n'en citer que quelques-unes parmi les plus courantes (Phonsa et Manu 2019).

Lors de la segmentation basée sur les seuils, les pixels de l'image sont triés en groupes en fonction du niveau d'intensité des couleurs ou de la valeur des niveaux de gris qu'ils possèdent, à l'aide des seuils établis précédemment. Les pixels dont les valeurs sont supérieures ou inférieures à ces seuils sont séparés en segments distincts.

Les pixels sont organisés en sections homogènes à l'aide de la segmentation par région, qui repose sur des facteurs tels que la similitude de leurs couleurs, de leurs textures, de leurs formes et d'autres propriétés visuelles. Cette méthode permet de diviser l'image en sections cohérentes qui représentent chacune un objet ou une partie de l'image Figure 3.

L'objectif de la segmentation basée sur les contours est d'identifier et de suivre les contours des nombreux éléments visibles dans l'image (Figure 3). Cette méthode localise les bords des objets et les isole du reste de l'image en utilisant des techniques de détection des bords telles que les gradients d'intensité et les opérateurs de détection des bords.

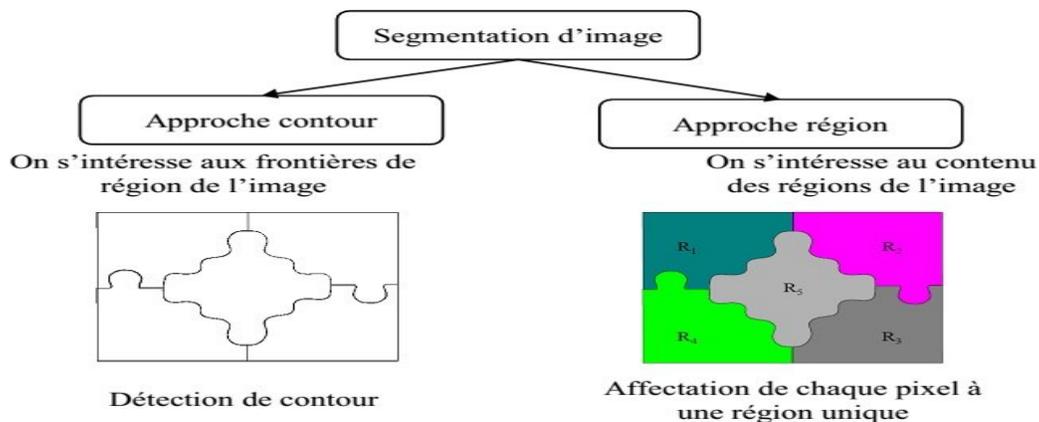


Figure 3 : Différence entre segmentation par contour et segmentation par région (Boudjehem 2016)

Une méthode améliorée connue sous le nom de segmentation sémantique (Liu, Deng et al. 2019) a été développée dans le but d'appliquer des étiquettes sémantiques à chaque pixel d'une image. La segmentation sémantique est une technique qui, plutôt que de simplement séparer l'image en sections, attribue des interprétations sémantiques distinctes à chaque région de l'image comme indiqué dans la Figure 4. Cela permet une compréhension plus approfondie du contenu de l'image.

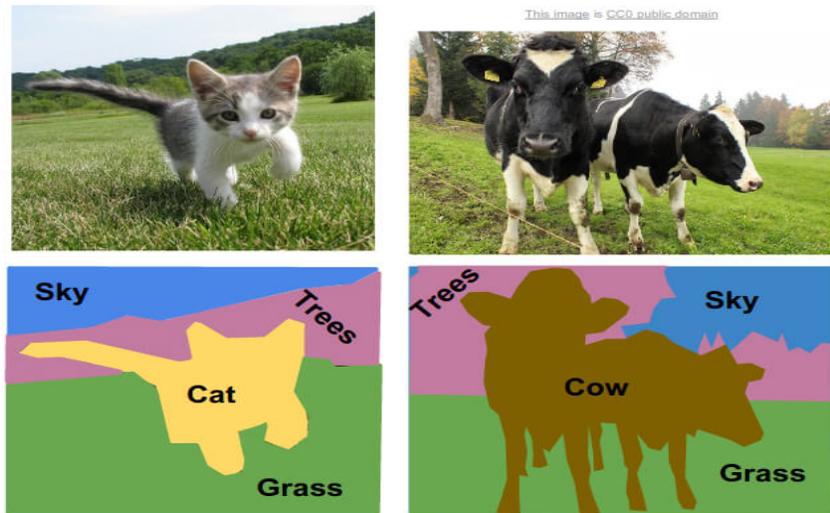


Figure 4 : Segmentation sémantique (Vahagn Tumanyan 2023)

Dans le cadre de notre étude, nous avons utilisé la segmentation sémantique pour localiser de différentes pièces externes de la voiture et mieux analyser les caractéristiques de l'image. L'objectif de la détection des pièces externes de la voiture est de pouvoir analyser chaque pièce du véhicule séparément.

3.1.3 Approche par la détection

Notre troisième approche se concentre sur la détection d'objets. Cette tâche consiste à repérer et identifier des éléments spécifiques au sein d'une images ou d'une vidéos. Dans notre contexte, nos objectifs sont les suivants :

- D'abord, nous cherchons à détecter les parties endommagées des véhicules en analysant plusieurs photos qui fournissent une vue complète de la voiture. Nous utilisons des modèles de détection d'objets pour accomplir cette étape.

- Ensuite, nous nous concentrons sur les zones affectées détectées précédemment afin de déterminer la localisation spécifique des dommages sur le véhicule.

L'objectif de cette détection des dommages est de localiser les parties endommagées du véhicule et d'évaluer par la suite l'état des dommages.

La Figure 5 montre le schéma du processus d'estimation du véhicule.

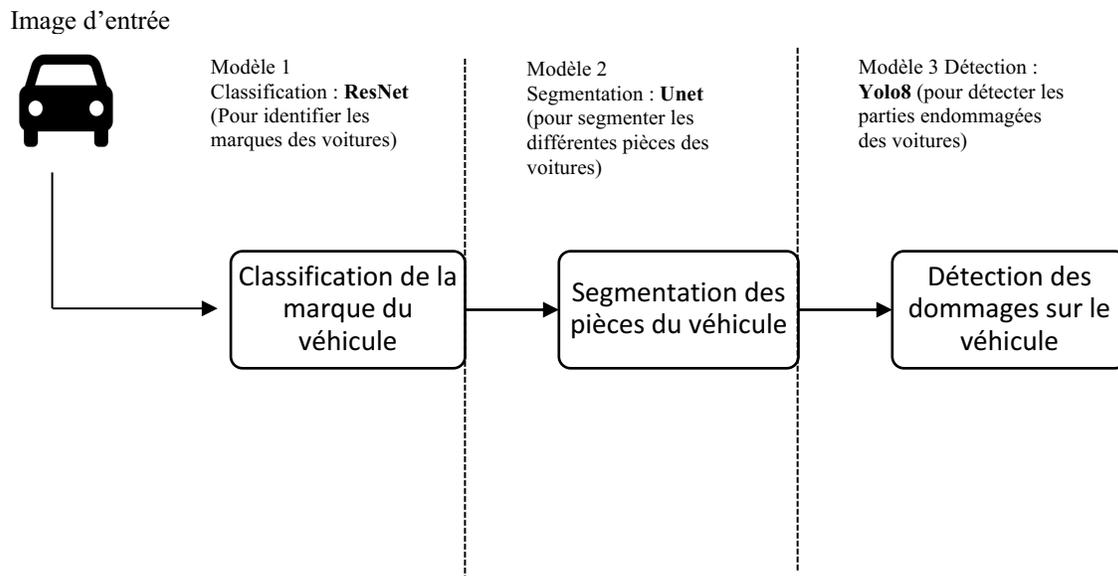


Figure 5 : Organigramme du processus de la détection des dommages

3.2 Présentation des modèles d'apprentissages automatiques utilisés

Le machine Learning, également appelé apprentissage automatique, est une branche de l'intelligence artificielle (Dhall, Kaur et al. 2020) qui se concentre sur la création d'algorithmes et de techniques permettant aux ordinateurs d'apprendre à partir de données. Les systèmes d'apprentissage automatique sont conçus pour analyser des données, identifier des modèles, prendre des décisions et accomplir des tâches sans avoir besoin d'être explicitement programmés pour chacune d'entre elles (Mahesh 2020). Il comprend plusieurs sous-domaines tels que l'apprentissage supervisé, l'apprentissage non supervisé, l'apprentissage par renforcement, l'apprentissage en profondeur (Deep Learning) et bien d'autres.

L'apprentissage supervisé (Dridi 2021) est une méthode qui consiste à entraîner l'algorithme sur un jeu de données étiqueté. C'est-à-dire que chaque exemple dans l'ensemble de données a une étiquette ou une valeur de sortie connue. L'objectif de l'apprentissage supervisé est de permettre à l'algorithme d'apprendre à partir d'exemples étiquetés, afin de prédire avec précision les étiquettes pour de nouvelles données inconnues (Dridi 2021). Il est couramment utilisé dans des applications telles que la classification, la régression et la détection d'anomalies.

L'apprentissage non supervisé (Borgholt, Havtorn et al. 2022) est une méthode qui consiste à utiliser un algorithme pour découvrir des structures, des modèles ou des informations utiles dans un ensemble de données non étiqueté, sans avoir recours à des données étiquetées préalables comme c'est le cas pour l'apprentissage supervisé. En effet, dans ce dernier, l'algorithme est formé sur un ensemble de données étiqueté afin de prédire des étiquettes ou des résultats spécifiques. Il est souvent utilisé pour l'exploration des données, la segmentation, l'analyse de texte et d'autres domaines où les données ne sont pas étiquetées (Borgholt, Havtorn et al. 2022).

L'apprentissage en profondeur (LeCun, Bengio et al. 2015), également connu sous le nom de « Deep Learning » en anglais, est un sous-domaine de l'apprentissage automatique (machine learning) qui se concentre sur l'utilisation de réseaux de neurones artificiels profonds. Il est souvent utilisé pour résoudre des problèmes d'apprentissage automatique qui impliquent des données massives et des tâches complexes telles que la reconnaissance d'images, la traduction automatique, la compréhension du langage naturel, la génération de contenu créatif, et bien d'autres (Fan, Ma et al. 2021).

Le cœur de l'apprentissage profond réside dans les réseaux de neurones profonds c'est-à-dire qu'il utilise des réseaux de neurones multicouches pour comprendre des données complexes, comme des images. Cette approche permet de capturer des caractéristiques à différents niveaux de détail, ce qui la distingue des méthodes statistiques traditionnelles. En d'autres termes, les réseaux profonds sont excellents pour analyser des données compliquées et en extraire des informations utiles (Fan, Ma et al. 2021).

En raison de la complexité de nos données et de la nature de notre problème, nous avons opté pour l'utilisation de modèles d'apprentissage profond dans cette étude. En outre, il convient de noter que les réseaux de neurones profonds ont prouvé leur efficacité en dépassant les performances des méthodes traditionnelles, comme cela a été expliqué dans le chapitre 2 de notre travail.

3.2.1 Les réseaux de neurones convolutifs

Le réseau de neurones convolutifs, également appelé CNN « Convolutional Neural Network » en anglais, est un type spécialisé de modèle de réseau de neurones utilisé principalement pour la classification d'images (Li, Liu et al. 2021). Ces réseaux sont capables d'apprendre à extraire des caractéristiques locales, c'est-à-dire des structures qui se répètent à travers l'image. Ils sont composés de plusieurs couches de neurones connectées entre elles (Figure 6).

La première partie du modèle est appelée la partie convolutive, qui est responsable de l'extraction des caractéristiques de l'image en utilisant des opérations de convolution (Figure 6). La deuxième partie est responsable de la classification de l'image, plus précisément la couche dense prend en entrée un vecteur et produit un nouveau vecteur en sortie. Elle applique une combinaison linéaire suivie éventuellement d'une fonction d'activation aux valeurs reçues en entrée. Cette couche permet de classifier l'image en attribuant une probabilité à chaque classe. Par exemple, si le problème consiste à distinguer les chats des chiens, le vecteur de sortie aura une taille de 2, où chaque élément représente la probabilité d'appartenir à la classe "chat" ou "chien" (Li, Liu et al. 2021).

L'un des avantages clés des réseaux de neurones convolutifs est leur capacité à partager les poids et les biais entre les neurones cachés d'une même couche. Cela signifie que tous les neurones cachés détectent la même caractéristique dans différentes régions de l'image, ce qui rend le réseau tolérant aux translations d'objets dans l'image. Par exemple, un réseau entraîné à reconnaître des voitures sera capable d'identifier un véhicule, quel que soit son emplacement dans l'image.

Dans un réseau convolutif, certains paramètres sont utilisés pour extraire les caractéristiques de l'image. Parmi ces caractéristiques nous avons :

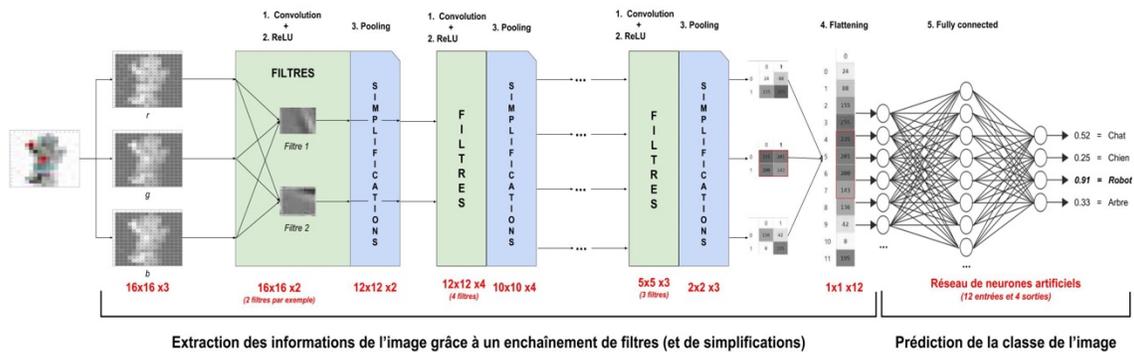


Figure 6 : Exemple de réseau de neurones convolutifs (artificiel).

- Les filtres : également appelés noyaux de convolution, sont des matrices de poids utilisées pour effectuer des opérations de convolution sur l'image d'entrée (Li, Liu et al. 2021). Chaque filtre est appliqué à l'image pour extraire des caractéristiques spécifiques, telles que les contours, les textures ou les motifs.
- Le padding : Le padding (Figure 7) fait référence à la technique d'ajout de zéros autour des bords de l'image d'entrée avant de l'envoyer à la couche de convolution. Le padding permet de préserver la taille spatiale de l'image lors de la convolution et de conserver les informations des bords de l'image. Il existe deux types de padding couramment utilisés : "valid" (aucun padding) et "same" (ajout de zéros pour maintenir la taille de l'entrée)

0	0	0	0	0	0
0	35	47	19	8	0
0	11	18	15	32	0
0	9	28	10	5	0
0	4	67	49	50	0
0	0	0	0	0	0

Figure 7 : Exemple de "padding"

- Le max pooling : est une technique de réduction de dimension (Figure 8) utilisée en apprentissage automatique pour réduire la taille du volume de données en

conservant les caractéristiques les plus importantes. Cette technique consiste à diviser le volume en fenêtres et à ne conserver que la valeur maximale de chaque fenêtre. Cela permet de réduire la taille du volume tout en conservant les informations importantes pour le modèle.

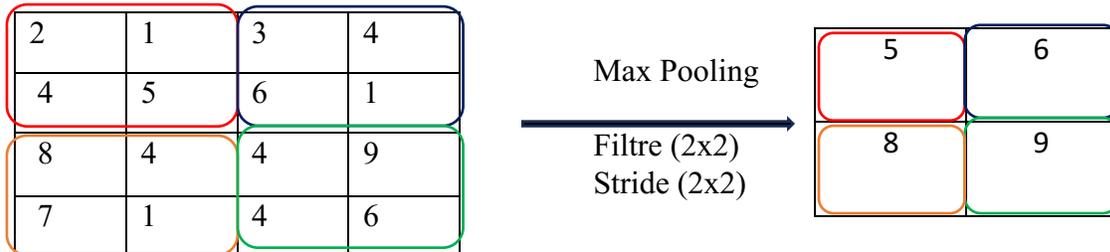


Figure 8 : Exemple de technique de max pooling

3.2.2 Le Modèle ResNet

Le ResNet, également connu sous le nom de Residual Network, est une architecture de réseau neuronal convolutif (CNN) introduit en 2015 par (He, Zhang et al. 2016). La vision par ordinateur est une technique couramment employée pour traiter diverses tâches comme la classification d'images, la détection d'objets et la segmentation d'images. En 2015 il remporte le challenge ImageNet Large Scale Visual Recognition Challenge (ILSVRC) (Fan, Ma et al. 2021) devant les modèles AlexNet, VGG19 et Google-Net avec plus de couches et un taux d'erreur beaucoup plus faible par rapport aux autres modèles comme montré dans le Tableau 1.

Modèle	Année	Couches	Paramètres	Top 5 Erreurs
Peu Profond	< 2012	-	-	>25%
AlexNet	2012	8	61M	16,4%
VGG19	2014	19	144M	7,3%
GoogleNet	2014	22	7M	6,7%
ResNet-152	2015	152	60M	3,6%

Tableau 1 : Les modèles gagnant lors du concours ImageNet Large Scale Visual Recognition Challenge source(Fan, Ma et al. 2021)

Le facteur distinctif entre ResNet et les architectures CNN conventionnelles réside dans l'incorporation de blocs résiduels. Les blocs résiduels sont des modules de réseau qui permettent de contourner les couches du réseau, favorisant ainsi l'apprentissage en profondeur tout en évitant le problème de la disparation du gradient. Les blocs résiduels permettent la formation de réseaux profonds avec de nombreuses couches sans que les performances du réseau se dégradent.

L'architecture ResNet est caractérisée par :

- Sa profondeur : Les réseaux ResNet sont définis par le nombre de couches qu'ils comportent. Les versions de ResNet-18 à ResNet-152 comportent un grand nombre de couches. Cela permet d'apprendre des représentations de caractéristiques complexes à partir de données brutes, ce qui peut améliorer les performances du réseau.
- Son utilisation de blocs résiduels : Les blocs résiduels sont des modules de réseau qui permettent aux informations de sauter certaines couches du réseau. Ils facilitent l'apprentissage en profondeur en évitant le problème de l'évanouissement du gradient. Cela permet de former des réseaux profonds sans compromettre les performances.
- Efficacité : comparés à d'autres modèles de réseaux neuronaux profonds, les réseaux ResNet sont connus pour être les plus efficaces en termes de nombre de paramètres et de temps nécessaire pour effectuer les calculs. Cette caractéristique les rend bien adaptés à un déploiement en temps réel ou dans des scénarios à grande échelle.
- Une précision remarquable : Les réseaux ResNet ont fait preuve d'une précision exceptionnelle dans l'exécution de tâches complexes de vision par ordinateur, notamment la classification d'images, la détection d'objets et la segmentation d'images. Le système obtient fréquemment des résultats exemplaires dans les concours de référence et est largement utilisé par la communauté des chercheurs en vision par ordinateur.
- Adaptabilité : Les réseaux ResNet présentent un degré élevé d'adaptabilité, ce qui leur permet d'être ajustés de manière flexible pour s'adapter à divers ensembles de

données et aux exigences des tâches. L'utilisation de ces modèles peut servir de cadre fondamental pour l'apprentissage par transfert, dans lequel les poids préexistants obtenus pour différentes tâches sont utilisés pour lancer le processus d'apprentissage d'une nouvelle tâche, même lorsque les données d'apprentissage disponibles sont limitées.

En outre, ResNet résout efficacement le problème de la fuite de gradient. Dans le contexte des réseaux neuronaux, il a été observé qu'à mesure que la profondeur du réseau augmente, les gradients utilisés pour calculer la fonction de perte tendent à diminuer rapidement vers zéro après de multiples itérations de la règle de la chaîne. Par conséquent, les valeurs de poids ne sont pas mises à jour et il n'y a pas d'apprentissage puisque les gradients sont trop faibles pour entraîner le réseau. Le problème de la perte de gradient est résolu efficacement par ResNet grâce à la mise en œuvre de connexions de saut, également connues sous le nom de connexions résiduelles, qui fournissent des connexions directes entre les différents niveaux du réseau.

3.2.2.1 Architecture

Dans la Figure 11, on peut voir l'architecture du modèle ResNet, en particulier la version ResNet-34 avec 34 couches, présentée dans la colonne de droite. Il est communément appelé "réseau résiduel" en raison de l'utilisation de blocs résiduels. Les blocs résiduels incorporés dans l'architecture ResNet sont spécifiquement conçus pour permettre au flux d'informations de contourner des niveaux spécifiques au sein du réseau. Ce mécanisme contribue efficacement à la propagation des gradients et résout le problème de perte du gradient.

Le bloc résiduel, tel qu'il est représenté à la Figure 9, utilise une connexion de saut, également connue sous le nom de connexion résiduelle, pour combiner l'entrée d'une couche avec la sortie d'une couche suivante. Ce mécanisme établit une voie directe reliant les niveaux, permettant la transmission d'informations sans passer par des couches intermédiaires.

Selon (He, Zhang et al. 2016), le calcul de la sortie d'un bloc résiduel consiste à ajouter l'entrée de la couche précédente à la sortie de la couche actuelle. La somme est ensuite traitée par une fonction d'activation, telle que ReLU (Rectified Linear Unit), afin d'obtenir

la sortie finale du bloc résiduel. Cette fonction Rectified Linear Unit (ReLU) prend des valeurs réelles en entrée et renvoie en sortie des valeurs dans l'intervalle $[0, \infty]$. Sa formule est :

$$f(x) = \max(0, x)$$

En observant la Figure 11, il devient évident qu'une connexion directe est présente, permettant de contourner des couches spécifiques au sein du modèle. Ce lien est essentiel pour les blocs résiduels, car il modifie la sortie. En l'absence de cette connexion, l'entrée "X" est soumise à un processus de multiplication par les poids de la couche, qui est ensuite suivi par l'ajout d'un terme de biais.

Après cela, nous appliquons la fonction d'activation $F(x)$ à notre entrée x pour obtenir la sortie $H(x)$.

$$H(x) = F(wx + b) \text{ ou } H(x) = F(x)$$

Avec l'introduction d'une nouvelle connexion de saut, la sortie $H(x)$ est changée en

$$H(x) = F(x) + x$$

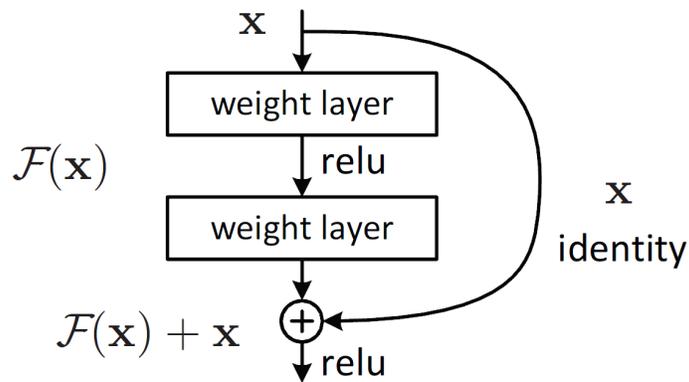


Figure 9 : Connexion de sauts de ResNet (He, Zhang et al. 2016).

Grâce à cette méthodologie, il est désormais possible de construire des réseaux neuronaux profonds comportant jusqu'à plusieurs centaines de couches, tout en résolvant le problème de la perte du gradient. Les modèles ResNet fréquemment utilisés ont généralement une profondeur de 18, 34, 50, 101, 152, voire plus. La principale méthode utilisée par l'architecture ResNet pour extraire les caractéristiques des images d'entrée est appelée couches de convolution, et la mise en commun maximale est utilisée pour réduire la taille.

En outre, la normalisation par lots permet de stabiliser les distributions d'activation et d'accélérer le processus d'apprentissage. Lorsque l'on tente d'ajouter de la non-linéarité à un réseau, il est courant d'utiliser une fonction d'activation non linéaire telle que ReLU. Dans la plupart des cas, la classification finale des classes d'objets est gérée par des couches entièrement liées situées à la fin des modèles ResNet. L'utilisation de connexions résiduelles, décrites dans la Figure 9, est un élément essentiel de l'architecture ResNet.

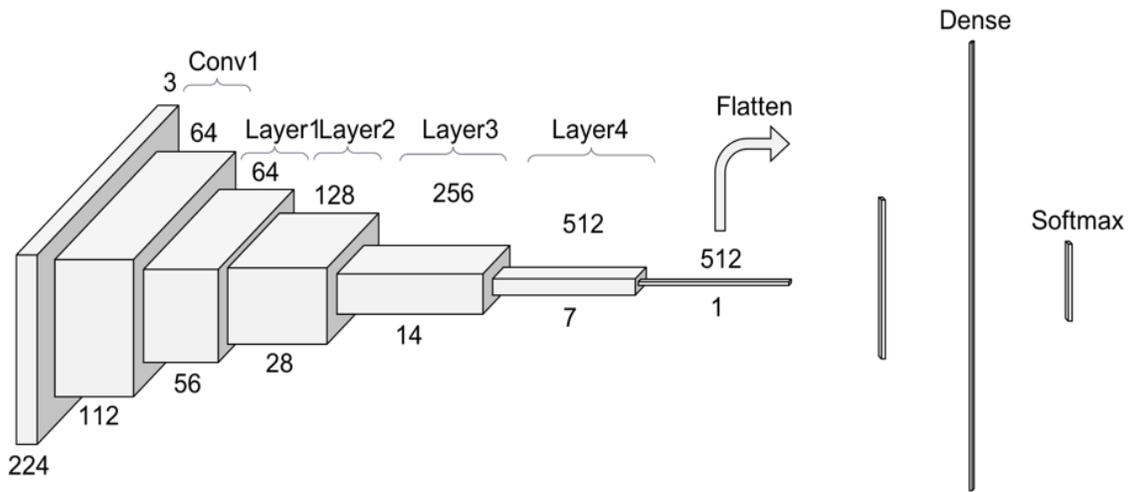


Figure 10 : Vue détaillée du modèle ResNet couche par couche. Source (Ruiz 2018)

Dans la Figure 10 nous avons les couches du modèle ResNet. La première étape du modèle, nous avons un bloc appelé Conv1. Ce bloc comprend une opération de convolution, une normalisation par lots et un max pooling. La convolution utilise un noyau de taille 7 et une carte de caractéristiques de 64, avec un remplissage de zéros de 3 fois sur chaque dimension. La taille de sortie de cette opération est un volume de 112x112x64. Chaque filtre de convolution fournit un canal dans le volume de sortie. La normalisation par lots est utilisée pour normaliser les valeurs de chaque canal tandis que le max pooling réduit la taille du volume en prenant le maximum dans chaque région.

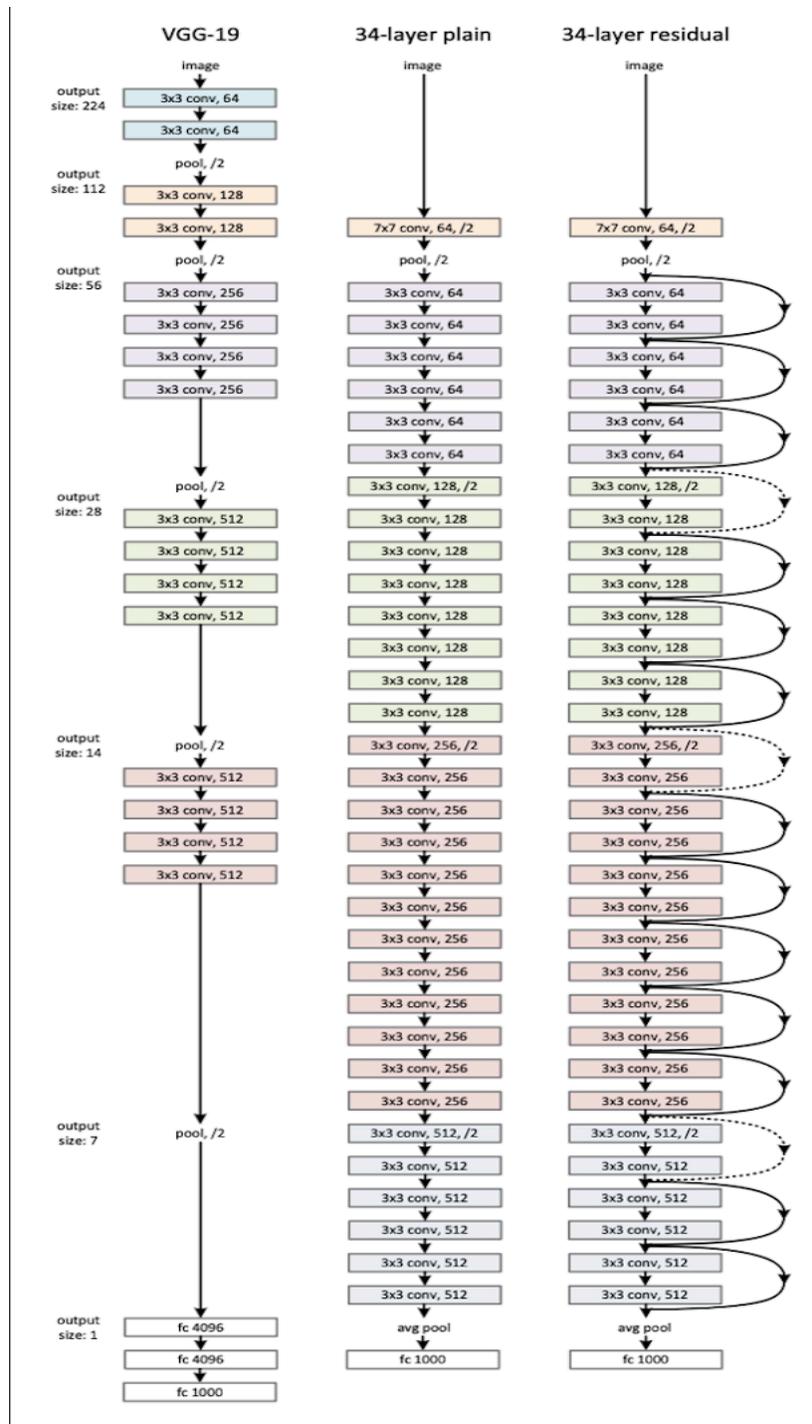


Figure 11 : Architecture de ResNet 34 à partir du papier original. Source (He, Zhang et al. 2016)

La prochaine étape est la normalisation par lots, qui est une opération effectuée sur chaque élément du volume et qui ne change pas sa taille. Ensuite, il y a l'opération de Max Pooling (réduction de dimension) avec une taille de fenêtre de 3x3 et un pas de 2. Il est également

indiqué que ces opérations sont effectuées sur le volume d'entrée pour obtenir le volume final avec les dimensions souhaitées.

La normalisation par lots est une technique de prétraitement des données utilisée en apprentissage automatique pour normaliser les données. Cette technique est utilisée pour accélérer la convergence du modèle et améliorer sa précision.

3.2.3 Le Modèle U-net

Lorsqu'il s'agit de segmentation d'images, en particulier de segmentation sémantique, le modèle U-Net est une sorte de réseau neuronal convolutionnel (CNN) qui est souvent utilisé. Créé en 2015 par Olaf Ronneberger, Philipp Fischer et Thomas Brox (Ronneberger, Fischer et al. 2015), il doit son nom à la forme en U qui le caractérise. Ce modèle, qui se compose de deux éléments distincts appelés chemin de contraction (encodeur) et chemin d'expansion (décodeur), permet de diminuer la résolution spatiale d'une image tout en augmentant simultanément la profondeur des caractéristiques qui ont été extraites. Le chemin d'expansion utilise des couches de convolution transposées pour reconstruire l'image segmentée à l'aide des caractéristiques extraites, tandis que le chemin de contraction combine des couches de convolution et de mise en commun pour extraire des caractéristiques à différentes échelles. En outre, les connexions de saut sont utilisées dans le processus de fusion des caractéristiques situées à différentes échelles, ce qui aboutit finalement à une segmentation précise et complète. En raison de sa capacité à segmenter les images avec des contours précis et à traiter efficacement des objets de différentes tailles, le modèle U-Net a trouvé une large application dans une variété de domaines, y compris la biologie médicale, la vision par ordinateur et la télédétection.

3.2.3.1 Architecture

Le modèle U-Net se caractérise par sa structure unique en forme de U, comme la montre la Figure 12. Cette structure comprend un chemin de contraction (encodeur) et un chemin d'expansion (décodeur) distincte.

Les couches de convolution sont utilisées dans la voie de contraction (partie codeur) afin d'extraire les informations de l'image tout en réduisant progressivement sa résolution

spatiale. Après chaque couche de convolution, une fonction d'activation typique telle que ReLU est appliquée afin d'injecter de la non-linéarité dans le réseau. Afin de réduire la taille d'une image tout en conservant ses caractéristiques fondamentales, des opérations de pooling sont effectuées. À chaque étape, les caractéristiques extraites sont stockées dans des connexions de saut en vue d'une utilisation ultérieure au cours du processus d'expansion.

Dans le chemin d'expansion (le décodeur), le suréchantillonnage des caractéristiques est effectué à l'aide de couches de convolution transposées. Les caractéristiques suréchantillonnées et celles qui ont été sauvegardées dans les connexions sautées du chemin de contraction sont ensuite mélangées. Lorsque des caractéristiques de différentes échelles sont combinées de cette manière, les prédictions qui en résultent sont précises et détaillées.

Afin de produire un effet non linéaire, des fonctions d'activation sont utilisées sur les couches de convolution.

La dernière couche, également appelée couche de sortie de convolution, est utilisée pour construire la couche de segmentation finale. Pour obtenir des probabilités de classe ou des masques de segmentation, il est généralement nécessaire d'appliquer une fonction d'activation appropriée à la sortie.

L'architecture U-Net est bien connue pour ses connexions par saut, qui relient entre elles des caractéristiques de résolutions différentes. Ces connexions permettent d'envoyer des informations contextuelles et spatiales entre le chemin de contraction et le chemin d'expansion afin de capturer les détails précis et de localiser avec exactitude les éléments importants de l'image.

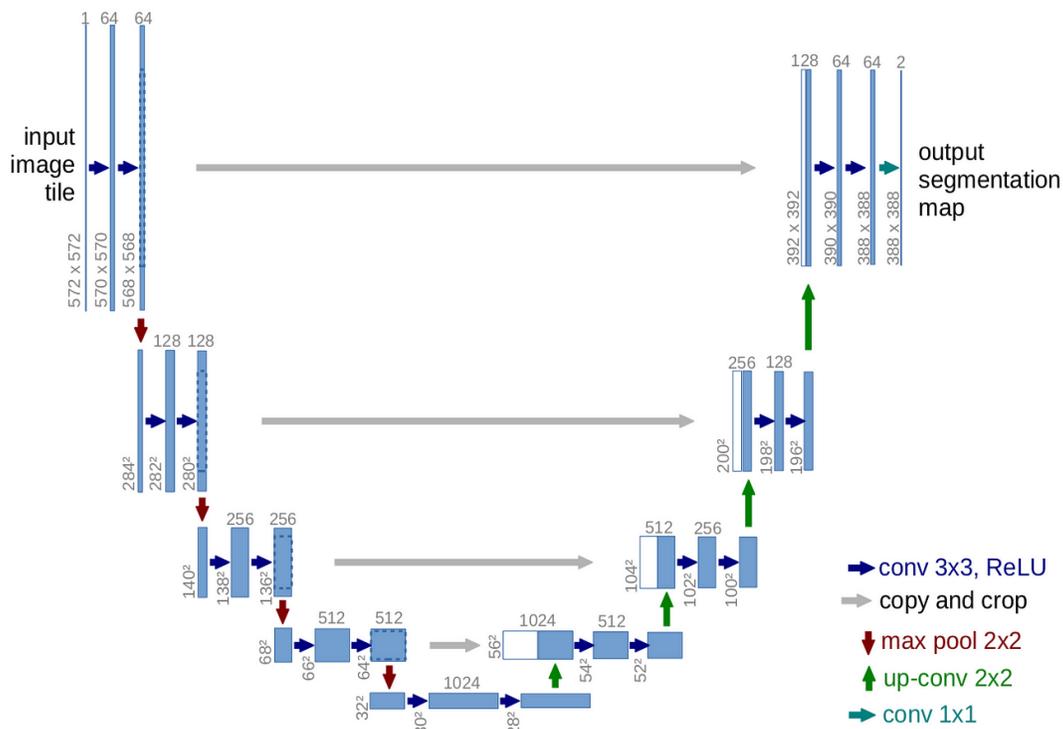


Figure 12 : Architecture du modèle U-net. (Ronneberger, Fischer et al. 2015)

3.2.4 Yolo version 8

Yolo est un système d'identification d'objets qui fonctionne en temps réel et repose sur les principes de l'apprentissage profond, tels que décrits par (Liu, Tao et al. 2018). Les réseaux convolutifs sont utilisés pour la détection d'objets dans les images et les vidéos. Le lancement initial du projet a eu lieu en 2016, comme l'indiquent (Redmon, Divvala et al. 2016). La méthodologie employée par cette approche est fondée sur une technique connue sous le nom de "You Only Look Once" (YOLO), qui permet l'identification simultanée de tous les éléments d'une image par le biais d'une analyse singulière.

Yolov8, développé par l'équipe d'Ultralytics, est une itération améliorée de l'algorithme YOLO et à ce jour il n'y a pas d'article officiel publié. L'itération actuelle du système présente des avancées notables par rapport à ses prédécesseurs, grâce aux améliorations apportées à sa structure sous-jacente et à l'utilisation de procédures optimisées. L'utilisation d'une architecture résiduelle, inspirée de ResNet, permet d'améliorer la formation et la

représentation des attributs complexes des objets. En outre, cette approche intègre de nombreuses techniques, y compris l'utilisation de la normalisation des instances, la détection multi-échelle et l'augmentation des couches de regroupement, afin d'améliorer la précision de la détection. En outre, il convient de noter que Yolov8 est très polyvalent, car il est capable d'exécuter efficacement plusieurs tâches de vision par ordinateur, notamment la classification, la segmentation, la détection et la pose, comme indiqué dans le Tableau 2

Type	Poids pré-entraînés	Tâche
Yolov8	yolov8n.pt, yolov8s.pt, yolov8m.pt, yolov8l.pt, yolov8x.pt	Détection
Yolov8-seg	yolov8n-seg.pt, yolov8s-seg.pt, yolov8m-seg.pt, yolov8l-seg.pt, yolov8x-seg.pt	Segmentation des instances
Yolov8-pose	yolov8n-pose.pt, yolov8s-pose.pt, yolov8m-pose.pt, yolov8l-pose.pt, yolov8x-pose.pt, yolov8x-pose- p6.pt	Poses
Yolov8-cla	yolov8n-cla.pt, yolov8s-cla.pt, yolov8m-cla.pt, yolov8l-cla.pt, yolov8x-cla.pt	Classification

Tableau 2 : YoloV8 (Jocher 2023)

Une autre amélioration qui peut être observée est la capacité d'identification d'objets en temps réel, même sur des systèmes qui possèdent des ressources limitées. Des techniques de compression de modèles sont employées pour maintenir un haut niveau de performance, ce qui implique la réduction du nombre de canaux à l'intérieur des couches.

Le Tableau 3 présente les différentes performances des modèles YOLO testés sur l'ensemble de données COCO.

Modèle	Taille	Carte val	Processeur	A100	Params	FLOPs
YOLOv8n	640	37,3	80,4	0,99	3,2	8,7
YOLOv8s	640	44,9	128,4	1,20	11,2	28,6
YOLOv8m	640	50,2	234,7	1,83	25,9	78,9
YOLOv8l	640	52,9	375,2	2,39	43,7	165,2
YOLOv8x	640	53,9	479,1	3,53	68,2	257,8

Tableau 3 : Performance des modèles. (Jocher 2023)

- **Modèle** : dans cette colonne nous avons les différentes variantes de YOLOv8, allant du modèle le plus petit YOLOv8n (nano) qui est le plus rapide et le plus petit en termes d'entraînement au modèle le plus grand YOLOv8x (Extra large) qui est le plus lent en termes d'entraînement.
- **Taille** signifie la taille des images en pixel qui sont de 640 pixels.
- **Carte de validation** : connu sous le nom de mAP(mean Average Precision 50-95) en anglais en une mesure utilisée pour évaluer les performances des modèles de détection(Hui 2018).
- **Processeur et A100** : signifie la vitesse du processeur et celle du GPU A100 (TensorRT) en millisecondes
- **Params** : signifie le nombre de paramètres de chaque modèle (en milliers ou millions ?).
- **FLOPs** : signifie en anglais (Floating Point Operations) quantifié le nombre d'opérations mathématiques pour effectuer un calcul spécifique.

YOLOv8 est très facile à implémenter. Il suffit d'installer en premier la bibliothèque Ultralytics et en suite spécifier le chemin du fichier de configuration. Ce fichier est au format yaml et il comprend le nombre de classes qu'on utilise pour l'entraînement et les chemins d'accès de nos données d'entraînement et de validation.

Ensuite on choisit le modèle qu'on veut utiliser pour notre cas il s'agit de YOLOv8m(medium) pour l'entraînement de nos données (voir Figure 13).

```
[ ] 1 from ultralytics import YOLO

[ ] 1 ROOT_DIR = '/content/drive/MyDrive/Yolov8/train_yolov8_customdataset/google_colab'

[ ] 1
2 import os
3
4 # Load a model
5 model = YOLO("yolov8m.yaml") # build a new model
6 model = YOLO("yolov8m.pt") # load a pretrained model
7
8 # Use the model
9 results = model.train(data=os.path.join(ROOT_DIR, "google_colab_damage_config.yaml"), epochs=30) # train the model
```

Figure 13: Implémentation de YOLOv8 sur Google Colab.

• Mesures d'évaluation du process de classification et de segmentation

Pour assurer la précision de l'identification de la marque ainsi que la segmentation des pièces du véhicule, il est crucial d'évaluer les performances des modèles de classification et de segmentation utilisés.

Le processus de classification est évalué par les mesures de performances suivantes :

- Précision : Est la fraction de données correctement classifiées par rapport au nombre total d'instances. Cela donne une vue d'ensemble de la performance du modèle.
- Matrice de confusion : C'est le tableau qui présente les vrais positifs, les faux positifs, les vrais négatifs et les faux négatifs pour chaque classe. Cela aide à visualiser où le modèle fait des erreurs.
- Précision : Différent de la première mesure, il s'agit du pourcentage d'exemples classés dans une catégorie donnée qui appartiennent réellement à cette catégorie. Cette mesure est cruciale lorsque les faux positifs peuvent entraîner des coûts importants.
- Rappel : La proportion d'instances réellement appartenant à une classe spécifique qui sont correctement identifiées comme telles. C'est crucial lorsque les faux négatifs sont coûteux.

Pour le processus de segmentation les mesures utilisées sont :

- Précision : La proportion de pixels correctement classifiés par rapport au nombre total de pixels. C'est une mesure globale qui peut ne pas refléter les performances sur des classes spécifiques.

- Rappel : La proportion de pixels d'une classe spécifique qui sont correctement identifiés comme appartenant à cette classe. Elle mesure la capacité du modèle à identifier tous les objets d'une classe donnée.
- Précision : C'est la proportion de pixels classifiés comme appartenant à une classe spécifique qui sont réellement de cette classe. Elle évalue la précision des détections faites par le modèle.
- F-mesure : C'est une moyenne qui combine la précision et le rappel pour donner une évaluation globale de la performance de la segmentation.

Après avoir suivi les étapes méthodologiques que nous avons définies, nous entamons l'étape de l'analyse des données. Cette phase revêt une importance capitale pour la suite de notre travail. Elle sera donc examinée en profondeur dans la prochaine section de ce chapitre, où nous nous pencherons en détail sur les résultats et les conclusions qui découleront de cette analyse.

3.3 Analyse de données

Une collection de 16185 images de véhicules de différentes marques a été obtenue à partir de la base de données publiquement accessible de l'université de Stanford (ensemble de données sur les voitures de Stanford) (Li 2018). Les véhicules ont ensuite été classés, réorganisés et affinés afin d'inclure exclusivement les véhicules qui sont accessibles dans la région géographique de l'Amérique du Nord. Le modèle de reconnaissance d'images ResNet a ensuite été utilisé pour déterminer la marque des automobiles. La détection des pièces de voiture dans la phase alternative nécessite une procédure de formation qui implique l'utilisation d'images distinctement délimitées et annotées. L'importance de cette étape pour garantir la précision de la détection d'images a été soulignée par (Wu et Chen 2015). Le traitement de nos données sera divisé en trois étapes distinctes pour s'adapter aux différentes tâches des modèles utilisés. Dans la première étape, nous nous concentrerons sur la détection des marques de voitures. Cela signifie que lors de la collecte des données, notre principal objectif sera d'identifier la marque de chaque voiture, et nous classerons nos

images en conséquence. Nous devons être très méticuleux dans le choix des images pour cette phase afin de garantir la qualité de notre modèle.

Ensuite, dans la deuxième étape, nous aborderons la segmentation des pièces externes des véhicules. Ici, l'objectif est de découper les différentes parties extérieures de chaque voiture, sans tenir compte de la marque spécifique. Notre priorité sera donc de bien segmenter et identifier ces composantes.

Enfin, la troisième étape sera consacrée à la détection des dommages. Pendant cette phase, nous nous concentrerons sur l'utilisation d'images où les dommages sont visibles pour former notre modèle de détection des dommages. Il y aura également des traitements supplémentaires à effectuer lors de la phase de segmentation et de localisation des dommages.

Ces trois étapes distinctes nous permettront de préparer nos données de manière précise pour chaque aspect de notre approche en s'assurant ainsi son efficacité pour chaque tâche.

3.3.1 Traitement des données pour la reconnaissance des marques de voitures

D'après (Owestfrance-auto), il est répertorié plus de 700 marques de véhicules à l'échelle mondiale. Toutefois, prendre en considération la totalité de ces marques serait impraticable, étant donné les considérables ressources en termes de puissance de calcul et de capacité de stockage requise. Afin de résoudre ce défi, nous avons opté pour un échantillonnage restreint des marques automobiles les plus utilisées en Amérique du Nord pour l'entraînement de notre modèle.

3.3.1.1 Collecte des données pour la reconnaissance des marques de véhicules

La phase initiale du projet revêt une importance significative, car elle implique la création d'une base de données méticuleusement organisée englobant divers constructeurs automobiles. Notre étude s'est concentrée sur l'examen des véhicules accessibles en Amérique du Nord. Les images utilisées dans cette étude ont été obtenues à partir de l'ensemble de données Stanford Cars (li 2018), accessible au public. La base de données comprend une collection complète de 16 185 images de véhicules. Les données ont ensuite fait l'objet d'un processus de filtrage au cours duquel les doublons, ainsi que les images

floues ou de mauvaise qualité, ont été éliminés. Cette étape a été prise en considération en raison de l'impact potentiel que ces facteurs peuvent avoir sur l'entraînement des modèles. Progi (Progi) nous a également fourni une sélection d'images de véhicules provenant de leurs bases de données. La compilation collective de ces images a permis l'établissement d'une base de données complète englobant diverses marques de voitures.

3.3.1.2. Processus de sélection des marques de voitures.

Notre étude est basée sur l'examen des automobiles les plus populaires au Canada au cours des trois dernières années (2020, 2021 et 2022) afin d'identifier les marques automobiles dominantes. Le Tableau 4 présente les marques de véhicules les plus vendus au Canada en 2020 (l'auto), 2021 (Aubaine) et 2022 (aubaine). L'unité représente le nombre de ventes annuel.

2020			2021			2022		
Rang	Marques	Unités	Rang	Marques	Unités	Rang	Marques	Unités
1	Honda Civic	42996	1	Honda Civic	43556	1	Ford série F	114729
2	Toyota Corolla	33181	2	Toyota Corolla	40019	2	Ford Ram	75740
3	Hyundai Elantra	22000	3	Hyundai Elantra	26312	3	Toyota Rav 4	55921
4	Kia Forte	14373	4	Kia Forte	14428	4	Chevrolet Silverado	53281
5	Volkswagen Golf	13112	5	Tesla Model 3	13000	5	GMC Sierra	52318
6	Mazda 3	12769	6	Volkswagen Jetta	12585	6	Toyota Corolla	33096
7	Tesla Model3	11200	7	Toyota Camry	11896	7	Honda CR-V	32096
8	Volkswagen Jetta	10558	8	Mazda 3	11804	8	Honda Civic	29722
9	Toyota Camry	10178	9	Nissan Sentra	10117	9	Hyundai Kona	24579
10	Kia Soul	8144	10	Volkswagen Golf	8589	10	Hyundai Elantra	24599

Tableau 4 : Tableau comparatif des voitures les plus vendues ces trois dernières années au Canada

Nous avons constitué un échantillon de 17 classes(marques) pour la détection des marques de voitures. Parmi celles-ci, 14 correspondent aux marques mentionnées en haut du tableau, tel que Toyota, Mazda, Hyundai, Chrysler, Nissan, Honda, Tesla, Volkswagen, Acura, Chevrolet, Dodge, Jeep, Ford et GMC. Les 3 autres catégories concernent des marques de voitures haut de gamme, à savoir Mercedes, BMW, Audi.

3.3.1.3 Statistiques descriptives

Après avoir choisi les marques de voitures que nous voulons détecter, passons aux statistiques descriptives des données. Le Tableau 5 nous donne des informations sur le nombre d'images dans chaque classe, ainsi que les différentes couleurs dominantes associées à chaque classe. Les couleurs répertoriées dans le tableau sont celles qui sont principalement présentes dans chaque classe, tandis que "autres" représente les couleurs moins fréquentes.

Classes	Nd d'images	Gris	Bleu	Rouge	Noir	Blanc	Autres
Acura	128	50	21	17	24	16	0
Audi	121	26	15	12	39	20	9
BMW	128	21	4	42	8	30	23
Chevrolet	119	40	3	14	16	25	21
Chrysler	122	39	7	11	26	12	32
Dodge	121	24	15	24	33	15	10
Ford	121	24	12	29	20	22	12
GMC	125	9	5	25	27	39	20
Honda	124	40	20	10	17	13	24
Hyundai	125	37	12	10	19	19	28
Jeep	116	20	6	15	32	17	26
Mazda	122	19	17	29	13	32	12
Mercedes	116	44	3	16	26	17	20
Nissan	105	12	14	25	17	21	16
Tesla	119	19	10	17	23	40	10
Toyota	121	28	8	32	15	27	11
Volkswagen	108	26	13	21	22	26	0

Tableau 5: Nombre d'images et couleurs dominant sur chaque classe

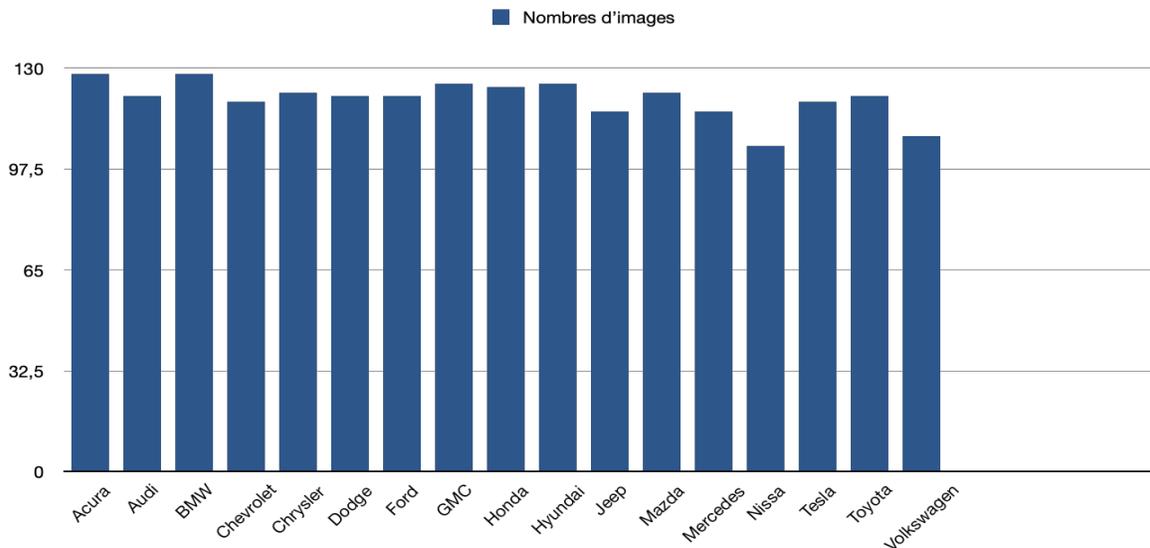


Figure 14 : Graphique du nombre d'images en fonction de la classe

Nos données d'entraînement comprennent un total de 2041 images, comme indiqué dans le

Tableau 5. L'histogramme présenté dans la Figure 14 représente la distribution du nombre d'images par classe, mettant en évidence la différence quantitative entre les différentes classes. La Figure 15 nous présente les pourcentages pour chaque couleur dans notre base de données. Elle nous donne également des indications sur la manière d'équilibrer notre base de données afin qu'elle soit plus homogène.

Notre intention est de poursuivre le processus de création d'ensembles de données afin d'acquérir une base de données bien équilibrée qui englobe tous les modèles de véhicules, y compris les différentes marques et années. Cette base de données sera conçue pour accueillir les futurs modèles, ce qui améliorera la précision de la reconnaissance de la marque et du modèle de voiture.

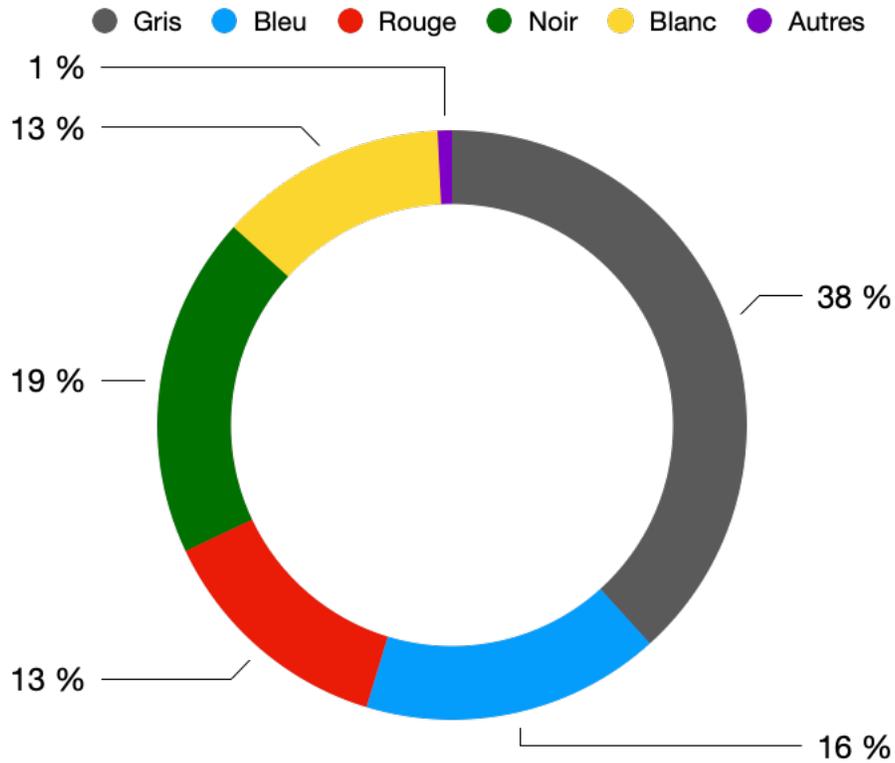


Figure 15 : Pourcentage des différentes couleurs dans notre base de données

Le Tableau 6 montre les statistiques descriptives de notre base de données

Nb. classes	Total	Moyenne	Min	Max	Écart type
17	2041	120,06	105	128	6,17

Tableau 6 : Statistiques descriptives pour la détection des marques

Nombre de Classes: Il y a un total de 17 classes dans l'ensemble de données.

Total: Le nombre total d'instances est de 2041. Cela représente la somme de toutes les images d'entraînement pour les 17 classes.

Moyenne: La moyenne du nombre de véhicules/marque est d'environ 120,058. Cela signifie que, en moyenne, chaque classe a une valeur d'environ 120,058.

Min: Le nombre minimum de véhicules/marque est de 105.

Max: Le nombre maximum de véhicules/marque est de 128. Cela indique que la classe ayant la valeur la plus élevée a une valeur de 128.

Écart Type: L'écart type est d'environ 6,169. Il mesure la dispersion ou la variabilité des valeurs par rapport à la moyenne. Plus l'écart type est élevé, plus les valeurs sont dispersées autour de la moyenne.

Ces statistiques donnent la répartition du nombre de véhicule de chacune des 17 classes avec une moyenne de 120,06. L'écart type, de 6,17 indique qu'il existe une certaine variation entre les valeurs des différentes classes, bien que cette variation soit relativement faible. Les valeurs minimales et maximales (105 et 128) permettent d'avoir une idée des plages dans lesquelles se situent nos données pour ces classes.

3.3.2 Préparation des données pour la segmentation des pièces externes de la voiture

Dans cette étape, nous procédons au traitement de la segmentation d'image, qui diffère du traitement précédent, car notre attention se porte désormais sur les composants externes du véhicule. Contrairement à la couleur ou à la marque, qui n'ont pas d'incidence sur notre ensemble de données, la position revêt une grande importance, car il est essentiel d'examiner tous les angles de la voiture.

3.3.2.1 Collecte des données pour la segmentation des pièces externes de la voiture

Dans cette étape, nous avons extrait une partie de nos données à partir de la base de données de Stanford (Li 2018). Afin d'assurer la qualité et la pertinence de notre ensemble de données, nous avons mis en place un processus de filtrage des données semblable à celui effectué dans la section 3.3.1.1. En outre, afin d'enrichir notre ensemble de données, nous avons également téléchargé gratuitement certaines images depuis le site Shutterstock (Shutterstock 2023), qui propose une vaste collection de plus de 734.000.000 images en 3D. L'avantage principal de ces images en 3D réside dans l'absence d'arrière-plan, ce qui nous offre une plus grande variété d'images non bruitées dans notre ensemble de données. Cette diversité permettra à notre modèle de s'adapter à diverses situations de détection. Nous avons utilisé 14 classes de composantes (pièces) pour la segmentation. Ces classes sont représentées dans le Tableau 7

Classes
Roues
Phares_Av
Phares_Arr
Roues
Portes_Av
Porte_Av
Portes_Arr
Parechoc_Avant
Parechoc_Arrier
Phares_Arr
Parebrise_Arr
PareBrise_Av
Ailes_Av
Ailes_Arri

Tableau 7 : Classes pour la segmentation

3.3.2.2 Statistiques descriptives

Après la collecte des données, nous allons effectuer les statistiques descriptives des données pour la segmentation.

Le Tableau 8 présente la répartition du nombre d'images pour chaque couleur dans notre jeu de données destiné à la segmentation d'images.

Couleurs	Rouge	Bleu	Blanc	Noir	Gris	Autres
Nombre d'images	40	56	80	54	93	27

Tableau 8 : Nombre d'images de chaque couleur pour la segmentation

La Figure 16 montre le pourcentage que représentent les différentes couleurs dans notre base de données pour la segmentation. Et on remarque que la couleur gris et blanc sont les plus représentés avec respectivement 27% et 23% de notre base donnée et ensuite nous avons la couleur noire avec 15% et la couleur bleue représente 16%. La couleur rouge représente 11% des images de notre base de données. Les autres couleurs qui ne sont pas

spécifiquement énumérées représentent 8%. Ces pourcentages nous donnent un aperçu de la représentation des couleurs dans notre base de données.

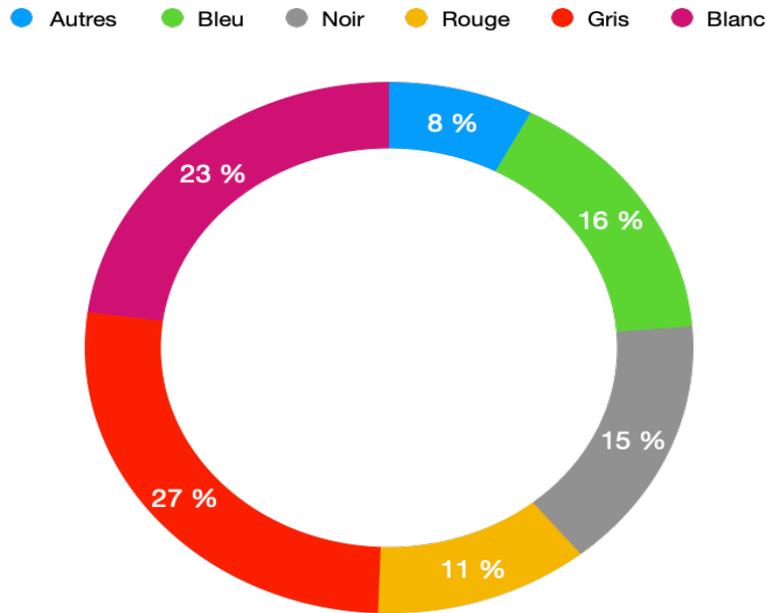


Figure 16 : Pourcentage de représentation de chaque couleur dans nos données de segmentation

Dans le Tableau 9 nous avons les statistiques descriptives des différentes couleurs représentatives dans notre ensemble de données de segmentation.

Nb. de couleurs de voitures dans notre base de données	Somme	Moy	Min	Max	Écart type
6	350	58,33	27	93	24,54

Tableau 9 : Statistique descriptive pour la segmentation

Nb. de couleurs de voitures dans notre base de données: Représente les couleurs disponibles dans notre base de données. Elles sont illustrées dans la Figure 16

Somme: La somme totale de toutes les valeurs dans cet ensemble de données est de 350, représentant ainsi le nombre total d'images d'entraînement.

Moy: La moyenne des valeurs est d'environ 58,33,

Min: La valeur minimale parmi toutes les couleurs est de 27, indiquant la couleur contenant le moins d'images.

Max: La valeur maximale parmi toutes les couleurs est de 93, révélant que la couleur la plus représentée dans l'ensemble de données compte 93 images qui correspond au gris.

Écart Type: L'écart type est d'environ 24,54.

Ces statistiques descriptives nous donnent une idée de la répartition des couleurs au sein de notre ensemble de données de segmentation. Elles sont essentielles pour détecter d'éventuelles erreurs de prédiction et pour ajuster nos données en conséquence, afin d'améliorer la performance de nos modèles.

Après cette étape nous allons passer à l'annotation des données.

Pour résumer, le schéma présenté à la Figure 17 montre un aperçu du processus de traitement des données pour la segmentation.

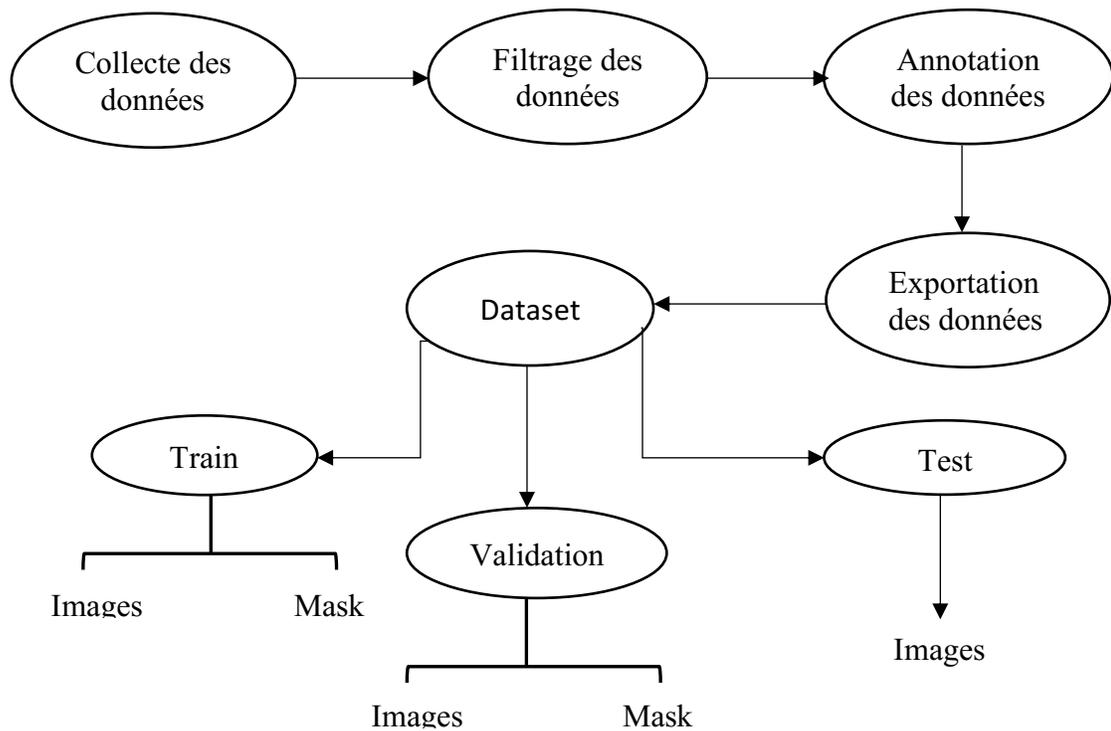


Figure 17 : Processus de segmentation d'images

3.3.3 Distribution des données pour la détection des dommages

Dans cette section, nous procédons à un traitement de données distinct pour la détection des dommages sur les véhicules. Notre attention portera sur les parties endommagées des voitures. Dans cette démarche, la couleur et la marque de l'automobile ne sont pas des facteurs importants pour la collecte de données. En revanche, la position est d'une grande importance, car elle nous permettra ultérieurement de localiser avec précision les dommages sur le véhicule.

3.3.3.1 Collecte des données pour la détection des dommages

Comme mentionné précédemment, chaque nouvelle étape du processus de traitement des données nécessite une nouvelle collecte de données, car les tâches à accomplir sont différentes. Pour obtenir des images de véhicules endommagés, nous avons fait appel à Progi (Progi), qui dispose d'une vaste base de données plus de 500 images de véhicules accidentés. Nous avons effectué quelques étapes de prétraitement pour assainir la collection de données. Nous avons, dans un premier temps, éliminé les images de mauvaise qualité ainsi que les doublons. Ensuite, nous avons écarté les images où les dommages étaient difficilement identifiables dû à l'angle ou à la qualité de la photo. Finalement, nous avons exclu les images montrant des véhicules complètement détruits où aucune partie distinctive (roue, porte, rétroviseur, etc.) n'était plus visible à l'œil nu. Après ce prétraitement, notre base de données ne compte désormais plus que 226 images de véhicules endommagés. Parmi ces images, 80 présentent des dommages à l'avant, 75 sur le côté et 71 à l'arrière. Avec cette collecte achevée, nous pouvons passer à l'étape suivante d'annotation de nos images.

3.4 Autres Outils.

Pour faire de l'apprentissage automatique, il est nécessaire d'avoir des ressources matérielles significatives. Pour notre part, étant donné que nos données se composent principalement d'images, cela requiert des ressources matérielles considérables, rendant indispensable l'utilisation d'une machine de grande puissance pour mener à bien nos tests.

Pour notre environnement de développement, nous avons opté pour Google Colaboratory, utilisé le langage Python pour la programmation, stocké nos données sur Google Drive, et fait usage des outils d'annotation comme CVAT et Apeer pour annoter nos images.

3.4.1 Google Colaboratory

Google Colaboratory (Research), souvent abrégé en "Colab," est un produit de Google Research qui offre la possibilité d'écrire et d'exécuter du code Python directement à partir de son navigateur web. Cet environnement est particulièrement bien adapté pour l'apprentissage machine ainsi que l'analyse de données. Colab fonctionne comme un service de notebooks Jupyter hébergé qui se distingue par sa simplicité d'utilisation, ne requérant aucune configuration préalable, et il offre un accès gratuit à des ressources informatiques, notamment des GPU. Il dispose de plusieurs bibliothèques intégrées telles que Numpy, Pandas, Keras, TensorFlow, et bien d'autres. De plus, il permet d'installer facilement des bibliothèques supplémentaires à l'aide de la commande *pip*. Un autre avantage majeur est sa compatibilité avec Google Drive, simplifiant ainsi le stockage et le partage des données. En cas de besoin de ressources supplémentaires, il existe des options d'abonnement payantes offrant des capacités étendues. Dans l'ensemble, cette plateforme a joué un rôle essentiel dans nos projets de recherche en fournissant les ressources et l'environnement nécessaires.

3.4.2 Python

Python (Python) est un langage de programmation open source créé en 1991 par Guido Van Rossum (Van Rossum 2007). Il est très utilisé dans le domaine de la science des données, l'apprentissage machine, ainsi que le développement des logiciels. Comme l'apprentissage automatique nécessite une gestion continue des données, l'avantage d'utiliser Python est qu'il possède une vaste bibliothèque standard qui permet de traiter et de gérer facilement les données. De plus sa syntaxe est facile à comprendre et c'est un langage multiparadigmes (Van Rossum 2007). Il possède également des packages intégrés fournissant un code de base, ce qui veut dire qu'on n'aura pas besoin d'écrire tout le code à partir de zéro, ce qui accélère le développement et l'analyse des données.

3.4.3 Google drive

Google drive (Google) est un service de stockage en ligne créé par Google en 2012. Dans notre cas il nous permet de stocker et partager nos fichiers. Il nous permet aussi d'accéder à nos fichiers depuis n'importe quel appareil connecté à internet. Il propose également un espace gratuit de 15 Go et si vous dépassez cette limite il est possible d'en acheter jusqu'à 30 To.

3.4.4 Apeer

Apeer (Apeer) est une plateforme pour le traitement d'images et la vision par ordinateur. Il offre des compétences sophistiquées applicables à la segmentation d'images et à toute une série d'autres activités d'analyse d'images. Il nous permet de charger, de prétraiter, d'annoter et d'analyser des ensembles de données d'images en utilisant soit des modèles préentraînés, soit des modèles qu'ils ont eux-mêmes construits, ainsi que les deux options simultanément.

3.4.5 CVAT

CVAT (Cvat) est un logiciel d'annotation d'images et de vidéo open source généralement utilisé pour étiqueter les données des algorithmes de vision par ordinateur. Il prend en compte plusieurs formats de données lors de l'importation et l'exportation et est facile à utiliser, car elle ne nécessite aucune configuration préalable.

3.5 Conclusion

Nous avons exposé dans ce chapitre la méthodologie que nous avons employée pour aborder notre problématique. Nous avons exploité divers modèles et techniques pour traiter des données complexes et hétérogènes, en nous concentrant sur des aspects spécifiques des véhicules, qu'il s'agisse de leur identification, de la localisation de leurs composants externes ou de la détection des dommages. Ces approches et modèles ont été choisis en

fonction des besoins de chaque tâche et ont été soutenus par des traitements de données soignés.

Le prochain chapitre dévoilera les résultats que nous avons obtenus, ce qui nous permettra d'évaluer l'efficacité de nos approches et de nos modèles.

Chapitre 4 : Résultats et discussions

4.1 Introduction

Au cours de ce chapitre, nous vous dévoilerons les résultats que nous avons obtenus à la suite de l'application des divers modèles que nous avons préalablement décrits dans le chapitre 3. Ces modèles ont été employés pour résoudre différentes tâches, à savoir la reconnaissance des marques de véhicules, la segmentation d'images, ainsi que la détection des pièces endommagées sur ces véhicules. Nous allons commencer par présenter de manière détaillée les résultats obtenus dans le cadre de chaque tâche. Cela inclura des informations sur la précision, la performance, et la fiabilité de nos modèles dans leur application respective. Ensuite, nous procéderons à une interprétation approfondie de ces résultats. Nous tenterons de comprendre les raisons derrière les performances de nos modèles, en analysant les aspects positifs et les limitations de chacun d'entre eux. Cette évaluation nous aidera à identifier les points forts et les faiblesses de nos méthodes. Enfin, nous discuterons des pistes d'amélioration possibles pour nos modèles, ainsi que des perspectives de notre recherche. Il sera question de l'optimisation de nos approches existantes et de l'exploration de nouvelles méthodes.

4.2 Reconnaissance des marques de véhicules.

4.2.1 Performance des modèles

Pour identifier les marques de véhicules, nous avons exploité trois modèles de la famille ResNet : ResNet50, ResNet101 et ResNet152. Ces modèles ont été entraînés sur un jeu de données contenant 17 classes distinctes correspondant à différentes marques de véhicules. La principale différence entre ces trois modèles ResNet réside dans leur profondeur, déterminée par le nombre de couches qu'ils comportent. Ainsi, ResNet152 est le plus profond des trois, avec un nombre de couches supérieur à celui de ResNet50 et ResNet101. Cette différence de profondeur peut influencer les performances des modèles dans la tâche de classification des marques de véhicules

Le Tableau 10 montre les performances des modèles après entraînement avec les différentes métriques.

Performance de la classification des marques de véhicules			
Métriques	Précision (%)	Recall	F1-score
ResNet 50	79,23	0,95	0,91
ResNet 101	83,18	0,92	0,89
ResNet 152	89,27	0,90	0,87

Tableau 10 : Performance de la classification des marques de véhicules

Dans la Figure 18 on constate qu’au fur et à mesure que les itérations avancent, la perte (train loss) diminue. Cela signifie que le modèle s’améliore et s’adapte mieux aux données d’entraînement. La précision (accuracy) augmente également (Figure 19) ce qui signifie que le modèle fait des prédictions de plus en plus précises sur les données d’entraînement.

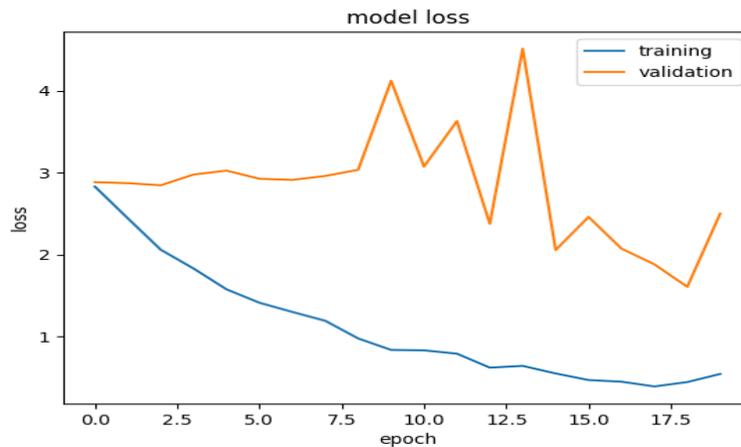


Figure 18 : Évolution de la fonction perte (apprentissage)

Cependant le modèle a du mal avec les données de validation, ce qui peut être signe de surapprentissage. Cela signifie que le modèle s’améliore sur les données d’entraînement, mais a du mal avec de nouvelles données (données de validation) comme la montre aussi la Figure 18 et 19

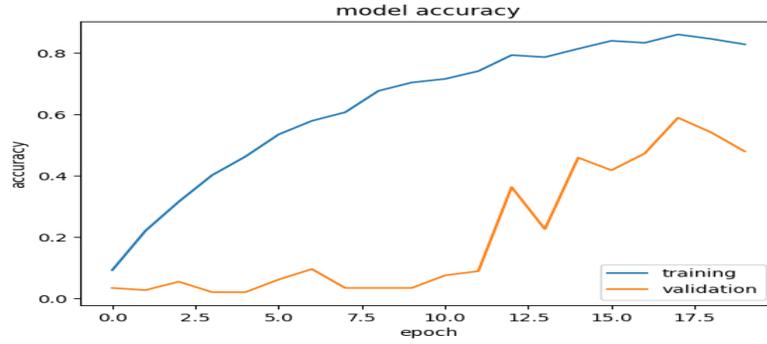


Figure 19 : Évolution de l'exactitude, apprentissage

La Figure 20 représente la matrice de confusion du modèle après avoir effectué des prédictions sur nos données de tests. Elle nous permet d'évaluer la performance de notre modèle de classification en comparant les prédictions du modèle avec les valeurs réelles.

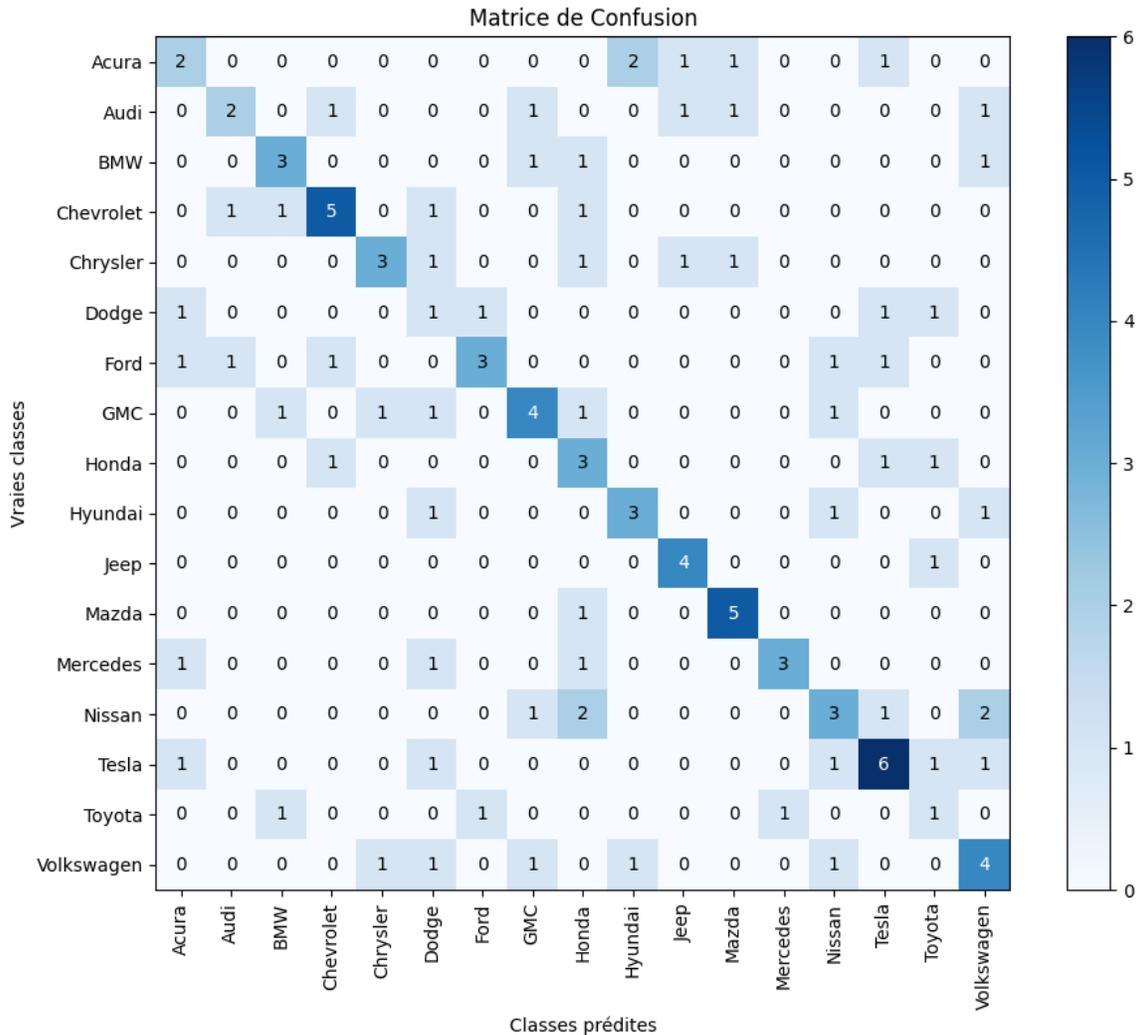


Figure 20 : Matrice de confusion

Pour les résultats nous avons :

Pour les marques Acura et Audi, le modèle a correctement prédit 2 véhicules sur 6. Les marques BMW, Chrysler, Ford, Honda, Hyundai et Mercedes ont obtenu un résultat de 3 véhicules correctement identifiés sur 6. Les marques Volkswagen, GMC et Jeep ont chacune obtenu un résultat de 4 véhicules correctement prédits sur 6. La Chevrolet et la Mazda ont eu un meilleur taux de réussite que les précédents avec 5 véhicules correctement classifiés sur 6. La Tesla a été parfaitement classifiée, avec un taux de réussite de 6 sur 6. En revanche, les marques Dodge et Toyota ont eu une performance faible, avec seulement 1 véhicule correctement identifié sur 6.

4.2.2 Résultats des prédictions

Dans cette partie nous avons toutes les différents résultats sur la prédiction de nos véhicules.

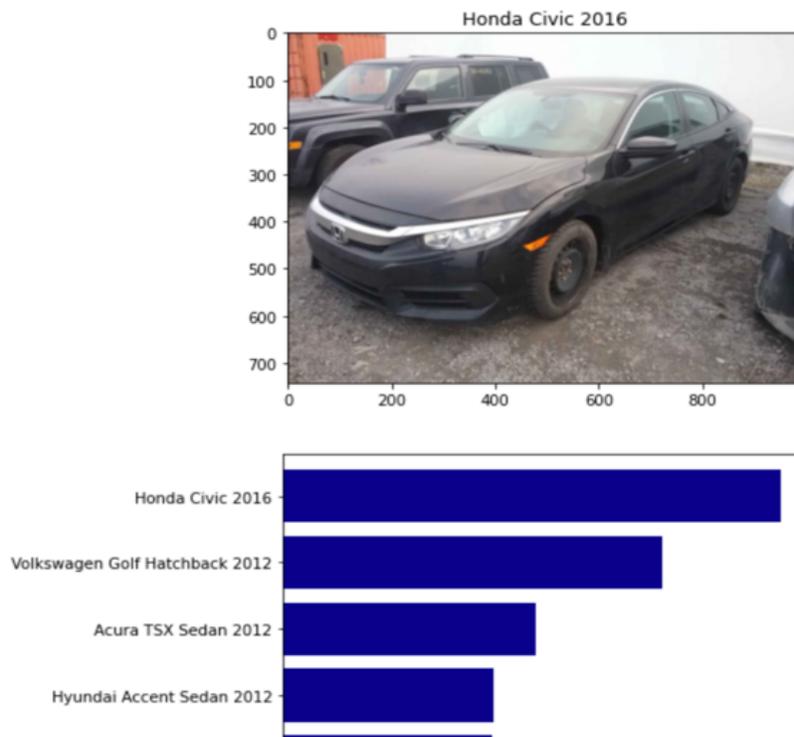


Figure 21 : Prédiction sur une Honda côté avant

Figure 21 et 22, montrent quelques résultats de prédiction. Nous constatons que notre modèle prédit très bien une Honda Civic que ce soit de devant ou derrière. Ceci peut être dû au fait que les Honda Civic pourraient avoir des caractéristiques plus distinctives et homogènes dans le jeu de données, ce qui facilite leur classification par le modèle par rapport aux autres marques de véhicules.

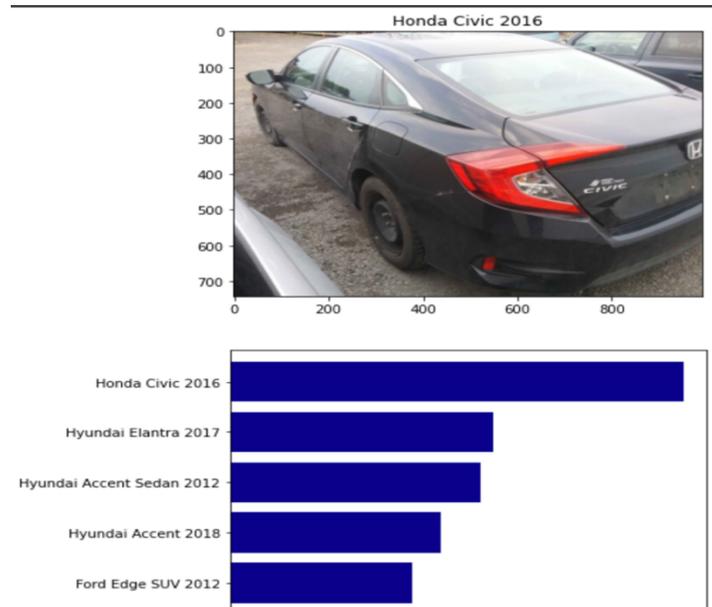


Figure 22 : Prédiction sur une Honda sur le côté arrière

Pour la Figure 23 le modèle fait également une bonne prédiction sur le véhicule.

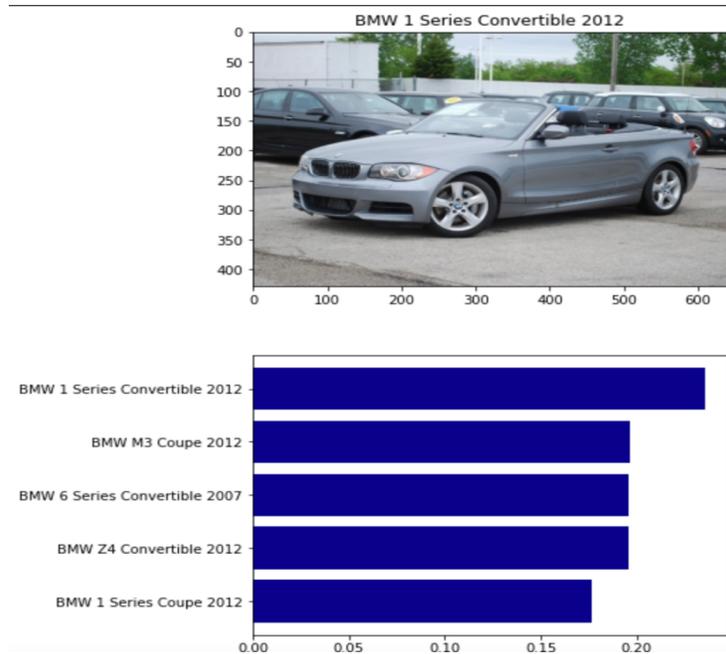


Figure 23 : Prédiction sur la BMW

Dans la Figure 24, notre modèle fait une mauvaise prédiction en attribuant une Volkswagen Golf à une Honda Civic. Cela peut être dû à des facteurs tels que la couleur de la voiture ou l'arrière-plan, auxquels le modèle est sensible. De plus, nous avons également constaté que le modèle place la Honda Civic en dernière position.

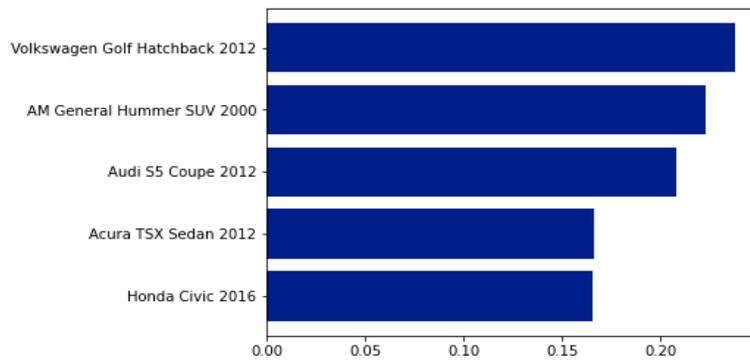
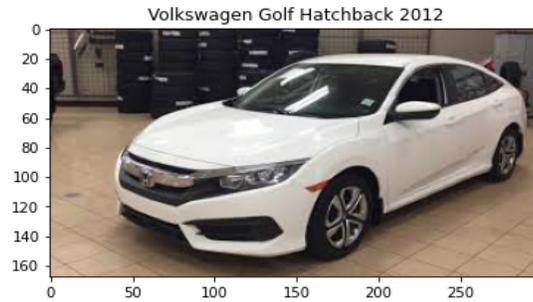


Figure 24 : Mauvaise prédiction sur une Honda

Dans la Figure 25 le modèle fait une bonne prédiction sur une Hyundai Elantra accidentée.

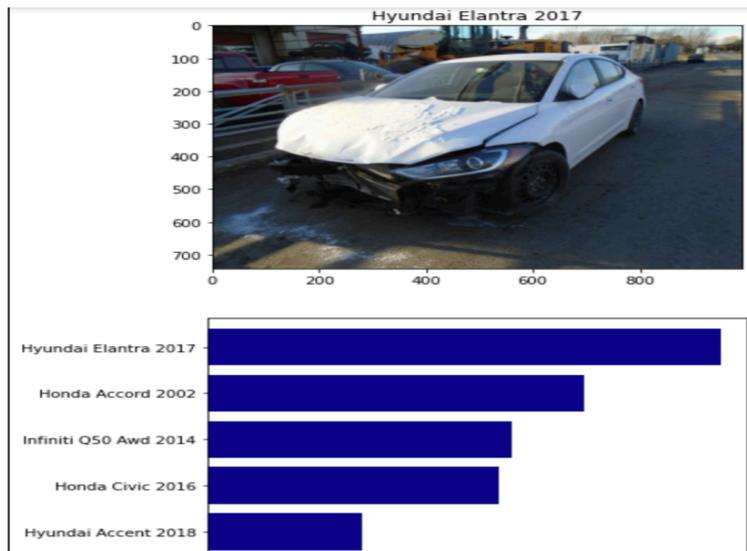


Figure 25 : Prédiction sur une Hyundai Elantra accidentée

Parfois, le modèle de classification peut présenter des difficultés lors de la prédiction en se basant uniquement sur la forme de la voiture. Cela peut être dû à la similarité de certains modèles de voitures en termes de carrosserie, ce qui rend difficile la distinction entre eux. C'est l'un des problèmes rencontrés dans ce domaine.

Ce problème montre les limitations potentielles des algorithmes d'apprentissage profond, notamment en raison de la similarité visuelle entre certains modèles ou marques de voitures (Figure 26) et de la difficulté à distinguer les années spécifiques d'un modèle. Des approches supplémentaires, telles que l'utilisation de données supplémentaires ou de techniques spécifiques de prétraitement des images, peuvent être nécessaires pour surmonter ces défis et améliorer la précision des prédictions.



Figure 26 : Ressemblance entre deux véhicules de marques différentes

Afin de résoudre ce problème, nous avons employé une méthodologie dans laquelle nous utilisons un ensemble de quatre photos distinctes de véhicules (Figure 27) prises sous différentes perspectives, qui servent d'entrées à notre modèle. Au lieu d'utiliser une seule image comme base pour la prédiction finale, nous générons des prédictions individuelles pour chaque image. Par la suite, les prédictions susmentionnées sont fusionnées afin d'obtenir une prédiction complète et concluante.

Cette méthodologie permet d'examiner les différents composants de la voiture à l'aide de nombreuses images, ce qui permet de comprendre le véhicule dans sa globalité et dans ses moindres détails.

En utilisant cette méthode, nous donnons à notre réseau la capacité de prendre en compte toutes les informations disponibles avant de faire une prédiction finale. Cela nous permet de mieux traiter les variations visuelles et les angles spécifiques qui peuvent influencer la détection des pièces externes d'un véhicule.

```
↳ (744, 992, 3)  
Text(0.5, 1.0, 'Honda Civic 2016')
```



```
[ ] 1 print(predicted_label)
```

```
Honda Civic 2016
```

Figure 27 : Prédiction sur quatre angles d'une Honda Civic

4.3 Résultats de la segmentation d'image pour les pièces externes du véhicule.

Dans cette phase de notre étude, nous avons procédé à la segmentation des images dans le but d'identifier et de localiser les différents composants de la voiture. Pour atteindre cet objectif, nous avons utilisé le modèle U-net, dont l'efficacité dans les tâches de segmentation est largement reconnue. De plus, nous avons exploré une approche plus élaborée en intégrant le modèle U-net au modèle ResNet. La combinaison de ces deux modèles a permis de modifier l'architecture de l'encodeur du réseau neuronal qui est responsable de l'extraction des informations visuelles. L'objectif principal était d'évaluer si cette modification améliorerait la précision de la segmentation. Grâce à cette analyse comparative, nous avons pu évaluer l'efficacité de l'approche conventionnelle (utilisant uniquement U-net) par rapport à l'approche améliorée (U-net combiné à ResNet), ce qui a permis d'obtenir des résultats plus complets.

4.3.1 Segmentation avec le modèle U-net

4.3.1.1 Performance du modèle

Dans nos premiers tests avec le modèle U-net nous avons utilisé 14 classes mentionnées dans le chapitre 3, comme élément de carrosserie externe à identifier. Et nous constatons un taux de perte relativement bas avec 14 epochs (Figure 28).

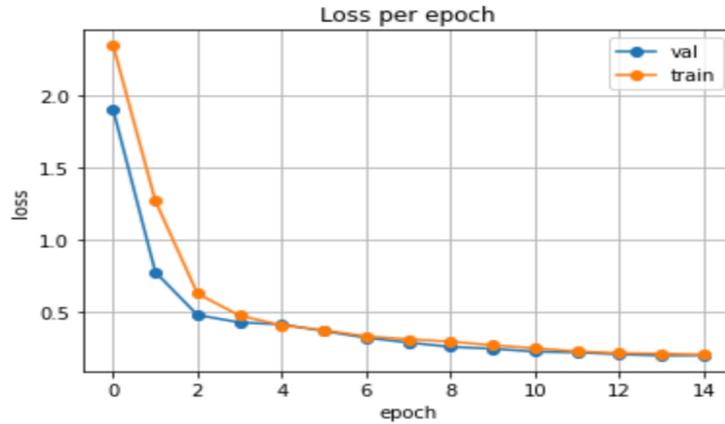


Figure 28 : Courbe de perte du modèle Unet

Nous constatons aussi que notre taux de précision est relativement élevé (Figure 29)

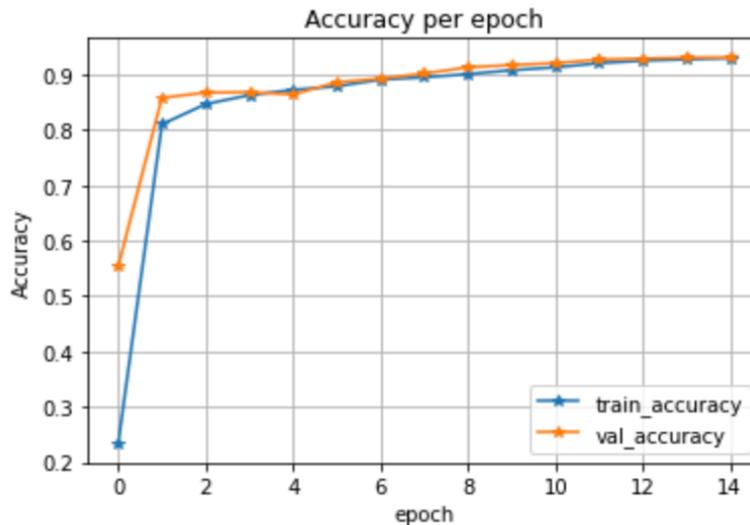


Figure 29 : Courbe de précision du modèle Unet

Maintenant, voyons voir les résultats des prédictions sur différents véhicules.

4.3.1.2 Résultats

Lorsque nous analysons les prédictions générées par notre modèle (voir Figure 30 et 31), nous constatons que certaines classes, telles que les roues, les phares arrières et le pare-brise arrière, sont bien détectées dans la Figure 30 et pareille pour le capot, les phares avant et le parebrise avant sont aussi bien représentés dans la Figure 31. Cela dit, nous avons

constaté que certaines parties ne sont pas représentées avec précision en termes de forme lors de leur prédiction comme les portes. La façon dont nous avons étiqueté les portes est différente du résultat de la prédiction comme le montre la Figure 30 et 31

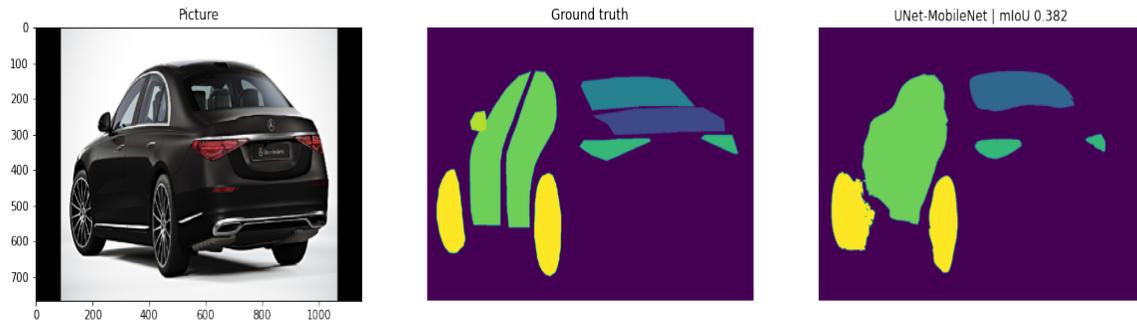


Figure 30 : Première prédiction avec le modèle Unet

L'identification des petites pièces, comme les rétroviseurs, est un autre défi auquel nous avons été confrontés. Le modèle a du mal à les détecter avec précision, ce qui peut entraîner des imprécisions ou des omissions dans les prédictions liées aux rétroviseurs (voir Figure 30 et 31). Certaines pièces n'ont pas été détectées comme le pare-choc avant et arrière ce qui est un problème.

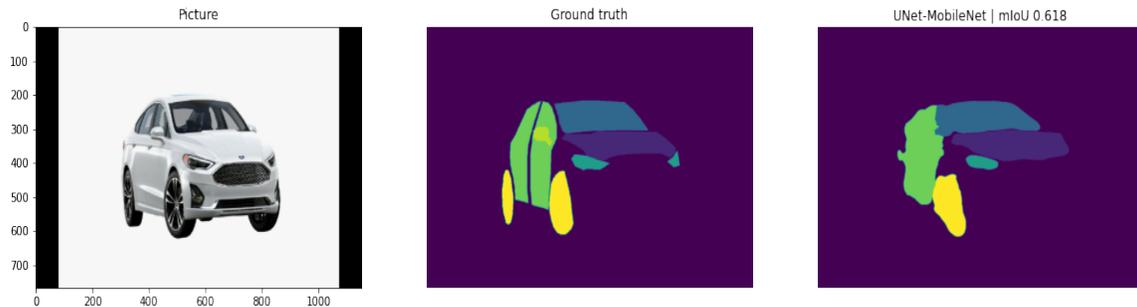


Figure 31 : Deuxième prédiction avec le modèle Unet

Ces questions mettent en évidence les difficultés que nous rencontrons pour détecter efficacement les nombreux composants situés à l'extérieur d'un véhicule. Afin de résoudre ces problèmes particuliers et d'améliorer la précision de la prédiction des pièces extérieures dans son ensemble, nous allons devoir continuer à travailler pour améliorer notre modèle en modifiant les annotations et en trouvant les valeurs optimales pour les paramètres.

4.3.2 Segmentation avec le modèle Unet-ResNet

Afin de résoudre les problèmes rencontrés dans la partie précédente, nous avons utilisé une combinaison des modèles Unet et ResNet pour améliorer les résultats des prédictions. Plus précisément, dans ce modèle, nous avons remplacé la partie encodeur (également appelée "backbone") par le modèle ResNet. Cette substitution nous permet de tirer pleinement parti des avantages offerts par ResNet.

Voyons les performances du modèle.

4.3.2.1 Performance du modèle

La Figure 32 et 33 montrent la courbe de perte et de précision après entraînement. Et on constate un taux de précision élevé et un taux de perte relativement faible.

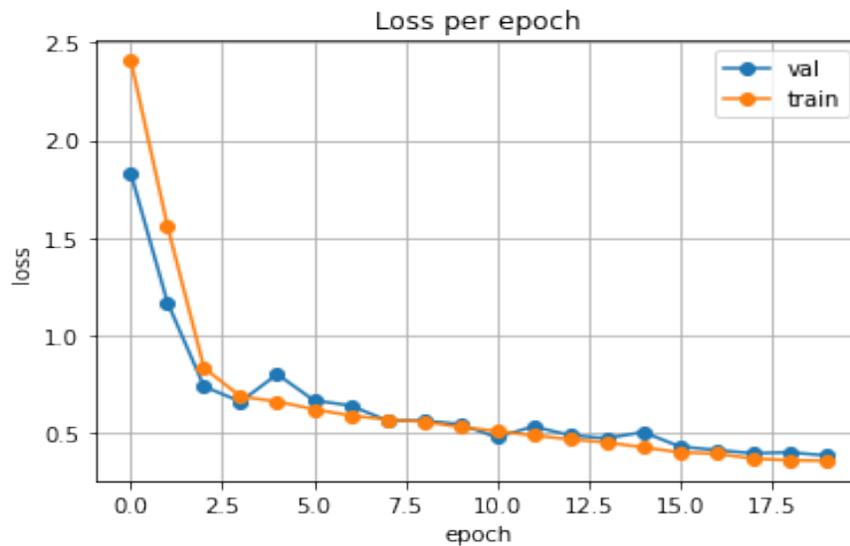


Figure 32 : Courbe de perte du modèle Unet-ResNet

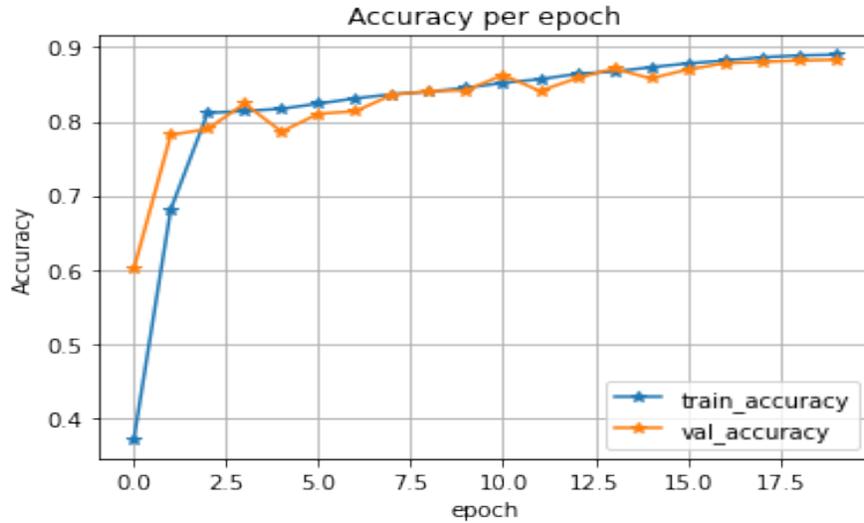


Figure 33 : Courbe de précision du modèle Unet-ResNet

4.3.2.2 Résultats

Bien que nous ayons un taux de perte faible et un taux de perte élevé, le modèle éprouve toujours des difficultés à faire des prédictions précises sur certaines pièces externes de la voiture (Figure 34).

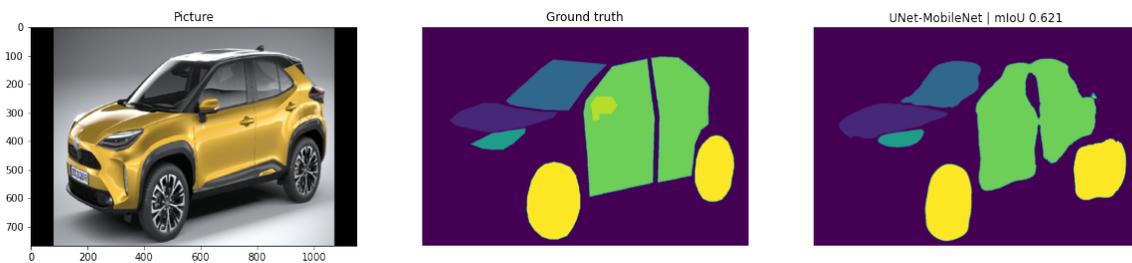


Figure 34 : Prédiction avec le modèle Unet-ResNet

Par rapport au modèle Unet, nous avons remarqué qu'il a tendance à produire des prédictions moins précises à mesure que le nombre de classes dans l'ensemble de données augmente. Cela souligne les défis liés à la complexité et à la précision de la tâche de détection des pièces externes ainsi que les limites du modèle dans l'exécution cette tâche.

Afin de résoudre ces problèmes, nous avons eu comme idée de tester avec le modèle Yolov8 la détection de pièces externes et ensuite passer à la détection des dommages sur le véhicule.

4.4 Détection des pièces externes avec le modèle Yolov8

Dans cette partie nous avons décidé de tester Yolov8 pour la détection des pièces externes afin de voir si nous obtenons de meilleurs résultats par rapport au modèle Unet.

Voyons les performances du modèle

4.4.1. Performance du modèle

La Figure 35 fournit les résultats d'entraînement du modèle Yolov8 sur la détection des pièces externes.

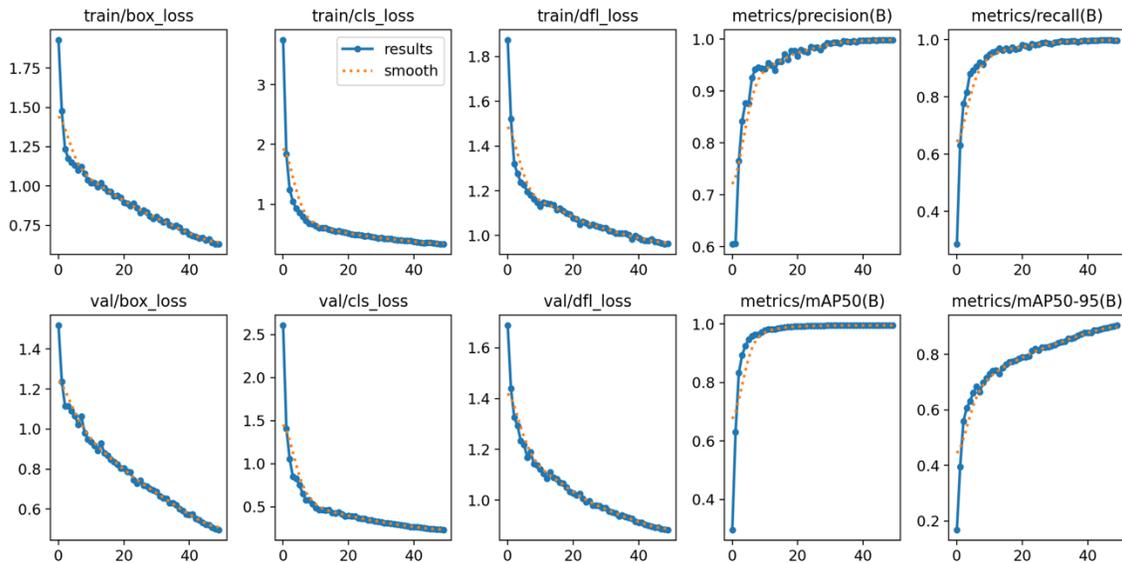


Figure 35 : Courbes de précision et de perte

Les courbes « train/box_loss », « train/cls_loss », « train/df_l_loss » sont les valeurs de pertes sur l'ensemble d'entraînement. Elles mesurent l'erreur sur les bounding box « train/box_loss », les classes « train/cls_loss » et les caractéristiques de détection « train/df_l_loss ». Elles diminuent régulièrement au fil des époques. Les valeurs de perte

sur l'ensemble de validation (« val/box_loss », « val/cls_loss », « val/df_l_loss ») montrent une tendance similaire, ce qui indique que le modèle ne semble pas souffrir d'un surapprentissage (« overfitting »), car il généralise bien aux données de validation.

Les métriques d'évaluation telles que la précision (« metrics/précision(B) ») et le rappel (« metrics/recall(B) ») semblent s'améliorer au fur et à mesure de l'entraînement, ce qui est un signe positif.

Les valeurs de précision moyenne (« metrics/mAP50(B) », « metrics/mAP50-95(B) ») montrent également une amélioration constante, ce qui signifie que le modèle devient meilleur pour détecter des objets dans les images.

La Figure 36 nous montre la matrice de confusion du modèle sur la détection des différentes pièces de la voiture.

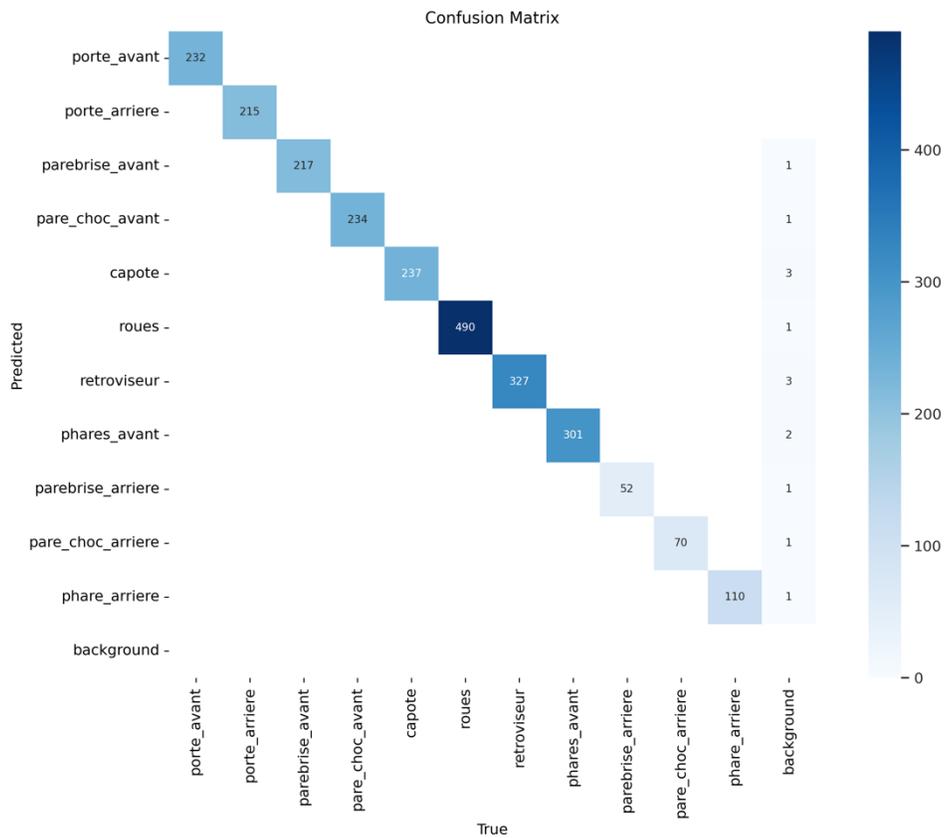


Figure 36 : Matrice de confusion de la détection des pièces du véhicule

Cette matrice nous montre les résultats sur les prédictions dans nos différentes classes. Les valeurs le long de la diagonale de la matrice de confusion (comme 232 pour

« porte_avant », 215 pour « porte_arriere », 217 pour « parebrise_avant », 490 pour « roues », 327 pour « retroviseur » etc.) indiquent les prédictions correctes pour chaque catégorie. Cela signifie que, par exemple, 232 « porte_avant » ont été correctement identifiées comme 'porte_avant'.

La plupart des valeurs en dehors de la diagonale sont nulles ou très faibles, indiquant un faible taux de fausses classifications. Cela démontre que le modèle possède un bon niveau de précision dans la majorité des cas, d'où l'absence de valeurs significatives en dehors de la diagonale.

Pour les cas d'erreurs on constate que pour la classe « capot » il y'a une valeur de 3 hors de la diagonale ce qui signifie que trois instances de « capot » ont été incorrectement classées sous une autre catégories. De même que pour « rétroviseur » avec trois valeurs en dehors de la diagonale et « phares_avant » avec deux valeurs hors de la diagonale.

4.4.2. Résultats de la détection

La Figure 37 et 38, montrent la prédiction du modèle pour la détection des différentes pièces externes sur un véhicule. Nous pouvons constater que le modèle fait de bonnes prédictions, que ce soit pour les petites ou les grandes pièces. Il est intéressant de noter que le modèle est capable de faire des prédictions précises même sur une vidéo, ce qui présente un avantage significatif. La vidéo nous permet d'obtenir des angles de vue variés sur la voiture, ce qui facilite l'analyse des différentes pièces et leur état.

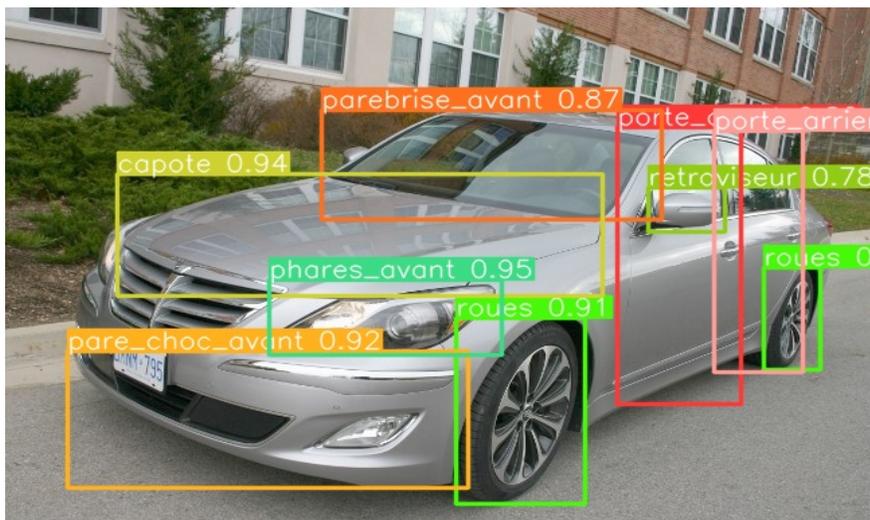


Figure 37 : Première prédiction du modèle

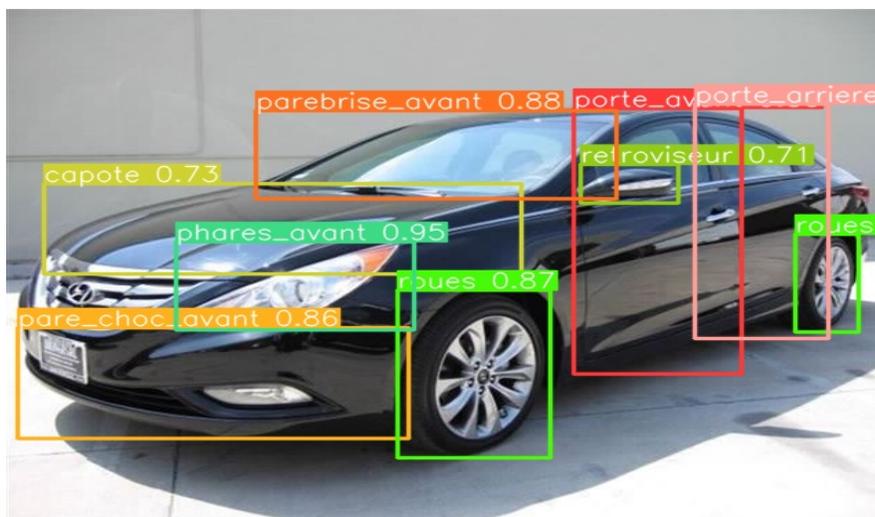


Figure 38 : Deuxième prédiction du modèle

4.5 Détection des dommages avec le modèle Yolov8

Dans cette partie nous allons effectuer la détection des dommages sur le véhicule à l'aide de Yolov8. Pour entraîner le modèle Yolov8 il faut suivre les étapes suivantes :

- Préparation des données : Chaque image doit être annotée avec des boîtes englobantes pour les parties endommagées. Les annotations doivent inclure les coordonnées des boîtes ainsi que les labels correspondants.
- Prétraitement des données : dans le prétraitement des données il faut redimensionner si nécessaire les images pour qu'elles correspondent à la taille attendue par Yolov8. Ensuite utiliser des techniques d'augmentations de données pour améliorer la robustesse du modèle. Cela inclut : la rotation (pour faire pivoter les images pour simuler différents angles de vue), le recadrage (pour découper des parties des images pour simuler différents champs de vision), le changement de luminosité (pour modifier la luminosité et le contraste pour varier les conditions d'éclairage).
- Configuration du modèle : YOLOv8 propose différentes versions en fonction de la taille et de la complexité (YOLOv8s, YOLOv8m, YOLOv8l et YOLOv8x). On choisit la version qui convient le mieux à notre tâche et à la capacité de calcul

disponible. Ensuite on configure le modèle avec des paramètres adaptés à notre tâche comme définir le nombre de classe et indiquez le chemin d'accès vers nos ensembles de données d'entraînement et de validation.

- Entraînement du modèle : pour démarrer le processus d'entraînement on utilise le script suivant :

```
« python train.py --data data.yaml --cfg yolov8.cfg --weights yolov8.weights --batch-size 16 --epochs 50 »
```

Avec data.yaml qui signifie le fichier fichier avec les chemins des images et des annotations, yolov8.cfg qui est le fichier de configuration du modèle et yolov8.weights : les poids pré-entraînés.

Pour les invariances à l'échelle nous avons utilisé la technique d'augmentation des données. Pendant l'entraînement, l'augmentation des données (comme le redimensionnement, le recadrage, le zoom) aide également à simuler des objets à différentes échelles, ce qui permet au modèle de s'adapter à ces variations lors de l'entraînement.

4.5.1 Performances du modèle

La Figure 39 illustre les performances de l'entraînement du modèle de détection des dommages en fonction de diverses métriques. On peut dire que le modèle a été entraîné de manière satisfaisante, atteignant des niveaux de précision et de mAP dépassant les 90 %. Le mAP est une mesure couramment utilisée en apprentissage automatique qui donne une mesure globale du modèle en tenant compte de la capacité à détecter correctement tous les types d'objets présents dans les images.

De plus, les taux de perte restent faibles tant pour les données d'entraînement que pour les données de validation.

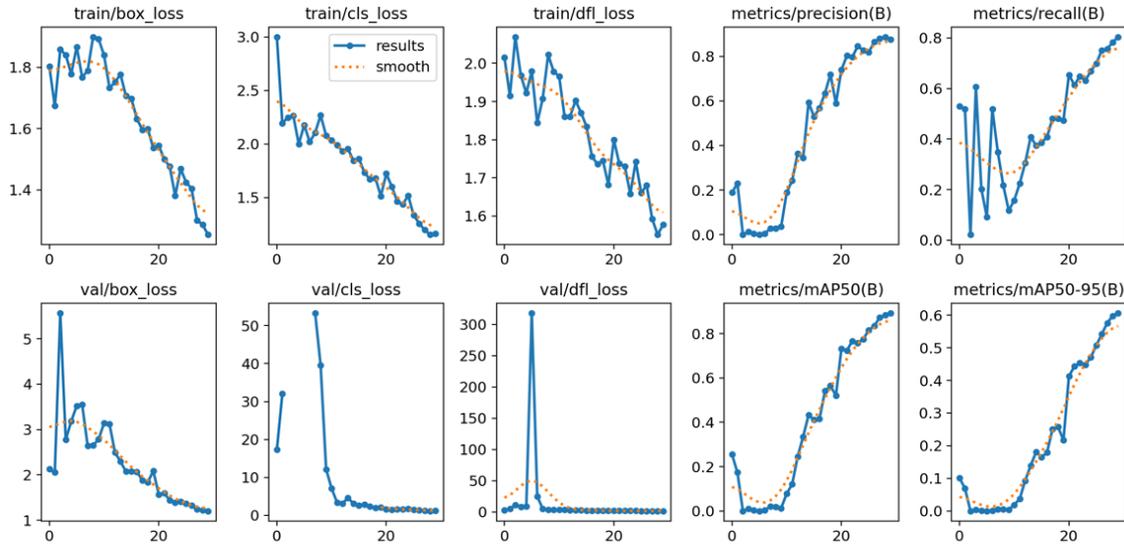


Figure 39 : Résultats des entraînements pour la détection des dommages

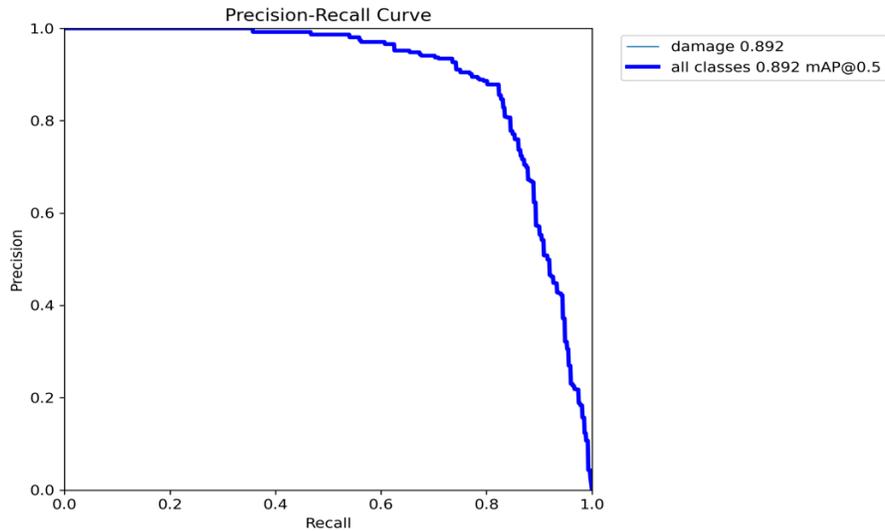


Figure 40 : Courbe de précision mAP du modèle

La Figure 40 montre le taux moyen de la précision sur nos classes. Et on voit que ce taux atteint une moyenne de 0.892. Voyons voir maintenant le résultat des prédictions sur différentes images.

4.5.2. Résultats de la détection

Dans cette partie nous montrons les résultats obtenus sur différents véhicules endommagés.



Figure 41 : Prédiction sur des véhicules endommagés

La Figure 41 montre le résultat des prédictions sur des véhicules endommagés. On peut constater que le modèle parvient à entourer la partie endommagée de la voiture avec une bonne précision.

Nous avons ensuite effectué un test sur une seule voiture sous plusieurs angles différents. Le but de ce test est de voir si le modèle est capable de faire de bonne prédiction même si la partie photographiée n'a pas subi de dommage.

Voici les résultats (voir Figure 42)



Figure 42 : Détection de dommage sur une voiture à partir de plusieurs angles

On constate que sur certaines images (Figure 42) le modèle parvient à détecter avec précision la partie endommagée. En outre dans d'autres images le modèle ne fait pas de prédiction ceci dû au fait qu'il n'y a pas de dommage sur ces parties de la voiture. Dans les résultats des prédictions, Yolo affiche pour les images 2, 3, 5 et 6 le message « no detections » pour spécifier qu'il n'y a pas eu de détection sur ces images (voir Figure 43)

```
↳ Ultralytics YOLOv8.0.147 🚀 Python-3.10.12 torch-2.0.1+cu118 CUDA:0 (Tesla V100-SXM2-16GB, 16151MiB)
Model summary (fused): 218 layers, 25840339 parameters, 0 gradients

image 1/7 /content/drive/MyDrive/Yolov8/Testing_car/bgr73891.jpeg: 480x640 1 damage, 59.0ms
image 2/7 /content/drive/MyDrive/Yolov8/Testing_car/bgr738910.jpeg: 480x640 (no detections), 7.6ms
image 3/7 /content/drive/MyDrive/Yolov8/Testing_car/bgr738912.jpeg: 480x640 (no detections), 7.8ms
image 4/7 /content/drive/MyDrive/Yolov8/Testing_car/bgr738913.jpeg: 480x640 1 damage, 7.9ms
image 5/7 /content/drive/MyDrive/Yolov8/Testing_car/bgr738915.jpeg: 480x640 (no detections), 9.4ms
image 6/7 /content/drive/MyDrive/Yolov8/Testing_car/bgr738917.jpeg: 480x640 (no detections), 7.7ms
image 7/7 /content/drive/MyDrive/Yolov8/Testing_car/bgr73895.jpeg: 480x640 1 damage, 7.6ms
Speed: 4.6ms preprocess, 15.3ms inference, 6.4ms postprocess per image at shape (1, 3, 480, 640)
Results saved to runs/detect/predict
```

Figure 43 : Résultat du modèle après prédiction

- Détection des pièces externes

Comme dernière étape de ce test, nous avons détecter les pièces externes sur un véhicule (Figure 44). Ceci nous permettra dans des études futures de faire des estimations sur chaque pièce endommagée.

L'importance de prendre plusieurs photos de la voiture permettra à notre réseau de pouvoir inspecter toutes les parties de cette auto et au fur et à mesure des étapes se concentrer uniquement sur les parties endommagées de la voiture.

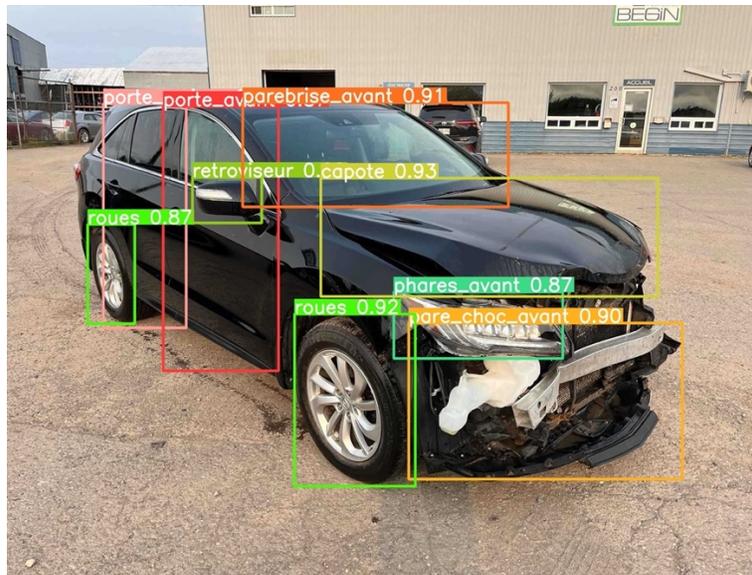
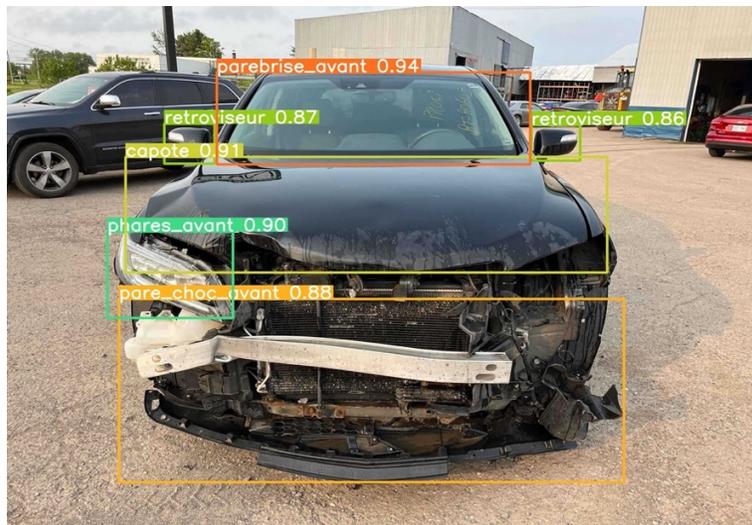
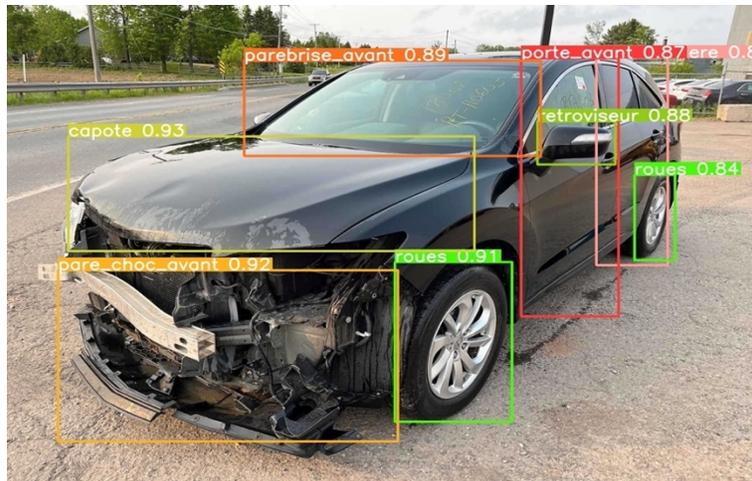


Figure 44 : Détection des pièces externes de la voiture

4.6 Discussion des résultats

La présente étude visait à développer une technologie d'intelligence artificielle pour détecter les dommages de véhicules après une collision. Nous avons utilisé des algorithmes d'apprentissage profond tels que ResNet pour classer les véhicules par marques, et le modèle U-Net pour la segmentation des pièces externes et YOLO pour détecter les différentes parties des véhicules ainsi que celles qui sont endommagées.

Les résultats obtenus ont montré que les modèles d'apprentissage profond ont réussi à classer avec précision les véhicules par marques et modèles, avec des taux de précision supérieurs à 85%. Cependant, nous avons également constaté que certains modèles de véhicules se ressemblaient beaucoup par leur carrosserie, ce qui a entraîné quelques mauvaises prédictions. Pour la Honda Civic, le modèle réalise des prédictions précises aussi bien pour l'avant que pour l'arrière du véhicule. Bien que ce niveau de précision soit prometteur, il est crucial de souligner que des véhicules ayant une carrosserie similaire à celle des Civic pourraient avoir été mal classés.

En ce qui concerne la détection des dommages, le modèle YOLO a montré des performances prometteuses. Il a réussi à détecter avec précision les différentes parties des véhicules, même celles qui sont endommagées. Cependant, avec le modèle U-Net nous avons observé quelques erreurs de prédiction, notamment au niveau des portes où le modèle a du mal à séparer les deux portes et a du mal à détecter les petites pièces dans certains cas. Nous avons également remarqué que plus le nombre de classes augmentait, plus les modèles avaient tendance à faire de mauvaises prédictions. Cela suggère qu'il est important de bien choisir les classes à détecter pour obtenir de meilleurs résultats.

Malgré ces quelques limitations, nous pensons que notre approche méthodologique est prometteuse pour la détection des dommages de véhicules. En combinant les algorithmes de classification et de détection, nous avons obtenu des résultats encourageants qui pourront être utilisés ultérieurement pour estimer les dommages de véhicules après une collision.

Pour améliorer notre modèle, nous suggérons d'explorer d'autres architectures d'apprentissage profond pour la détection des dommages comme Faster R-CNN, RetinaNet

et Yolov9. Nous recommandons également d'investiguer les modèles hybrides qui pourraient combiner les avantages des différentes architectures. Nous prévoyons également d'élargir la gamme de dommages traités en incluant des catégories supplémentaires telles que les dommages au coffre arrière, à la calandre, ainsi qu'aux vitres, avec une détection plus détaillée.

En résumé, cette étude a montré que les modèles d'apprentissage profond sont prometteurs pour la détection des dommages de véhicules. Malgré quelques limitations, nous sommes convaincus que notre approche peut être améliorée et utilisée dans le domaine de l'assurance automobile pour faciliter l'estimation des dommages après une collision. Des recherches futures dans ce domaine pourraient permettre de développer des modèles encore plus précis et performants.

Chapitre 5 : Conclusion finale et perspectives

L'objectif principal de notre projet était de mettre en place un système automatique pour estimer les dommages survenus après une collision impliquant des véhicules. Plus précisément, notre étude visait à détecter les dommages sur les véhicules à la suite d'une collision, en passant par plusieurs étapes. Tout d'abord, nous avons entrepris la classification des marques de voitures, puis la segmentation des différentes pièces externes de ces véhicules, et enfin la détection des dommages.

Pour la classification des marques de véhicules, nous avons testé plusieurs modèles de la famille ResNet, à savoir ResNet50, ResNet101 et ResNet152. Après avoir entraîné ces modèles sur notre jeu de données, nous avons constaté que ResNet152 surpassait les autres en termes de performances pour l'identification des marques de véhicules.

En ce qui concerne la segmentation des pièces de véhicules, nous avons utilisé un modèle Unet simple et un autre modèle Unet combiné à ResNet152. Cependant, le modèle Unet présentait des difficultés à détecter les petites pièces, tandis que le modèle combinant Unet avec ResNet152 déformait parfois la forme de certaines pièces dans ses prédictions.

Pour la détection des dommages, nous avons opté pour le modèle YOLOv8, qui nous a également servi à détecter les pièces externes des véhicules, étant donné que les résultats obtenus avec Unet n'étaient pas satisfaisants. Avec YOLOv8, nous avons obtenu des résultats plus précis aussi bien pour la détection des dommages que pour celle des pièces externes des véhicules.

Cependant, nous reconnaissons qu'il reste encore des défis à relever dans ce domaine. Certains modèles de véhicules se ressemblent beaucoup par leur carrosserie, ce qui peut entraîner des erreurs de prédiction. De plus, lorsque le nombre de classes augmente, les modèles peuvent montrer des performances moins satisfaisantes.

Pour améliorer notre modèle, nous suggérons d'explorer d'autres architectures d'apprentissage profond et d'utiliser des techniques d'augmentation de données pour entraîner nos modèles sur un plus grand nombre d'images. Nous recommandons également

D'investiguer les modèles hybrides qui pourraient combiner les avantages des différentes architectures. Malgré ces limitations, nous sommes convaincus que notre approche méthodologique est prometteuse pour la détection des dommages de véhicules. Nos résultats montrent que les modèles d'apprentissage automatique peuvent être utilisés pour détecter les dommages de véhicules avec une précision satisfaisante.

Nos recherches futures s'orientent vers les objectifs suivants à savoir: l'estimation comparative des dommages en comparant les différentes pièces externes du véhicule, permettant ainsi une estimation globale de l'étendue des dommages. Nous visons également à développer des capacités de reconnaissance du modèle et de l'année d'un véhicule en analysant des caractéristiques spécifiques. De plus, nous envisageons de détecter la présence de neige, un facteur qui peut affecter la détection et l'estimation des dommages, afin d'améliorer la fiabilité de notre modèle dans des conditions météorologiques défavorables.

Enfin, cette étude ouvre des perspectives prometteuses pour l'application de techniques d'apprentissage profond dans le domaine de la détection des dommages sur les véhicules. Des avancées futures dans ce domaine pourraient conduire à des modèles plus précis et performants, améliorant ainsi les services offerts par les compagnies d'assurance et facilitant la gestion des accidents automobiles. La technologie développée pourrait également s'étendre à d'autres domaines tels que la maintenance préventive des véhicules et la gestion des flottes automobiles, contribuant ainsi à la sécurité et à l'efficacité de l'industrie automobile.

Bibliographies

Vahagn Tumanyan (2023). "Complete guide to semantic segmentation [Updated 2023]." from <https://www.superannotate.com/blog/guide-to-semantic-segmentation>.

Apeer. "Apeer." from <https://www.apeer.com/home/>.

artificielle, P. "Focus : Le Réseau de Neurones Convolutifs - Pensée Artificielle." from https://www.google.com/url?sa=i&url=https%3A%2F%2Fpenseeartificielle.fr%2Ffocus-reseau-neurones-convolutifs%2F&psig=AOvVaw1_OgtqU4c0isSD6hdBJsCA&ust=1701288227341000&source=images&cd=vfe&opi=89978449&ved=0CBMQjhxqFwoTCMCS8dW-54IDFQAAAAAdAAAAABAJ.

aubaine, A. "TOP 10 DES VÉHICULES LES PLUS VENDUS AU CANADA EN 2022." from <https://www.autoaubaine.com/actualite-automobile/chronique-Top-10-des-vehicules-les-plus-vendus-au-Canada-en-2022-55627.html>.

Aubaine, A. "TOP 10 DES VOITURES LES PLUS VENDUES AU CANADA EN 2021." from <https://www.autoaubaine.com/actualite-automobile/chronique-Top-10-des-voitures-les-plus-vendues-au-Canada-en-2021-55193.html>.

Bochkovskiy, A., et al. (2020). "Yolov4: Optimal speed and accuracy of object detection." arXiv preprint arXiv:2004.10934.

Borgholt, L., et al. (2022). "A brief overview of unsupervised neural speech representation learning." arXiv preprint arXiv:2203.01829.

Boudjehem, R. (2016). "Behavior control of an adaptive multi-agent system (Maîtrise du comportement d'un Système Multi-agents adaptatif)."

Cao, Y., et al. (2019). "Pixel-wise PolSAR image classification via a novel complex-valued deep fully convolutional network." *Remote Sensing* **11**(22): 2653.

Castonguay, A. (2023). "Le bénéfice net chute de 79 % pour Intact au deuxième trimestre de 2023." from <https://portail-assurance.ca/article/le-benefice-net-chute-de-79-pour-intact-au-deuxieme-trimestre-de-2023/>.

Chen, X., et al. (2020). "Residual attention u-net for automated multi-class segmentation of covid-19 chest ct images." arXiv preprint arXiv:2004.05645.

Cvat. from <https://app.cvat.ai>.

Deng, J., et al. (2009). Imagenet: A large-scale hierarchical image database. 2009 IEEE conference on computer vision and pattern recognition, Ieee.

Dhall, D., et al. (2020). "Machine learning: a review of the algorithms and its applications." Proceedings of ICRIC 2019: Recent Innovations in Computing: 47-63.

Dridi, S. (2021). "Supervised learning-a systematic literature review."

Dwivedi, M., et al. (2021). Deep learning-based car damage classification and detection. Advances in Artificial Intelligence and Data Engineering: Select Proceedings of AIDE 2019, Springer.

Faghihi, U., et al. (2020). From Association to Reasoning, an Alternative to Pearls' Causal Reasoning. The Thirty-Third International Flairs Conference.

Fan, J., et al. (2021). "A selective overview of deep learning." Statistical science: a review journal of the Institute of Mathematical Statistics **36**(2): 264.

Girshick, R. (2015). Fast r-cnn. Proceedings of the IEEE international conference on computer vision.

Girshick, R., et al. (2014). Rich feature hierarchies for accurate object detection and semantic segmentation. Proceedings of the IEEE conference on computer vision and pattern recognition.

Google. "Google driver ". from <https://accounts.google.com/AccountChooser?service=writely>.

Grosse, R. (2017). "Lecture 15: Exploding and vanishing gradients." University of Toronto Computer Science.

Haradji, Y. M., Zouinar; Catherine, Delgoulet; Alexandre, Morais (2020). "IA, robotique, automatisaion : quelles évolutions pour l'activité humaine ?".

He, K., et al. (2016). Deep residual learning for image recognition. Proceedings of the IEEE conference on computer vision and pattern recognition.

Hesamian, M. H., et al. (2019). "Deep learning techniques for medical image segmentation: achievements and challenges." Journal of digital imaging **32**: 582-596.

Hui, J. (2018). "mAP (mean Average Precision) for Object Detection." from <https://jonathan-hui.medium.com/map-mean-average-precision-for-object-detection-45c121a31173>.

infoassurance (2019). "Comment se déroule l'estimation des dommages automobiles?". from <https://infoassurance.ca/blogue/automobile/comment-se-deroule-l-estimation-des-dommages-automobiles/>.

Intelligence, M. (2022). "Part de marché de l'assurance automobile au Canada, valeur, taille de l'industrie (2022-27)." from <https://www.mordorintelligence.com/fr/industry-reports/canada-motor-insurance-market>.

Jocher, G. (2023). "Ultralytics YOLOv8." from <https://docs.ultralytics.com/>.

Kaur, P., et al. (2020). "Facial-recognition algorithms: A literature review." Medicine, Science and the Law **60**(2): 131-139.

Krizhevsky, A., et al. (2012). "Imagenet classification with deep convolutional neural networks." Advances in neural information processing systems **25**.

Kyu, P. M. and K. Woraratpanya (2020). Car damage detection and classification. Proceedings of the 11th international conference on advances in information technology.

l'auto, L. G. d. from <https://www.guideautoweb.com/galleries/58344/top-10-les-voitures-les-plus-vendues-au-canada-en-2020/>.

Learning, D. (2020). "Différences entre classification d'image, localisation/détection & segmentation d'objets." from <https://deeplylearning.fr/cours-theoriques-deep-learning/differences-entre-classification-dimage-localisation-detection-segmentation-dobjets/>.

LeCun, Y., et al. (2015). "Deep learning." nature **521**(7553): 436-444.

li, J. (2018). "Stanford Cars Dataset." from <https://www.kaggle.com/datasets/jessicali9530/stanford-cars-dataset>.

Li, Z., et al. (2021). "A survey of convolutional neural networks: analysis, applications, and prospects." IEEE transactions on neural networks and learning systems.

Lin, T.-Y., et al. (2017). Focal loss for dense object detection. Proceedings of the IEEE international conference on computer vision.

Liu, C., et al. (2018). Object detection based on YOLO network. 2018 IEEE 4th information technology and mechatronics engineering conference (ITOEC), IEEE.

Liu, X., et al. (2019). "Recent progress in semantic image segmentation." Artificial Intelligence Review **52**: 1089-1106.

Lu, H., et al. (2018). "Brain intelligence: go beyond artificial intelligence." Mobile Networks and Applications **23**(2): 368-375.

Mahesh, B. (2020). "Machine learning algorithms-a review." International Journal of Science and Research (IJSR). [Internet] **9**(1): 381-386.

Ouestfrance-auto. from <https://www.ouestfrance-auto.com/marques-voitures/#:~:text=On%20peut%20estimer%20%C3%A0%20plus,marques%20automobiles%20dans%20le%20monde>.

Phonsa, G. and K. Manu (2019). A survey: image segmentation techniques. Harmony Search and Nature Inspired Optimization Algorithms: Theory and Applications, ICHSA 2018, Springer.

Praveen, K., et al. (2020). "Inventory management using machine learning." Int J Eng Res **9**(06): 866-869.

Progi. "Progi." from <https://progi.com/>.

Python. "Python." from <https://www.python.org/>.

Qaddour, J. and S. A. Siddiqa (2023). "Automatic damaged vehicle estimator using enhanced deep learning algorithm." Intelligent Systems with Applications **18**: 200192.

Redmon, J., et al. (2016). You only look once: Unified, real-time object detection. Proceedings of the IEEE conference on computer vision and pattern recognition.

Redmon, J. and A. Farhadi (2018). "Yolov3: An incremental improvement." arXiv preprint arXiv:1804.02767.

Research, G. from <https://research.google.com/colaboratory/>.

Ronneberger, O., et al. (2015). U-net: Convolutional networks for biomedical image segmentation. International Conference on Medical image computing and computer-assisted intervention, Springer.

Rovelloti, O. (2016). "Traitement d'images et deep learning." from <https://www.natural-solutions.eu/blog/traitement-d-images-et-deep-learning-quelles-avancees->.

Ruiz, P. (2018). "Understanding and visualizing ResNets." from <https://towardsdatascience.com/understanding-and-visualizing-resnets-442284831be8>.

Shutterstock (2023). from <https://www.shutterstock.com/fr/search/3d-automobile>.

Simonyan, K. and A. Zisserman (2014). "Very deep convolutional networks for large-scale image recognition." arXiv preprint arXiv:1409.1556.

Tan, M. and Q. Le (2019). Efficientnet: Rethinking model scaling for convolutional neural networks. International conference on machine learning, PMLR.

Tontiwachwuthikul, P., et al. (2020). "Recent progress and new developments of applications of artificial intelligence (ai), knowledge-based systems (kbs), and machine learning (ml) in the petroleum industry." Special Issue on Artificial Intelligence (AI) in Petroleum, Petroleum Journal 6(4): 319-320.

Van Rossum, G. (2007). Python Programming Language. USENIX annual technical conference, Santa Clara, CA.

van Ruitenbeek, R. and S. Bhulai (2022). "Convolutional Neural Networks for vehicle damage detection." Machine Learning with Applications 9: 100332.

Vault, M. (2020). "My experiment with UNet – building an image segmentation model." from <https://analyticsindiamag.com/my-experiment-with-unet-building-an-image-segmentation-model/>.

Waheed, S. R., et al. (2021). Deep learning algorithms-based object detection and localization revisited. journal of physics: conference series, IOP Publishing.

Wu, M. and L. Chen (2015). Image recognition based on deep learning. 2015 Chinese Automation Congress (CAC), IEEE.

Wu, X., et al. (2020). "Recent advances in deep learning for object detection." Neurocomputing 396: 39-64.

Zhao, Z.-Q., et al. (2019). "Object detection with deep learning: A review." IEEE transactions on neural networks and learning systems 30(11): 3212-3232.