REMERCIEMENTS

*Je remercie vivement mon directeur de thèse **Pr. Noureddine ABOUTABIT**, Professeur à l'Ecole Nationale des Sciences Appliquées (ENSA) Khouribga, pour l'excellence de son accompagnement ainsi que pour le suivi permanant qu'il a fait tout au long de la réalisation de ce travail. Je n'oublierai jamais sa méthode pédagogique, ses conseils judicieux de haut niveau scientifique et pour ses nombreux encouragements prodigués tout au long de ce travail. Sa disponibilité, sa confiance et son humanisme méritaient d'être chaleureusement remerciés.*

*Je tiens à remercier très chaleureusement les membres du jury, **Pr. Cherki DAOUI**, **Pr. Mohamed YOUSSFI**, **Pr. Hicham BELHADAOUI**, **Pr. Badr ABOU EL MAJD**, **Pr. Lekbir AFRAITES**, qui ont accepté de lire et d'évaluer mon travail.*

*J'aimerais exprimer toute ma gratitude à l'équipe scientifique du laboratoire LIPIM (Laboratoire d'Ingénierie des Procédés, Informatique et Mathématiques) et notamment au Professeur **Imad HAFIDI**, au Professeur **Abdelmoutalib METRANE**, au Professeur **Abdelghani GHAZDALI**, ainsi qu'à la Professeure **Kaoutar KHALLAKI** pour leur soutien, leurs encouragements et leur générosité tout au long de cette thèse.*

*Je voudrais exprimer aussi ma gratitude et ma reconnaissance au Professeur **Mohamed SAJIEDDINE**, Directeur de l'ENSA, et au Professeur **My Saddik KADIRI**, Directeur adjoint, pour leur soutien et leurs encouragements.*

*Mes remerciements vont également à tous les enseignants qui ont contribué à ma formation depuis l'école primaire jusqu'au cycle doctoral ainsi qu'à tout le corps administratif de l'**ENSA Khouribga** et l'**Université Sultan Moulay Slimane Béni-mellal**.*

*Je suis enfin sincèrement reconnaissante à **mes parents** pour leur sacrifice et leur soutien moral durant tout mon cursus. Mes remerciements vont aussi à ma sœur jumelle **Hajar**, mes sœurs **Asmae** et **Meryam**, et mon frère **Khalid** pour leurs encouragements et leur soutien. Je remercie aussi tous mes amis, qui m'ont soutenue et encouragée au cours de cette thèse.*

جامعة السلطان مولاي سليمان
ⵜⴰⵙⴷⴰⵡⵉⵜ ⵓⵙⵍⵟⴰⵏ ⵎⵓⵍⴰⵢ ⵙⵍⵉⵎⴰⵏ
Université Sultan Moulay Slimane

ENSA KHOURIBGA ÉCOLE NATIONALE DES SCIENCES APPLIQUÉES

Solutions d'intelligence artificielle pour la détection des dépassements interdits de véhicules

Sara BAGHDADI

Résumé

Avec l'augmentation rapide du nombre de véhicules dans le trafic routier, les accidents de voiture sont également devenus de plus en plus fréquents. Récemment, la plupart des villes du monde ont commencé à intégrer diverses technologies comme les systèmes de transport intelligents afin d'accroître le confort et la sécurité. L'objectif de cette thèse est de développer un système de vision par ordinateur qui détecte les dépassements interdits des véhicules. Ce système aide à faire respecter le code de la route, à identifier et à sanctionner les contrevenants. Ainsi, nous avons proposé deux approches pour résoudre ce problème. Nous avons utilisé les techniques d'apprentissage automatique et profond, ainsi que les techniques de traitement d'image et de vidéo pour détecter les lignes de la chaussée, détecter et suivre les véhicules, corriger l'illumination de leur environnement, classer leurs catégories et leurs vues, etc. De nombreux tests sont effectués sur le système en utilisant des vidéos réelles tournées dans différentes circonstances. Une fois le système déployé, il s'avérera bénéfique pour la société.

Mots-clés : STI, Sécurité routière, Dépassements interdits, Vision par ordinateur, Apprentissage automatique, Détection, Classification, Suivi.

Abstract

With the rapid increase in the number of vehicles in road traffic, car accidents have also become more and more frequent. Recently, most cities in the world have started to integrate various technologies like intelligent transportation systems in order to increase comfort and safety. The objective of this thesis is to develop a computer vision system that detects prohibited vehicle overtaking. This system helps to enforce traffic laws, identify and sanction violators. Thus, we proposed two approaches to solve this problem. We used Machine Learning and Deep Learning techniques, as well as image and video processing techniques to detect the roadway lines, detect and track the vehicles, correct the illumination of their environment, classify their categories and views, etc. Numerous tests are performed on the system using real videos taken in different circumstances. Once the system is deployed, it will prove to be beneficial to society.

Keywords : ITS, Traffic Safety, Prohibited Passing, Computer Vision, Machine Learning, Detection, Classification, Tracking.

ملخص

مع الزيادة السريعة في عدد المركبات في حركة المرور على الطرق ، أصبحت حوادث السيارات أكثر وأكثر تواترا .في الآونة الأخيرة ، بدأت معظم مدن العالم في دمج تقنيات مختلفة مثل أنظمة النقل الذكية من أجل زيادة الراحة والأمان .الهدف من هذه الرسالة هو تطوير نظام رؤية حاسوبي ، يكتشف التجاوزات المحظورة للمركبات .يساعد هذا النظام في تطبيق قانون الطريق السريع والتعرف على المخالفين ومعاقبتهم .وبالتالي ، فقد اقترحنا طريقتين لحل هذه المشكلة .استخدمنا تقنيات التعلم الآلي والعميق ، بالإضافة إلى تقنيات معالجة الصور والفيديو لاكتشاف خطوط الطرق واكتشاف المركبات وتتبعها ، وتصحيح إضاءة محيطها ، وتصنيف فئاتها وزوايا رؤيتها ، إلخ .لقد ، تم إجراء الكثير من الاختبارات على النظام باستخدام مقاطع فيديو حقيقية تم تصويرها في ظل ظروف ، مختلفة .بمجرد أن يدرك السائقون وجود نظام آلي يضمن عدم تمكنهم من الإفلات من المخالفة سيؤدي ذلك إلى تحسن كبير في كفاءة شبكة الطرق من خلال تقليل معدل حوادث الطرق

الكلمات المفتاحية :أنظمة النقل الذكية ، السلامة الطرقية ، التجاوز المحظور ، الرؤية الحاسوبية التعلم الآلي ، الكشف ، التصنيف ، التتبع

Table des matières

Table des Figures

Table des Tableaux

"Artificial intelligence has the potential to help humanity thrive more than any invention that has come before it".

Dileep George -Co-Founder Vicarious

1. Introduction Générale

Les routes tuent dans le monde plus de 1,35 million de personnes et blessent chaque année entre 20 et 50 millions de personnes dont beaucoup sont devenues handicapées [1]. En fait, aujourd'hui les blessures causées par les accidents de la route sont la huitième cause de décès dans le monde (il est prévu qu'elle devienne la cinquième en 2030) et la première cause de décès chez les jeunes de 15 à 29 ans [2]. « Ces morts sont un prix inacceptable à payer pour la mobilité », a déclaré le Dr Tedros Adhanom Ghebreyesus, directeur général de l'OMS [3]. Plus de 90% de la mortalité routière se produit dans les pays à faible et moyen revenu, en particulier dans la région africaine [1]. Les accidents de la route entraînent des pertes économiques considérables pour les victimes, leurs familles et leurs pays. Ces pertes sont dues au coût des traitements et à la perte de productivité des personnes qui meurent ou restent handicapées à la suite de leurs blessures, ainsi que des membres de la famille qui doivent interrompre leur travail ou leurs études pour s'occuper des blessés [1]. Cela montre la gravité de la situation et la nécessité de déployer des efforts importants pour la mise en place des nouveaux systèmes, ce qui pourrait entraîner une réduction du nombre de décès dus aux accidents de la route. Toutefois, « la sécurité routière est une question qui ne reçoit pas partout l'attention qu'elle mérite et c'est l'une de nos grandes possibilités de sauver

des vies partout dans le monde », a expliqué Michael R Bloomberg, fondateur et PDG de Bloomberg Philanthropies et ambassadeur mondial de l'OMS pour les maladies non transmissibles et les traumatismes [3].

Aujourd'hui, la technologie peut réduire le nombre énorme des accidents. Différents systèmes et technologies ont été développés pour améliorer le confort et la sécurité routière. Ces systèmes sont référencés sous le nom de STI (Systèmes de Transport Intelligents). Les STI représentent de multiples et diverses technologies dans le domaine des transports. Ces technologies peuvent être utilisées dans de nombreuses applications comme : l'application automatique de la loi routière (Automatic road law enforcement), les ADAS (Advance Driver Assistant Systems ou systèmes avancés d'aide à la conduite) etc. [4].

Plus de 90 % des accidents de la route sont dus aux erreurs humaines [5], c'est-à-dire à une application insuffisante du code de la route. Donc, s'il n'y a pas d'autorité chargée de faire respecter le code de la route, le nombre de blessés et de morts sur les routes ne diminuera pas [6]. Ainsi, la surveillance du comportement humain pour trouver les conducteurs qui ont violé le code de la route peut rapidement réduire le grand nombre d'accidents.

Les dépassements interdits représentent un pourcentage élevé du total des causes des accidents qui se produisent sur la route, juste après les excès de vitesse [7]. Ils sont dus à un véhicule qui tente d'en dépasser un autre sur une route à deux voies. Le dépassement interdit est généralement indiqué par une ligne blanche (ou jaune) continue située à gauche de la voie dans laquelle on circule. Tout véhicule ne doit pas dépasser cette ligne afin d'éviter de passer dans le sens inverse de la circulation. Ainsi, les accidents correspondants sont souvent assez graves en raison de la nature frontale des collisions. Ils représentent en fait un risque majeur pour le conducteur lui-même et les autres utilisateurs de la route.

Malheureusement, le contrôle humain des dépassements interdits est particulièrement difficile, surtout que les agents de police ne peuvent pas être toujours surplace pour surveiller toutes les lignes continues pendant toute la journée pour voir si quelqu'un dépasse la ligne. Donc cette tâche est pratiquement impossible vu le grand nombre de routes.

Pour cette raison, les pays développés ont installé des caméras de surveillance sur les routes afin que les agents puissent suivre le comportement des conducteurs en direct

depuis les salles de surveillance. Cette solution reste difficile à mettre en œuvre en raison de la taille du réseau de caméras. Il est donc nécessaire d'automatiser les tâches ; il faut disposer de systèmes qui détectent automatiquement les erreurs des conducteurs et envoient des alarmes aux agents pour qu'ils vérifient l'infraction, avant de délivrer des étiquettes électroniques à ces conducteurs. Sans aucun doute, si cette solution est appliquée, le risque d'accident peut être réduit de manière significative. Car, les conducteurs seront conscients de l'existence d'un système automatique qui garantit qu'ils ne peuvent pas s'en tirer en cas d'infraction.

C'est dans cette perspective que s'inscrit l'objectif principal de cette thèse. Il s'agit de développer un système de vision par ordinateur visant à détecter automatiquement les dépassements interdits à partir d'images ou de séquences d'images capturées par une caméra fixe (qui est utilisée dans les systèmes de surveillance du trafic, Figure 1) et caméra mobile (dashcam).

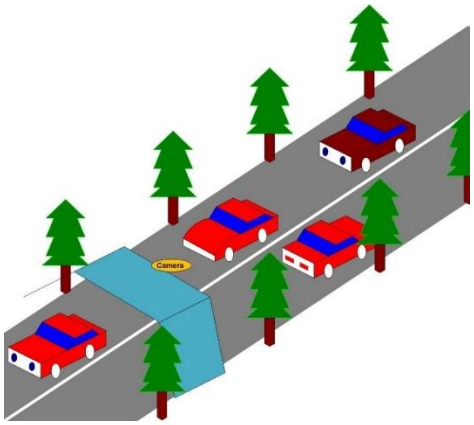


Figure 1. Position de la caméra de surveillance

Le système proposé dans cette thèse pour l'analyse et le traitement de cette problématique fait relever plusieurs problèmes adjacents qui doivent être étudiés en amont. Notre système suppose une caméra fixe filmant une scène où le dépassement est interdit (ligne continue). Ainsi, un véhicule filmé de vue arrière et se trouvant dans la zone de dépassement est considéré en infraction. Pour réaliser un tel système, il est ainsi primordial d'étudier plusieurs défis techniques liés. Dans ce sens, notre système doit intégrer des processus de détection des véhicules, de classification des vues (face ou arrière), de classification des types de véhicules (voitures, camions, bus, motos…) et de reconnaissance des véhicules d'urgences tels que les ambulances et les voitures de police.

La détection de véhicules est l'un des problèmes de recherche les plus difficiles en matière de vision par ordinateur en raison de la diversité des angles de vue, des variations d'éclairage, le camouflage, les ombres et de la complexité de l'arrière-plan. En raison de l'existence de plusieurs vues pour un véhicule, les études de recherche de détection dépendent considérablement de la vue du véhicule. De nombreuse études sont proposées pour le cas de la vue latérale tandis que peu de travaux se focalisent sur les vues d'avant et d'arrière.

Vu notre objectif de détecter les dépassements interdits, nous avons utilisé plusieurs techniques efficaces de traitement d'image et de vision par ordinateur. Les techniques de vision par ordinateur ont connu un développement accéléré au cours des dernières années. Ces techniques sont appliquées récemment à la sécurité routière. Il devient donc de plus en plus pratique de détecter, classifier et suivre automatiquement les usagers de la route sur des données visuelles (vidéo ou image) [2]. Parmi ces techniques, on trouve : l'apprentissage automatique, l'apprentissage profond, et l'extraction des caractéristiques visuelles. En outre, les capacités de combinaison de ces techniques sont explorées et une méthodologie est présentée pour la fusion des classifieurs construits à partir de ces techniques, en tenant compte également de la pose du véhicule. L'étude dévoile les limites de la classification basée sur une seule caractéristique.

Les contributions qui ont été apportées au cours de ce travail de thèse sont :

- Nos premiers travaux portent sur la détection et la classification des véhicules. Plusieurs types d'algorithmes de classification d'objets par apprentissage automatique et profond sont comparés. Les paramètres de ces algorithmes sont choisis de manière à assurer la crédibilité des résultats expérimentaux, et les algorithmes de classification sont comparés en ce qui concerne le runtime, l'exactitude, les taux de détection. Nous avons créé des descripteurs et classifieurs en fusionnant d'autres fameux algorithmes. Dans ce contexte, la combinaison des différentes techniques apparaît comme le moyen naturel de surmonter les limites de chacune d'entre elles et d'exploiter leur nature hétérogène dans un cadre commun. Malheureusement, la fusion de ces techniques n'a été que très peu explorée dans la littérature. En outre, aucune étude fusionnant les mêmes techniques que nous avons utilisées pour la classification des véhicules n'est rapportée dans la littérature. Les résultats des tests obtenus sont analysés et résumés, qui ont une certaine importance d'orientation pour la recherche dans le domaine de la détection des véhicules.

- Pour corriger l'illumination de la route, nous avons proposé d'utiliser de nombreuses techniques, dont l'algorithme Variational Retinex. Ce dernier a donné de très bons résultats par rapport aux autres algorithmes, mais malheureusement, son utilisation dans le domaine de la détection de véhicules est très rare.

- Nous avons travaillé aussi sur la classification des vues. Il n'y a pas beaucoup de travaux qui traitent la classification des vues [8]. De nombreux facteurs rendent cette classification des vues avant et arrière très difficile, y compris la similitude de forme, de taille et de couleur.

- Nous avons proposé aussi un système qui vise à classifier les catégories de véhicules indépendamment de l'angle de vue (angles avant et arrière). La classification des véhicules, à partir de leur vue avant ou/et arrière, est plus difficile. La plupart des catégories de véhicules ont les mêmes caractéristiques dans leurs structures de vue telles que le pare-chocs, le pare-brise, l'éclairage, les rétroviseurs, etc. De plus, personne n'a encore développé un système de classification des catégories de véhicules indépendant des angles de vue.

- Nous avons proposé aussi une solution de vision par ordinateur pour détecter de manière robuste les ambulances. Dans la littérature, nous ne trouvons aucun travail impliquant la détection des ambulances (ou autre véhicule d'urgence) à partir des images. Notre système est le premier système qui effectue cette tâche. La majorité des recherches utilisent des technologies comme Bluetooth, ZigBee, et des capteurs électroniques afin de les détecter.

- La dernière contribution concerne le développement de deux approches pour détecter les dépassements interdits. Dans la littérature, il n'y a pas beaucoup de travaux qui tentent de résoudre ce problème à l'exception de d'environ trois études qui sont toutes presque basées sur le même principe. Cependant, aucune étude précédente n'a utilisé comme les approches que nous avons développées pour résoudre ce problème, donc nous pensons que cette thèse constitue le premier travail qui aborde sérieusement les dépassements interdits. En plus de la détection automatique des dépassements interdits, le système peut être utilisé aussi dans toutes les situations où les conducteurs peuvent utiliser la mauvaise direction de la circulation. Par exemple, il peut être installé dans des rues à sens unique, dans des parkings automatiques, etc.

- De plus, nous avons construit plusieurs bases de données d'images pour la détection et la classification des véhicules (vues arrière/avant). Nous avons construit aussi une base de données vidéo pour tester le système de détection des dépassements, car il n'existe pas de bases de données publiques pour les vidéos des dépassements interdits.

Dans le premier chapitre, nous avons traité la partie de la détection des véhicules ainsi que les différentes techniques utilisées pour résoudre les problèmes d'illumination et d'ombres. Dans le deuxième chapitre, nous avons travaillé sur la classification des vues de véhicules. Pour le troisième chapitre, nous avons étudié le problème de la classification des catégories de véhicules (bus, voiture, moto et camion) indépendamment de l'angle de vue (angles avant/arrière). Dans le quatrième chapitre, nous avons présenté le système global de la détection des dépassements interdits des véhicules.

"Computer vision and machine learning have really started to take off, but for most people, the whole idea of what is a computer seeing when it's looking at an image is relatively obscure".

Mike Krieger- CTO and co-founder of Instagram

2. Chapitre 1 : Détection des véhicules par vue arrière

Sommaire

2.1. Introduction

La détection des véhicules est l'un des problèmes de recherche les plus difficiles dans le domaine de la vision par ordinateur dû aux divergences de vues, aux variations d'éclairage et à la complexité de l'arrière-plan. Il existe de nombreuses vues possibles d'un véhicule : la vue latérale, la vue de face et la vue arrière [9] [10]. Les vues latérales des véhicules présentent des caractéristiques clairement reconnaissables et régulières dans leurs structures, telles que les roues, les vitres obliques et les pare-chocs, qui fournissent des indices cruciaux pour la détection [11]. La détection des véhicules à vue latérale est un domaine de recherche très actif par rapport à la détection des véhicules à vue arrière ou de face, qui représente néanmoins un problème important puisque la plupart des caméras de surveillance captent l'arrière ou l'avant des véhicules. Les vues face/arrière sont moins discriminantes et donc plus difficiles à détecter [12].

Dans le domaine de la vision par ordinateur, on désigne par le terme "détection d'objets" (ou classification d'objets) le fait de détecter la présence d'une classe d'objets dans une image numérique. Généralement, il existe plusieurs concepts à savoir la détection (ou la classification) d'objet, la reconnaissance d'objet, la localisation d'objet, et le suivi d'objet. La détection (ou la classification) comme on a déjà expliqué, c'est la présence d'une classe d'objet (une voiture par exemple) dans l'image. D'autre part, la reconnaissance est l'identification d'une instance spécifique (par exemple un individu spécifique parmi d'autres personnes). La localisation (ou parfois détection) : positionnement précis de l'instance détectée dans l'image (boite englobante ou segmentation de la zone) [13]. De plus, le suivi d'objet est le processus de localisation d'un objet en mouvement dans le temps dans une vidéo. Il associe les détections d'un objet sur plusieurs images. Toutes ces méthodes se fondent souvent sur les techniques de l'apprentissage supervisé.

Dans ce chapitre, nous nous concentrons sur la phase de détection des véhicules (vue arrière). Nous avons déjà expliqué le choix de ce type de vue (dans nos expériences, nous avons envisagé le scénario d'une violation d'un véhicule filmé depuis la vue arrière). Différentes méthodes peuvent être utilisées pour la détection des véhicules à partir des images. Elles sont principalement basées sur l'apprentissage d'un classificateur pour la création d'un modèle de classes prédéfinies. L'apprentissage automatique est effectuée en deux phases principales: l'extraction des caractéristiques

et la classification. La première étape consiste à extraire des caractéristiques pertinentes et informatives de l'image. Les caractéristiques extraites mesurent en effet plusieurs propriétés de l'image telles que la couleur, la texture, la forme, le mouvement, l'emplacement, etc. Dans la littérature, il existe de nombreux algorithmes conçus pour extraire ces caractéristiques telles que : SURF (Speeded up Robust Features), HOG (Histogram of Oriented Gradient), SIFT (Scale Invariant Feature Transform), LBP (Local Binary Patterns). Après cette étape d'extraction de caractéristiques, il est nécessaire d'avoir une décision qui prédit la classe de l'objet en fonction de ses caractéristiques. Pour ce faire, un classifieur utilise des données d'apprentissage afin de comprendre comment les variables d'entrée sont liées à la classe. On trouve deux types d'apprenants dans la classification : les apprenants paresseux (lazy learners) et les apprenants enthousiastes (eager learners). L'apprenant paresseux stocke les données d'entraînement en mémoire, comme k-NN (k-Nearest Neighbor), car il n'apprend rien pendant la phase d'entraînement. Pour prédire la classe d'une nouvelle instance, il trouve donc ses k-voisins les plus proches à partir des données d'entraînement et sélectionne leur classe majoritaire. L'apprenant enthousiaste, après avoir reçu un ensemble de données d'entraînement, construit un modèle de classification avant de recevoir de nouvelles données (test) à classer (Ex : Arbre de décision, Bayes naïf, réseaux neuronaux artificiels) [14].

Dans ce chapitre, nous avons commencé par réaliser une étude comparative des modèles construits en utilisant des descripteurs et classifieurs les plus connus. Ensuite, nous avons appliqué quelques techniques afin d'améliorer les performances de ces modèles de classification. Pour cela, nous avons divisé ce travail en deux grandes parties. Dans la première partie, nous avons construit dans un premier temps les modèles en utilisant quatre descripteurs : HOG (Histogramme de gradient orienté), SURF (Caractéristiques robustes accélérées), filtre de Gabor et LBP (Motif binaire local), et trois classifieurs comme SVM (Support Vector Machines), kNN et Arbre de décision. Ensuite, nous avons fusionné les descripteurs utilisés pour combiner leurs avantages dans le but de construire un modèle performant et robuste. Dans la seconde partie, nous nous sommes attaqués à la résolution des problèmes d'illumination et d'ombre dans les images de la base de données. C'est une étape extrêmement importante car la performance de l'ensemble du système dépend de la qualité des données.

En effet, les systèmes de détection et de classification doivent faire face à une variété de défis dans les scènes extérieures, comme les changements d'illumination, camouflage et les ombres. Ces problèmes ont été traités dans quelques études utilisant différentes méthodes d'amélioration de l'éclairage. Dans ce travail, nous avons essayé de résoudre ces problèmes en appliquant les techniques suivantes : Image du quotient morphologique, correction de l'éclairage basée sur l'algorithme de rétinex variationnel et techniques d'amélioration du contraste (filtrage Top-Hat, amélioration du contraste local et ajustement de l'intensité). Nous avons également essayé d'éliminer ou du moins réduire les zones d'ombre dans les images des véhicules en utilisant l'effet de minimisation de la fonction énergétique.

Ce chapitre est organisé comme suit : la section 2.2 présente les principaux travaux du domaine de la détection et classification des véhicules. La section 2.3 décrit le système suivi pour la classification des véhicules. Cette section est divisée en trois parties. La première partie présente le système global. La deuxième partie présente les descripteurs et classifieurs utilisés. La troisième partie présente toutes les méthodes utilisées pour résoudre les problèmes d'éclairage et d'ombre. La section 2.4 décrit la série des expériences réalisées et présente les résultats obtenus en les discutant.

2.2. Détection des véhicules par vue arrière dans la littérature

Les systèmes de transport intelligents (STI) sont devenus le centre d'intérêt de la recherche scientifique, afin d'offrir aux gens des bonnes conditions de conduite et une assistance meilleure et plus sûre. Le cœur de chaque système de transport intelligent est la reconnaissance et la détection des véhicules [15]. Dans cette section, nous abordons les recherches liées à la détection et la classification des véhicules. En effet, l'utilisation de la vision par ordinateur pour détecter les véhicules sur la route avec précision est une tâche difficile et a été un sujet de recherche très actif au cours des deux dernières décennies [16]. L'étude de ce problème diffère d'une situation à l'autre en fonction de nombreux critères. Par exemple, les environnements de conduite sont visuellement dynamiques avec un arrière-plan et un éclairage qui changent constamment [17]. La taille, la forme et la couleur des véhicules rencontrés sur la route varient également [18]. De plus, la position de la caméra joue un rôle important dans l'obtention de différentes vues du véhicule, qu'elle soit statique ou mobile. Nous commençons par aborder des travaux similaires à notre étude sur la détection de véhicules par la vue arrière. De plus, nous abordons certaines études qui ont traité le problème de l'illumination sur les systèmes de détection de véhicules.

Dans les études précédentes, diverses techniques d'apprentissage automatique ont été utilisées. Dans [15], les auteurs ont proposé une approche pour détecter les véhicules à partir de différentes vues : avant, arrière, latérale et oblique. Pour l'avant et l'arrière du véhicule, ils ont utilisé la caractéristique symétrique du véhicule. Pour le côté du véhicule, ils ont utilisé un algorithme de détection basé sur la détection de templates. Le template est construit en fonction de la connaissance de la forme du véhicule. L'algorithme peut être divisé en deux étapes : la génération d'hypothèses et la vérification des hypothèses. Pour les véhicules obliques, ils ont utilisé la méthode de correspondance des templates (template matching) pour générer l'hypothèse. Le template est modifié dynamiquement en fonction de la taille réelle du véhicule potentiel dans l'image donnée. Une fois l'hypothèse générée, ils utilisent des caractéristiques de ligne (line features) dans chaque sous-partie du modèle pour effectuer la vérification. [19] a proposé une méthode de détection et de suivi des véhicules en vue arrière. Les auteurs ont d'abord localisé les parties saillantes du véhicule, notamment la plaque d'immatriculation et les feux arrières. Ensuite, ils ont construit un modèle MRF (Markov Random Field) en traitant les parties du véhicule comme des nœuds de graphe. Le descripteur LBP a été utilisé pour déduire le graphe MRF afin d'obtenir la localisation des véhicules. Après la détection des véhicules, ils ont mis en œuvre le suivi des véhicules en utilisant le filtre de Kalman (kalman Filter, KF). Ils ont réalisé une technique de détection par suivi dans laquelle les emplacements prédits par KF ont été ajoutés au modèle MRF en tant que nœuds de graphe. Dans [20], les auteurs ont proposé un algorithme de détection de véhicules en mouvement et de classification par type, appelé consensus amélioré d'échantillons spatio-temporels. Ils utilisent d'abord l'algorithme classique de consensus d'échantillons spatio-temporels pour détecter les objets en mouvement, en éliminant les interférences de la variation de l'éclairage et des ombres sur l'identification des véhicules. Ensuite, ils classent les objets à l'aide de méthodes de fusion de caractéristiques tenant compte de la symétrie du véhicule, du numéro de plaque, de la surface, de la netteté et des caractéristiques de face. En 2018, [21] a présenté un modèle basé sur une caméra fixe pour la surveillance du trafic, la détection des véhicules qui intègre la manipulation des occlusions, le comptage, le suivi et la classification par OC-SVM (One Class SVM). Dans cette approche, les objets en mouvement sont d'abord segmentés de l'arrière-plan en utilisant le modèle de mélange gaussien (Gaussian Mixture Model, GMM) adaptatif. Ensuite, plusieurs caractéristiques géométriques sont extraites, telles que la surface, la hauteur, la largeur, le centroïde et le bounding box. En cas d'occlusion, un algorithme a été

implémenté pour la réduire. Le suivi est effectué avec un filtre de Kalman adaptatif. Enfin, les caractéristiques géométriques sélectionnées (surface, hauteur et largeur estimées) sont utilisées par différents classifieurs afin de classer les véhicules en trois catégories : petits, moyens et grands. [22] propose un système de détection et de classification des véhicules basé sur une zone de détection virtuelle (VDZ). Le système proposé comprend quatre étapes principales : l'extraction du premier plan, la détection du véhicule, l'extraction des caractéristiques du véhicule et la classification du véhicule. Un véhicule en mouvement est d'abord détecté à l'aide de GMM. Ensuite, plusieurs techniques, dont la sélection de la région d'intérêt, l'opération morphologique adaptative et le traitement des contours, sont appliquées pour obtenir des objets de premier plan corrects. Ensuite, les caractéristiques des véhicules sont calculées lorsque le centroïde d'un véhicule se trouve sur la VDZ. Enfin, les véhicules sont classés à l'aide d'un classifieur k-NN. Dans [23], une méthode de détection de véhicules basée sur la vidéo et utilisant des opérations morphologiques et l'histogramme du gradient HOG est proposée. La région ROI est sélectionnée et les pixels à l'intérieur des régions ROI sélectionnées sont seuls détectés. L'opérateur Sobel est utilisé pour l'identification des pixels de contour. Le gradient est obtenu en trouvant les pixels communs dans le contour détecté et les régions ROI. Enfin, le processus de détection d'objet est employé en utilisant des opérateurs morphologiques et le descripteur d'histogramme de gradient. Ainsi, un taux de réussite d'environ 83% est atteint dans la détection de véhicules en utilisant la méthode proposée.

Dans la littérature, différents travaux ont proposé de multiples techniques d'extraction de caractéristiques (descripteurs) et des algorithmes de classification pour la détection des véhicules. Les trois descripteurs les plus couramment utilisés sont Haar [24], HOG [25] et LBP [26]. En général, ces descripteurs sont combinés principalement avec les classifieurs SVM, et AdaBoost. Les caractéristiques de type Haar combinées avec Adaboost sont très efficaces pour la détection des visages. Cette méthode a été conçue par Viola et Jones [27] [28]. Elle commence par calculer des caractéristiques très simple (les caractéristiques pseudo-Haar ou Haar-like features en anglais) en introduisant les images intégrales. Ces dernières rendent le calcul de ces caractéristiques plus rapide. Ensuite, la méthode effectue une cascade pour les classifieurs forts formés par Adaboost [27]. Étant une méthode d'apprentissage supervisé, la méthode est constituée de deux phases : la première phase consiste en l'apprentissage du classifieur à partir de nombreux exemples positifs (images d'objets) et négatifs, et la seconde en l'application de ce classifieur à des nouvelles images. Haar combiné avec Adaboost a été appliqué

aussi à la détection de véhicules et s'est avéré avoir de bonnes performances. En effet, [29] a proposé un algorithme amélioré basé aussi sur les caractéristiques de type Haar et le classifieur AdaBoost pour la reconnaissance des véhicules. Dans un premier temps, les caractéristiques étendues de type Haar sont extraites à l'aide de la méthode de l'image intégrale. Puis un petit nombre de caractéristiques critiques parmi un très grand ensemble de caractéristiques de type Haar sont sélectionnées lors de l'apprentissage d'AdaBoost. Enfin, une classification en deux classes est effectuée à l'aide du classifieur AdaBoost et des caractéristiques sélectionnées. Les résultats expérimentaux démontrent que l'algorithme proposé a de meilleures performances en termes de reconnaissance et de temps d'exécution que les méthodes traditionnelles. Dans [30], les auteurs ont proposé une méthode hybride de détection de véhicules sur deux étapes. La première étape consiste à vérifier d'abord l'existence du véhicule par la détection des zones d'ombre de la partie basse du véhicule en utilisant des caractéristiques de type Haar combinées avec AdaBoost. L'étape suivante est la vérification de la vue arrière du véhicule en utilisant un descripteur HOG et un classifieur SVM.

La caractéristique HOG combinée avec le classifieur SVM a été largement utilisée dans la reconnaissance d'images, et a obtenu un grand succès dans la détection des piétons. Elle a été aussi appliquée aux tâches de détection de véhicules [16]. [31] propose une nouvelle méthode de détection et de suivi des véhicules à l'aide d'une caméra monoculaire montée sur un véhicule. Dans cette méthode, les caractéristiques des véhicules sont apprises en tant que modèle d'objet par la combinaison d'une machine à vecteurs de support latents (LSVM) et d'histogrammes de gradients orientés (HOG). Le détecteur de véhicules utilise des caractéristiques globales et locales. Les véhicules détectés sont suivis à l'aide d'un filtre particulaire avec des vraisemblances intégrées, telles que la probabilité des véhicules estimée à partir du modèle et la corrélation d'intensité entre les différentes images.

D'autres méthodes d'extraction de caractéristiques ont été utilisées pour la détection des véhicules. [32] présente une méthode de détection et de suivi des véhicules à partir d'images ou de données vidéo acquises par des capteurs installés à bord du véhicule. Les caractéristiques du véhicule sont extraites par l'algorithme de traitement d'image GIST suivi du classifieur SVM. Le processus de suivi a été effectué sur la base d'une approche de correspondance des caractéristiques des bords (edge features matching). Le filtre de Kalman a été utilisé aussi pour corriger les mesures. Dans [33], les auteurs ont utilisé un filtre de Gabor optimisé. Plus précisément, les paramètres des filtres de

Gabor ont été optimisés à l'aide d'une approche global (GA), suivis d'un regroupement des filtres ayant des caractéristiques similaires. Chaque groupe de filtres est représenté par un seul filtre dont les paramètres correspondent à la moyenne des paramètres des filtres du groupe. Cette étape élimine les filtres redondants, ce qui permet d'obtenir un ensemble compact et optimisé de filtres. Les filtres moyens sont évalués à l'aide des machines à vecteurs de support (SVM). La méthode est testée le problème de la détection des véhicules à partir d'images en niveaux de gris. Elle donne de bonnes performances que l'utilisation de banques de filtres traditionnelles. Le travail [34] traite le sujet de la détection de véhicules en vue arrière. Plus précisément, en traitant le problème de la détection de véhicules comme un problème de classification à deux classes. Les auteurs ont utilisé quelques méthodes d'extraction de caractéristiques telles que l'analyse en composantes principales (ACP), les ondelettes et les filtres de Gabor. Pour évaluer les caractéristiques extraites, ils ont expérimenté deux classifieurs populaires, les réseaux neuronaux (NNs) et les machines à vecteurs de support (SVMs). La transformation de caractéristiques visuelles invariante à l'échelle (Scale-Invariant Feature Transform, SIFT) [35] est aussi une méthode d'extraction des caractéristiques qui remplace une image par un ensemble de descripteurs de points clés (caractéristiques) invariants en ce qui concerne l'illumination, la rotation et la mise à l'échelle [36]. Elle est utilisée dans le domaine de la reconnaissance d'images, dans un large éventail d'applications incluant la détection de véhicules, la détection de logos de véhicules. Le travail [37] a proposé un système de classification des marques et modèles de véhicules. Les auteurs ont utilisé les caractéristiques SIFT et un modèle de sac de mots pour extraire les caractéristiques visuelles de l'image d'entrée et les représenter sous forme de vecteur de longueur fixe. Un sac de caractéristiques SIFT est utilisé pour l'entraînement et le test. Le SVM avec un noyau linéaire est utilisée comme classifieur dans ce travail. La méthode proposée est évaluée en utilisant la base de données NTOU-MMR et les résultats montrent une exactitude de 89%. [38] propose un algorithme pour la reconnaissance des logos de véhicules, basé sur une transformée de caractéristiques invariantes d'échelle améliorée (Merge-SIFT ou M-SIFT). L'algorithme est évalué sur un ensemble de 1500 images de logos appartenant à 10 constructeurs automobiles distincts. Une série d'expériences est menée, divisant les 1500 images en un ensemble d'entraînement (base de données) et un ensemble de test. Il a été démontré que l'approche M-SIFT améliore l'exactitude de la reconnaissance par rapport à la méthode SIFT standard. Les résultats rapportés indiquent un taux de reconnaissance réelle moyen de 94,6 % pour les logos de véhicules, tandis que le temps de traitement

reste faible (~0,8 s). Cependant, en raison de l'utilisation de filtres de différence de gaussienne (DOG), la méthode SIFT n'est pas assez rapide pour les applications en ligne. [39] ont proposé un nouveau détecteur de caractéristiques invariant à l'échelle appelé Speeded-Up Robust Features (SURF) pour calculer et comparer rapidement les points caractéristiques. Il s'appuie sur des images intégrales pour les convolutions d'images et est plus efficace que la méthode SIFT. [40] propose un descripteur SURF symétrique afin d'enrichir la puissance de SURF pour identifier les points symétriques et ces points sont la version miroir les uns des autres. Une application de reconnaissance de marques et de modèles de véhicules est ensuite réalisée pour prouver la praticabilité et la faisabilité de la méthode. [41] applique aussi un descripteur SURF pour détecter les véhicules dans un système de vidéo-surveillance. Pour détecter les véhicules sur la route, le descripteur symétrique proposé est d'abord appliqué pour déterminer la région d'intérêt de chaque véhicule sur la route sans utiliser de caractéristiques de mouvement. Ce schéma présente deux avantages : il n'est pas nécessaire de soustraire le fond et il est extrêmement efficace pour les applications en temps réel.

Deux défis de la reconnaissance de marques et modèles de véhicules, à savoir les problèmes de multiplicité et d'ambiguïté, sont ensuite abordés. Le problème de multiplicité découle du fait qu'un modèle de véhicule a souvent des formes différentes sur la route. Le problème d'ambiguïté résulte du fait que les véhicules de différents constructeurs partagent souvent des formes similaires. Pour résoudre ces deux problèmes, un schéma de division en grille est proposé pour séparer un véhicule en plusieurs grilles ; différents classifieurs faibles entraînés sur ces grilles sont ensuite intégrés pour construire un classifieur d'ensemble fort. L'histogramme du gradient et les descripteurs SURF sont adoptés pour entraîner les classifieurs faibles à l'aide d'un algorithme SVM.

La fusion de plusieurs descripteurs est un moyen de surmonter les limites de chacun et d'exploiter leur nature hétérogène dans un cadre commun. Malheureusement, la fusion de ces techniques n'a été que très peu explorée dans la littérature. Par exemple, dans [42], les caractéristiques de Haar et SURF sont combinées avec un classifieur en cascade et un classifieur Gentle AdaBoost pour construire un modèle de détection de véhicules en temps réel. Tout d'abord, la détection des voies est utilisée pour réduire l'espace de recherche à une région d'intérêt. Ensuite, le classifieur en cascade est appliqué pour générer quelques candidats. Enfin, le classifieur Gentle AdaBoost avec les caractéristiques Haar et SURF évalue les candidats et indique la position finale du

véhicule avec les informations sur la voie. Cependant, ce système présente quelques inconvénients. L'un d'eux est que le détecteur de voie peut parfois donner des voies totalement fausses, donc le détecteur de véhicule va chercher dans une mauvaise zone. Un autre problème est que lorsque la luminosité est trop faible ou trop forte, le détecteur de véhicules peut ne pas fonctionner correctement. Dans [34], les caractéristiques de Gabor et d'ondelettes sont combinées en supposant qu'elles produisent des résultats complémentaires pour détecter les véhicules. Pour évaluer ces caractéristiques, les auteurs ont réalisé des expériences en utilisant deux classifieurs : SVM et Réseaux de neurones. En dehors de cela, aucune étude de la fusion des caractéristiques populaires pour la détection ou la classification des véhicules n'a été rapportée dans la littérature. La méthode la plus utilisée est la fusion de plusieurs algorithmes basés chacun sur un type de caractéristiques, c'est-à-dire qu'elle fusionne les décisions issues des différents modèles en utilisant la méthode du vote majoritaire. De nombreuses informations sont donc perdues le long de la chaîne de classification.

Dans quelques études, les chercheurs ont travaillé sur le problème des ombres et des changements d'illumination dans la détection des véhicules. Les systèmes de détection de véhicules dans le trafic urbain et interurbain utilisant la vision par ordinateur sont souvent basés sur des méthodes de soustraction d'arrière-plan. Les ombres en mouvement représentent une difficulté sérieuse pour ces méthodes, car elles apparaissent comme faisant partie des véhicules de premier plan. [43] propose un algorithme de suppression des ombres, adapté aux méthodes de soustraction d'arrière-plan, où seules les informations en niveaux de gris sont nécessaires. La méthode est basée sur le calcul de la densité des bords sur une image quotient, obtenue à partir de l'image courante et du modèle d'arrière-plan. [44] analyse les méthodes de génération d'arrière-plan et de suppression des ombres dans l'approche traditionnelle de soustraction d'arrière-plan, et présente un algorithme simple de génération et de mise à jour de l'image d'arrière-plan en couleur. Il définit un histogramme pour chaque canal de couleur de chaque pixel. Ensuite, les candidats d'arrière-plan seront sélectionnés par l'information de la fréquence d'apparition. La fonction d'évaluation détermine quel candidat correspond à l'arrière-plan. L'approche de la suppression des ombres combine la méthode de la corrélation croisée normalisée (Normalized Cross-correlation, NCC) et la différence entre les images. [45] a étudié l'application du concept de minimisation des fonctions d'énergie dans le traitement des images en utilisant les équations différentielles pour résoudre deux problèmes : les changements d'illumination, et l'existence des ombres. Dans le premier problème, il a examiné

l'algorithme Variational Retinex proposé par [46] qui essaie de créer l'effet Rétinex en extrayant l'image d'illumination. Dans le deuxième problème, il a essayé de détecter et de supprimer les ombres des images. Dans le processus de minimisation de la fonction d'énergie, il a utilisé l'équation différentielle d'Euler-Lagrange dans le premier problème. Dans le deuxième problème, la fonction d'énergie n'était pas une fonction intégrale, donc il n'était pas nécessaire d'utiliser une équation différentielle pour la minimiser. Il a suffi de définir la dérivée de la fonction d'énergie pour obtenir la valeur nécessaire. [47] a étudié l'algorithme variational and multi-scale Retinex de Kimmel pour éliminer l'ombre du véhicule, et proposé une méthode Retinex basée sur l'estimation des bords de l'anisotropie qui considère les bords de l'ombre comme des valeurs aberrantes et les lisse plage par plage. L'expérience montre que l'algorithme peut éviter l'effet de halo, et que les ombres peuvent être éliminées. [48] ont développé un système pour améliorer les séquences vidéo capturées dans des conditions de visibilité extrêmement faible, comme le phénomène de voile blanc dans les tempêtes de neige ou de sable, ce qui permettra d'améliorer la visibilité et la sécurité du conducteur. Le système proposé est basé sur l'algorithme Retinex avec une technique intégrée de détection d'objets et d'estimation de la distance dans le champ de vision du conducteur. [49] ont examiné la méthode d'égalisation d'histogramme et ont constaté qu'elle est très simple et efficace pour améliorer le contraste de l'image. Cependant, les méthodes traditionnelles d'égalisation d'histogramme entraînent généralement une amélioration excessive du contraste, ce qui donne à l'image traitée un aspect peu naturel et des artefacts visuels.

2.3. Méthodologie

Dès notre naissance, nous, les humains, sommes naturellement capables de voir, comprendre et interpréter le contenu d'une image. Cependant, tout ce que l'ordinateur voit est une grande matrice de chiffres ; il n'a aucune idée sur le contenu ou la description de l'image. Afin de comprendre le contenu d'une image, nous devons appliquer une classification des images, qui consiste à utiliser la vision par ordinateur et les algorithmes d'apprentissage automatique pour extraire le contenu d'une image [50]. La classification des images est la tâche d'attribuer un label à une image en fonction d'un ensemble prédéfini de catégories. La classification des images et la compréhension des images sont actuellement (et continueront d'être) le sous-domaine le plus populaire de la vision par ordinateur [50].

Alors, l'image doit être représentée de façon à ce qu'un ordinateur puisse la comprendre et l'analyser. On peut décrire une image en se basant sur la texture, la couleur, ou la structure. Pour extraire ces informations, on extrait des caractéristiques avec un processus qui consiste à appliquer un algorithme sur une image d'entrée et à obtenir un vecteur de caractéristiques qui décrit notre image. Pour accomplir ce processus, nous pouvons appliquer des descripteurs, telles que HOG, LBP, SURF, etc. Alors, ces caractéristiques peuvent être utilisées pour décrire et finalement étiqueter le contenu de l'image elle-même [50].

Nous allons présenter un aperçu général sur les fameuses techniques de la classification d'images dans l'apprentissage automatique ainsi que les différents descripteurs connus. Bien que l'apprentissage automatique n'ait commencé à se développer que dans les années 1990, il est rapidement devenu le sous-domaine le plus populaire et le plus réussi de l'intelligence artificielle (IA), une tendance qui s'explique par la disponibilité de matériel et de bases de données [51].

Cette section présente donc toutes les méthodes utilisées dans le cadre de ce premier travail visant à détecter les véhicules. Nous commençons par définir les descripteurs d'images, puis les méthodes de classification, puis les techniques utilisées pour résoudre les problèmes d'éclairage et d'ombres.

2.3.1. Descripteurs d'images

La clé d'une classification efficace des véhicules est de sélectionner un bon descripteur de caractéristiques qui peut caractériser la forme du véhicule. Comme nous l'avons dit, nous avons utilisé quatre types de descripteurs : HOG, SURF, LBP, et Gabor.

a- Histogramme de gradient orienté (HOG)

Introduit en 2005 par Dalal et Trigg, le HOG est un descripteur de caractéristiques utilisé pour détecter des objets [52]. Le descripteur HOG repose sur l'idée que l'apparence et la forme locale de l'objet dans une image peuvent être représentées par la distribution de l'intensité du gradient ou la direction des contours [53].

Pour une image I, le processus d'extraction des caractéristiques du HOG est basé sur le calcul du gradient de chaque pixel (x,y). Le gradient correspond à la première dérivée de l'image selon les deux axes : horizontal (Équation 1) et vertical (Équation 2).

$$G_x(x,y) = I(x,y) - I(x-1,y) \qquad \text{Équation 1}$$

$$G_y(x,y) = I(x,y) - I(x,y-1) \qquad \text{Équation 2}$$

Où $G_x(x,y)$ est le gradient horizontal et $G_y(x,y)$ est le gradient vertical au pixel (x,y).

Alors la valeur du gradient (ou la magnitude) G et la valeur de la direction du gradient (ou l'orientation) θ de chaque pixel *(x,y)* d'une cellule sont calculées comme suit (Équation 3, Équation 4, respectivement):

$$G(x,y) = \sqrt{G_x^2(x,y) + G_y^2(x,y)} \qquad \text{Équation 3}$$

$$\theta(x,y) = \arctan\left(\frac{G_x(x,y)}{G_y(x,y)}\right) \qquad \text{Équation 4}$$

Pour les images en couleur RGB, le gradient est calculé indépendamment pour chaque canal (R, G, et B). Et donc pour chaque pixel, la norme la plus élevée des trois valeurs est choisie [53].

Comme le montre la Figure 2, on utilise donc les valeurs de la magnitude et de l'orientation de tous les pixels pour construire un histogramme d'orientation des gradients. Cela se fait en appliquant un vote de gradient de chaque pixel. Alors, chaque pixel de la cellule vote pour une classe dans l'histogramme qui varie de 0 à 180◦ (20◦ par classe ou bin (en anglais)) [54].

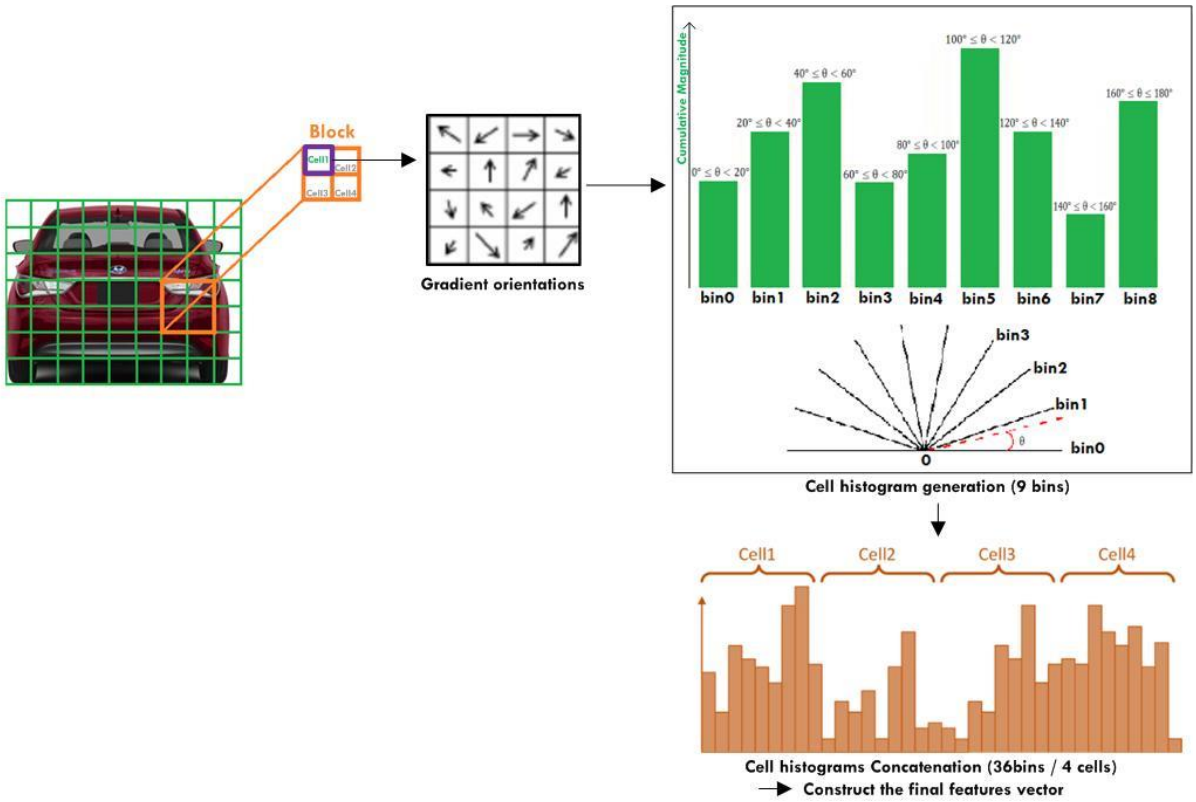


Figure 2. Calcul des caractéristiques par le descripteur HOG

Une cellule contient n ∗ n pixels, avec l'augmentation de n, le nombre d'éléments dans l'image diminue comme le montre la Figure 3.

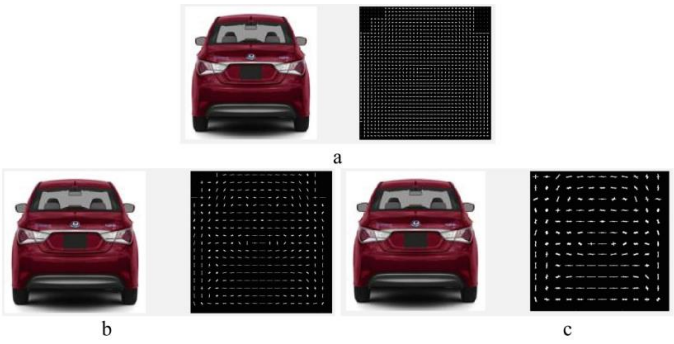


Figure 3. a. Caractéristiques HOG du véhicule/vue d'arrière (taille de cellule 4 * 4), b. Caractéristiques HOG (taille de cellule 8 * 8), c. Caractéristiques HOG (taille de cellule 12 * 12)

Dans la dernière étape, le vecteur de caractéristiques est obtenu en utilisant les histogrammes de chaque bloc. A titre d'exemple, la longueur de ce vecteur ici est de 36 caractéristiques, car chaque bloc a 2x2 cellules et chaque cellule est représentée par 9 bins. C'est-à-dire que chaque bloc est représenté par 2x2x9=36

caractéristiques. Si le bloc se compose de 3x3 cellules, on aura 3x3x9=81 caractéristiques/bloc, et 4x4 cellules donne 4x4x9=144 caractéristiques par bloc.

En combinant tous les vecteurs de tous les blocs en série, on obtient le vecteur de caractéristiques HOG final pour l'image entière.

b- Caractéristiques robustes accélérées (SURF)

SURF est une méthode qui permet de détecter et de décrire les caractéristiques locales d'une image. Il est proposé par des chercheurs de l'ETH Zurich et de la Katholieke Universiteit Leuven en 2006 [*55*], puis révisé en 2008 [*56*]. SURF a été construit à partir du descripteur SIFT (Scale Invariant Feature Transform, en Français : transformation de caractéristiques visuelles invariante à l'échelle). SIFT se compose de plusieurs étapes. Il commence par la création de l'espace d'échelle. Au cours de cette étape, l'image est remise à l'échelle (création d'octaves) afin de détecter les caractéristiques les plus importantes et les plus résistantes. Après cette étape, une pyramide d'espace d'échelle est obtenue. Cette pyramide est constituée d'octaves, triées de la plus grande à la plus petite. Dans l'étape suivante, le flou gaussien est appliqué. Ensuite, le Laplacien est calculé, afin de détecter les bords. Cela devrait être fait par le calcul de la dérivée seconde, mais cette opération est coûteuse en temps de calcul. La différence de gaussiennes (DOG) est donc effectuée. L'étape suivante de l'algorithme SIFT est la localisation des points clés. Chaque descripteur de point clé SIFT résultant est constitué de deux vecteurs. Le premier contient la position du point (x, y), l'échelle, la réponse (réponse de la caractéristique détectée, force), l'orientation, le Laplacien (signe du Laplacien). Le second contient le descripteur de 128 caractères.

D'autre part, le SURF est basé sur la somme des réponses des ondelettes de Haar et exploite efficacement les images intégrales qui sont utilisées à la place de DOG (Différence de Gaussien), ce qui lui permet de travailler beaucoup plus rapidement que SIFT. Un avantage important de SURF est qu'il génère moins de données que SIFT (SURF a un descripteur plus court de longueur 64), ce qui accélère le traitement ultérieur. SURF génère des points clés dans l'image (points d'intérêt). Pour chaque point clé (qui indique une caractéristique locale de l'image), un vecteur est généré. Il décrit l'environnement du point clé et permet de déterminer son orientation [*57*].

Le processus de SURF comprend quatre étapes principales [*57*]:

- Calcul d'images intégrales.

- Détecteur rapide de Hesse.

    - Le Hessois.

    - Construction de l'espace d'échelle.

    - Localisation précise des points d'intérêt.

- Descripteur de point d'intérêt.

    - Affectation d'orientation.

    - Composantes des descripteurs.

- Génération de vecteurs décrivant le point clé.

Dans une première étape, les images intégrales sont calculées. Cela permet d'augmenter l'efficacité [57]. Cette méthode est très simple et consiste à calculer la somme des pixels dans une zone donnée. Elle peut être décrite par la formule suivante [58] :

$$\mathrm{I}_{\mathrm{Somme}}(\mathrm{x},\mathrm{y}) = \sum_{\mathrm{i}=0}^{\mathrm{i}<\mathrm{x}} \sum_{\mathrm{j}=0}^{\mathrm{j}<\mathrm{y}} \mathrm{I}(\mathrm{i},\mathrm{j}) \qquad \text{Équation 5}$$

où I est l'image traitée et $\mathrm{I}_{\mathrm{Somme}}(\mathrm{x},\mathrm{y})$ est une somme de pixels dans une zone donnée. L'utilisation d'images intégrales pour le calcul de la surface permet de réduire les calculs à quatre opérations. Pour illustrer ceci, considérons un rectangle décrit par les sommets E, F, G, H (L'exemple est présenté à la Figure 4).

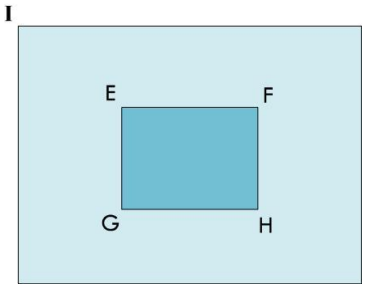


Figure 4. Calcul de la surface à l'aide d'images intégrales

La somme des intensités des pixels est calculée par la formule [58] :

$$\sum = (\mathrm{E} + \mathrm{H}) - (\mathrm{F} + \mathrm{G}) \qquad \text{Équation 6}$$

L'étape suivante consiste à calculer le déterminant de la matrice hessoise. La matrice de Hesse est présentée ci-dessous [58].

$$H(f(x,y)) = \begin{bmatrix} \frac{\partial^2 f}{\partial x^2} & \frac{\partial^2 f}{\partial x\,\partial y} \\ \frac{\partial^2 f}{\partial x\,\partial y} & \frac{\partial^2 f}{\partial y^2} \end{bmatrix} \qquad \text{Équation 7}$$

Le déterminant de cette matrice peut être calculé par [58] :

$$\det(H) = \frac{\partial^2 f}{\partial x^2}\frac{\partial^2 f}{\partial y^2} - \left(\frac{\partial^2 f}{\partial x\,\partial y}\right)^2 \qquad \text{Équation 8}$$

Le calcul du maximum local par le déterminant de la matrice de Hesse dépend du signe de ce déterminant. Si sa valeur est supérieure ou égale à 0, la zone est déterminée comme le maximum local. Dans l'étape suivante, l'espace d'échelle est construit. Cette étape est utilisée afin de rendre le point clé insensible aux changements d'échelle et à la rotation. Dans l'étape de localisation du point d'intérêt, le paramètre minHessian est nécessaire pour déterminer la valeur seuil. Le minHessian est un seuil permettant de décider à partir de quelle valeur les points clés sont acceptés. La localisation est calculée en comparant le déterminant hessois avec ses voisins.

Le processus de création des descripteurs de points clés est effectué en utilisant des ondelettes de Haar (voir Figure 5) qui décrivent les gradients des points clés. Afin de calculer l'orientation des descripteurs, l'algorithme recherche la plus grande somme d'ondelettes de Haar dans la fenêtre π/3 (60) et l'étape +-15 (voir Figure 6) [57].

Figure 5. Ondelettes de Haar. Ces filtres calculent les réponses pour les directions x (gauche) et y (droite) [57]

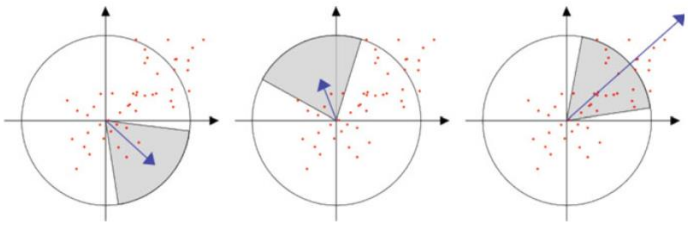


Figure 6. Affectation de l'orientation des descripteurs. La fenêtre de taille π/3 se déplace autour de l'origine et détermine la somme des plus grandes ondelettes, ce qui permet d'obtenir le vecteur le plus long [58]

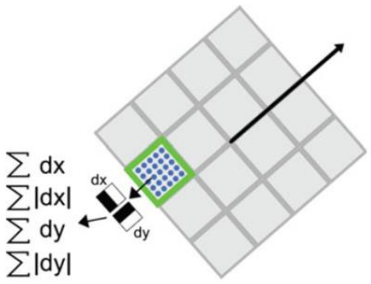


Figure 7. Représentation visuelle du descripteur SURF. Le descripteur contient 16 sous-régions et chacune d'entre elles est constituée d'ondelettes de Haar calculées 5×5 [58]

Le descripteur est constitué d'une matrice de groupes d'ondelettes 4×4. Chacune d'entre elles est composée des ondelettes de Haar calculées et est divisée en 5×5 éléments (Figure 7). Le point clé SURF de sortie est constitué de deux vecteurs. Le premier contient : la position du point (x, y), l'échelle (échelle détectée), la réponse (réponse de l'élément détecté, force), l'orientation, le Laplacien (le signe). Le second décrit la distribution de l'intensité des pixels dans le voisinage du point d'intérêt (64 valeurs) [57].

c- Motif binaire local (LBP)

Le LBP (*Local Binary Pattern*) est un descripteur de texture. Son but général est de créer une caractéristique basée sur l'ordre pour chaque pixel en comparant sa valeur d'intensité avec celle de ses pixels voisins [59]. Les zones d'intérêt telles que les coins ou les bords peuvent alors être détectées [60].

Le vecteur de caractéristique LBP est créé en suivant le processus suivant :

L'image est divisée en cellules de taille 3x3. LBP étiquette le pixel central de chaque cellule avec un nombre binaire appelé code LBP. Ce code est calculé comme décrit dans la Figure 8. Le pixel central pc est soustrait des 8 pixels voisins pi (voisinage 3 × 3, i = 0, 1, ..., 7). Si le résultat est négatif, le pixel voisin est codé avec « 0 ». S'il est positif, il est codé avec « 1 », comme le suivant :

$$s(p_i - p_c) = \begin{cases} 1, si\ p_i \geq p_c \\ 0, si\ p_i < p_c \end{cases} \qquad \text{Équation 9}$$

Où p_i et p_c sont les valeurs en niveaux de gris des pixels voisin et central respectivement.

Le code résultat est un nombre de 8 bits, il est donc converti en une valeur décimale (Label), comme le montre l'équation suivante [59] :

$$\mathrm{LBP}(p_c) = \sum_{i=0}^{7} s(p_i - p_c).\,2^i \qquad \text{Équation 10}$$

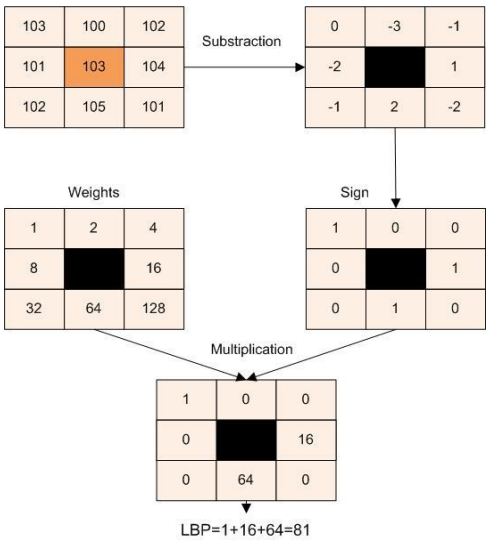


Figure 8. Opérateur LBP

En outre, le descripteur LBP est calculé dans sa forme générale comme suit [59].

$$\mathrm{LBP}(p_c) = \sum_{i=0}^{M-1} s(p_i - p_c).\,2^i,\ \ s(d) = \begin{cases} 1, si\ d \geq 0 \\ 0, \quad sinon \end{cases} \qquad \text{Équation 11}$$

où pc correspond à la valeur en niveau de gris du pixel central et pi est les valeurs des M pixels voisins.

Ensuite, les valeurs décimales obtenues sont utilisées pour construire l'histogramme pour chaque cellule en calculant les fréquences des valeurs obtenues de tous les pixels. Cet histogramme est considéré comme un vecteur de caractéristiques à $256 = 2^8$ dimensions. On concatène les histogrammes de toutes les cellules pour donner un vecteur de caractéristiques pour toute l'image [61].

Le descripteur LBP présente l'avantage de la tolérance aux changements d'illumination et de la simplicité de calcul. De plus, le LBP et ses variantes remportent un grand succès dans la description des textures [59].

d- Filtre de Gabor

Le **Filtre de Gabor** (nommé comme le nom du physicien anglais Dennis Gabor) est un filtre linéaire utilisé pour l'analyse de texture, la détection des contours, l'extraction de caractéristiques, etc [62]. Il est défini par le produit d'une sinusoïde complexe (connue sous le nom de porteuse) et une fonction gaussienne (connue sous le nom d'enveloppe)

[63], (voir Figure 9 [64]). Alors, dans un domaine spatial de dimension 2 (s'il s'agit d'une image), la fonction de Gabor est présentée comme suit :

$$G(x,y) = s(x,y).g(x,y) \qquad \text{Équation 12}$$

avec s(x,y) est la sinusoïde complexe. Elle est définie comme suit [65]:

$$\mathrm{s(x,y)} = \exp\left(2\mathrm{j}\pi.(u_0.x + v_0.y) + \varphi\right) \qquad \text{Équation 13}$$

(u_0, v_0) et φ définissent respectivement les fréquences spatiales et la phase de la sinusoïde. La partie réelle et la partie imaginaire de cette sinusoïde sont [63] :

$$Re\big(s(x,y)\big) = \cos\left(2\pi(u_0.x + v_0.y) + \varphi\right) \qquad \text{Équation 14}$$

$$Im\big(s(x,y)\big) = \sin\left(2\pi(u_0.x + v_0.y) + \varphi\right) \qquad \text{Équation 15}$$

g(x,y) est la fonction gaussienne. Sa formulation est donnée par :

$$g(x,y) = \exp\left(-\left(\frac{(x-x_0)^2}{\sigma_x^2} + \frac{(y-y_0)^2}{\sigma_y^2}\right)\right) \qquad \text{Équation 16}$$

où (x0, y0) est le pic l'enveloppe gaussienne g, σ_x (respectivement σ_y) est l'écart type du g par rapport à l'axe des abscisses (resp. des ordonnées) [63].

La fonction Gabor peut être représenté par une composante réelle (Équation 17) et une composante imaginaire (Équation 18) dont les directions sont perpendiculaires (déphasage de 90 degrés).

$$\mathrm{Gr}(x,y) = \cos(2\pi(u_0.x + v_0.y) + \varphi).\exp\left(-\left(\frac{(x-x_0)^2}{\sigma_x^2} + \frac{(y-y_0)^2}{\sigma_y^2}\right)\right) \quad \text{Équation 17}$$

$$\mathrm{Gi}(x,y) = \sin(2\pi(u_0.x + v_0.y) + \varphi).\exp\left(-\left(\frac{(x-x_0)^2}{\sigma_x^2} + \frac{(y-y_0)^2}{\sigma_y^2}\right)\right) \quad \text{Équation 18}$$

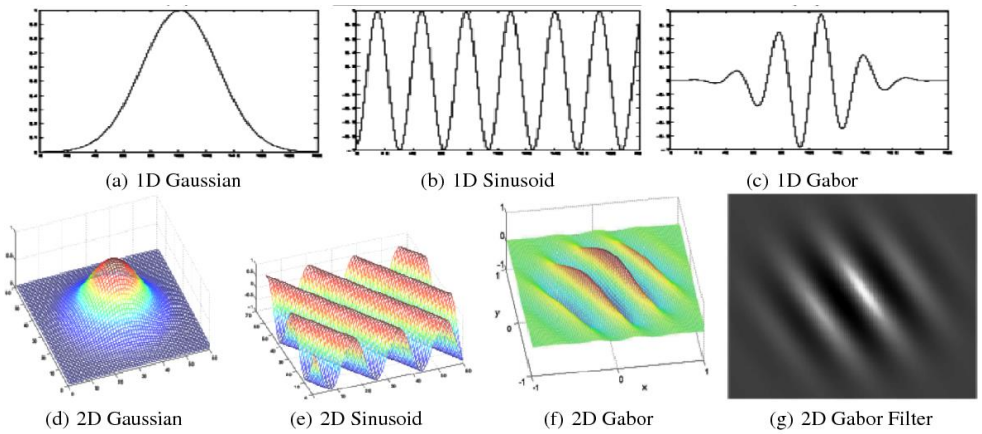


Figure 9. Filtre de Gabor [64]

Cette fonction Gabor est appliquée à un masque de convolution, pour définir un filtre de convolution appelé filtre de Gabor [65]. L'application de ce filtre *G* à une image *I* est donc la convolution de l'image avec le masque ou le noyau de Gabor *N* (Figure 10), comme le montre la formule suivante [65] :

$$G(I) = I * N \quad \text{Équation 19}$$

où "*" est l'opérateur de convolution.

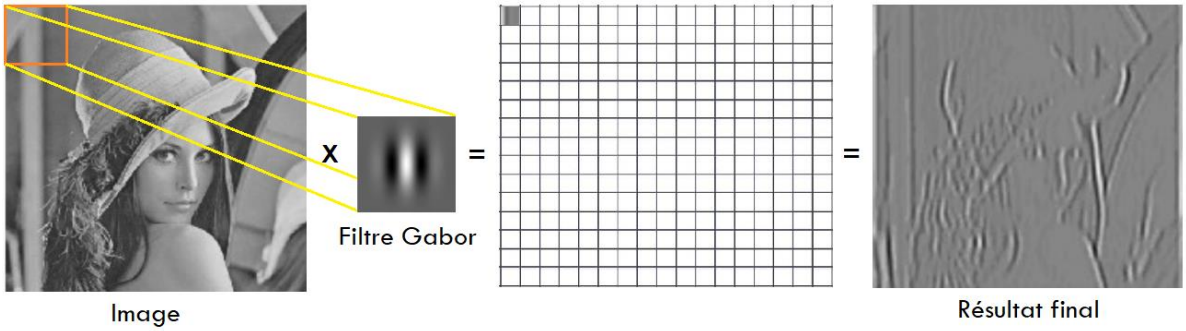


Figure 10. Convolution d'une image par un filtre de Gabor

Comme le montre la Figure 11, pour chaque filtre, on fait remonter les contours orientés selon l'angle auquel le filtre de Gabor est orienté.

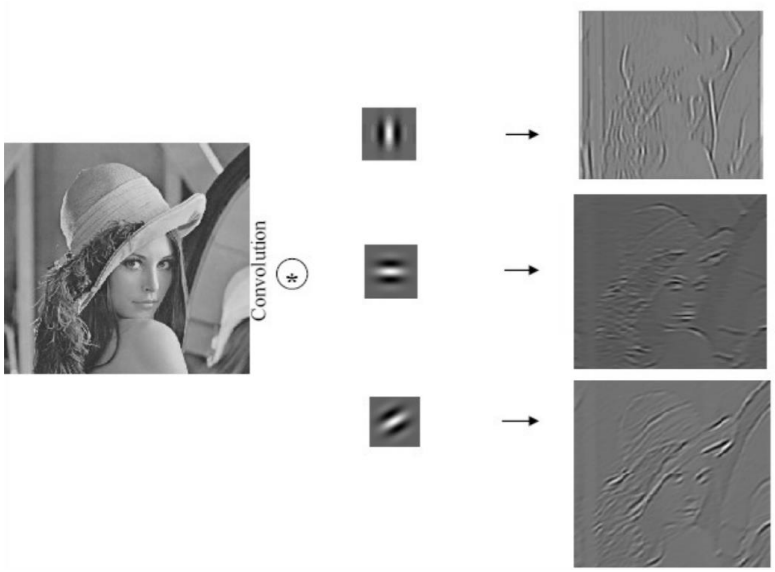


Figure 11. L'application de filtres avec différentes orientations

En pratique, pour analyser la texture ou obtenir des caractéristiques de l'image, on utilise un ensemble des filtres de Gabor à différentes fréquences et un certain nombre d'orientations différentes (exemple : voir Figure 12). Pour chaque noyau de Gabor, la magnitude *A* et la phase *P* sont calculées par les équations (Équation 20 et Équation 21), respectivement.

$$A(x,y) = \sqrt{Gr^2(x,y) + Gi^2(x,y)} \qquad \text{Équation 20}$$

$$P(x,y) = \arctan(Gr(x,y) / Gi(x,y)) \qquad \text{Équation 21}$$

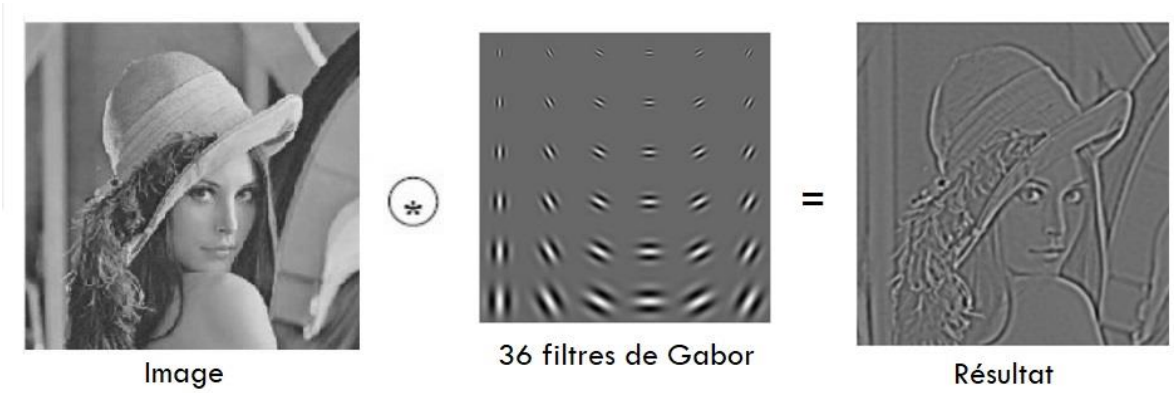


Figure 12. L'application de 36 filtres de Gabor sur une image

2.3.2. Méthodes de classification

Dans la programmation classique, les humains entrent des règles (un programme) et des données à traiter selon ces règles, et en ressortent avec des réponses (voir Figure 13). Avec l'apprentissage automatique, les humains entrent des données ainsi que les réponses attendues des données, et les règles (le modèle) en ressortent. Ces règles peuvent ensuite être appliquées à de nouvelles données pour produire des réponses originales [51].

Figure 13. La programmation classique vs l'apprentissage automatique

En bref, l'apprentissage automatique est un ensemble de techniques de modélisation qui nécessitent des données et permettent de créer des "modèles". Ici, les données peuvent être des documents, des sons, des images, etc [66]. Le "modèle" est le produit final de l'apprentissage automatique, comme le montre la Figure 14. Le terme "apprentissage" signifie que le processus ressemble à un entraînement avec les données pour résoudre le problème de la recherche d'un modèle. La technique analyse les données et trouve le modèle par elle-même plutôt que de le réaliser par un humain [66].

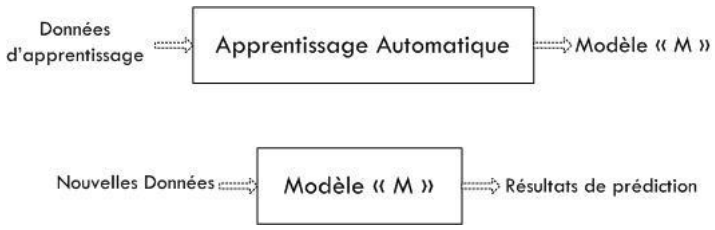


Figure 14. Principe de l'apprentissage automatique

L'apprentissage automatique est important dans des domaines tels que la reconnaissance d'images et la reconnaissance vocale, lorsque les lois physiques ou les équations mathématiques ne parviennent pas à produire un modèle [66]. Autrement dit, lorsqu'il n'est pas possible d'écrire des algorithmes pour effectuer la tâche [67].

Les techniques d'apprentissage automatique peuvent être classées en trois types selon les informations disponibles pendant la phase d'apprentissage : l'apprentissage

supervisé, l'apprentissage non supervisé, l'apprentissage par renforcement. Si les données sont labellisées (les sorties ou les réponses à la tâche sont connues), on parle d'apprentissage supervisé. Dans ce cas, il existe deux types de problèmes : la classification (si les sorties sont de type catégorique), et la régression (si les sorties sont des valeurs numériques continues) [68]. Contrairement à l'apprentissage supervisé, l'apprentissage non supervisé utilise des données non labelisées. Son objectif est de regrouper ces données pour créer des catégories en fonction de leurs similitudes.

Le présent travail rentre dans l'apprentissage supervisé et concerne particulièrement le problème de classification. Nous présentons en détail les méthodes de classification utilisées : SVM, k-NN et l'arbre de décision.

a- Machine à Vecteurs de Support (SVM)

Le classifieur SVM (Support Vector Machine) a été développée par Vladimir Vapnik et Corinna Cortes au début des années 1990 aux Bell Labs et publiée en 1995 [69]. Il est l'un des algorithmes d'apprentissage supervisé les plus populaires, qui est utilisé pour les problèmes de classification ainsi que de régression (il est surtout utilisé pour les problèmes de classification) [70].

Les SVM sont basés sur deux notions clés : la notion de marge maximale et la notion de fonction du noyau [71].

Dans la première, la marge est la distance entre la frontière de séparation et les points de données (ou vecteurs) les plus proches [71]. Le SVM utilise ces vecteurs pour créer la frontière séparatrice optimale (appelée un hyperplan) qui maximise la marge. Ces vecteurs extrêmes qui affectent la position de l'hyperplan sont appelés vecteurs de support (parce qu'ils soutiennent ou supportent l'hyperplan). D'où l'appellation de l'algorithme Machine à vecteurs de support [70].

Dans le cas où les données ne sont pas linéairement séparables, la deuxième idée clé des SVM est de transformer l'espace de représentation des données d'entrées en un espace de dimension supérieure dans lequel il existe une séparation linéaire. Pour ce faire, on utilise une fonction de noyau [71].

La résolution d'un problème de classification passe généralement par la construction d'une fonction f qui fait correspondre à un vecteur d'entrée *e* une sortie *s* :

$$s=f(e)$$ Équation 22

On se limite à un problème de classification binaire (à deux classes), c'est-à-dire s∈{-1,1}. Le vecteur d'entrée e appartenant à un espace E.

La classification multi-classes peut être considérée comme une combinaison d'un certain nombre de classifications binaires. Ceci sera expliqué dans le chapitre 3.

- Cas 1: Ensemble de données linéairement séparable

Le cas simple est celui d'une fonction linéaire qui est obtenue par combinaison linéaire du vecteur d'entrée e=(e1, e2, …,eN), avec un vecteur de poids $w = (w_1, w_2, \dots, w_N)$ [71].

$$f(e) = w.e + w_0$$ Équation 23

Si f(e)>0, e est affecté à la classe +1, si f(e)<0, e est affecté à la classe -1.

La frontière de séparation f(e)=0 est l'hyperplan séparateur, ou séparatrice.

Le principe d'un algorithme d'apprentissage supervisé vise à construire la fonction f(e) en utilisant un ensemble d'apprentissage $\{(e_i, l_i)\}_{i=1,\dots,m}$ où $e_i \in E$ (souvent $E=\mathbb{R}^N$) et $l_i \in \{-1,1\}$. Les l_i sont les labels, m est la taille de l'ensemble des données d'apprentissage, N la dimension des vecteurs d'entrée. Lorsque le problème est linéairement séparable, on doit avoir :

$$l_i.f(e_i) \geq 0 \quad 1 \leq i \leq m, \quad \textit{autrement dit} \quad l_i.(w.e_i + w_0) \geq 0 \quad 1 \leq i \leq m$$

Équation 24

Supposons que nous avons un ensemble de données (Figure 15) qui a deux classes (C1 et C2), et que l'ensemble de données ait deux caractéristiques x1 et x2. Nous voulons un classifieur qui peut classer un point de données en C1 ou en C2 [70]. Comme il s'agit d'un espace à deux dimensions, il suffit donc d'utiliser une ligne droite pour séparer facilement ces deux classes. Mais même dans ce cas simple, le choix de l'hyperplan séparateur ne semble pas évident. Il existe en effet toute une infinité de lignes [70].

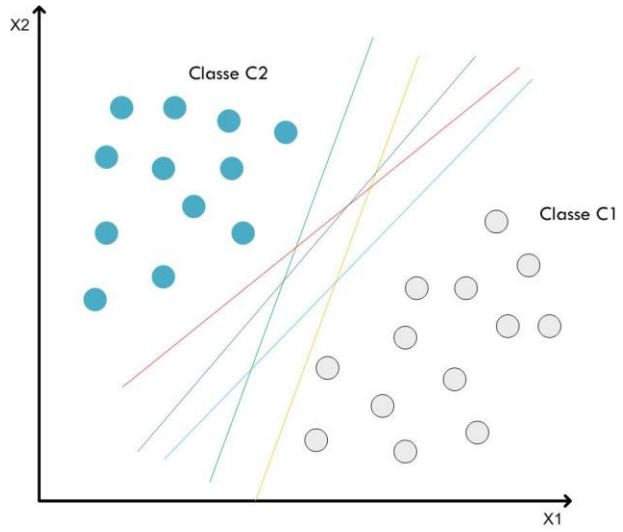


Figure 15. Données linéairement séparables

Pour résoudre ce problème, il a été démontré qu'il existe un seul hyperplan optimal, défini comme l'hyperplan qui maximise la marge entre les points de données extrêmes des deux classes et l'hyperplan séparateur (Figure 16).

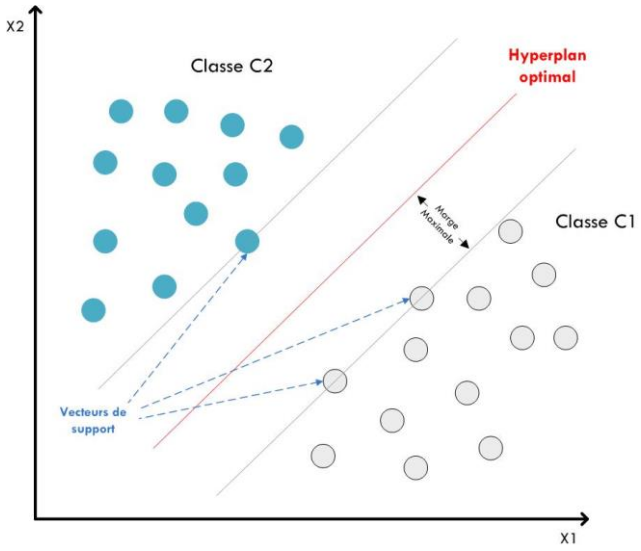


Figure 16. L'hyperplan optimal

L'hyperplan optimal est obtenu par :

$$arg\max_{w,w0} \min_{i} \{\|e - e_i\| : e \in R^m, w.e + w_0 = 0\} \qquad \text{Équation 25}$$

Il suffit donc de trouver w et w0 vérifiant cette condition pour définir l'équation du séparateur de l'hyperplan [71] :

$$f(e) = w.e + w_0 = 0 \qquad \text{Équation 26}$$

Seuls les *vecteurs de support* interviennent dans la définition de l'hyperplan optimal. Seul un petit nombre de points est suffisant pour le déterminer, les autres points de données ne participant absolument pas à sa définition. Ceci est donc avantageux en termes de complexité. De plus, le fait d'élargir l'ensemble d'apprentissage a moins d'influence que dans un classifieur où tous les points participent à la solution. Notamment, l'ajout d'échantillons à l'ensemble d'apprentissage qui ne sont pas des vecteurs de soutien n'a aucune influence sur la solution finale [71].

- Cas 2: Ensemble de données non linéairement séparable

Si un ensemble de données ne peut pas être classé en utilisant une ligne droite, alors ces données sont appelées données non linéaires [70].

Les SVM utilisent une fonction de noyau pour mapper les données d'apprentissage dans un espace de dimension supérieure de sorte que le problème soit linéairement séparable [72]. L'espace d'arrivée est nommé espace de redescription [70].
Considérons l'exemple suivant (*Figure 17*), les points de données sont non linéairement séparables ; on ne peut pas tracer une seule ligne droite.

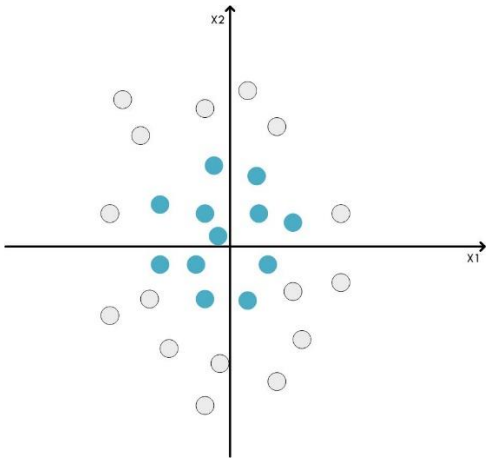


Figure 17. Données non linéairement séparables

Pour séparer ces points de données, on doit donc ajouter une dimension supplémentaire. Pour les données linéaires, on a utilisé deux dimensions x1 et x2, donc pour les données non linéaires, on ajoute une troisième dimension x3.

En ajoutant la troisième dimension, l'espace de l'échantillon deviendra comme dans la *Figure 18*. Le SVM va donc maintenant diviser les ensembles de données en classes de la manière suivante :

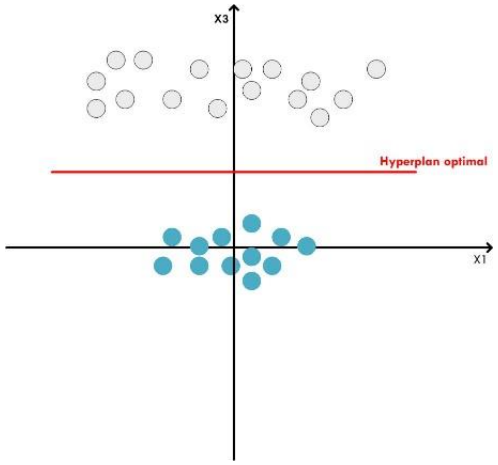


Figure 18. Séparation devient plus simple dans les dimensions supérieures

L'hyperplan ici est un plan (à deux dimensions). La dimension de l'hyperplan dépend du nombre de caractéristiques. Si le nombre de caractéristiques d'entrée est de 2 (comme indiqué sur la Figure 16: x1 et x2), alors l'hyperplan est une ligne droite. Et si le nombre de caractéristiques d'entrée est de 3, alors l'hyperplan sera un plan à 2 dimensions [70]. En général, un hyperplan d'un espace E de dimension n sera de dimension (n-1) [73].

b- Méthode des k plus proches voisins (k-NN)

Le classifieur des k plus proches voisins (k-NN pour k-Nearest Neighbors) est un algorithme simple d'apprentissage automatique et de classification d'images. En fait, il est si simple qu'il ne "apprend" rien. Il stocke tout simplement les points de données de l'ensemble d'apprentissage et il classifie chaque nouveau point de données en le comparant à chaque point de l'ensemble d'apprentissage en calculant une distance (voir Figure 19). Ensuite, il établit les distances calculées par ordre décroissant et sélectionne k points de données les plus proches du nouveau point. Parmi ces k voisins, on compte le nombre de points de données dans chaque catégorie. La classe ayant obtenu le plus grand nombre de votes "gagne" [50]. Ça nous rappelle le proverbe : "Dis-moi qui sont tes voisins, et je te dirai qui tu es".

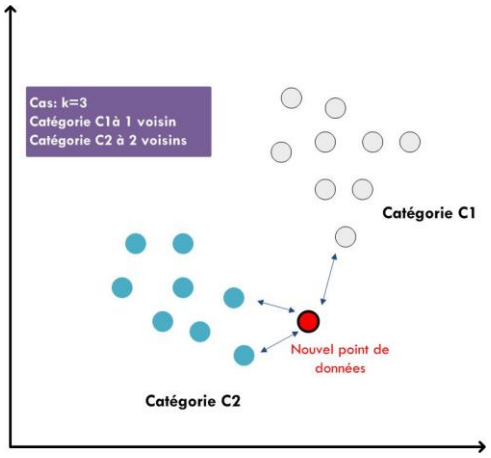


Figure 19. Classifieur k-NN (k=3)

Pour que l'algorithme k-NN fonctionne, il suppose avant tout que des images ayant un contenu visuel similaire sont proches les unes des autres dans un espace à n dimensions. Cela implique que la distance entre deux images de même classe est beaucoup plus petite que la distance entre deux classes différentes [50].

Pour calculer la distance, il faut choisir une fonction de mesure de distance ou de similarité comme la distance euclidienne, la distance de Manhattan, et la similarité cosinus. Comme nous le savons, la distance euclidienne est la distance entre deux points dans l'espace euclidien. Si sa dimension est de deux, la distance est entre deux points dans l'espace plan xy. Elle est présentée sur la Figure 20.

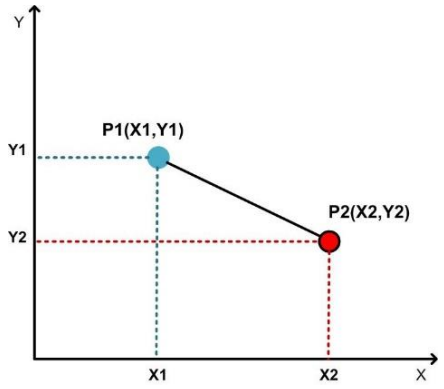


Figure 20. La distance euclidienne entre deux points dans un plan xy

La distance euclidienne entre deux points est calculée comme suit [74]:

$$D(P1, P2) = \sqrt{(X_2 - X_1)^2 + (Y_2 - Y_1)^2} \qquad \text{Équation 27}$$

Cette formule est étendue au cas où l'espace est de dimension *N*, [50]:

$$D(A,B) = \sqrt{\sum_{i=1}^{N}(A_i - B_i)^2} \qquad \text{Équation 28}$$

La distance de Manhattan ou de City Block est présentée par la formule suivante [50]:

$$D(A,B) = \sum_{i=1}^{N}|A_i - B_i| \qquad \text{Équation 29}$$

D'autre part, la similarité cosinus détermine la similarité entre deux vecteurs à n dimensions en calculant le cosinus de l'angle entre eux [75], comme le montre la Figure 21.

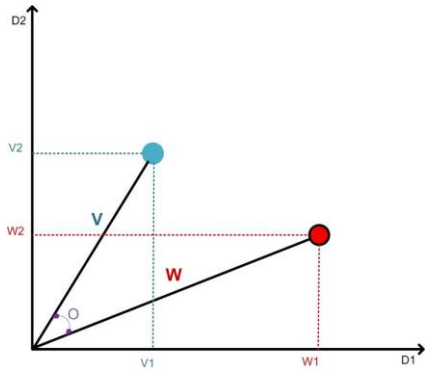


Figure 21. La similarité Cosinus

La similarité Cosinus entre V et W est calculée par la formule suivante [75] :

$$SimCos(V,W) = Cos(O) = \frac{V.W}{\|V\|\|W\|} \qquad \text{Équation 30}$$

Alors, plus la distance entre les deux points de données est petite, plus la similarité est grande (plus ils sont similaires) [76].

Il existe évidemment deux hyperparamètres à prendre en compte lors de l'exécution de l'algorithme k-NN. Le premier est la valeur de k : Quelle est la valeur optimale de k ? Si la valeur choisie est trop petite (par exemple k = 1), on gagne en efficacité mais le classifier devient sensible au bruit et aux points de données aberrants. En revanche, si k est trop grand, on risque de lisser exagérément les résultats de classification et d'augmenter les biais [50]. Le deuxième paramètre à prendre en compte est la métrique de distance utilisée. La distance euclidienne est-elle le meilleur choix ? Que dire de la distance de Manhattan ou de la similarité Cosinus ?

En général, on doit jouer avec différentes valeurs de k ainsi qu'avec différentes mesures de distance, en notant comment les performances changent.

c- Arbre de décision

L'arbre de décision (Decision-Tree) est une technique d'apprentissage supervisé qui peut être utilisée à la fois pour les problèmes de classification et de régression [77]. Il existe donc deux types d'arbres de décision : les arbres de classification (*Figure 22*) et les arbres de régression. Les arbres de classification produisent des résultats catégoriels (le résultat attendu est une classe). Les arbres de régression produisent des résultats numériques [67].

Il s'agit d'un classifieur à structure arborescente ; il commence par le nœud racine qui se développe sur d'autres branches. Les nœuds internes représentent les caractéristiques d'un ensemble de données, les branches représentent les règles de décision et chaque nœud de feuille représente le résultat (les étiquettes de classe) [77].

Dans un arbre de décision, il y a deux nœuds, qui sont le nœud de décision et le nœud de feuille. Les nœuds de décision sont utilisés pour prendre n'importe quelle décision et ont de multiples branches, tandis que les nœuds de feuille sont le résultat de ces décisions et ne contiennent pas d'autres branches [77].

Un arbre de décision pose simplement une question, et en fonction de la réponse (Oui/Non), il divise l'arbre en sous-arbres. Les décisions sont basées sur les caractéristiques de la base de données utilisée (données d'apprentissage) [77].

Pour construire un arbre, on utilise l'algorithme d'arbre de classification et de régression (CART pour Classification and Regression Tree algorithm) [77].

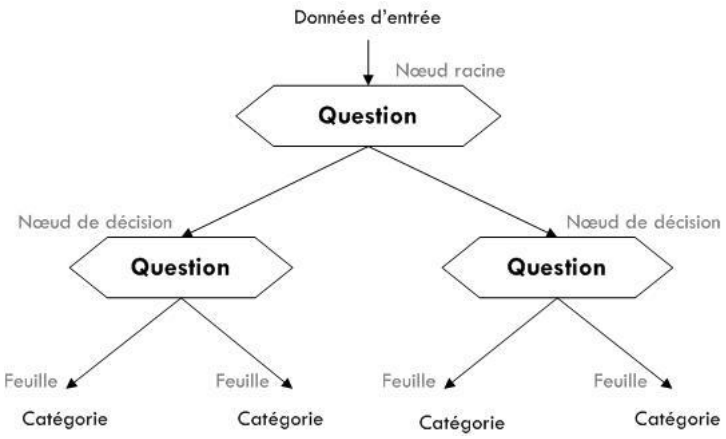


Figure 22. La structure générale d'un arbre de classification

2.3.3. Solutions de normalisation de l'éclairage et d'élimination de sombres

Les conditions d'éclairage modifient la visibilité globale des véhicules sur la route en fonction du temps. Dans les scènes extérieures, le problème de l'éclairage est alors un défi. La grande variation de l'éclairage influence souvent les performances des algorithmes de détection des véhicules. Il est donc important d'améliorer l'effet visuel des images. Bien que les descripteurs classiques tels que le Local Binary Pattern (LBP) et Gabor soient généralement considérés comme robustes à de légères variations d'éclairage, leurs performances diminuent lorsque les conditions d'éclairage sont mauvaises [78]. Diverses méthodes avancées ont été proposées en extrayant la composante de réflectance et la composante d'illumination, comme les algorithmes Retinex. Des techniques d'amélioration du contraste et des algorithmes d'image de quotient peuvent être utilisés aussi pour résoudre ce problème. La suppression des ombres sur les images peut aussi améliorer le taux de détection.

a- Méthode de normalisation de l'éclairage

Dans ce travail, nous avons utilisé les méthodes : Correction de l'illumination basée sur l'algorithme Rétinex Variationnel, Image du Quotient Morphologique et techniques d'amélioration du contraste (filtrage Top-Hat, amélioration du contraste local et ajustement de l'intensité). Les descriptions de ces méthodes sont données dans la suite.

- *Algorithme de Rétinex Variationnel*

La théorie du rétinex aborde le problème de la séparation de l'illumination et la réflectance dans une image donnée et donc de la compensation d'un éclairage non uniforme [46]. Elle suppose que la valeur d'intensité d'image I(x, y) d'un pixel (x,y) est composée de deux composantes : l'illumination L(x, y) et la réflectance R(x,y) des images. La relation entre ces trois composantes est définie comme suit :

$$I(x,y) = L(x,y) \times R(x,y) \qquad \text{Équation 31}$$

Où I, L, R représentent respectivement l'intensité, la composante d'illumination et la composante de réflectance.

Les avantages d'une telle décomposition comprennent la capacité d'éliminer les effets de l'éclairage arrière/avant des véhicules et de corriger les variations de l'éclairage dans l'espace (zones intérieures et extérieures).

Une première étape effectuée par la plupart des algorithmes Retinex est la conversion dans le domaine logarithmique par i = log I, l = log L, r = log R, et donc l'Équation 31 devient :

$$i = l + r \qquad \text{Équation 32}$$

Les différents algorithmes Retinex suivent généralement le même diagramme que celui de la Figure 23.

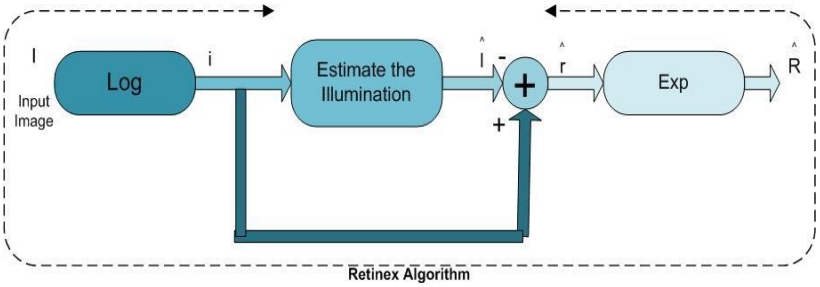


Figure 23. Diagramme de Rétinex

L'algorithme que nous avons utilisé dans notre étude est l'algorithme du Rétinex Variationnel proposé par [46]. Cet algorithme utilise l'équation différentielle d'Euler-Lagrange dérivée d'une fonction énergétique connexe. Comme tous les autres algorithmes de Rétinex, l'algorithme de Rétinex Variationnel est basé sur certaines hypothèses détaillées dans [45]. Ainsi, l'équation d'Euler-Lagrange devient :

$$\Delta E = \alpha.(l-i) - \Delta l - \beta.\Delta(l-i) = 0$$

Équation 33

Où α et β sont des constantes positives.

L'algorithme Rétinex Variational utilise cette équation pour estimer l'image d'illumination l.

Ici, nous avons traité un seul canal. Lorsque notre image est en couleur, il y a deux façons d'extraire l'image d'illumination à l'aide de l'algorithme Rétinex Variational selon l'espace de couleur de l'image ooriginale : Le Rétinex RGB et le Rétinex HSV. Dans le Rétinex RGB, l'image colorée est décomposée en trois canaux : Rouge R, Vert G et Bleu B. Alors, l'algorithme du Rétinex Variationnel est appliqué sur chaque canal de couleur séparément pour obtenir lR, lG, lB. Ensuite, ces trois images sont combinées pour obtenir l'illumination colorée l [45]. Pour le Retinex HSV, l'image colorée est décomposée en trois couches : La teinte H, la saturation S et la valeur V. La couche V

correspond à la luminosité du pixel ; il est donc logique d'appliquer la variation du Retinex à V uniquement, car nous essayons d'extraire l'image d'illumination. Dans ce travail, nous avons utilisé le RGB Retinex.

- *Algorithme de correction de l'illumination :*

Comme le montre la Figure 24, nous commençons par séparer les images d'illumination et de réflectance à l'aide de l'algorithme Rétinex Variationnel. Ensuite, nous corrigeons l'illumination par un facteur Gamma pour obtenir une nouvelle image d'illumination et nous la multiplions par R, ce qui donne la version améliorée de l'image originale I'.

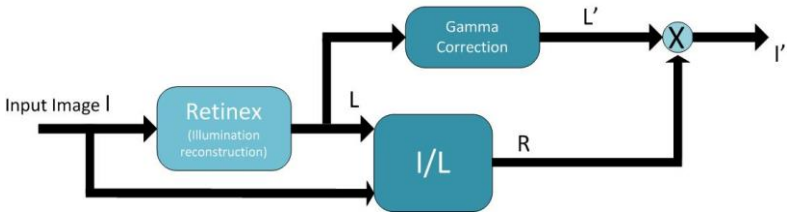


Figure 24. Algorithme de correction de l'éclairage

Le facteur Gamma [46] est donné par :

$$L' = W.\left[\frac{L}{W}\right]^{\frac{1}{\delta}} \qquad \text{Équation 34}$$

Où W est la valeur du blanc « White » (égale à 255 dans les images 8 bits).

L'image de sortie I' est :

$$\begin{aligned} I' &= L'.R \\ &= \frac{L'}{L}.I \\ &= W.\frac{(L/W)^{1/\delta}}{L}.I \\ &= \frac{I}{(L/W)^{1-1/\delta}} \end{aligned}$$

Équation 35

- *Image du quotient morphologique (IQM) :*

Récemment, les méthodes basées sur l'image du quotient sont considérées comme une solution simple et efficace au problème des variations d'éclairage. [79] a proposé une méthode basée sur l'image du quotient morphologique (IQM). Elle utilise la théorie morphologique mathématique et la technique de l'image du quotient pour la normalisation de l'éclairage.

L'IQM est définie par la relation :

$$MQI = I(x,y)/M(x,y) \qquad \text{Équation 36}$$

Où I(x,y) est l'image originale et M(x,y) est sa version lissée. Nous utilisons une opération morphologique pour lisser l'image. Dans notre cas, nous avons utilisé l'opérateur d'ouverture, qui est défini comme une érosion suivie d'une dilatation avec le même élément structurant.

- *Techniques d'amélioration du contraste :*

Il existe certaines méthodes d'amélioration du contraste pour résoudre les problèmes d'éclairage :

La Transformation Top-Hat est l'une des techniques d'amélioration du contraste basée sur la morphologie. En morphologie mathématique et en traitement numérique des images, la transformation Top-Hat est une opération qui permet d'extraire de petits éléments et détails d'images données. Le filtrage Top-Hat calcule l'ouverture morphologique de l'image (l'ouverture est la dilatation de l'érosion), et soustrait le résultat de l'image originale [80].

Dans l'**Amélioration du contraste local**, nous prétraitons les images originales en exploitant l'amélioration du contraste local. La gamme d'intensité est très large pour une image éclairée par une source lumineuse irrégulière. Et la distribution de la valeur des données est déséquilibrée. Le contraste local pour un pixel (m, n) avec la valeur de luminance I (m, n) est donné par :

$$Y(m,n) = \begin{cases} \log(I(m,n)/\overline{I(m,n)}), I(m,n) > \theta, \overline{I(m,n)} > \theta \\ 0, otherwise \end{cases} \qquad \text{Équation 37}$$

θ est un seuil prédéfini, $\overline{I(m,n)}$ indique la valeur de luminance moyenne du voisinage du pixel (m, n) [81].

L'**Ajustement de l'intensité** est une technique d'amélioration de l'image qui fait correspondre les valeurs d'intensité d'une image à une nouvelle plage. Elle renvoie une image de taille égale à I avec les valeurs d'intensité ajustées afin d'augmenter le contraste de l'image. Dans notre cas, nous ajustons les valeurs d'intensité dans les images de notre base de données sur Matlab en utilisant la fonction Imadjust [82].

b- Élimination des ombres

L'existence de l'ombre d'un véhicule a un effet négatif sur la détection du véhicule. Si l'ombre n'est pas éliminée, elle va se confondre avec le véhicule et affecter la détection du véhicule. Comme les ombres se produisent par manque de lumière dans certaines régions, nous essayons d'éliminer les ombres en fournissant plus de lumière aux régions ombrées. Nous détectons et séparons l'ombre de la région non ombragée, puis nous essayons de produire de la lumière pour la région ombragée.

Nous utilisons l'effet de minimisation de la fonction énergétique pour supprimer les ombres. La lumière nécessaire est supposée être une constante. Ainsi, la lumière constante est un vecteur à trois composantes, une composante pour chaque couleur : Rouge (c_R), Vert (c_G) et Bleu (c_B). Lorsque la valeur du vecteur $\vec{c}$ est ajoutée à la zone d'ombre, elle minimise la norme de la différence entre la lumière moyenne à l'intérieur (u_{in}) et la lumière moyenne à l'extérieur (u_{out}) de la zone d'ombre. La fonction énergétique est donnée par :

$$E(\vec{c}) = \left[(c_R.u_{in}^R - u_{out}^R)^2 + (c_G.u_{in}^G - u_{out}^G)^2 + (c_B.u_{in}^B - u_{out}^B)^2\right] \quad \text{Équation 38}$$

La solution de cette équation est la suivante :

$$\vec{c} = \left(\frac{u_{out}^R}{u_{in}^R}, \frac{u_{out}^G}{u_{in}^G}, \frac{u_{out}^B}{u_{in}^B}\right) \quad \text{Équation 39}$$

La méthode a été présentée dans [45] avec plus de détails.

2.4. Expérimentations & Résultats

Dans cette section, nous présentons les expériences réalisées et les résultats obtenus pour étudier le problème de la détection des véhicules. En effet, nous avons mené une série d'expériences pour évaluer les performances de différentes algorithmes utilisées pour la détection de véhicules. Deux expériences ont été envisagés dans notre travail. La première expérience visait à comparer les modèles construits en combinant les classifieurs SVM, kNN et l'arbre de décision avec les descripteurs HOG, SURF, LBP, Gabor Magnitude et Gabor Phase, afin de trouver les modèles qui fournissent des résultats significatifs et plus rapides. Ensuite, nous avons fusionné des descripteurs afin d'améliorer les résultats. Dans la seconde expérience, l'idée est d'appliquer des techniques de normalisation de l'illumination sur les images dans le but de rehausser les performances des modèles cités précédemment.

Nous présentons la structure du système de classification des images illustrée à la Figure 25. Il faut donc réaliser deux étapes principales : un processus d'apprentissage et un processus de test. Dans le processus d'apprentissage, tout d'abord, les caractéristiques sont extraites des données d'apprentissage pour stocker des informations discriminantes sur chaque image de véhicule, en utilisant les descripteurs tels que HOG, SURF, LBP, Gabor. Ensuite, il y a l'étape d'apprentissage ou de modélisation, dont l'objectif est de créer un modèle basé sur des algorithmes d'apprentissage automatique tels que SVM, k-NN ou l'arbre de décision. Dans le processus de test, on extrait des caractéristiques de l'image d'entrée puis on utilise le modèle déjà construit pour prédire, en conséquence, la classe appropriée (Véhicule / Non-véhicule).

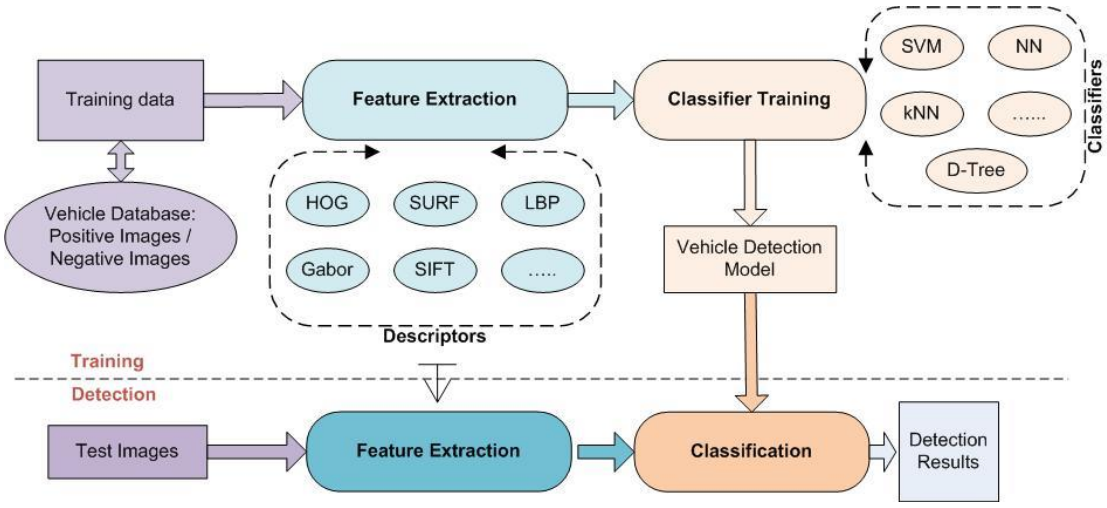


Figure 25. Système de détection des véhicules

La technique de la fusion des descripteurs se base sur la concaténation des caractéristiques. Elle donnerait à priori de meilleurs résultats que l'utilisation d'un seul descripteur, car elle vise à combiner les avantages complémentaires de tous les descripteurs utilisés pour construire un seul descripteur plus puissant, comme le montre la Figure 26. Ensuite, les caractéristiques obtenues sont entrées dans l'étape de classification.

Enfin, pour l'évaluation de la performance de nos modèles, nous avons calculé plusieurs critères présentés dans la suite.

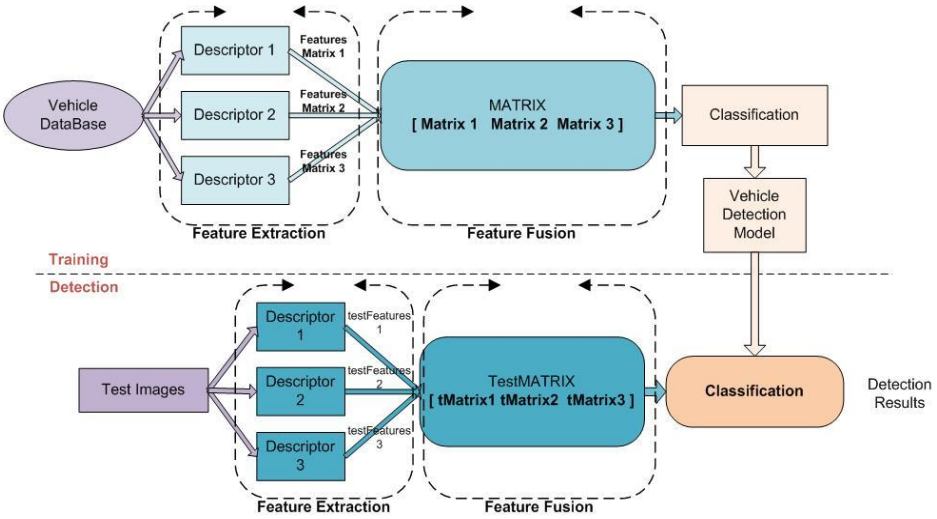


Figure 26. Système de fusion de descripteurs

2.4.1. Banc expérimental

Nous avons mené nos expériences en utilisant une base de données qui intègre plusieurs bases de données et images collectées sur le web. En particulier, nous avons utilisé la fameuse base de données GTI [83] comportant 7325 images : 3425 images de véhicules vue d'arrière, et 3900 images extraites de routes ne contenant pas de véhicules. En plus, nous avons ajouté aussi un ensemble d'images provenant d'autres bases de données (la base de données Caltech [84] [85] et de la base de données TU Graz-02 [86] [87], et des images collectées d'internet) afin d'augmenter le nombre d'images à 16 753 images réparties en deux classes. La première classe contient 7785 images positives de véhicules (vue arrière, Figure 27) et la deuxième classe contient 8968 images négatives (images qui ne contiennent pas de véhicules).

Les images de la base de données sont des images RGB. Nous avons redimensionné chaque image en 150×150 pixels. Donc, chaque image de l'ensemble de données est représentée par 150×150×3 = 67 500 entiers.

Toutes les images sont prises dans des conditions météorologiques différentes. En fait, 20 % de ces images ont été prises durant des journées ensoleillées, 20 % pendant les jours nuageux, 20 % dans des conditions moyennes (ni très ensoleillé ni nuageux), 20 % avec un mauvais éclairage, 10 % avec une pluie légère, 5 % avec des caméras de mauvaise résolution et 2,5 % dans des tunnels (avec une lumière artificielle) [88].

Le nombre d'images pour chaque classe doit être à peu près uniforme. Si nous avons deux fois, trois fois ou plus d'images d'une classe que d'une autre, notre classifieur sera naturellement biaisé à ces catégories fortement représentées.

Le déséquilibre des classes est un problème courant dans l'apprentissage automatique. Il existe plusieurs méthodes pour le surmonter mais la meilleure méthode pour éviter les problèmes dus au déséquilibre des classes est tout simplement de l'éviter.

La base de données est divisée en deux ensembles : un ensemble d'apprentissage et un ensemble de test. Quatre-vingt pour cent des images ont été inclus dans l'ensemble d'apprentissage. Les 20 % restants ont été inclus dans l'ensemble de test. Notre classifieur utilise l'ensemble d'apprentissage pour "apprendre" à quoi ressemble chaque catégorie. Une fois que le classifieur a été créé, on peut l'évaluer en utilisant l'ensemble de test.
Il est extrêmement important que l'ensemble d'apprentissage et l'ensemble de test soient indépendants l'un de l'autre et ne se chevauchent pas ! Si l'ensemble de test fait partie des données d'apprentissage, le classifieur va avoir un avantage injuste puisqu'il a déjà vu les exemples de test auparavant et en a appris.

Les algorithmes sont développés *sous* Matlab et implémentés sur une machine Lenovo ThinkPad avec un processeur Intel® Core™ i5 7ème génération de CPU @ 2.50GHz 2.71GHz, RAM 8Go.

Figure 27. Échantillons de la classe de véhicule (images positives)

2.4.2. Critères d'évaluation

Pour évaluer la performance des modèles construits, nous avons calculé les critères décrits ci-dessous. Pour cela, nous avons commencé par calculer les métriques suivantes : Vrais Positifs (TP pour True Positives), Faux Positifs (FP pour False Positives), Faux Négatifs (FN pour False Negatives) et Vrais Négatifs (TN pour True Negatives). On dit qu'un résultat est vrai positif lorsqu'un élément est correctement prédit comme positif, faux positif lorsqu'un élément est prédit comme positif alors qu'il ne l'était pas, faux négatif lorsqu'un élément est prédit comme négatif alors qu'il était en réalité positif, et vrai négatif lorsqu'un élément est correctement prédit comme négatif [89].

Ensuite, le taux de détections correctes (Équation 40), le taux d'extra-détections (Équation 41) et le taux de détections manquées (Équation 42) sont calculés comme suit :

$$Taux\ Détections\ correctes = \frac{TP+TN}{TP+TN+FP+FN} \quad \text{Équation 40}$$

$$Taux\ d'extra-détections = FP \quad \text{Équation 41}$$

$$Taux\ de\ détections\ manquées = FN \quad \text{Équation 42}$$

2.4.3. Résultats et discussion

a- Expérience 1 : Etude comparative

Cette expérience vise à comparer les modèles construits et à fusionner les descripteurs afin d'améliorer les résultats. Pour évaluer les modèles, nous les avons tous testés avec la même liste d'images afin de conserver les mêmes conditions d'évaluation. Cette évaluation était basée sur la durée d'exécution et les taux de détection (voir Section 2.4.2).

Dans notre expérience, les paramètres des descripteurs (HOG, LBP et Gabor) sont définis comme valeurs par défaut. Pour les classifieurs, les paramètres sont définis comme suit : la fonction de noyau du SVM est défini comme un noyau linéaire, le paramètre k du classifieur k-NN est fixé à 4 et 1, et la métrique de distance est soit euclidienne (par défaut), soit cosinusoïdale.

Les durées moyennes d'exécution (par image) de tous les modèles sont affichées dans la Figure 28.

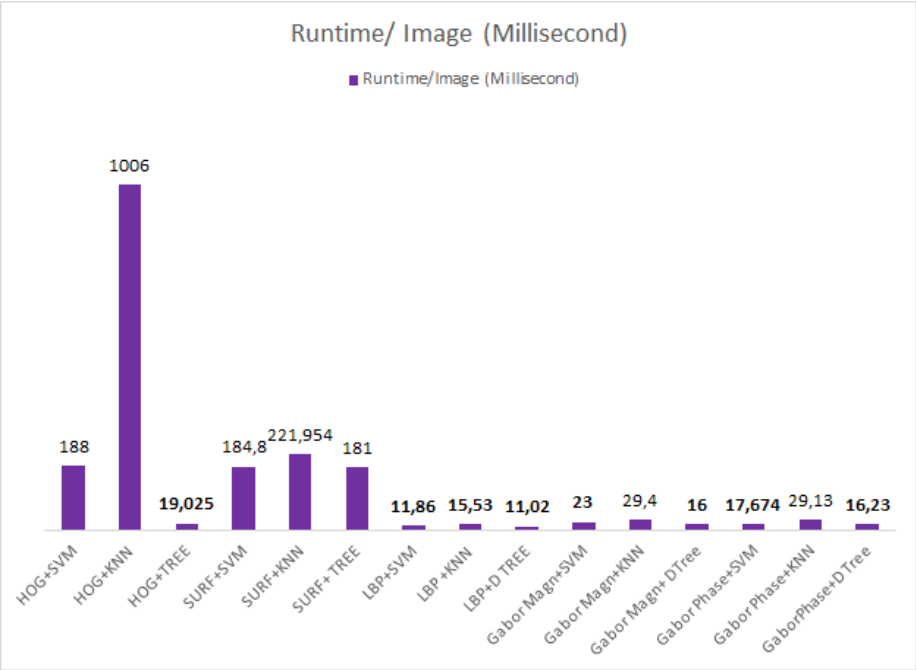


Figure 28. Comparaison des temps moyens d'exécution des modèles

Concernant le temps d'exécution, pour le cas de filtres de Gabor, chaque pixel est représenté par un ensemble de caractéristiques contenant plusieurs magnitudes avec leurs valeurs de phase correspondantes. Nous avons donc séparé deux types de caractéristiques : Les valeurs de phase ainsi que celles de magnitude.

Le modèle LBP+Arbre de décision est le plus rapide vu sa simplicité. Comme la complexité de cette méthode dépend de la profondeur et que la profondeur de cet arbre est ici de 1 (un arbre de décision de profondeur 1 est appelé souche de décision), il est donc plus simple qu'un classifieur linéaire. Pour les autres classifieurs, nous voyons que le SVM (classifieur linéaire) est un peu plus rapide que le classifieur k-NN. Plus le paramètre k augmente, plus le nombre de voisins augmente, et donc plus de temps est nécessaire pour classer les données. Même constat si on utilise la distance cosinusoïdale au lieu de la distance par défaut.

Les caractéristiques de LBP, Gabor Magnitude et Gabor Phase nécessitent moins de temps pour être extraites que celles de HOG et SURF. En effet, les histogrammes ont besoin de plus de temps pour être calculés. Le descripteur SURF est le plus lent ici, car il est basé sur un paquet de caractéristiques calculé à travers une série d'étapes telles que l'extraction des caractéristiques, le codage des caractéristiques, le regroupement des caractéristiques et la préservation des informations spatiales.

On constate que les modèles les plus rapides sont, par ordre décroissant : LBP+Arbre de décision, LBP+SVM, LBP+kNN (D=Euclidien), GaborMagnitude+Arbre de décision, GaborePhase+Arbre de décision, GaborPhase+SVM et HOG+Arbre de décision.

La Figure 29 résume la comparaison des performances de tous les modèles en indiquant les pourcentages de détections correctes, d'extra-détections et de détections manquées. Comme le montre cette figure, les descripteurs LBP, HOG et GaborPhase sont beaucoup plus performants que les autres descripteurs. Cela est dû au fait que, les caractéristiques de texture comme celles de LBP sont résistantes aux changements d'éclairage et aux ombres. Le HOG est une grille dense ; il est utilisé comme caractéristiques de bas niveau. Il est insensible aux changements géométriques et photométriques [90]. Pour le filtre de Gabor, les phases ont l'avantage d'être très résistantes aux changements d'illumination par rapport aux magnitudes [91].

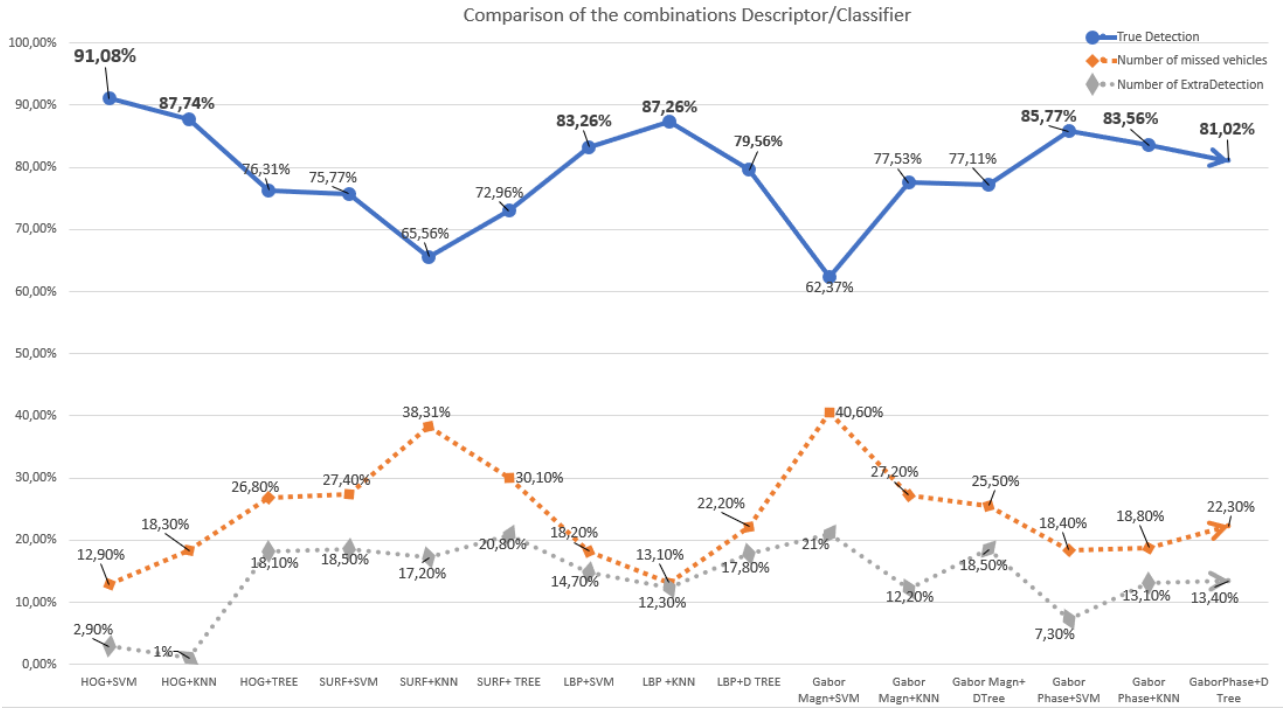


Figure 29. Comparaison des modèles construits (données de test)

Tous les descripteurs fournissent de très bonnes performances en combinaison avec les classifieurs SVM et k-NN sauf Gabor Magnitude.

Durant notre expérience, nous remarquons que : Pour le classifieur k-NN, si nous augmentons légèrement le paramètre k, le classifieur fonctionne un peu mieux. Le classifieur k-NN avec la distance cosinus se comporte de manière légèrement similaire à celui qui utilise la fonction de distance euclidienne. Ainsi, le classifieur k-NN avec les

paramètres (k=1 & Distance=Euclidienne) ou (k=4 & Distance=Cosine) donne les meilleurs résultats.

L'un des principaux avantages de l'algorithme k-NN est qu'il est extrêmement simple à implémenter et à comprendre. De plus, le classifieur ne prend absolument pas de temps à former, puisqu'il suffit de stocker nos points de données dans le but de calculer ultérieurement les distances et d'obtenir la classification finale [50]. Cependant, cette simplicité se paie au moment de la classification. La classification d'un nouveau point de test nécessite une comparaison avec chaque point de données d'apprentissage, ce qui rend le travail avec des grandes ensembles de données plus prohibitif sur le plan du calcul [50]. On note aussi que l'algorithme k-NN ne "apprend" rien ; l'algorithme n'est pas capable de se rendre plus intelligent s'il fait des erreurs ; il se base simplement sur les distances dans un espace à n dimensions pour faire la classification [50].

Nous avons aussi observé que si nous augmentons le paramètre k du classifieur k-NN, l'erreur relative aux données d'apprentissage commence à augmenter et qu'avec l'augmentation des données d'apprentissage, l'exactitude du modèle augmente aussi et le taux d'erreur diminue.

Le descripteur SURF est le moins efficace parmi tous les descripteurs que nous avons testés. SURF est moins efficace lorsqu'il est combiné avec k-NN, mais il fournit de meilleurs résultats lorsqu'il est combiné avec le SVM ou avec l'arbre de décision. Nous avons observé aussi que le classifieur de l'arbre de décision fonctionne bien en combinaison avec GaborPhase.

En se basant sur les résultats des données de test (Figure 29), les dix meilleurs modèles par ordre décroissant sont les suivants : HOG+SVM, HOG+kNN (k=1), LBP+kNN (k=1), LBP+kNN (k=4&D=Euclidien), GaborPhase+SVM, GaborPhase+kNN (k=4&D=Euclidien), HOG+kNN (k=4&D=Euclidien), GaborPhase+kNN (k=1 & D=Euclidien), LBP+SVM et GaborPhase+Arbre de décision.

Les changements de lumière et le bruit ont une grande influence sur l'image et surtout sur l'extraction des caractéristiques. Un seul descripteur risque d'être faible. La combinaison de plusieurs descripteurs serait donc capable à priori d'améliorer l'exactitude de la détection ou la classification des véhicules car chaque descripteur a ses forces et ses faiblesses face à chaque problème. Les meilleurs descripteurs HOG, LBP et Gabor Phase sont utilisés pour combiner leurs différentes puissances afin de construire un modèle de classification plus performant. Le HOG est un descripteur

robuste mais il présente des inconvénients lorsque l'éclairage varie. Les descripteurs LBP et Gabor Phase peuvent compenser le manque de HOG grâce à leur efficacité en cas de changement d'éclairage [91].

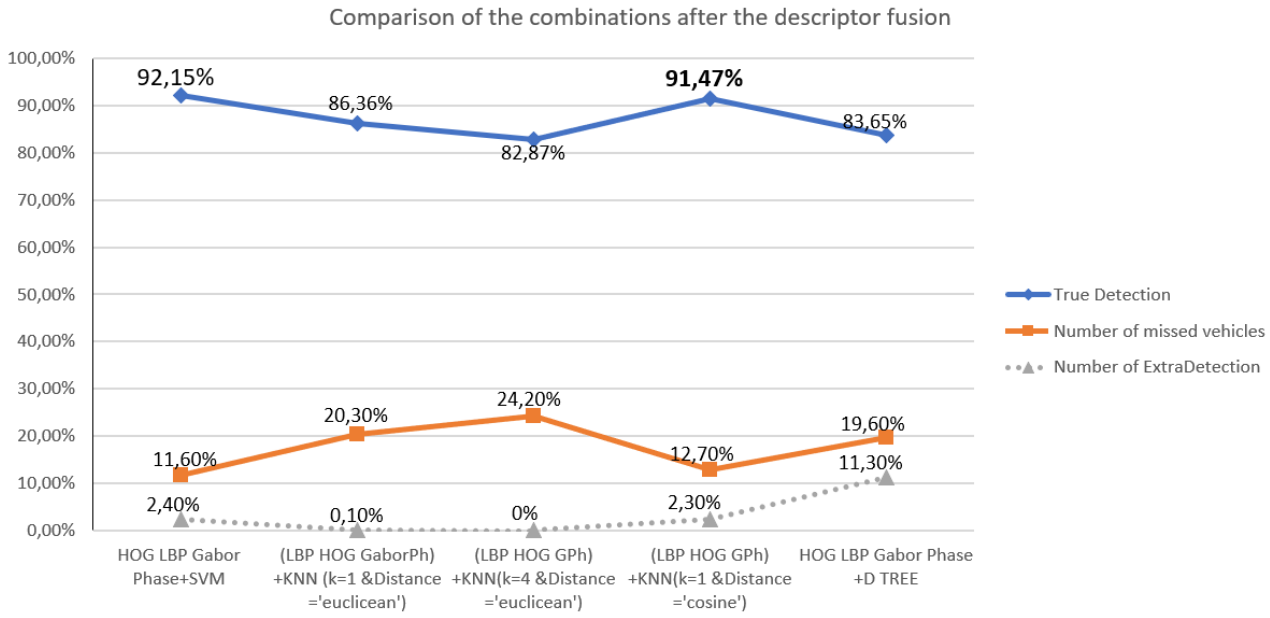


Figure 30. Comparaison des modèles après l'étape de fusion des descripteurs

Tous les descripteurs utilisés produisent des détections fausses positives, causées par les changements d'éclairage et les ombres, et des détections fausses négatives dues au camouflage (lorsque l'arrière-plan et le premier plan partagent des couleurs similaires). Alors, la fusion de ces descripteurs réduit le taux de détections fausses positives (extra-détections).

Comme l'environnement routier subit de nombreuses variations telles que les conditions météorologiques, les changements de lumière au cours de la journée, les changements des sources d'éclairage naturelle ou artificielle, il est nécessaire de développer des algorithmes de traitement d'images capables de détecter les véhicules de manière robuste et fiable et d'extraire des informations efficaces.

b- Expérience 2 : Correction de l'illumination

Dans cette expérience, notre objectif est d'améliorer les performances du système de détection des véhicules en traitant les problèmes d'éclairage et d'illumination. Alors, nous avons commencé par l'amélioration des images de la base de données en améliorant les effets visuels et éliminant l'impact d'éclairage inégal. Pour cela, nous avons appliqué les méthodes de normalisation de l'éclairage sur toutes les images de

la base de données. Ainsi, nous avons corrigé l'éclairage en se basant sur l'algorithme de Rétinex Variationnel et l'Image du Quotient Morphologique. Nous avons utilisé aussi trois techniques d'amélioration du contraste : Filtrage Top-Hat, amélioration du contraste local et ajustement de l'intensité. Ce traitement permet de réduire l'écart entre la route sombre et les autres véhicules sur la route afin d'améliorer le pouvoir de description des caractéristiques extraites pour détecter le véhicule, quelles que soient la lumière et les conditions de la route. Pour les problèmes d'ombre, nous essayons d'éliminer les zones d'ombre en se basant sur le concept de minimisation de la fonction énergétique.

La performance de la normalisation de l'éclairage est évaluée en fonction des taux de détection en utilisant les meilleures combinaisons obtenues dans la première expérience : (LBP, HOG, Gabor Phase) +SVM, (LBP, HOG, Gabor Phase) +KNN (*K*=1 & Distance=Cosine).

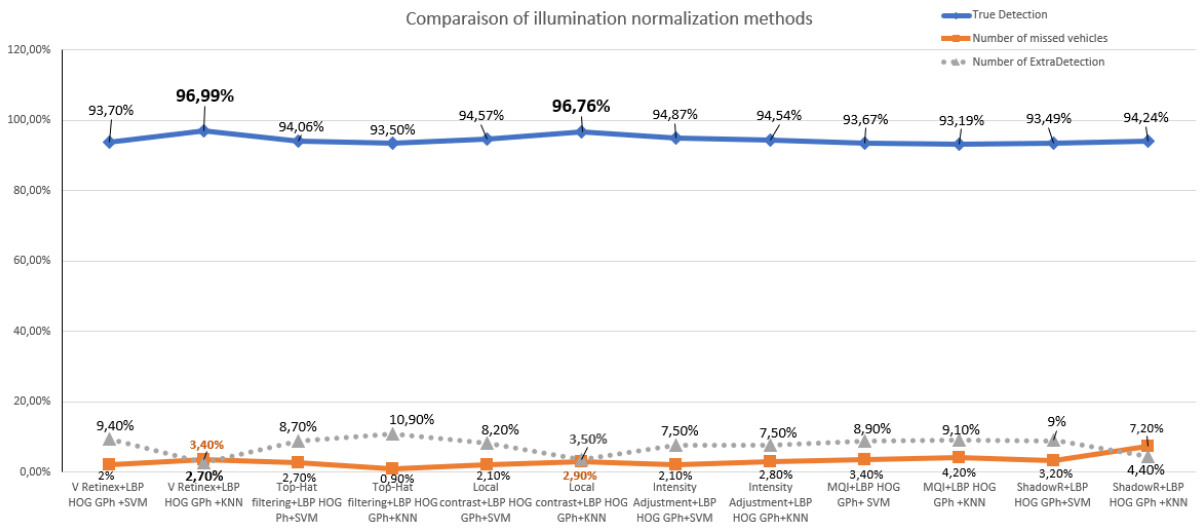


Figure 31. Comparaison des méthodes de normalisation de l'éclairage

La Figure 31 affiche les résultats obtenus dans cette expérience. On remarque que, pour toutes les méthodes utilisées, les taux de la détection sont améliorés par rapport à la Figure 30. Les méthodes de correction de l'illumination basée sur l'algorithme de Rétinex Variationnel et l'algorithme d'amélioration du contraste local ont de meilleurs résultats que les autres (les taux de détections passent à 96.99% et 96.76% respectivement pour les deux techniques). Plus précisément, les taux ont été améliorées comme suit : 5.52% et 5.29% pour les détections correctes, 3.4% et 2.9% pour les détections manquées, 2.7% et 3.5% pour les extra-détections en utilisant le modèle

LBP-HOG-GaborPhase +kNN. Ces deux méthodes permettent d'éviter efficacement les erreurs de détection.

La méthode basée sur l'algorithme de Rétinex Variationnel est très efficace pour corriger l'illumination. La Figure 32 montre quelques exemples lorsque l'illumination est très faible.

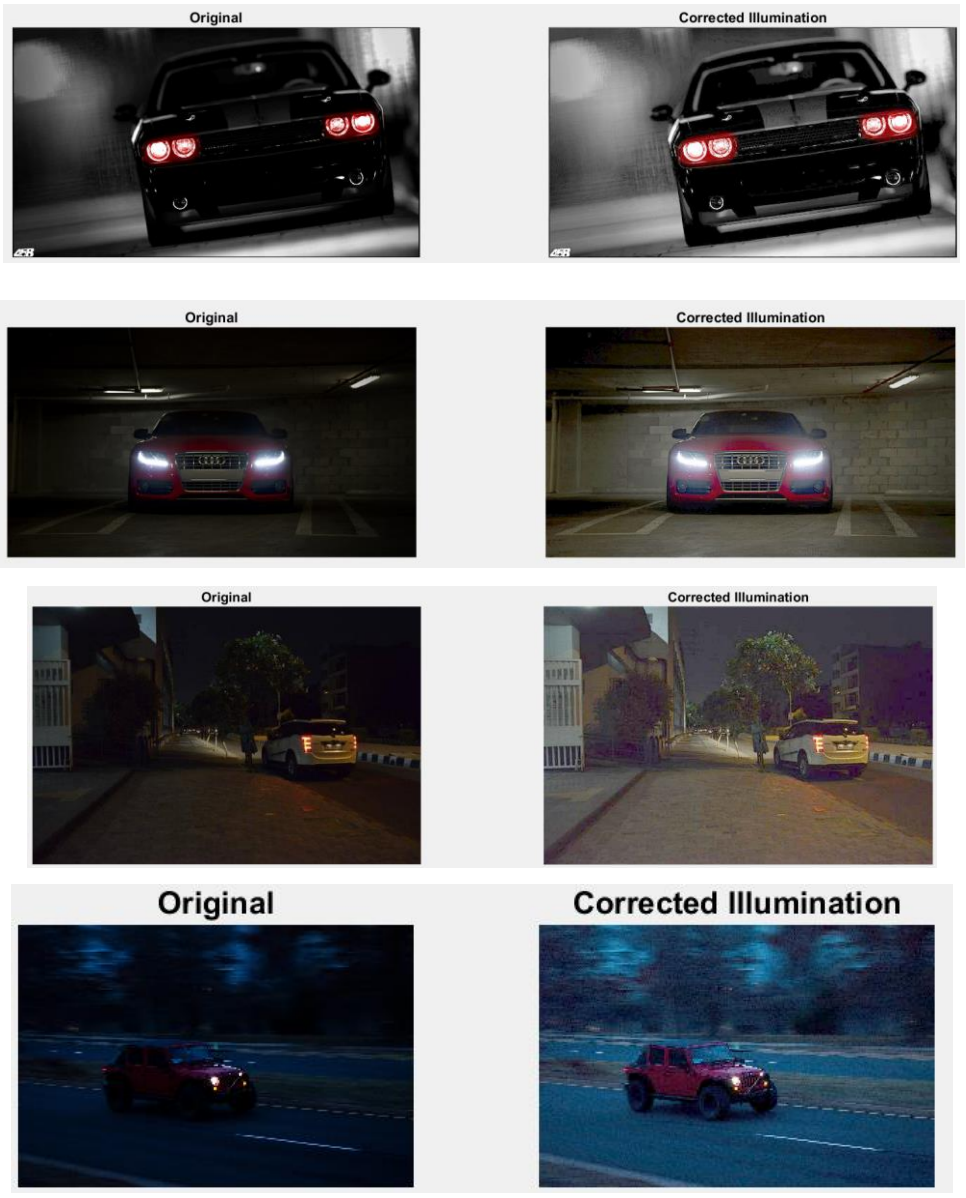


Figure 32. L'application de la méthode de la correction d'illumination basée sur l'algorithme de Rétinex Variationnel

Il n'est pas toujours possible de comparer les performances de plusieurs travaux, car ils ne sont pas tous évalués sur les mêmes bases de données. Cependant, [92] ont utilisé les bases de données CalTech et MIT pour entraîner les modèles ZF et VGG16 afin de détecter les types de véhicules. Ils ont trouvé pour le modèle ZF une exactitude de

détection de 79.9% et 84.0% pour 3052 et 5042 images, respectivement. Et pour le modèle VGG16, ils ont trouvé 82.3% et 84.4% pour 3052 et 5042 images, respectivement. [23] a atteint une exactitude de détection de véhicules de 83%.

2.5. Conclusion

Dans presque tous les systèmes de véhicules intelligents, la détection des véhicules constitue souvent la première étape. Pour cela, nous avons commencé par une étude comparative des méthodes de détection des véhicules. Nous avons comparé différents modèles en utilisant les descripteurs : HOG, LBP, SURF, Gabor Magnitude et Phase, et les classifieurs comme SVM, k-NN et l'arbre de décision. À partir des résultats obtenus, nous avons conclu que le modèle basé sur HOG et SVM fournit les meilleurs résultats sur les données de test avec une exactitude et une durée d'exécution par image d'environ 91,08% et 0,188s respectivement. Nous avons amélioré les performances du système en fusionnant plusieurs descripteurs. La meilleure fusion de descripteurs est HOG+ LBP+ Gabor Phase. Il donne de bons résultats avec le classifieur SVM avec une exactitude de 92,15 %. En outre, l'un des problèmes les plus importants de la détection ou la classification des véhicules est l'irrégularité de l'illumination. Ainsi, nous avons proposé d'utiliser plusieurs méthodes de la correction d'illumination. Les résultats obtenus montrent que les taux de détection sont améliorés. Les deux méthodes Retinex Variationnel et Local Contrast sont les plus efficaces et elles améliorent de manière significative l'exactitude du système (96,99%) et peuvent être très utiles pour notre système de détection global. Dans le prochain chapitre, nous allons travailler sur la classification des deux vues : vue arrière et vue avant des véhicules.

"I have worked all my life in Machine Learning, and I've never seen one algorithm knock over benchmarks like Deep Learning".

Andrew Ng- Professor at Stanford & Founder of deeplearning.ai & Co-Founder Coursera

3. Chapitre 2 : Classification des vues des véhicules

Sommaire

3.1. Introduction

La classification des véhicules est une partie essentielle des Systèmes de Transport Intelligents STI. Elle est récemment devenue un sujet d'étude actif pour les problèmes de la surveillance et contrôle de la circulation [93]. Différents modèles ont été développés pour détecter et classifier les véhicules selon plusieurs critères (catégorie, marque, modèle). Cependant, il n'y a pas de travaux qui traitent la classification des vues. Dans ce chapitre, nous visons à classifier les vues avant et arrière des véhicules, y compris les bus, les voitures, les motos et les camions. De nombreux facteurs rendent cette classification des vues avant et arrière très difficile, comprenant la similitude de forme, de taille et de couleur (Figure 33). Cette classification est l'une des principales parties de notre système global de détection des dépassements interdits. Cette classification peut être utilisée aussi dans différents buts tels que le stationnement automatique, la surveillance des rues à sens unique, le contrôle automatique des routes, les systèmes ADAS (Advanced Driver Assistance Systems), etc.

Figure 33. Vues arrière et avant : (a) vue arrière, (b) vue avant

Pour traiter le problème de la classification des vues, nous proposons deux approches : une approche descripteur-classifieur et une approche apprentissage profond (DL pour Deep Learning). Dans la première approche, deux types d'extracteurs de caractéristiques sont utilisés : HOG et LBP associés aux classifieurs : SVM, k-NN. La deuxième approche est une sous-catégorie de l'apprentissage automatique. Elle utilise plus précisément les réseaux neuronaux convolutifs en se fondant sur le training from scratch et l'apprentissage par transfert.

La classification des objets est un vaste domaine de recherche. Son but est de classer les objets en images dans des catégories significatives [94]. Il s'agit de prédire la classe d'un nouvel objet dans une image en se basant sur les informations obtenues à partir des données d'apprentissage avec des classes de catégories connues [95]. La classification des objets s'effectue essentiellement en deux étapes : l'extraction des caractéristiques et la classification. Dans les techniques traditionnelles d'apprentissage automatique, les caractéristiques sont extraites manuellement, mais les réseaux

neuronaux convolutifs les extraient automatiquement en passant par les couches. En fait, il existe de nombreuses architectures d'apprentissage profond comme les réseaux neuronaux profonds (Deep Neural Networks, DNN), les réseaux neuronaux récurrents (Recurrent Neural Networks, RNN) et les réseaux neuronaux convolutifs (Convolutional Neural Networks, CNN). Un réseau neuronal convolutif est l'un des réseaux neuronaux profonds les plus connus. Il est largement utilisé dans le domaine de la vision par ordinateur [96]. Son nom signifie que le réseau utilise une opération mathématique linéaire appelée convolution. Un CNN est constitué de paires de couches de convolution et de Pooling (regroupement), suivies d'un réseau neuronal entièrement connecté (Fully Connected layers, FC), constitué d'au moins une couche [97].

Les réseaux peuvent être conçus et entraînés à partir de zéro (training from scratch) ou à partir de modèles pré-entraînés. Dans la littérature, plusieurs réseaux pré-entraînés ont été créés pour des tâches telles que la classification d'images, de textes et de sons [98]. Réutiliser un réseau pré-entraîné pour une nouvelle tâche à travers le transfert de connaissances est appelé "Apprentissage par Transfert (Transfer Learning, TL)" [99]. Nous allons définir le TL dans les sections suivantes et expliquer comment les connaissances peuvent être transférées entre les tâches. L'apprentissage par transfert est récemment devenu de plus en plus populaire [98]. Il est généralement efficace, plus rapide et plus facile que la création à zéro d'un réseau avec des poids initialisés de manière aléatoire [100].

Dans ce chapitre, nous avons mené trois types d'expériences. Dans la première expérience, nous avons utilisé l'approche descripteur-classifieur pour la classification des vues de voitures. Dans la deuxième expérience, nous avons utilisé les deux approches (descripteur-classifieur) et l'apprentissage profond pour la classification des vues de véhicules. Dans la troisième expérience, nous avons utilisé l'apprentissage profond en exploitant le TL. Il y a deux scénarios dans cette expérience. Le premier scénario vise à appliquer le TL en utilisant le modèle Alexnet qui est un CNN pré-entraîné sur plus d'un million d'images (1000 catégories d'objets). Le second scénario vise à utiliser la création à partir de zéro (training from scratch) du réseau afin de prouver la robustesse de l'apprentissage par transfert en comparant les deux scénarios. Pour garder les mêmes conditions d'évaluation, dans la création à partir de zéro, nous avons essayé de concevoir un CNN inspiré de la structure d'AlexNet. Finalement, afin d'améliorer les résultats, nous avons décidé d'intégrer le classifieur SVM à la place du réseau entièrement connecté (FC) ou de le fusionner avec.

Le reste de ce chapitre est structuré comme suit. Dans la section 1, nous avons présenté certains travaux qui sont étroitement liés à la classification des véhicules. La section 3 décrit toutes les techniques utilisées dans ce travail. Nos résultats expérimentaux et notre discussion sont présentés dans la section 4. La dernière section conclut le travail et présente les travaux futurs.

La section suivante présente les travaux relatifs à la classification des vues de véhicules.

3.2. État de l'art

Dans la littérature, à notre connaissance, il n'y a pas de travaux antérieurs traitant la classification des vues de véhicules.

Nous présentons maintenant quelques travaux connexes qui sont étroitement liés au sujet de ce chapitre. Dans [101], les auteurs ont développé un système de reconnaissance des voitures qui combine deux méthodes. Toutes deux basées sur l'analyse des caractéristiques externes des voitures. La première met en évidence la forme de la vue arrière de la voiture, en explorant les caractéristiques : coefficient largeur/hauteur, une segmentation binaire de la forme de la voiture, une carte binaire des bords, obtenue par un filtrage de Sobel, et le contour extérieur de la voiture. La seconde méthode considère les caractéristiques calculées à partir des feux arrière de la voiture. Les auteurs ont expliqué que la plupart des feux de voiture ont le rouge comme couleur principale et plus pertinente. Par conséquent, la première étape est de trouver toutes les régions de couleur rouge apparaissant dans l'image segmentée de la voiture afin de détecter les feux arrière rouges et de les segmenter. Ces feux arrière peuvent être alors caractérisés en calculant plusieurs caractéristiques principalement liées à leur forme : excentricité, orientation, position, angle avec la plaque d'immatriculation et le contour extérieur. Finalement, les auteurs ont effectué la reconnaissance du fabricant et du modèle des voitures en se basant sur le calcul de la similarité des voitures qui combine la similarité de forme (utilisant des caractéristiques extraites des formes de la voiture) et la similarité de feux (utilisant des caractéristiques extraites des feux arrière). Les résultats des expérimentations montrent que les caractéristiques des feux sont plus efficaces car elles dépendent moins de la qualité de la segmentation de la voiture et sont donc censées être plus robustes que les caractéristiques des formes. Le système a atteint une exactitude de 89%. Alors ici, les auteurs n'ont pas donc exploité l'approche d'apprentissage automatique (classifieurs combinés avec des descripteurs). Ils ont plutôt choisi de calculer des caractéristiques spécifiques basées sur la forme et les feux arrière. Donc, pour reconnaître le modèle et le fabricant d'une voiture (classification),

ils ont calculé une métrique de similarité entre les caractéristiques de l'image d'entrée et celles des images de la base de données. Une valeur élevée signifie que les deux voitures ont le même modèle et le même fabricant. Toutefois [102] ont utilisé les algorithmes d'apprentissage automatique pour résoudre ce problème de classification des marques et des modèles de véhicules. Ils ont proposé une méthode basée sur le SVM linéaire et SIFT (Scale Invariant Transform Feature). La méthode proposée comprend les trois étapes : (1) l'extraction des caractéristiques à l'aide du SIFT afin de détecter les points d'intérêt. (2) Sac de mots : Il est utilisé pour représenter une image en tant que vecteur de caractéristiques de longueur fixe. (3) Machine à vecteur de support (SVM) : SVM est utilisé comme classifieur pour entraîner et tester le modèle proposé. Ainsi, la méthode proposée est évaluée sur une base de données publique NTOU-MMR sur la marque et le modèle de véhicule. Vingt-neuf marques et modèles de véhicules différents sont présents dans cet ensemble de données. L'ensemble de données d'entraînement contient 2 748 images tandis que 3 274 images sont données dans l'ensemble de données de test. L'algorithme a été implémenté à l'aide de MATLAB sur un processeur Intel Core i7 à 3,4 GHz et une mémoire de 16,00 Go. Le modèle a atteint une exactitude de 89 %.

Récemment, l'apprentissage par transfert est devenu de plus en plus populaire. Des modèles pré-entraînés ont été utilisés pour résoudre de nombreuses tâches, notamment dans le domaine de la classification des images. [103] a entraîné AlexNet et le VGG-16 en utilisant l'apprentissage par transfert et le training from scratch pour la reconnaissance des expressions faciales. Les résultats ont montré que la technique TL surpasse le training from scratch. Dans cette étude, les réseaux alexnet et vgg16 sont utilisés pour évaluer l'efficacité des méthodes d'apprentissage par transfert et de training from scratch sur le problème de la reconnaissance des expressions faciales. Les réseaux pré-entraînés alexnet et vgg16 ont utilisé pour la méthode d'apprentissage par transfert. En outre, une structure alexnet et un autre de vgg16 ont été régénérées avec des poids aléatoires afin d'être utilisées pour la méthode d'apprentissage à partir de zéro (training from scratch). Ainsi, quatre scénarios différents ont été élaborés : (1) training from scratch de alexnet, (2) apprentissage par transfert de alexnet, (3) training from scratch de vgg16, (4) apprentissage par transfert de vgg16. Les auteurs ont utilisé la base de données de visages Radboud (RaFD) qui a été divisée en trois parties, à savoir l'ensemble de formation (70%), de validation (%5) et de test (%25). RaFD contient 67 images de personnes avec différentes expressions et chaque expression a trois directions différentes du regard. L'ensemble de données contient des images

d'hommes, de femmes, d'enfants et d'adultes. Les résultats expérimentaux montrent que les approches d'apprentissage par transfert ont atteint une exactitude de 95 % et 98,33 %. L'architecture Alexnet a donné de bons résultats pour les approches d'apprentissage par transfert et de training from scratch. Mais l'architecture vgg16 avec le training from scratch n'a pas donné de bons résultats. Ils ont conclu que l'apprentissage par transfert surpasse l'approche de training from scratch pour alexnet et vgg16. Le meilleur score a été observé pour la méthode d'apprentissage par transfert de vgg16 avec une exactitude de 98,33 %.

Dans [104], les auteurs ont utilisé les CNNs en utilisant l'apprentissage par transfert pour surmonter le manque de données d'entraînement dans la tâche de classification des polypes coliques. Pour l'entraînement du CNN, les auteurs ont utilisé neuf bases de données différentes, dont trois bases de données endoscopiques, trois bases de données de texture et trois bases de données d'images naturelles. Ils ont confirmé que l'apprentissage par transfert peut être une alternative efficace pour extraire des caractéristiques pertinentes en exploitant les connaissances acquises sur d'autres ensembles de données, même dans des tâches très différentes. Ils ont prouvé aussi que lorsque le nombre de classes et la nature des images sont similaires à la base de données cible, les résultats sont meilleurs. Ensuite, les auteurs ont comparé les caractéristiques des CNNs avec les résultats obtenus par certaines méthodes d'extraction de caractéristiques pour la classification des polypes du côlon, à savoir : Blob Shape adapted Gradient using Local Fractal Dimension method (BSAG-LFD), Blob Shape and Contrast (Blob SC), Discrete Shearlet Transform using the Weibull distribution (Shearlet-Weibull), Gabor Wavelet Transform (GWT Weibull), Local Color Vector Patterns (LCVP) et Multi-Scale Block Local Binary Pattern (MB-LBP). Toutes ces méthodes d'extraction de caractéristiques (à l'exception de la méthode BSAG-LFD) ont été appliquées aux trois canaux RGB pour former l'espace vectoriel de caractéristiques final. Ils ont constaté que les CNN ont de meilleures performances que toutes les caractéristiques classiques, surtout lorsqu'ils sont entraînés avec plus d'images, ce qui est le cas du CNN AlexNet finement ajusté (fin-tuned) avec la base de données CALTECH101 avec deux classes (86,84% d'exactitude). L'application de la fusion de caractéristiques des deux meilleurs CNN (AlexNet fine-tuned avec dataset DTD, et AlexNet fine-tuned avec CALTECH101) avec les deux meilleures caractéristiques classiques (BSAG-LFD et GWT-Weibull), a présenté le meilleur résultat : 89,13% en moyenne, montrant que plusieurs caractéristiques de nature complètement différente peuvent se compléter.

[105] présente une étude comparative des performances entre les algorithmes d'apprentissage par transfert et les algorithmes d'apprentissage machine traditionnels sous la condition d'un déséquilibre des classes de domaines. La condition de déséquilibre des classes de domaines est caractérisée par le fait que les domaines source et cible ont des probabilités de classe différentes, ce qui peut entraîner des différences de distribution marginales entre les données source et cible. Les auteurs ont utilisé sept algorithmes d'apprentissage par transfert et cinq des algorithmes traditionnels d'apprentissage automatique. Les sept algorithmes d'apprentissage par transfert comprennent l'algorithme Adaptation Regularization Transfer Learning (ARTL), Graph Co-Regularization Transfer Learning (GTL), l'algorithme Geodesic Flow Kernel (GFK), l'algorithme Transfer Joint Matching (TJM), l'algorithme Joint Domain Adaptation (JDA), l'algorithme Transfer Component Analysis (TCA), et l'algorithme Transfer Kernel Learning (TKL). Tous les algorithmes d'apprentissage de transfert sont entraînés avec des données sources étiquetées et des données cibles non étiquetées. Dans les expériences, trois bases de données sont utilisées. La première est MAGIC Gamma Telescope dataset (appelé MAGIC). La deuxième est USPS Handwritten Digits dataset (appelé USPS). Chaque instance capture l'image d'un seul chiffre de zéro à neuf. Chaque chiffre (zéro à neuf) comporte 1 100 instances. Le chiffre 8 est sélectionné comme classe positive et les autres chiffres constituent la classe négative. La troisième est "Default of Credit Card Clients" dataset (appelé CCC). Les instances sélectionnées dans ce jeu de données sont celles qui représentent des individus âgés de 30 ans ou moins. Les cinq algorithmes traditionnels d'apprentissage automatique utilisés sont les suivants : SVM, k-NN, Random Forest (RF), Decision Tree with boosting (DT-R) et Logistic Regression (LR). Les auteurs ont implimenté leurs algorithmes avec Matlab, et les algorithmes RF et LR sont réalisés avec l'outil WEKA. Pour les algorithmes d'apprentissage par transfert avec une structure de conception non intégrée, leurs expérimentations ont montré que les performances sont faibles par rapport à l'apprentissage automatique traditionnel. Pour les algorithmes d'apprentissage par transfert avec une structure de conception intégrée, l'algorithme GTL montre une excellente performance globale.

Ces dernières années, les algorithmes de détection d'objets par apprentissage profond utilisant des images 2D sont devenus des outils puissants pour la détection d'objets routiers dans la conduite autonome. En fait, les techniques d'apprentissage profond appliquées à la détection des véhicules routiers ont obtenu des résultats remarquables. [16] compare cinq algorithmes d'apprentissage profond pour la détection de véhicules,

à savoir Faster R-CNN, R-FCN, SSD, RetinaNet et YOLOv3, sur l'ensemble de données KITTI en calculant le temps de détection et l'exactitude. Les auteurs ont divisé l'ensemble de données en deux parties, 90% des données ont été utilisées pour le training, et 10% des données ont été utilisées pour la validation. L'entraînement du modèle a duré 100 époques. Le deep learning framework Tensorflow a été utilisé pour le training sur un ordinateur portable équipé d'un processeur Core i5-7300HQ et d'un GPU GTX1050. L'algorithme de descente de gradient Adam a été sélectionné comme optimiseur pendant le training. Les résultats ont montré que le modèle RetinaNet a obtenu le meilleur taux d'exactitude et était très proche du modèle R-FCN. Cependant, RetinaNet avait un avantage énorme en termes de vitesse de détection par rapport à R-FCN. Elle a une précision de détection remarquable, car le modèle RetinaNet utilise une fonction de perte spéciale appelée FocalLoss, qui peut réduire efficacement le poids des échantillons facilement classés, de sorte que le modèle se concentre davantage sur les échantillons difficiles à classer pendant le training. Ils ont conclu aussi que RetinaNet est très adapté aux applications qui nécessitent une précision de détection et qui ont certaines exigences en matière de performances en temps réel de l'algorithme, telles que celles appliquées aux véhicules intelligents pour la détection des cibles routières. Ils ont constaté également que, bien que SSD soit un ancien algorithme de détection de cibles par apprentissage profond, il présente toujours un avantage considérable en temps réel par rapport à la version actualisée de YOLOv3. Car, ce modèle de détection, à part le réseau d'extraction de caractéristiques, est relativement simple, ce qui réduit considérablement la complexité du modèle. C'est aussi grâce à sa structure légère, qui a de faibles exigences en matière de puissance de calcul. Les auteurs considèrent que le modèle SSD est très adapté à divers types de plates-formes mobiles, comme les petits drones et les chariots logistiques, à la place des algorithmes traditionnels de détection de cibles comme la méthode de la fenêtre glissante.

3.3. Méthodologie

Le problème traité ici va au-delà d'un problème de reconnaissance de classes d'objets visuellement différentes (chat, voiture, personne, bus, chaise, moto, avion ...). Il s'agit de reconnaître des sous-classes de l'objet « véhicule » qui sont visuellement très proches. Comme nous avons déjà expliqué, nous avons construit un système de classification des vues avant et arrière des véhicules en utilisant deux approches. Cette section présente les méthodes utilisées dans la classification des vues. Nous commençons par présenter

la première approche (descripteur-classifieur) et puis la deuxième (apprentissage profond).

3.3.1. Approche 1 : Descripteur-Classifieur

Plusieurs facteurs rendent la reconnaissance des vues de face et de derrière très difficile, notamment la similitude de la forme, de la couleur et de la taille. Dans cette approche, nous avons utilisé les techniques classiques d'apprentissage automatique. La Figure 34 montre le processus de la construction de chaque modèle à partir des données. Ici, nous avons utilisé un ensemble de données contenant deux classes (vues avant et arrière). Pour les techniques de ML traditionnelles, les caractéristiques sont extraites manuellement. Plusieurs caractéristiques sont calculées pour caractériser la forme de la vue du véhicule à l'aide de descripteurs tels que HOG, LBP ou Gabor. Le vecteur de caractéristiques associé à chaque image entre dans le classifieur tel que SVM ou k-NN. Cependant, Les CNN extraient automatiquement les caractéristiques et utilisent la dernière couche pour la classifier les vues.

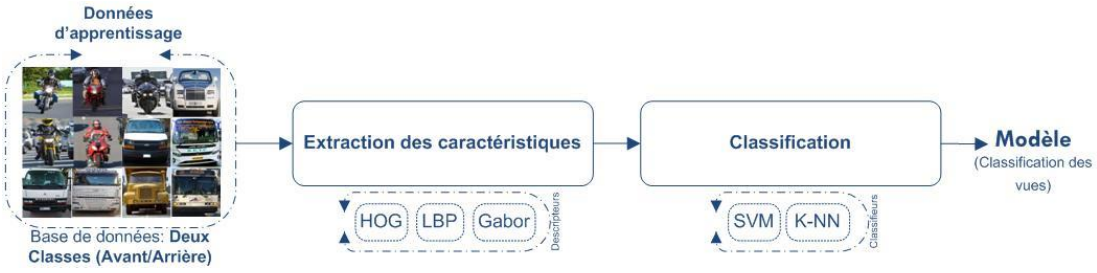


Figure 34. Schéma de construction d'un modèle de classification des vues de véhicules

Dans cette approche, nous avons utilisé les classifieurs (SVM et k-NN) combinés avec les descripteurs (HOG, LBP, et Gabor). Ces techniques sont toutes définies en détail dans le Chapitre 1.

3.3.2. Approche 2 : Apprentissage profond

L'apprentissage profond est un sous-domaine de l'apprentissage automatique, qui est lui-même un sous-domaine de l'intelligence artificielle (IA). Cette relation est représentée graphiquement par Figure 35. L'apprentissage profond appartient à la famille des algorithmes de réseaux neuronaux artificiels (Artificial Neural Networks) à couches multiples [106]. Les ANNs sont une classe d'algorithmes d'apprentissage

automatique qui apprennent à partir de données en s'inspirant de la structure et du fonctionnement du cerveau [50].

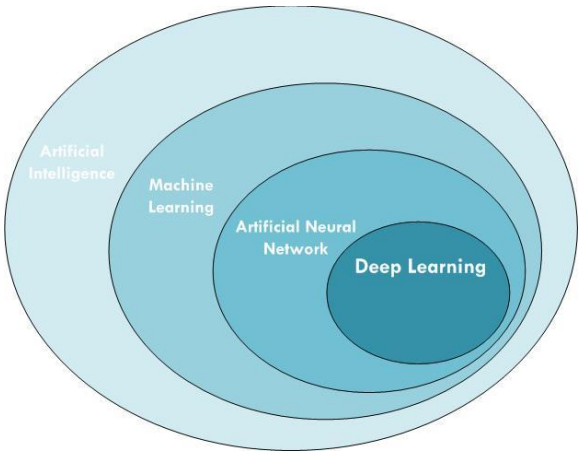


Figure 35. La relation entre l'intelligence artificielle (AI), l'apprentissage automatique (ML) et l'apprentissage profond (DL)

L'apprentissage profond a récemment suscité beaucoup d'intérêt, notamment dans le domaine de la vision par ordinateur, en particulier la classification des images.

Dans le contexte de l'apprentissage automatique appliqué à la classification d'images, l'objectif d'un algorithme d'apprentissage automatique est de prendre les ensembles d'images et d'identifier des motifs qui peuvent être utilisés pour distinguer diverses classes d'images/objets les uns des autres. Auparavant, nous utilisions des caractéristiques conçues à la main pour quantifier le contenu d'une image. Pour chaque image de l'ensemble de données, nous avons effectué une extraction de caractéristiques à l'aide d'un extracteur de caractéristiques ou descripteur d'image et à renvoyer un vecteur (c'est-à-dire une liste de nombres) visant à déterminer le contenu d'une image. L'apprentissage profond, et plus particulièrement les réseaux neuronaux convolutifs, adoptent une approche différente. Au lieu de définir manuellement des algorithmes pour extraire des caractéristiques d'une image, ces caractéristiques sont automatiquement extraites par les couches pendant le processus de training [50].

En fait, vous serez peut-être surpris d'apprendre que le domaine de l'apprentissage profond existe depuis plus de 60 ans et qu'il a connu différents noms et incarnations en fonction des tendances de la recherche, du matériel et des ensembles de données disponibles, ainsi que des options populaires des chercheurs éminents de l'époque. Dans la section suivante, nous présenterons un bref historique de l'apprentissage profond.

Historique :

L'histoire des réseaux neuronaux et de Deep Learning est longue et quelque peu confuse. Le DL remonte aux années 1940. Le DL n'apparaît comme nouveau que parce qu'il a été relativement impopulaire pendant plusieurs années avant sa popularité actuelle, et parce qu'il a connu de nombreux noms différents, pour ne s'appeler que récemment « Deep Learning » [107].

De manière générale, le DL a connu trois vagues de développement : DL connu sous le nom de « cybernetics » dans les années 1940-1960, DL connu sous le nom de « connexionnism » dans les années 1980-1990, et la renaissance actuelle sous le nom de Deep Learning à partir de 2006 [107]. La Figure 36 illustre cette évolution de manière quantitative. Elle montre ces trois vagues historiques de recherche sur les réseaux neuronaux artificiels, mesurées par la fréquence des expressions « cybernetics », « connexionnism » ou « neural networks », et « Deep Learning » selon Google Books Ngram Viewer.

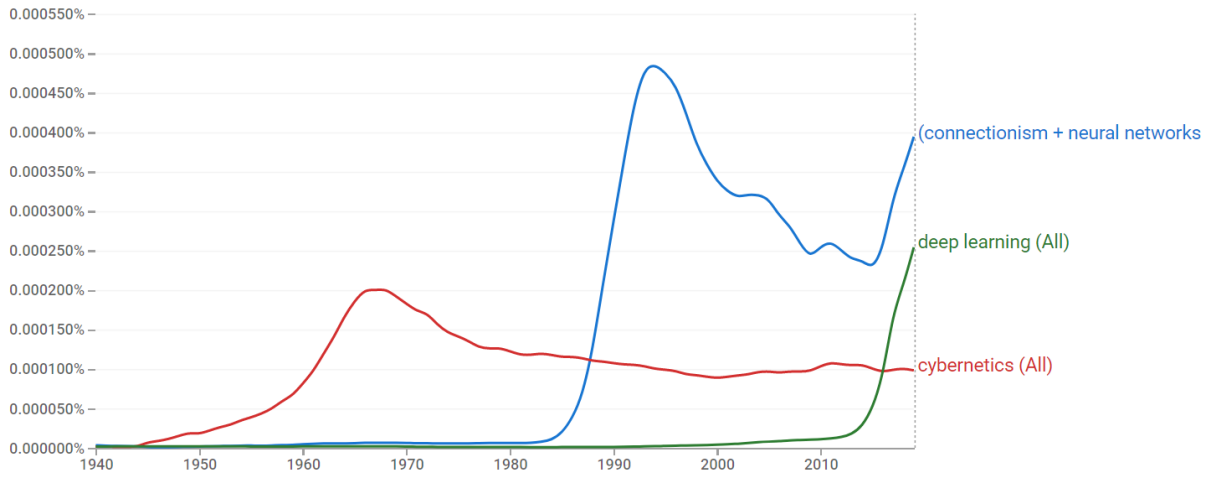


Figure 36. L'évolution de Deep Learning

Les ANNs sont inspirés du cerveau biologique (qu'il s'agisse du cerveau humain ou du cerveau d'un autre animal) et la façon dont ses neurones interagissent entre eux. Certes, ils ne sont pas des modèles réalistes du cerveau [50].

Les premiers prédécesseurs de l'apprentissage profond moderne étaient de simples modèles linéaires motivés par une perspective neuroscientifique. Ces modèles étaient conçus pour prendre un ensemble de n valeurs d'entrée *e1,..., en* et les associer à une sortie *s*. Ces modèles apprenaient un ensemble de poids *p1,..., pn* et calculaient leur sortie $f(e,p) = e_1.p_1 + \cdots + e_n.p_n$ Cette première vague de recherche sur les réseaux neuronaux était connue sous le nom de cybernétique [107], comme l'illustre la Figure 36.

Le premier modèle de réseau neuronal a été proposé par Warren McCulloch et Walter Pitts en 1943 [108]. Ce réseau était un classifieur binaire. Le problème était que les poids utilisés pour déterminer l'étiquette de classe pour une entrée donnée devaient être réglés manuellement par un humain [50].

Puis, dans les années 1950, Rosenblatt [109] [110] a publié l'algorithme fondamental du perceptron, un modèle capable d'apprendre automatiquement les poids nécessaires pour classer une entrée (sans intervention humaine). Un exemple de l'architecture du perceptron est présenté à la Figure 37 [50].

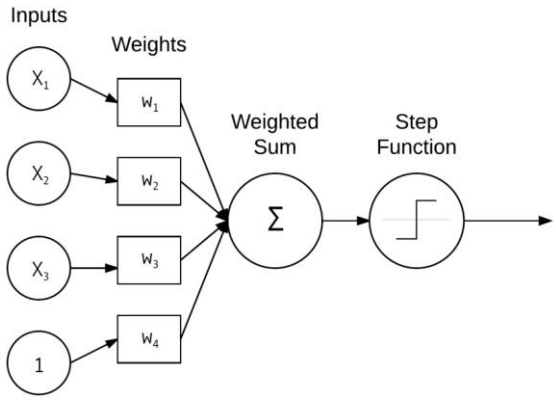


Figure 37. Un exemple de l'architecture simple du réseau Perceptron [50]

À cette époque, les techniques basées sur les perceptrons connaissaient un grand succès dans la communauté des réseaux neuronaux. Cependant, une publication de 1969 par Minsky et Papert [111] a effectivement bloqué la recherche sur les réseaux neuronaux pendant près d'une décennie. Leur travail a démontré qu'un Perceptron avec une fonction d'activation linéaire (quelle que soit la profondeur) n'était qu'un classifieur linéaire, incapable de résoudre des problèmes non linéaires.

De plus, les auteurs ont expliqué qu'à l'époque, ils ne disposaient pas des ressources informatiques nécessaires pour construire de grands réseaux neuronaux profonds. Ce seul article a presque tué la recherche sur les réseaux neuronaux [50].

Heureusement, l'algorithme de rétropropagation (backpropagation) et les recherches de Werbos (1974) [112], Rumelhart (1986) [113] et LeCun (1998) [114] ont permis de sauver les réseaux neuronaux de ce qui aurait pu être une disparition prématurée. Leurs recherches sur l'algorithme de rétropropagation ont permis de créer des réseaux neuronaux multicouches feedforward.

Avec les fonctions d'activation non linéaires, les chercheurs pouvaient désormais apprendre des fonctions non linéaires, ouvrant ainsi la voie à un tout nouveau domaine de recherche sur les réseaux neuronaux [50].

L'algorithme de rétropropagation est la base des réseaux neuronaux modernes. Il nous permet de former efficacement les réseaux neuronaux et de leur « apprendre » à apprendre de leurs erreurs. Cependant, à cette époque, en raison de (1) la lenteur des ordinateurs (par rapport aux machines actuelles) ; il n'y avait pas de GPU pour faciliter l'entraînement, et même les CPU étaient lents [115] et (2) manque de grands ensembles d'apprentissage étiquetés, les chercheurs étaient incapables de former (de manière fiable) des réseaux neuronaux comportant plus de deux couches cachées - c'était tout simplement infaisable sur le plan informatique [50].

Aujourd'hui, la dernière incarnation des réseaux neuronaux tels que nous les connaissons est appelée Deep Learning. Ce qui distingue le Deep Learning de ses incarnations précédentes, c'est que nous disposons de plus de données d'apprentissage. Les données devenaient disponibles et accessibles aux personnes en rendant publiques les bases de données comme ImageNet, CIFAR et MNIST. Parallèlement, les CPU devenaient plus rapides, et les GPU sont devenus un outil de calcul polyvalent. Nous pouvons aujourd'hui entraîner des réseaux ayant beaucoup plus de couches cachées et capables d'un apprentissage hiérarchique où des motifs simples sont appris dans les premières couches et des motifs plus complexes dans les couches supérieures du réseau [50].

Le réseau neuronal convolutif [116], appliqué à la reconnaissance de caractères manuscrits, est peut-être l'exemple le plus représentatif de l'apprentissage profond appliqué à l'apprentissage de caractéristiques. Il apprend automatiquement des motifs discriminants à partir d'images en empilant séquentiellement des couches les unes sur les autres. Les filtres des premières couches du réseau extraient des caractéristiques primitives comme les bords et les coins, tandis que les couches de niveau supérieur utilisent ces caractéristiques primitives pour apprendre des motifs plus complexes utiles à la classification des images [50]. Nous discuterons les réseaux neuronaux convolutifs dans la section suivante.

A. Réseaux neuronaux convolutifs (Convolutional Neural Networks, CNNs) :

Les réseaux neuronaux acceptent un vecteur d'image en entrée comme le montre la Figure 38 (il faut aplatir l'image de sorte à obtenir une colonne : un nœud d'entrée pour chaque pixel) et le transforment à travers une série de couches cachées. Chaque

couche cachée est également constituée d'un ensemble de neurones, où chaque neurone est entièrement connecté à tous les neurones de la couche précédente. La dernière couche (couche de sortie) est également entièrement connectée et représente les classifications de sortie finales du réseau. Cependant, ces réseaux neuronaux fonctionnant directement sur les intensités brutes des pixels ne s'adaptent pas bien à l'augmentation de la taille de l'image (exemple : une base de données où chaque image est de 227×227 pixels avec un canal rouge, vert et bleu, ce qui donne un total de 227×227×3 = 154 587 entrées totales pour le réseau et ce chiffre ne concerne que la seule couche d'entrée) et donnent donc de mauvais résultats [50]. Dans le cas d'images binaires extrêmement basiques, la méthode pourrait afficher un score d'exactitude moyen lors de la prédiction des classes, mais n'aurait que peu ou pas d'exactitude lorsqu'il s'agit d'images complexes. Un réseau feed-forward standard sur CIFAR-10 n'a obtenu qu'une précision de 15 % et ceci uniquement avec des images couleurs de 32x32 pixels.

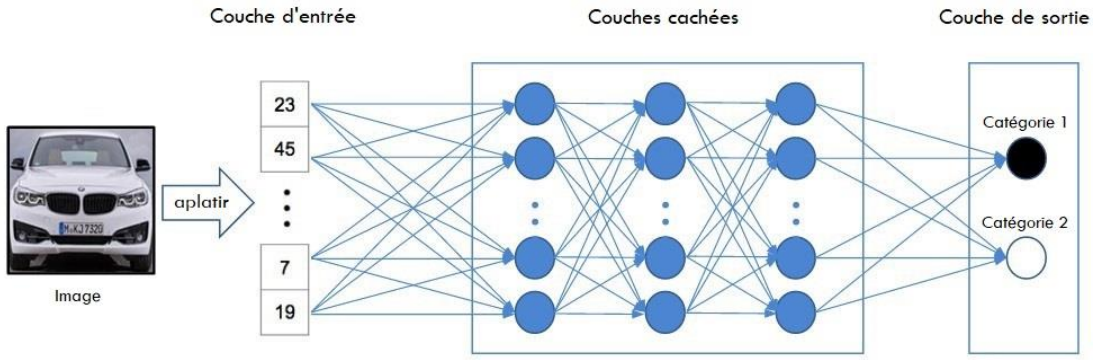


Figure 39. Intégration de l'image dans les réseaux neuronaux normaux

Au lieu de cela, nous pouvons utiliser des réseaux neuronaux convolutifs (CNN) qui utilisent, à la classification, des cartes de caractéristiques (feature maps) à la place des images originales. La « feature map » est l'image de sortie d'un filtre appliqué à l'entrée. Chaque élément de « feature map » est une sortie d'un neurone qui « regarde » seulement une petite région de l'entrée (Figure 40 inspirée de [106]). Les neurones sont « excités » et « activés » lorsqu'ils voient un motif particulier dans une image d'entrée. C'est pour ça le terme « activation map » est également utilisé comme un équivalent de « feature map ». Donc, ici chaque neurone est connecté à une région locale appelée le champ réceptif (receptive field) du neurone (au lieu de la structure entièrement connectée d'un réseau neuronal standard) - nous appelons cela la

connectivité locale, qui nous permet d'économiser une quantité énorme de paramètres dans notre réseau [50].

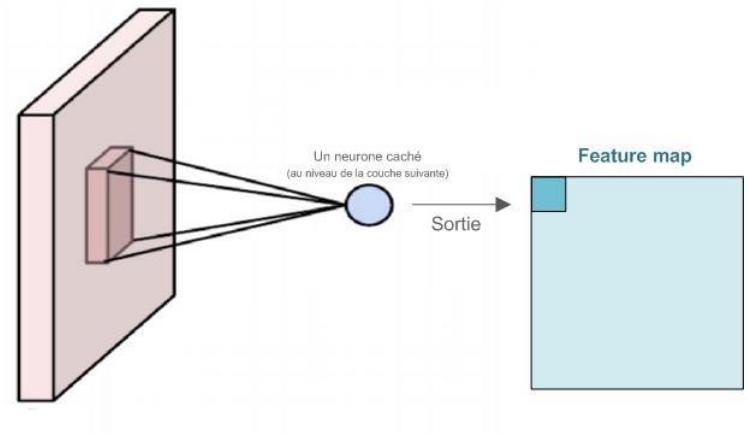


Figure 40. Un neurone dans un réseau neuronal convolutif

Si on a par exemple, l'image d'entrée a une taille de 227×227×3. Chaque image a donc une largeur de 227 pixels, une hauteur de 227 pixels et une profondeur de 3 (une pour chaque canal RGB). Si notre champ réceptif est de taille 5×5, chaque neurone de la couche de convolution se connectera à une région locale 5×5 de l'image pour un total de 5×5×3 = 75 poids.

Il existe plusieurs types de couches utilisées pour construire un réseau neuronal convolutif, notamment : Convolution, Activation, Pooling, Fully-Connected (FC), Batch Normalization (BN), Dropout. Il applique, par exemple (Figure 41), une couche de convolution, une couche d'activation, une couche de pooling, puis une couche entièrement connectée et, enfin, un classifieur softmax qui calcule les probabilités de classification en sortie. Le softmax est souvent omise du schéma du réseau car on suppose qu'elle suit directement le FC final.

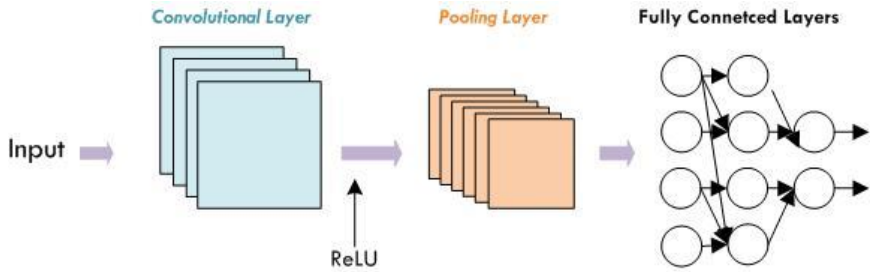


Figure 41. Une architecture simple d'un CNN

La couche de convolution convertit les images par l'opération de convolution. La couche de pooling réduit les dimensions de l'image. Elle combine les pixels voisins en un seul pixel [117]. Nous allons définir ces deux couches en détail dans les sections suivantes.

Parmi ces types de couches, la couche de convolution et de FC (et BN) sont les seules couches qui contiennent des paramètres appris pendant le processus de training - les autres couches sont simplement chargées d'effectuer une opération donnée. Les couches d'activation et de dropout ne sont pas ne sont pas techniquement des couches, mais sont souvent incluses dans les diagrammes de réseau pour rendre l'architecture explicitement claire. Les couches de pooling, d'importance égale à la couche de convolution et de FC, sont également incluses dans les diagrammes de réseau car elles ont un impact essentiel sur les dimensions spatiales d'une image lorsqu'elle se déplace dans un CNN [50].

De manière simple, le CNN est composé d'un extracteur de caractéristiques et d'un réseau neuronal de classification comme le montre la Figure 42. L'extracteur de caractéristiques est constitué de paires de couches de convolution et de Pooling. La sortie de la couche de convolution passe par une fonction d'activation comme ReLU, Sigmoïde ou Tanh. Le classifieur est un réseau de neurones entièrement connecté, constitué d'au moins une couche. Les résultats finaux de cette partie sont transformés en un vecteur unidimensionnel et entrent ensuite dans le réseau du classifieur qui génère la sortie [117].

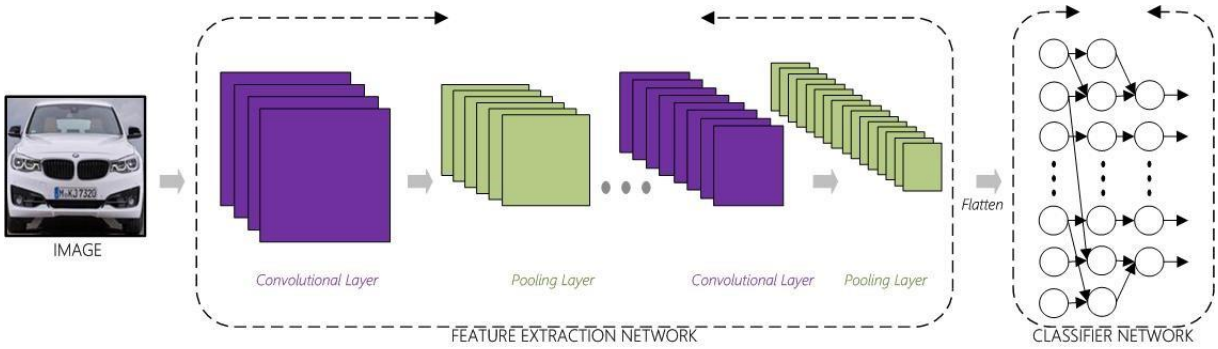


Figure 42. Architecture typique de CNN

a- Couche de convolution :

Le mot « convolution » semble être un terme compliqué, mais il ne l'est pas vraiment. Appliquer un flou ou un lissage à une image est une convolution. Et la détection des bords ? Oui, une convolution. Rendre une image plus nette ? est une convolution. Les

convolutions sont l'une des composantes les plus essentielles et fondamentales de la vision par ordinateur et du traitement des images [50].

La couche de convolution est le bloc de construction central d'un réseau neuronal convolutif [50]. Elle génère de nouvelles images appelées cartes des caractéristiques « Feature maps » en utilisant des filtres (des noyaux) de convolution. « Feature map » est une grille 2D des caractéristiques ; elle met en évidence les caractéristiques de l'image originale.

Les paramètres de la couche de convolution consistent en un ensemble de K filtres apprenables où chaque filtre est une matrice bidimensionnelle dont les éléments sont les poids cachés. Le nombre de Feature maps est identique à ce nombre K filtres (Figure 43).

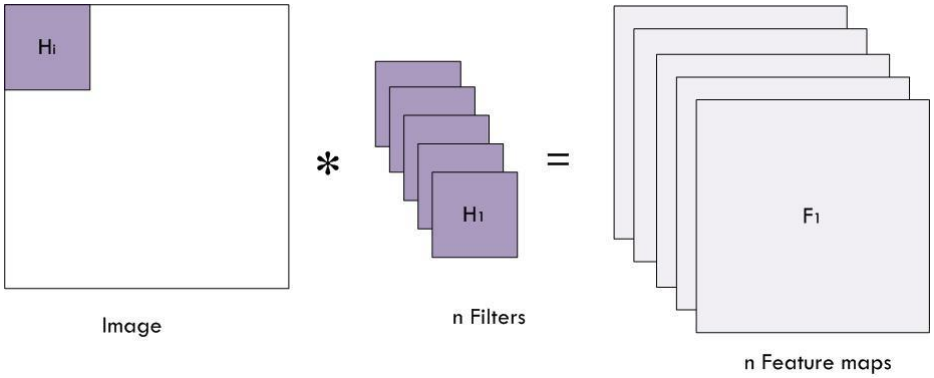


Figure 43. Dans chaque couche de convolution, n filtres sont appliqués à chaque image. Chaque filtre produit une carte de caractéristiques « Feature map »

L'opération de convolution est le plus souvent représentée par un astérisque. Comme le montre la Figure 44, le filtre commence par la région supérieure gauche. Chaque région est une sous-matrice qui a la même taille que le filtre convolutionnel. Ainsi, pour chaque sous-matrice, le résultat de la convolution est la somme des produits des valeurs ayant la même position avec les valeurs du noyau du filtre [117]. De la même manière, le filtre glisse sur l'image de gauche à droite et de haut en bas jusqu'à ce que la Feature map soit produite. La dimension de Feature map est présentée comme suit :

$$f = \frac{D - K}{S} + 1$$

D et K sont les dimensions de l'image et du filtre respectivement. S est le pas (Stride) qui correspond au nombre de pixels par lequel le filtre peut glisser sur l'image.

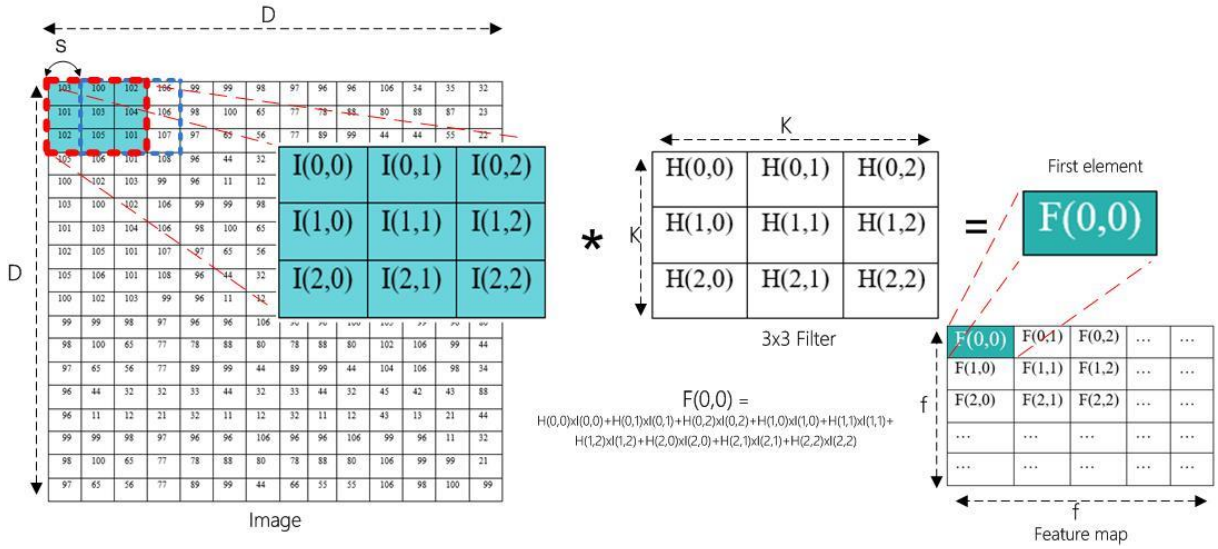


Figure 44. Opération de convolution

Dans les premières couches du réseau, les filtres peuvent s'activer lorsqu'ils voient des régions en forme de bord ou de coin. Ensuite, dans les couches plus profondes du réseau, les filtres peuvent s'activer en présence de caractéristiques de haut niveau, telles que des parties du visage, la roue d'une voiture, etc.

Si nous avons une image, nous lui appliquons une convolution, nous obtenons une combinaison de pixels en sortie, comme les bords (des caractéristiques primitives). Pour la deuxième couche de convolution, les entrées sont des bords, nous appliquons à nouveau la convolution, donc maintenant la sortie est une combinaison de bords, comme des coins et des contours, et puis des formes comme des carrées, des cercles et des rectangles, et ainsi de suite. Et finalement, ces formes sont regroupées pour former des parties d'objets jusqu'à identifier l'objet final.

Après chaque couche de convolution d'un CNN, nous appliquons une fonction d'activation non linéaire, telle que ReLU. Les couches d'activation ne sont pas techniquement des « couches » (du fait qu'aucun paramètre/poids n'est appris dans une couche d'activation) et sont parfois omises des diagrammes d'architecture de réseau car on suppose qu'une activation suit immédiatement une convolution [50].

b- Fonction d'activation :

Il est nécessaire d'utiliser les fonctions d'activation dans les CNN et les réseaux neuronaux artificiels. Sans elles, les CNN ne seraient qu'une série d'opérations linéaires. Une fonction d'activation est une transformation non linéaire de la sortie d'un neurone dans une couche, comme ReLU, Tanh et Sigmoid [118].

Unité linéaire rectifiée (Rectified Linear Unit, **ReLU**). La ReLU est la fonction d'activation la plus utilisée actuellement. Elle est présentée par l' Équation 43. Comme le montre la Figure 45, lorsque l'entrée est égale ou supérieure à zéro, la sortie est identique à l'entrée. Lorsqu'elle est inférieure à zéro, la sortie est égale à zéro [118].

$$f(x) = \max(0, x) \qquad \text{Équation 43}$$

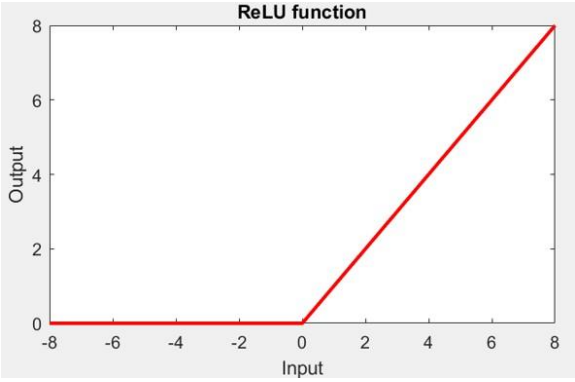


Figure 45. Rectified Linear Unit

Sigmoïde. La fonction Sigmoïde ou logistique est présentée par Équation 44 [118]. Comme le montre la Figure 46, le Sigmoïde a une sortie comprise entre les valeurs 0 et 1.

$$f(x) = \frac{1}{1+e^{-x}} \qquad \text{Équation 44}$$

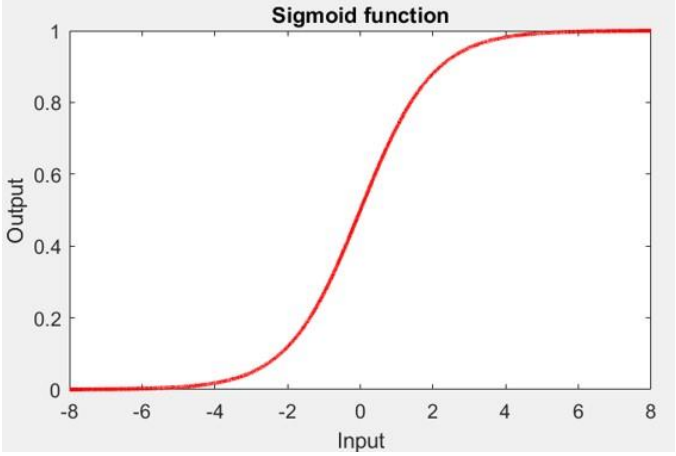


Figure 46. Fonction sigmoïde

Récemment, elle est devenue moins populaire en raison de plusieurs inconvénients, tels que le problème de la disparition de la pente.

Tanh. La fonction tangente hyperbolique (Tanh) est présentée par Équation 45 [118]. Comme nous le voyons sur la Figure 47, Tanh est très similaire à la

fonction sigmoïde, elle est centrée autour de 0. Tanh a une sortie comprise entre -1 et 1.

$$f(x) = \frac{1-e^{-2x}}{1+e^{-2x}}$$ Équation 45

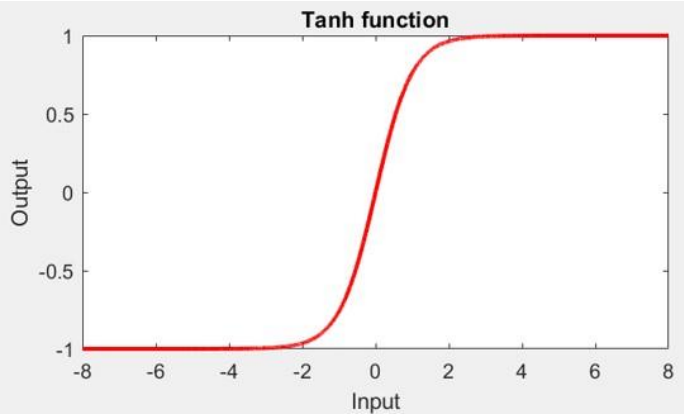


Figure 47. Tangente hyperbolique

En pratique, le Tanh est généralement préférable au sigmoïde, mais il souffre toujours du problème du gradient de disparition.

c- Couche Pooling :

La couche de Pooling réduit la taille des Feature maps. Elle remplace les pixels voisins par leur valeur maximale ou moyenne. La Figure 48 illustre les deux opérations.

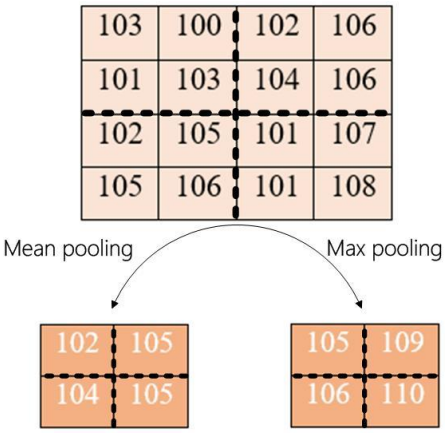


Figure 48. Les opérations de pooling

Mean pooling (Regroupement de moyennes). L'opération de Mean pooling compresse une « feature map » en prenant la valeur moyenne de chaque bloc (Équation 46) [119]. Le regroupement est effectué sur des blocs qui ne se chevauchent pas.

$$P_{i,x,y} = \frac{1}{d^2}\sum_{v,h} F_{i,(x+v),(y+h)}$$ Équation 46

$F_{i,x,y}$ est la valeur à la position *x,y* de la *ième* Feature map. *v* est un indice vertical dans le voisinage local. *h* est un indice horizontal dans le voisinage local.

Max Pooling. L'opération de max-pooling est similaire au mean-pooling, sauf qu'au lieu de prendre la moyenne, on prend le max exprimé par l'équation suivante (Équation 47) [119] :

$$P_{i,x,y} = max_{v,h}(F_{i,(x+v),(y+h)})$$ Équation 47

$F_{i,x,y}$ est la valeur à la position *x,y* de la *ième* feature map. *v* est un index vertical dans le voisinage local. *h* est un index horizontal dans le voisinage local.

d- Couches FC (Fully Connected):

Les neurones des couches FC sont entièrement connectés à toutes les activations de la couche précédente, comme c'est le cas pour les réseaux neuronaux feedforward. Les couches FC sont toujours placées à la fin du réseau (c'est-à-dire que nous n'appliquons pas une couche de convolution, puis une couche FC, suivie d'une autre de convolution) [50].

Il est courant d'utiliser une ou deux couches FC avant d'appliquer le classifieur softmax qui calcule les probabilités de sortie finales pour chaque classe.

e- Batch Normalization (La normalisation par lots):

Introduit pour la première fois par Ioffe et Szgedy dans leur article de 2015, Batch Normalization : Accelerating Deep Network Training by Reducing Internal Covariate Shift [120], les couches de Batch Normalization (ou BN en abrégé), comme leur nom l'indique, sont utilisées pour normaliser les activations d'un volume d'entrée donné avant de le transmettre à la couche suivante du réseau. Si nous considérons *x* comme notre mini-batch d'activations, alors nous pouvons calculer le $\hat{x}$ normalisé via l'équation suivante [50] :

$$\hat{x} = \frac{x_i - m_b}{\sqrt{{e_b}^2 + \varepsilon}}$$ Équation 48

Pendant le training, nous calculons m_b et e_b sur chaque mini-batch *b*, où :

$$m_b = \frac{1}{N}\sum_{i=1}^{N} x_i \qquad e_b = \frac{1}{N}\sum_{i=1}^{N}(x_i - m_b)^2$$

La Batch Normalization s'est avérée extrêmement efficace pour réduire le nombre d'époques nécessaires au training d'un réseau neuronal. Elle présente également l'avantage d'aider à « stabiliser » le training, La Batch Normalization permet également de réduire les pertes finales et d'obtenir une courbe de pertes plus stable [50].

Cela dit, il est préférable d'utiliser la normalisation par lot dans presque toutes les situations, car elle fait une différence significative. L'application de la normalisation par lots au réseau peut nous aider à éviter l'overfitting et nous permet d'obtenir une exactitude de classification supérieure en moins d'époques que pour la même architecture de réseau sans normalisation par lots [50].

Le principal inconvénient de la Batch Normalization est qu'elle peut ralentir le temps d'entraînement du réseau de 2 à 3 fois, en raison du calcul des statistiques et de la normalisation par lots.

f- Dropout :

Le dernier type de couche que nous allons aborder est le dropout. Le dropout est en fait une forme de régularisation qui vise à empêcher l'overfitting en modifiant explicitement l'architecture du réseau au moment du training. Pour chaque mini-lot (mini-batch) de notre ensemble de training, les couches de Dropout, avec une probabilité p, déconnectent aléatoirement les entrées de la couche précédente vers la couche suivante dans l'architecture du réseau. La Figure 49 visualise ce concept où nous déconnectons aléatoirement avec une probabilité p = 50% les connexions entre deux couches FC pour un mini-lot donné [50].

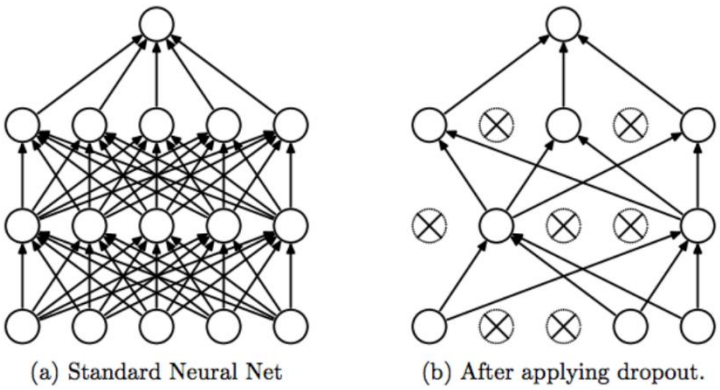


Figure 49. À gauche : un réseau neuronal standard sans dropout. Droite : Le même réseau après Dropout. Les unités barrées ont été supprimées.

Il est plus courant de placer des couches d'abandon avec p = 0,5 entre les couches FC d'une architecture. Toutefois, nous pouvons également appliquer le dropout avec des probabilités plus faibles (p = 10% : 25%) dans les couches précédentes du réseau [50].

Les réseaux neuronaux convolutifs sont un cas particulier de réseau neuronal « feedforward ». Dans ce type d'architecture, une connexion entre les nœuds est uniquement autorisée des nœuds de la couche i vers les nœuds de la couche i+1 (d'où le terme « feedforward »). Aucune connexion en arrière ou inter-couche n'est autorisée. Lorsque les réseaux feedforward incluent des *feedback connections* (connexions de sortie qui rétroagissent sur les entrées), ils sont appelés réseaux neuronaux récurrents. Nous présentons dans la section suivante ce type de réseaux [50].

B. Réseaux neuronaux récurrents (Recurrent Neural Networks, RNNs) :

Les réseaux neuronaux récurrents utilisent les informations séquentielles. Les séquences transmises au réseau peuvent se trouver en entrée, en sortie ou même dans les deux cas [121]. Pour traiter une séquence ou une série temporelle de points de données, on doit montrer la séquence entière au réseau en une seule fois : la transformer en un seul point de données [96].Le RNN traite les séquences de données à travers un "état" ou une "mémoire" [121]. Par contre, les réseaux entièrement connectés comme les CNNs n'ont pas de mémoire ; ils traitent chaque entrée indépendamment, sans conserver d'état entre les entrées. Les RNN contiennent des boucles dans la structure du réseau, comme l'illustrent les Figure 50 et Figure 51 [121].

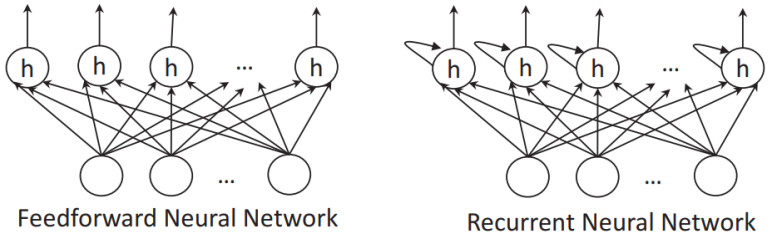


Figure 50. Les réseaux neuronaux récurrents comportent une boucle dans la structure du réseau et traitent les données par séquences [121].

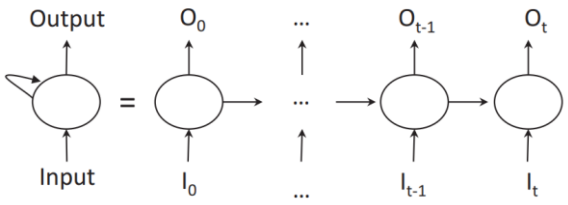


Figure 51. Le traitement d'information sur des séquences par les RNNs [121].

Les RNN ont été efficaces dans de nombreuses tâches de traitement du langage naturel (Natural Language Processing, NLP), car le sens d'un mot dans une phrase dépend des autres mots qui l'entourent. Les RNN ont également été utiles dans d'autres applications telles que la prédiction de séries temporelles, la reconnaissance vocale et la reconnaissance de l'écriture manuscrite [121].

La Figure 52 présente différents modèles de structure pour les RNN. La couche inférieure est constituée des entrées, au milieu se trouvent les états cachés, et la couche supérieure est constituée des sorties. La Figure 52(a) présente la structure typique d'un réseau neuronal classique de type feed forward, avec une entrée et une sortie ; la classification d'images en est un exemple. La Figure 52(b) montre le modèle « un-à-plusieurs », qui est la structure typique utilisée dans la description d'images, où l'entrée est une image et la sortie une séquence de mots décrivant l'image. La Figure 52(c) montre le modèle « plusieurs-à-un ». Une application de ce modèle est l'analyse des sentiments, où l'entrée est un texte et la sortie est un booléen (positif ou négatif). La Figure 52(d) montre le modèle synchrone « plusieurs-à-plusieurs ». Un exemple de ce modèle pourrait être le sous-titrage vidéo, où nous voulons attribuer une étiquette à chaque image vidéo. Enfin, la Figure 52(e) montre la représentation asynchrone « plusieurs-à-plusieurs », qui est le cas typique de la traduction automatique [121].

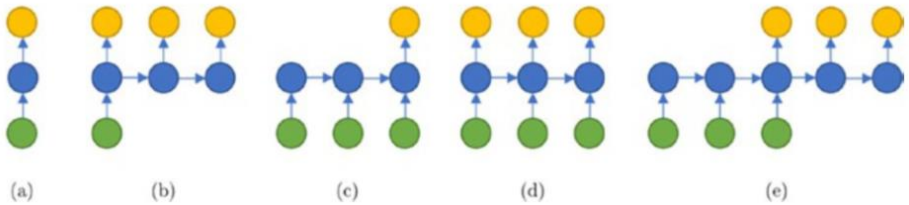


Figure 52. (a)One-to-one (b) One-to-many (c) Many-to-one (d) Many-to-many (e) Many-to-many [121].

Des variantes de RNN ont été proposées pour mieux traiter des séquences plus longues, comme les réseaux à mémoire à long terme (LSTMS) introduit par [122]. Aujourd'hui, le

LSTM est largement utilisé pour de nombreuses tâches de modélisation de séquences, y compris de nombreuses tâches de traitement du langage naturel chez Google [107].

C. Réseaux adverses génératifs (Generative Adversarial Networks, GANs) :

Les réseaux adverses génératifs (GAN), introduits en 2014 par [123], sont une approche de la modélisation générative utilisant des méthodes d'apprentissage profond.

Un GAN est un modèle génératif où deux réseaux distincts sont mis en compétition dans un scénario de théorie des jeux. Le premier réseau est le générateur, il génère de nouveaux exemples, tandis que son adversaire, le discriminateur, tente de classer les exemples comme réels (tirés de training data) ou bien générés. Le discriminateur émet une valeur de probabilité que l'exemple soit un réel échantillon de training plutôt qu'un faux échantillon tiré (généré) du modèle.

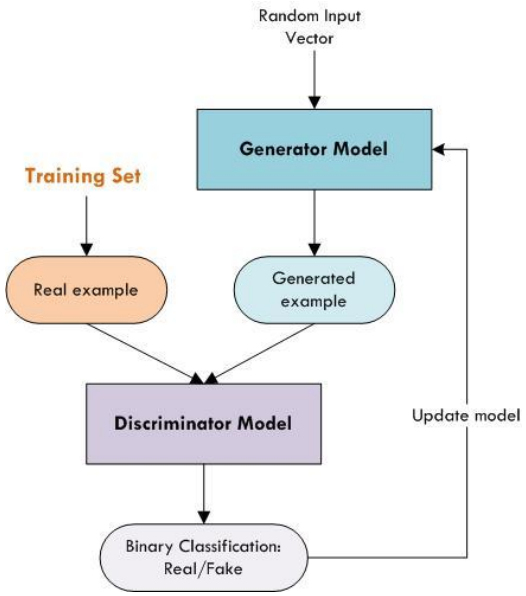


Figure 53. Architecture du modèle de GAN

Comme le montre la Figure 53, un GAN est constitué de deux parties :

- Réseau générateur : il prend en entrée un vecteur aléatoire, et le décode en une image synthétique.

- Le réseau discriminateur (ou adversaire) : il prend en entrée une image (réelle ou générée) et prédit si l'image provient de l'ensemble d'entraînement (training data) ou a été créée par le réseau générateur.

L'apprentissage dans les GANs peut être modélisé comme un jeu à somme nulle, dans lequel une fonction *f(g,d)* détermine le gain du discriminateur. Le générateur reçoit -*f(g,d)* comme son propre gain. Pendant l'apprentissage, chaque joueur tente de maximiser son propre gain.

Cela pousse le discriminateur à essayer d'apprendre à classer correctement les échantillons comme vrais ou faux (générés). Simultanément, le générateur tente de tromper le classifieur en lui faisant croire que ses échantillons sont réels. À la convergence, les échantillons du générateur sont indistinguables des données réelles [107].

Les GANs utilisent, comme modèles de générateur et de discriminateur, les réseaux neuronaux convolutifs (CNNs), les réseaux neuronaux récurrents (RNNs) ou simplement les réseaux neuronaux ordinaires (ANNs). Les GANs travaillent souvent avec des données d'image, donc les CNN sont les mieux adaptés à cette tâche.

Ces dernières années, les GAN ont montré un énorme potentiel et ont été appliqués dans divers scénarios, allant de la synthèse d'images, l'amélioration de la qualité des images (superresolution), les traductions d'image à image (image-to-image translations), à la génération de texte à image (text-to-image generation), et plus encore. En outre, les GAN sont les éléments constitutifs des avancées dans l'utilisation de l'IA pour l'art, la musique et la créativité (par exemple, la génération de musique, la génération de poésie, etc.) [121].

Selon Yann Lecun, directeur de la recherche en IA chez Facebook et professeur à l'université de New York, les GAN sont « l'idée la plus intéressante de ces dix dernières années en matière d'apprentissage automatique ».

La section suivante définit en détail l'approche d'apprentissage par transfert avec les réseaux neuronaux convolutifs.

D. Apprentissage par transfert

L'entraînement d'un modèle de Deep Learning à partir de zéro (training from scratch) nécessite une énorme quantité de données pour donner de meilleurs résultats. Malheureusement, la collecte et l'entraînement d'un grand ensemble de données prennent beaucoup de temps et nécessitent des ressources de calcul importantes. Une stratégie pour surmonter ce problème consiste à utiliser l'apprentissage par transfert (Transfer Learning) qui consiste à intégrer des connaissances préalables dans le développement du modèle afin qu'il n'apprenne pas uniquement à partir des données

du problème à résoudre [121]. Donc, grâce à cette technique, il devient possible d'obtenir de meilleurs résultats en utilisant seulement un petit ensemble de données.

L'apprentissage par transfert est une technique d'apprentissage automatique. Elle est particulièrement utile dans les tâches de vision par ordinateur telles que la classification d'images et la détection d'objets [121]. L'idée de base est d'appliquer et transférer les connaissances acquises lors de tâches précédentes pour résoudre une nouvelle tâche [124].

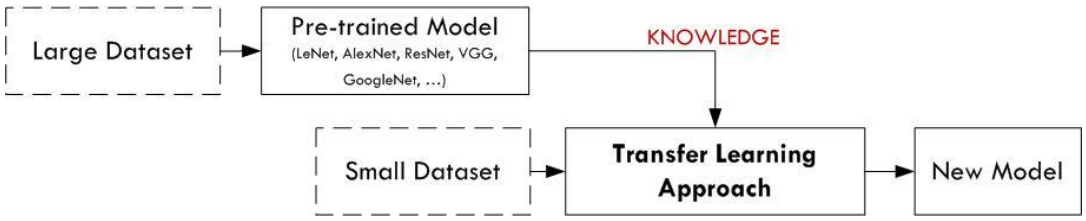


Figure 54. Processus d'apprentissage par transfert

En fait, il y a plusieurs réseaux pré-entraînés (LeNet, Alexnet, VGG, GoogleNet, ResNet, et bien d'autres encore) qui ont été formés sur de grands ensembles de données comme ImageNet, CIFAR-10, et COCO. Donc, comme l'illustre la Figure 54, on peut créer le nouveau modèle avec un petit ensemble de données en utilisant les connaissances de l'un de ces réseaux pré-entraînés.

Alors, comment les connaissances peuvent-elles être transférées de l'ancienne tâche à la nouvelle ? Tout d'abord, comme nous le savons, un modèle pré-entraîné a déjà été entraîné sur des millions d'images. Il s'est donc adapté aux images naturelles [121]. Il peut donc extraire efficacement des caractéristiques. Il a déjà appris à détecter les bords, les couleurs, les carrés, les cercles, les motifs, etc. Ainsi, ces caractéristiques apprises sont représentées par les valeurs des filtres de convolution qui sont les poids du réseau. Donc, pour transférer ces connaissances, il suffit de geler les poids de la partie d'extraction de caractéristiques du réseau (c'est-à-dire que les poids de ces couches d'extraction de features ne sont pas mis à jour pendant le training).

Presque toutes les topologies de réseau publiées sont pré-entraînés sur la base de données ImageNet, l'une des plus grandes bases de données de classification d'images et plus ou moins la norme pour les problèmes de classification d'images. Cet ensemble de données se compose de millions d'images couvrant plusieurs classes. L'utilisation de CNN pré-entraîné sur ImageNet est un moyen facile d'obtenir de très bons résultats pour les tâches de classification d'images [121].

Dans ce travail, nous avons utilisé le modèle AlexNet pour reconnaître les vues de véhicules. AlexNet est un réseau neuronal convolutif qui a été formé sur lmageNet par Alex Krizhevsky et son équipe [100]. Plus de détails dans la section des expérimentations.

3.4. Expérimentations & Résultats

Pour évaluer les performances des algorithmes que nous avons utilisés, nous avons construit plusieurs bases de données ; nous avons collecté les images sur lnternet. Ensuite, nous avons classé les images en deux catégories : lmages de vue avant et de vue arrière. Nous avons divisé chaque ensemble de données en un ensemble de Training et un ensemble de test.

Dans la première expérience, nous avons construit une base de données des images de vues (avant/arrière) de voitures à partir du site Web Caradisiac [125] et de la base de données GTI [126]. Cette base de données comprend 400 images de véhicules. Dans cette expérience, nous avons utilisé l'approche descripteur-classifieur (descripteurs : HOG et LBP, classifieurs : SVM et kNN).

Dans la deuxième expérience, nous avons construit une base de données de vues de véhicules (bus, moto, voiture, camion). Elle contient 4000 images avec deux classes : une pour les images de vue avant et l'autre pour les images de vue arrière. Nous avons utilisé l'approche descripteur-classifieur (descripteurs : HOG, LBP, et Gabor, classifieurs : SVM et k-NN) et les CNNs. Cette expérience fait partie du système de classification des catégories de véhicules indépendamment de vues (construit au Chapitre 3).

Dans la troisième expérience, nous avons utilisé la même base de données que la deuxième en utilisant l'apprentissage par transfert (TL) avec le modèle AlexNet.

Cette évaluation a été faite en calculant plusieurs métriques : exactitude globale (overall accuracy), classifications erronées (misclassification), classifications correctes (true classification) pour chaque vue et durée d'exécution par image (runtime per image). Dans toutes les expériences, nous avons fixé le nombre maximum d'époques à 10.

Les caractéristiques de la machine utilisée pour exécuter tous les algorithmes sont détaillées ci-dessous : Lenovo ThinkPad avec un processeur Intel R , CoreTM, i5 7ème

génération CPU @ 2.50GHz 2.71GHz, RAM 8Go. En outre, les algorithmes sont implémentés dans MATLAB R2017b.

3.4.1. Expérience 1 : Classification des vues de voitures/Approche descripteur-classifieur

Nous avons redimensionné toutes les images de la base de données à une taille de 150*150 pixels.

Les résultats de la comparaison sont présentés dans Tableau 1. Ces résultats montrent que le modèle kNN+HOG a obtenu les meilleures performances parmi les modèles comparés. Il a atteint une exactitude de 97,47%, ce qui est supérieur à kNN+LBP avec une différence d'exactitude de 12,66%. Nous avons également constaté que le SVM+HOG est supérieur au SVM+LBP et ce avec une différence de précision de 11,4%.

Les résultats montrent que les caractéristiques HOG caractérisent bien les orientations des vues de véhicules, ce qui les rend plus résistantes aux variations géométriques et d'illumination.

Tableau 1. Temps d'exécution et métriques d'évaluation des algorithmes sur les données de test

Modèle	Temps d'exécution/ Image (milliseconde)	Exactitude	FN	FP
Combinaison HOG+SVM	47,180	94.94 %	4.7 %	5.6 %
Combinaison HOG+KNN(k=4 &Distance ='Euclidean')	78,405	**97.47 %**	2.3 %	2.8 %
Combinaison LBP SVM	**17,737**	83.54 %	0 %	16.46 %
Combinaison LBP KNN(k=4 &Distance ='Euclidean')	38,802	84.81%	3.9%	11.29 %

Pour la métrique du temps d'exécution (Tableau 1), le modèle LBP+SVM est le plus rapide. En comparant les descripteurs, nous avons constaté que HOG est plus performant que LBP mais il est plus lent. Cela a été expliqué par le fait que les caractéristiques HOG décrivent bien la forme et l'apparence des objets par la distribution des directions des bords locaux. La comparaison des classificateurs a montré que le kNN est plus performant que SVM mais il est plus lent. La performance de kNN dépend du choix du paramètre k et de la fonction de distance. Au cours des

expériences, nous avons constaté que ce classifieur est plus performant lorsque le paramètre k est fixé à 4 et que la métrique de distance est euclidienne. Le SVM avec noyau linéaire était beaucoup plus rapide que le kNN. Cela a été expliqué par le fait que le noyau linéaire nécessite moins de calcul que le kNN. Dans cette expérience, nous avons effectué la classification des vues uniquement pour les voitures, et utilisé les algorithmes d'apprentissage automatique traditionnels. Cependant, dans les deux expériences 2 et 3, nous classifions les vues des quatre catégories de véhicules : voitures, bus, camions et motos, en exploitant aussi les méthodes d'apprentissage profond.

3.4.2. Expérience 2 : Classification des vues de véhicules /Approche descripteur-classifieur & CNNs

Dans cette expérience aussi, les images ont été redimensionnées à une taille fixe de 150*150 pixels. La Figure 55 présente un résumé de la comparaison des résultats obtenus en termes d'exactitude et d'erreur. Le graphique montre que CNN a obtenu de meilleurs résultats que les autres méthodes. Elle a atteint une exactitude de 94,29%, supérieure à celle de la méthode LBP+SVM par une différence de 30,65%. Nous pouvons également remarquer que HOG avec SVM et k-NN ont obtenu la même exactitude. Le Tableau 2 montre que CNN a produit 92,91% / vue arrière. Mais pour la classe/Vue Avant, le taux de TP de HOG+kNN était plus élevé que CNN (différence d'exactitude de 1,01%).

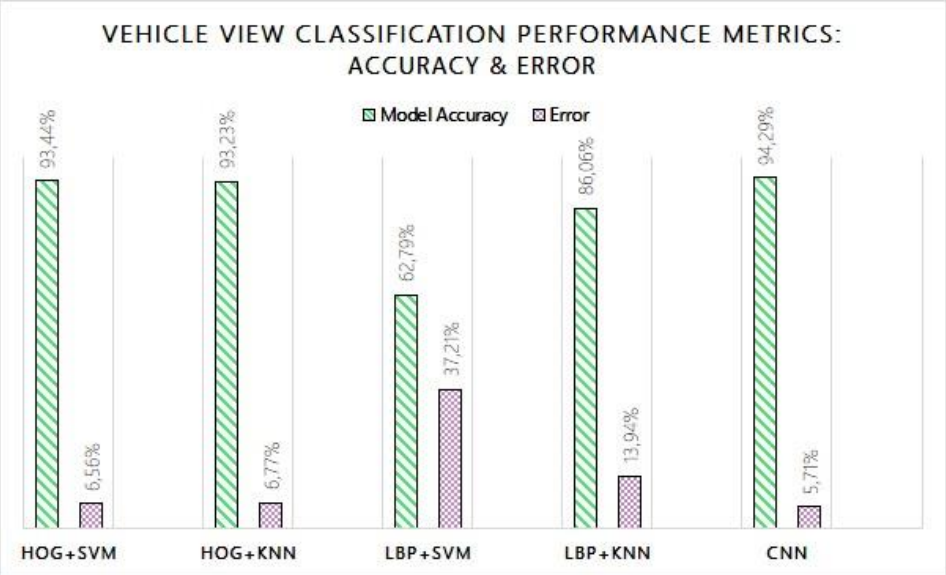


Figure 55. L'exactitude de la classification des vues des véhicules

Tableau 2. Les résultats de la classification des vues avant et arrière

	Classe : Vue arrière		Classe : Vue Avant	
Modèles	TP	FN	TP	FN
HOG+SVM	90.94%	9.06%	95.64%	4.06%
HOG+KNN	89.76%	10.24%	96.70%	3.30%
LBP+SVM	32.68%	67.32%	92.89%	7.11%
LBP+KNN	85.83%	14.17%	86.29%	13.71%
CNN	92.91%	7.09%	95.69%	4.31%

Lorsque nous avons entraîné les réseaux neuronaux convolutionnels, nous avons suivi les progrès de leur entraînement en traçant plusieurs mesures. Ainsi, nous pouvons déterminer si et comment l'exactitude et l'erreur du réseau s'améliorent. Les Figure 56 et Figure 57 montrent ces mesures à chaque itération. Chaque itération estime le gradient et met à jour les paramètres du réseau.

La Figure 56 présente la progression de l'entraînement d'un modèle CNN. Elle illustre l'évolution de l'exactitude pendant l'entraînement (training accuracy) et sa version lissée (smoothed version) en fonction du nombre d'époques. Chaque époque est marquée par un fond ombré. Une époque est un passage complet à travers l'ensemble des données.

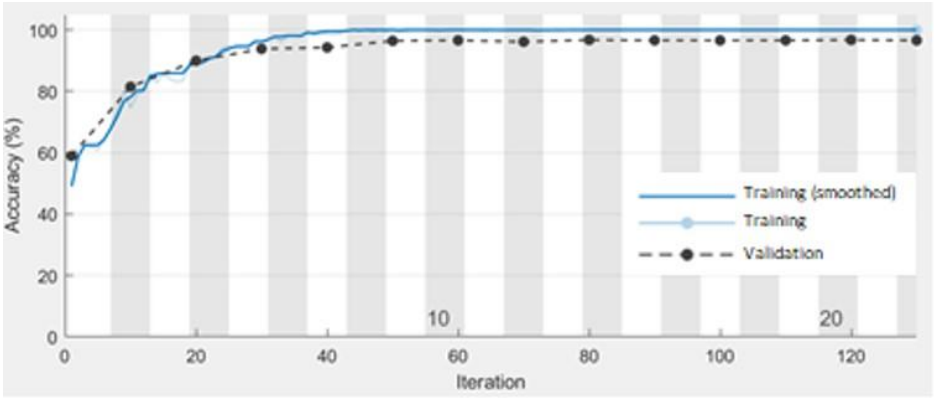


Figure 56. L'évolution de l'exactitude de l'entraînement

On peut voir que l'exactitude augmente progressivement en fonction du nombre d'époques (24 au total) pour atteindre son meilleur niveau. Le modèle est devenu plus généralisé, surtout à partir de l'époque 7. Le graphique montre également l'exactitude de la classification sur tout l'ensemble de la validation.

Dans la Figure 57 suivante, nous montrons l'évolution de la courbe de la perte d'entraînement et sa version lissée en fonction du nombre d'époques. Comme le montre la figure, la fonction de perte diminue au fur et à mesure que le nombre d'époques augmente.

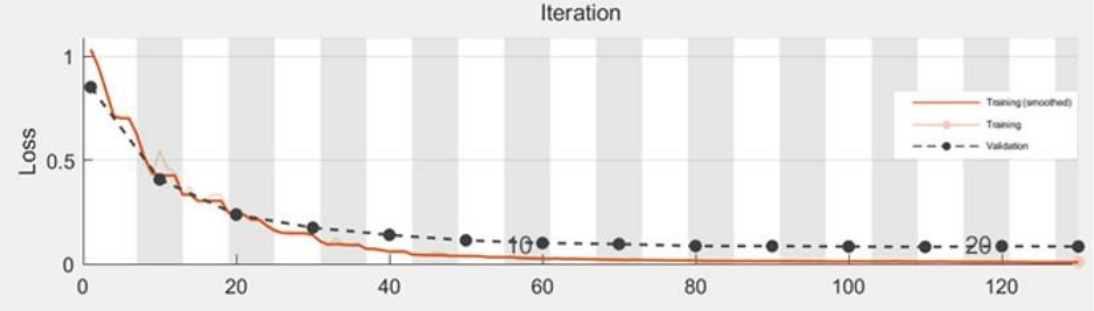


Figure 57. Fonction d'erreur

En exécutant nos algorithmes, nous avons remarqué que les CNNs prennent beaucoup de temps à être entraînés par rapport aux autres modèles. Si le nombre de couches augmente, le temps d'entraînement et de test augmente. D'autre part, les performances du réseau s'améliorent [127]. Dans les CNN, la couche la plus importante est la couche de convolution. C'est donc cette dernière qui prend la majeure partie du temps d'entraînement.

Dans la première expérience, nous avons travaillé sur la classification des vues avant et arrière mais seulement pour les voitures [128] (97,47% obtenus par HOG+kNN). Cependant, dans cette expérience, nous avons classé les vues des quatre catégories de véhicules. Cette classification est plus difficile, surtout pour les motos. Les deux vues (avant et arrière) ont presque la même forme.

3.4.3. Expérience 3 : classification des vues de véhicules / Approche TL

Dans cette expérience, il y a trois scénarios : l'apprentissage par transfert, training from scratch de CNN et TL-AlexNet+SVM. Dans le premier scénario, nous avons construit une architecture CNN à partir de zéro, inspirée d'AlexNet, afin de comparer ce scénario avec le suivant. Ensuite, nous avons appliqué l'apprentissage par transfert à l'aide d'AlexNet. Dans le troisième scénario, nous avons ajouté le classifieur SVM pour remplacer le FC et le fusionner avec.

a- Training from scratch

Pour commencer, les images présentées à la couche d'entrée doivent être carrées - ici 227x227 (sans le nombre de canaux). En général, les couches de convolution doivent

utiliser des filtres de plus petite taille, comme 3 × 3 et 5 × 5. De minuscules filtres 1×1 sont utilisés pour apprendre des caractéristiques locales, mais uniquement dans les architectures de réseau plus avancées [50].

Des filtres de plus grande taille, tels que 7×7 et 11×11, peuvent être utilisés comme première couche de convolution du réseau (pour réduire la taille spatiale d'entrée, à condition que les images soient suffisamment grandes > 200×200 pixels) ; toutefois, la taille du filtre doit diminuer considérablement, sinon les dimensions spatiales des entrées seront réduites trop rapidement [50].

On utilise généralement un stride de S = 1 pour les couches de convolution, du moins pour les petits volumes d'entrée spatiaux (les réseaux qui acceptent des volumes d'entrée plus importants utilisent un stride S >= 2 dans la première couche de convolution).

Le plus souvent, on voit le max pooling appliqué sur un champ réceptif de 2×2 et un stride de S = 2. On peut également voir un champ réceptif 3×3 au début de l'architecture du réseau pour aider à réduire la taille de l'image. Il est très rare de voir des champs réceptifs supérieurs à trois, car ces opérations sont très destructrices pour leurs entrées.

Pour comparer l'approche d'apprentissage par transfert avec le training from scratch, nous avons essayé de conserver les mêmes conditions d'évaluation : mêmes images de l'ensemble de données, mêmes dimensions de l'image (227x227x3), même structure du modèle que dans AlexNet. Ainsi, avant l'entrainement du modèle, nous avons configuré divers hyperparamètres comme le nombre de couches, la fonction d'activation, la taille du Batch, le nombre d'époques, les paramètres de la couche de convolution (taille et pas du filtre), les paramètres de la couche de pooling, etc. Tout d'abord, nous montrons l'influence de la qualité des images d'entrée sur le comportement du modèle. Nous avons entraîné le modèle pour deux dimensions (150x150) et (227x227) pixels. Donc, on peut facilement conclure du Tableau 3 que plus la qualité de l'image est bonne, plus le modèle est performant. Une bonne qualité d'image signifie qu'il y a plus d'informations à fournir au modèle.

Tableau 3. Résultats obtenus en modifiant la taille de l'image

		Vue : Arrière	Vue : Avant

Dimensions des images d'entrée	Exactitude globale	Classifications correctes	Classifications erronées	Classifications correctes	Classifications erronées
150x150 pixels	94.29%	92.91%	7.09%	95.69%	4.31%
227x227 pixels	95.85%	94.49%	5.51%	97.21%	2.79%

Le Tableau 4 présente les valeurs de précision et d'erreur pour CNN-training from scratch.

Tableau 4. Résultats de training from scratch

		Vue : Arrière		Vue : Avant		
Modèle	Exactitude globale	Classifications correctes	Classifications erronées	Classifications correctes	Classifications erronées	Temps d'exécution/Image
CNN-training from scratch	96.79%	94.09%	5.91%	99.49%	0.51%	1.117101

Comme le montre le Tableau 4, le modèle a atteint une exactitude de 96,79 % et une durée d'exécution de 1,117101 seconde par image.

b- Apprentissage par transfert

Comme indiqué, dans ce travail, nous avons réutilisé le modèle AlexNet. Il a été entraîné sur la fameuse base de données ImageNet avec 1000 catégories d'objets. Comme le montre la Figure 58, AlexNet contient 25 couches : cinq couches de convolution, trois couches de pooling, sept fonctions (ReLU), deux couches de normalisation, deux couches de dropout, trois couches FC, une couche Softmax [129].

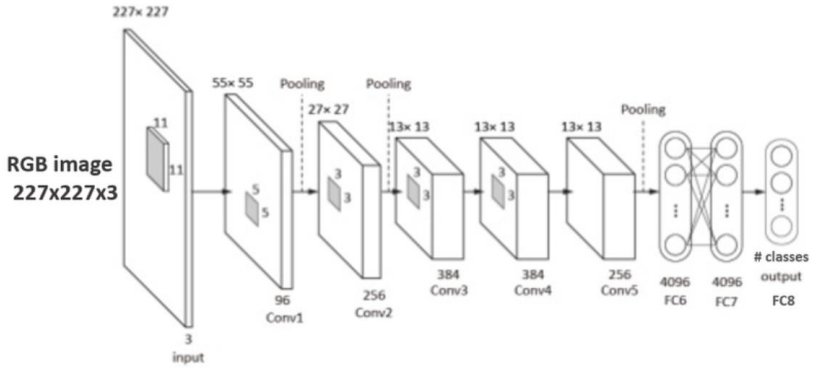


Figure 58. Modèle AlexNet [130]

La Figure 59 présente la manière dont nous avons ajusté le réseau AlexNet (fine-tuning) pour effectuer la classification de vue Arrière /Avant. En entrée, nous avons notre base de données. Nous avons donc gelé la première partie du réseau (les poids sont fixés) afin de réutiliser les connaissances relatives à l'extraction des caractéristiques. Car, les couches initiales détectent les lignes obliques, quel que soit l'objet de la classification. Il est donc inutile de les entrainer à chaque fois qu'on crée un CNN. Seules les couches finales de notre réseau, celles qui apprennent à identifier les classes spécifiques à chaque classification, nécessitent un entraînement. Nous avons donc réinitialisé ces dernières couches responsables de la classification (les couches jaunes). Nous avons défini comme sortie deux catégories avec des étiquettes (avant et arrière) au lieu de 1000 étiquettes. Et finalement, nous avons entraîné notre nouveau modèle.

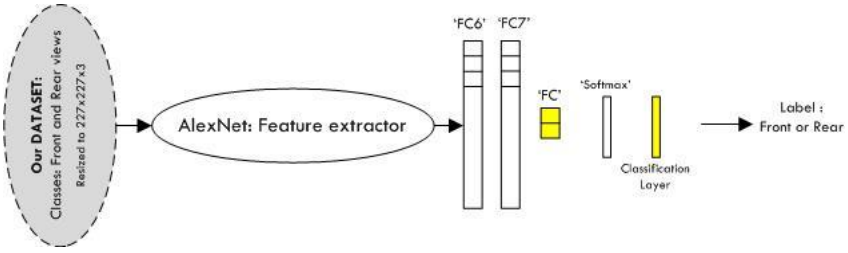


Figure 59. Fine-tuning le modèle AlexNet

Comme le montrent le Tableau 5 et le Tableau 6, les résultats sont meilleurs lorsqu'on utilise la fonction d'activation ReLU et miniBatch=32. Le modèle a atteint une exactitude de 99,75%.

Tableau 5. Résultats obtenus lors de l'utilisation des fonctions d'activation (ReLU et leakyReLU)

		Vue : Arrière	Vue : Avant	

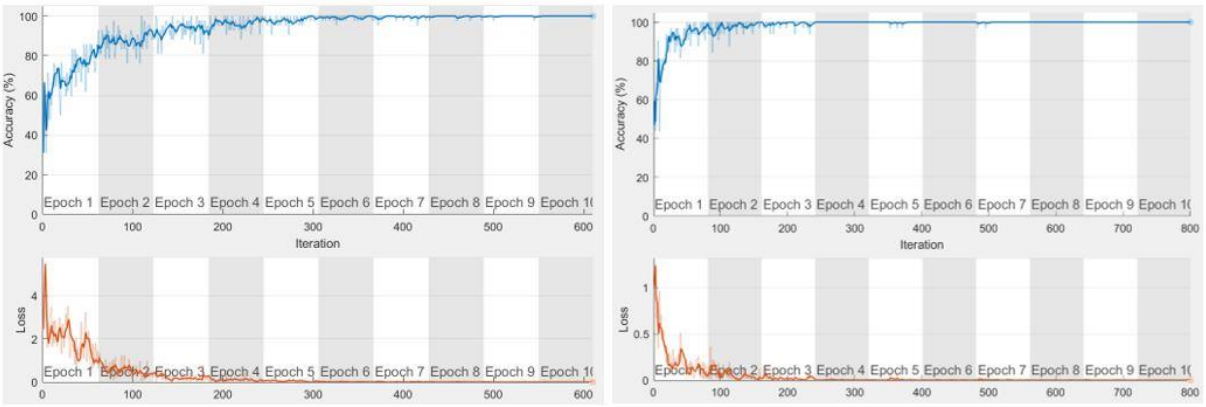
TL.AlexNet model	Exactitude globale	Classifications correctes	Classifications erronées	Classifications correctes	Classifications erronées	Temps d'exécution/ Image
ReLU	99.42%	99.61%	0.39%	99.24%	0.75%	1.239981
leakyReLU	99.09%	98.43%	1.57%	99.75%	0.25%	1.241131

Tableau 6. Résultats obtenus lors de la modification de la valeur de la taille du batch

		Vue : Arrière		Vue : Avant		
TL.AlexNet model	Exactitude globale	Classifications correctes	Classifications erronées	Classifications correctes	Classifications erronées	Temps d'exécution/ Image
MiniBatch=32, ReLU	99.75%	100%	0%	99.49%	0.51%	1.101806
MiniBatch=64, ReLU	99.42%	99.61%	0.39%	99.24%	0.75%	1.239981

Discussion

L'exactitude (lors de l'entraînement) et la fonction de perte sont indiquées ci-dessous. La Figure 60 donne la courbe de l'exactitude et de la perte de l'entraînement en fonction du nombre d'époques (a- Training from scratch, b- TL). Dans la Figure 60, la perte globale diminue avec l'augmentation du nombre d'époques. Ainsi, trois points permettent de différencier l'apprentissage par transfert de Training from scratch (Figure 61 [131]) : L'apprentissage par transfert a (1) un départ plus élevé (higher start), (2) une pente plus élevée (higher slope), et (3) une asymptote plus élevée (higher asymptote) [131].



a- training from scratch b- Transfer Learning

Figure 60. L'évolution de l'exactitude et de la fonction de perte lors de l'entraînement

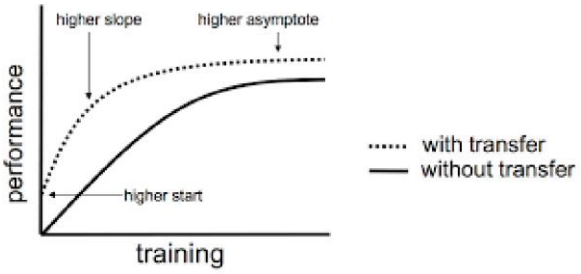


Figure 61. Le processus de l'entraînement pour l'apprentissage par transfert et training from scratch

Le principal avantage du TL est qu'on a besoin moins de données pour entraîner le modèle. Cependant, le training d'un modèle d'apprentissage profond à partir de zéro nécessite une grande quantité de données pour donner une grande exactitude. Malheureusement, on ne trouve pas de bases de données qui conviennent à notre travail. On ne trouve que quelques ensembles de données/vue de dessus ou seulement des bases de données pour la catégorie de voitures. Pour cette raison, nous avons collecté les images/vues avant et arrière sur Internet. Donc ici, l'application du TL est un bon choix pour surmonter cette limitation.

c- Apprentissage par transfert - Modèle AlexNet + SVM

Comme nous l'avons dit, CNN est composé d'un extracteur de caractéristiques et d'un réseau neuronal de classification. La sortie du réseau de l'extracteur de caractéristiques est transformée en un vecteur unidimensionnel qui entre dans le réseau du classifieur. Le classifieur est un réseau de neurones entièrement connecté [132]. Comme les CNNs pré-entraînés ont été entraînés avec de larges ensembles de données d'images naturelles, ils représentent un bon descripteur de caractéristiques. Nous avons donc

décidé d'utiliser le TL-AlexNet comme extracteur de caractéristiques et de remplacer les couches entièrement connectées par un classifieur SVM (Figure 62/Cas1). D'autre part, nous avons combiné le SVM avec les couches entièrement connectées de TL-AlexNet (Figure 62/Case2). L'entrée de toute couche entièrement connectée pourrait être utilisée comme entrée d'un classifieur SVM ; les caractéristiques passent par une ou deux couches entièrement connectées et entrent ensuite dans le classifieur SVM qui produit les labels.

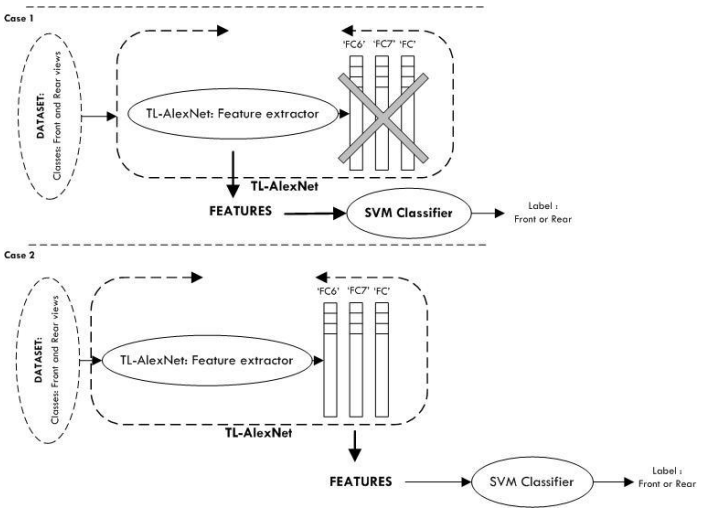


Figure 62. Combinaison des caractéristiques d'AlexNet avec le classifieur SVM

Tableau 7. Résultats obtenus en combinant les caractéristiques d'AlexNet avec le classifieur SVM

		Vue : Arrière		Vue : Avant		
TL.AlexNet model	Exactitude globale	Classifications correctes	Classifications erronées	Classifications correctes	Classifications erronées	Temps d'exécution/ Image
TL.AlexNet +SVM	99.48%	99.21%	0.79%	99.75%	0.25%	0.2822615
TL.AlexNet +(FC6+SVM)	99.87%	100%	0%	99.75%	0.25%	0.829891

TL.AlexNet +(FC7+SVM)	100%	100%	0%	100%	0%	0.791048

Comme le montre le Tableau 7, TL-AlexNet features+SVM ont atteint une exactitude de 99,48 %. Cela n'améliore pas les résultats. Toutefois, lorsque les deux classifieurs sont combinés, le premier cas améliore l'exactitude à 99,87 %. Le second surpasse tous les autres modèles, il atteint une exactitude de 100 % et permet de gagner du temps.

Nous pouvons conclure que l'apprentissage profond utilisant les réseaux neuronaux convolutifs est une très bonne approche pour la classification des vues de véhicules, et l'utilisation de l'apprentissage par transfert est un choix pertinent qui est amélioré par l'ajout du classifieur SVM.

3.5. Conclusion

Ce chapitre a présenté plusieurs types courants de réseaux neuronaux, notamment les CNNs, les RNNs et les GANs. Dans ce chapitre, nous avons utilisé les CNNs pour faire la classification de vues avant et arrière des véhicules qui est une partie principale de notre système globale. Ce type de réseaux est spécialisée dans la reconnaissance d'images. En prenant une image, nous fournissons les valeurs d'intensité des pixels comme entrées au CNN. Une série de couches cachées est utilisée pour extraire les caractéristiques de notre image d'entrée. Ces couches cachées sont construites les unes sur les autres de manière hiérarchique. Au début, seules les régions de type bord sont détectées dans les couches de niveau inférieur du réseau. Ces régions de bord sont utilisées pour définir les coins et les contours. La combinaison des coins et des contours peut conduire à des parties d'objet abstraites dans la couche suivante. Le concept de construction de caractéristiques de niveau supérieur à partir de caractéristiques de niveau inférieur est exactement la raison pour laquelle les réseaux CNN sont si puissants en vision par ordinateur [50]. Comme indiqué, nous avons utilisé l'approche de l'apprentissage par transfert en utilisant un CNN pré-entraîné appelé AlexNet. Pour montrer l'efficacité de cette approche, nous avons construit un autre modèle à partir de zéro (training from scratch), inspiré de l'architecture d'AlexNet, afin de comparer les deux modèles. Les meilleurs résultats ont été obtenus grâce à la technique de TL. Quelques raisons du succès du TL : le modèle préformé est déjà entraîné sur un très grand ensemble de données. Le modèle est donc adapté aux images naturelles ; il peut facilement détecter les bords, les couleurs, les carrés, les cercles, etc. Ces caractéristiques sont représentées par les poids du réseau. Ainsi, dans le TL, ces poids

sont gelés pour transférer et réutiliser ces connaissances acquises. Cependant, les couches finales sont réinitialisées pour classer les nouvelles catégories. Les principaux avantages de la TL sont donc les suivants : tout d'abord, elle permet de gagner du temps, ne nécessite qu'un petit ensemble de données et donne de bons résultats. Finalement, nous avons décidé d'utiliser le TL-AlexNet comme extracteur de caractéristiques, et de remplacer/combiner les couches entièrement connectées par/avec un classifieur SVM. En fait, les résultats que nous avons obtenus montrent que le système de classification des véhicules avant et arrière peut être intégré avec succès dans le système global de la détection des dépassements interdits des véhicules.

"Deep Learning is an algorithm which has no theoretical limitations of what it can learn; the more data you give and the more computational time you provide; the better it is"

Geoffrey Hinton -Google

4. **Chapitre 3 :** Classification de catégories de véhicules

Sommaire

4.1. Introduction

Avec le nombre croissant de véhicules et de caméras de surveillance, il est important d'automatiser la reconnaissance des véhicules pour collecter les données relatives au trafic [133]. Plusieurs systèmes informatiques ont été développés pour détecter et classer les véhicules selon différents critères (catégorie, couleur, marque, taille, vues) où le type (catégorie) d'un véhicule est la caractéristique la plus importante et essentielle [134].

La classification par catégorie joue un rôle essentiel dans les systèmes de circulation intelligents et peut être utilisée de manière extensive à des fins diverses telles que la surveillance des autoroutes, la perception de péages, les statistiques sur les flux de circulation [133], le stationnement automatique, la prévention de l'accès des poids lourds aux ponts des villes [135]. Il peut également être utilisé dans les systèmes de régulation du trafic pour spécifier les limites de vitesse pour chaque catégorie (exemple : les camions ont des limites de vitesse différentes de celles des autres véhicules). La classification par catégorie de véhicules peut également être utilisée dans les voitures à conduite autonome pour alerter, par exemple, le conducteur de l'approche d'un camion.

Cette classification fait partie de notre système qui vise à détecter automatiquement les dépassements interdits de véhicules. Car, ce système doit tenir compte de situations particulières comme le fait de permettre à des motocyclettes de dépasser la circulation sans envoyer d'avertissement pour violation des lois. De point de vue pratique, la classification des catégories de véhicules est influencée par les changements d'angles de vue ; ce qui la rend plus difficile. Dans ce chapitre, nous présentons un système qui vise à classifier les catégories de véhicules (bus, voiture, moto et camion) indépendamment de l'angle de vue (angles avant/arrière). Nous proposons, en effet, un système en deux phases : une phase de reconnaissance de l'angle de vue et une phase de classification des catégories de véhicules.

Notre motivation à réaliser ce travail est que, premièrement, de nombreuses études antérieures ont travaillé sur la classification des véhicules vue de profil. Cependant, il y a eu relativement peu de recherches sur la classification des véhicules selon leur vue avant et arrière [8]. En fait, la classification des véhicules, à partir de leur vue avant

ou/et arrière, est plus difficile. Car, la plupart des catégories de véhicules ont les mêmes caractéristiques dans leurs structures de vue telles que le pare-brise, l'éclairage avant, les rétroviseurs, le pare-chocs, etc (Exemple : Figure 63/ Vue de face). Deuxièmement, la plupart des chercheurs n'ont utilisé qu'une, deux ou trois méthodes dans leurs études comparatives. Troisièmement, personne n'a encore développé un système de classification des catégories de véhicules indépendant des vues.

Figure 63. Aperçu de la structure de la vue frontale des camions, des bus et des voitures

Aujourd'hui, avec l'augmentation du nombre de véhicules sur les routes, les ambulances doivent attendre dans le trafic, ce qui retarde leur arrivée à destination [136]. L'excellence du service d'urgence dépend de la rapidité avec laquelle les véhicules d'urgence peuvent atteindre le lieu de l'incident. Cette situation se produit souvent en raison de l'augmentation du nombre de véhicules [137] et parfois du manque de coopération des civils [138]. Si le véhicule d'urgence est coincé dans un embouteillage et que son arrivée sur le lieu de l'incident est retardée, cela peut entraîner la perte de vies humaines et de biens [139]. Réduire ce temps de réponse d'une seule minute augmente le taux de survie des patients victimes d'un arrêt cardiaque de 24 % [140].

Il est nécessaire de mettre en place des systèmes qui donnent la priorité aux ambulances. Donc la détection automatique des ambulances joue un rôle important pour ces systèmes. Elle peut être utilisée dans les systèmes de gestion et contrôle des réseaux de feux de signalisation ; ceux-ci visent à trouver les chemins les plus courts possibles pour les ambulances jusqu'à leur destination. Elle peut également être utilisé dans les véhicules autonomes pour alerter le conducteur de l'approche d'un véhicule d'urgence.

Le système de détection d'ambulance peut également être utilisée dans la surveillance des systèmes de règles et de régulation du trafic pour permettre, par exemple, l'accès aux limites de vitesse ou pour autoriser les dépassements pour ces véhicules d'urgence sans envoyer une alerte pour violation de la loi. En effet, cette partie de la détection des ambulances est considérée comme une situation particulière de notre système global (de détection des dépassements interdits des véhicules) à prendre en compte.

Donc la deuxième partie de ce chapitre, nous proposons une solution de vision par ordinateur pour détecter de manière robuste la catégorie des ambulances.

Dans les solutions proposées dans ce chapitre, nous avons utilisé des techniques d'apprentissage automatique et d'apprentissage profond pour classer les vues et les catégories de véhicules. Nous avons utilisé trois descripteurs (LBP, HOG et filtre de Gabor), deux classifieurs SVM et k-NN ainsi que des réseaux neuronaux convolutifs. Trois expériences sont donc réalisées. La première est consacrée à la classification des catégories de véhicules. La deuxième présente le système global en utilisant les meilleurs modèles que nous avons obtenus. La troisième propose une solution pour la détection des ambulances.

Le reste de ce document est structuré comme suit. La section 2 décrit les travaux précédents relatifs au problème de la classification des véhicules. Dans la section 3, nous présentons un aperçu du système que nous avons développé. La section 4 décrit en détail toutes les techniques utilisées pour la classification des vues et des catégories. Les résultats expérimentaux et les comparaisons sont présentés dans la section 5. La dernière section comprend les conclusions.

La section suivante présente les recherches précédentes sur la classification des catégories des véhicules.

4.2. État de l'art

De nombreux chercheurs ont actuellement effectué des travaux approfondis sur la classification des types de véhicules. Cette section présente quelques travaux antérieurs sur cette classification. Nous avons classifié ces travaux selon 3 approches : Machine Learning, non Machine Learning, et Deep Learning.

Nous commençons par les travaux utilisant la première approche. [[141], [142], [134], [135], [143], [144]] ont utilisé les algorithmes d'apprentissage automatique connues comme SVM, k-NN, Random Forest, Logistic Regression. [141] a utilisé des mesures de

dimensions et de forme comme caractéristiques avec les classifieurs SVM et les forêts d'arbres décisionnels (Random Forest, RF) pour classer les véhicules selon les catégories : voiture, vélo/moto, bus et camionnette. Dans la pratique, une segmentation manuelle est utilisée pour déterminer les véhicules sur les images et un ensemble de caractéristiques est extrait de chaque silhouette binaire. Treize mesures différentes composent le vecteur de caractéristiques. Les composantes comprennent des mesures de dimension et de forme à partir de la silhouette binaire et englobent la boîte de délimitation (largeur, hauteur et surface), la circularité (dispersion, diamètre équivalent), l'ellipticité (longueur du grand axe et du petit axe, excentricité) et la mesure du remplissage de la forme (surface remplie, surface convexe, étendue, solidité). La classification attribue chaque silhouette à l'une des quatre classes de véhicules (voiture, camionnette, bus et vélo/moto) à l'aide de deux classifieurs SVM et Random Forest. Les auteurs ont construit leur base de données de caractéristiques d'objets en segmentant manuellement des véhicules à partir de la vidéo d'une route urbaine très fréquentée. Chaque véhicule est détecté individuellement dans une zone de détection. Une frontière polygonale convexe fermée a été dessinée manuellement autour du contour du véhicule, et étiquetée avec l'une des quatre catégories de véhicules suivantes : voiture, camionnette, bus et moto/bicyclette. La base de données comprend 2055 contours de véhicules (voiture : 1033, van : 589, bus : 290, moto : 143). Les auteurs ont expliqué que, pour des raisons d'équilibre, le nombre de voitures a été limité car elles constituaient la grande majorité des véhicules vus pendant la période d'observation. Ils ont utilisé une stratégie de validation croisée en sélectionnant aléatoirement la moitié de l'ensemble de données pour la formation et l'autre moitié pour le test, en répétant le processus 10 fois et en calculant la moyenne des résultats afin d'évaluer la performance des méthodes de classification. Les résultats montrent que le SVM surpasse le RF. Les valeurs résultantes du taux de vrais positifs (TPR pour True Positive Rate) pour la voiture, la camionnette, le bus et la moto/bicyclette sont respectivement de 0,9661, 0,8913, 0,9931 et 1,0. Le TPR moyen est de 0,9626. Le coût de calcul du classifieur SVM est aussi significativement plus bas que celui du RF. Le plus grand nombre d'erreurs de classification se produit entre les catégories de voitures et de fourgonnettes, où les caractéristiques de taille et de forme présentent une similarité significative.

[142] a proposé une classification en temps réel des types de véhicules à partir des vidéos de surveillance sur les routes urbaines. Les véhicules sont classés Trois types de véhicules à savoir les petites voitures, les grandes voitures et les motos. Le descripteur

HOG est utilisé pour représenter les apparences et les formes locales de l'objet. Un classifieur SVM multi-classes est entrainé en utilisant les caractéristiques HOG pour la classification du type de véhicule. Deux métriques, le taux de précision et le taux de rappel, ont été calculées pour l'évaluation. En moyenne, le taux de précision est de 93,82%, et le taux de rappel de 88%. [134] a proposé une reconnaissance des types de véhicules basée aussi sur un modèle HOG-SVM amélioré. Les auteurs ont appliqué des méthodes de prétraitement d'image ciblées telles que l'étirement des niveaux de gris et le filtrage gaussien sur l'image originale pour réduire les facteurs d'interférence du fond. Ils ont utilisé ensuite HOG pour extraire les caractéristiques des images. Le processus de réduction de la dimension PCA est utilisé pour réduire la taille et la complexité des vecteurs de caractéristiques afin d'accélérer la reconnaissance de la caractéristique HOG. Pour la classification, les auteurs ont utilisé le SVM. Les images vidéo utilisées dans les expériences sont issues de la surveillance des routes urbaines. Pour les paramètres de la caméra, la fréquence d'images vidéo capturées est de 40 images par seconde, l'image vidéo est de 960*540 pixels. Ces images sont utilisées comme base de données de test pour évaluer les performances de l'identification des véhicules. L'ensemble de données d'entraînement utilise l'ensemble de données publique BIT-Vehicle qui contient 9 850 images de véhicules. Ces images présentent diverses conditions d'éclairage et des angles de vue. Tous les véhicules de la base de données sont divisés en six catégories : bus, microbus, minivan, berline, SUV et camion. Les résultats expérimentaux montrent que le taux de reconnaissance moyen atteint 92,6 %, ce qui est supérieur à 90,3 % de l'algorithme CNN (Convolutional Neural Network) et à 87,4 % de l'algorithme SURF. Le temps de reconnaissance moyen est également 20 à 38% plus élevé qu'avec ces deux méthodes. Dans [135], les auteurs ont utilisé un classifieur d'ensemble en cascade utilisant le Multiple Layer Perceptron (MLP) avec k-NN pour la classification de types de véhicules alimentés par des caractéristiques de type HOG, LBP et la combinaison HOG-LBP. Plus de 1800 images de véhicules différents ont été sélectionnées pour cinq types de véhicules, dont les voitures, les camionnettes, les bus, les camions moyens et les camions lourds. Toutes ces images contenaient des vues de face d'un ou plusieurs véhicules. La taille originale de chaque image est de 1920x1152 pixels. Pour chaque type de véhicule, les auteurs ont sélectionné aléatoirement 150 échantillons pour l'entraînement et 100 échantillons pour le test de chaque classifieur. Toutes les images d'entrée sont redimensionnées à la taille de 64x64. Pour les caractéristiques HOG et LBP, un vecteur de caractéristiques avec une dimension de 1764 et 944 est généré, respectivement. Le classifieur MLP a trois

couches ; une couche d'entrée de n_i nœuds égale à la dimension des vecteurs de caractéristiques. La couche de sortie à n_o nœuds qui sont égaux au nombre de types de véhicules. Les résultats montrent que pour le classifieur MLP, le résultat de reconnaissance du HOG est meilleur que celui du LBP, alors qu'un résultat contraire est obtenu par le classifieur k-NN. De plus, les auteurs ont obtenu de meilleurs résultats en concaténant HOG et LBP. Le taux de classification du système global atteint 94,8%. [143] a utilisé deux extracteurs de caractéristiques, l'histogramme pyramidal des gradients orientés (PHOG pour Pyramid Histogram of Oriented Gradients) et la transformée en curvelet, et les classifieurs MLP, SVM, k-NN et Random Forest pour classer le type et la marque des véhicules. Les auteurs ont combiné les caractéristiques PHOG avec Curvelet Transform pour améliorer l'exacitude de la classification en utilisant leurs informations complémentaires. Ils ont appliqué, ensuite, le classifieur d'ensemble Rotation Forest (RF) utilisant un vote majoritaire pour la décision. Les expériences sont réalisées sur plus de 600 images de 21 marques de voitures ou de camionnettes. Les résultats obtenus montrent que les caractéristiques combinées sont meilleures que des caractéristiques individuelles en termes d'exactitude de classification et le modèle d'ensemble produit de meilleures performances par rapport à n'importe quel classifieur de base de réseau neuronal individuel. Avec une taille d'ensemble modérée de 20, les ensembles de Rotation Forest produisent un taux de classification proche de 96,5 %.

Dans la deuxième approche, les chercheurs utilisent d'autres méthodes que les algorithmes de la première approche. [145] utilise une méthode d'estimation visuelle des dimensions pour la classification des types de véhicules : les taxis, les véhicules à deux étages et les mini-bus. La classification est faite en estimant la largeur, la longueur et la hauteur du véhicule à l'aide d'un ensemble de fonctions de mapping de coordination dérivées d'un modèle de caméra calibré. Le concept de la méthode proposée consiste à ajuster une projection 2D d'un modèle 3D simple sur une représentation binaire du véhicule extraite de la séquence d'images du trafic, dont la dimension est déterminée. Cette représentation binaire est appelée masque du véhicule. Le masque du véhicule est extrait en soustrayant de la séquence d'images du trafic un fond stationnaire estimé. Ensuite, une méthode d'élimination des ombres est utilisée pour supprimer toute ombre portée extraite avec le masque du véhicule. Puis, la modélisation du véhicule est effectuée en 2D en utilisant un modèle de caméra calibré pour fournir une correspondance entre les coordonnées la scène en 3D et les coordonnées de l'image en 2D. Enfin, les dimensions du véhicule, notamment la largeur,

la longueur et la hauteur, sont calculées en coordonnées 3D. A partir de ces dimensions, les véhicules sont classés dans leurs types respectifs. Pour l'évaluation, trois séries d'images de circulation d'un taxi, d'un mini-bus et d'un bus à deux étages ont été obtenues à partir d'une séquence d'images de circulation. Les images ont été prises de jour, lorsque les ombres sont proéminentes. Les auteurs ont montré que l'algorithme de suppression des ombres supprime une partie de l'arrière des taxis. Ils ont expliqué cela par le fait que cette zone est sombre et que la densité des pixels du bord est faible. Cependant, cela n'affecte pas beaucoup le processus de modélisation. Pour les mini-bus et les véhicules à deux étages, l'ombre est retirée avec succès sans affecter le masque du véhicule. Les résultats montrent que, pour les taxis, l'erreur d'estimation de la longueur est d'environ 7,4 %, tandis que les erreurs d'estimation de la largeur et de la hauteur sont légèrement supérieures à 10 %. Pour les minibus et les véhicules à deux étages, les erreurs sont moins importantes dans les estimations de la longueur (~ 4%) et de la largeur (<3%). Cependant, des erreurs plus importantes pour l'estimation de la hauteur de >10% pour le mini-bus et de >7% pour le bus à deux étages sont obtenues. L'inconvénient de cette méthode est que sa performance est affectée directement par la précision du masque du véhicule.

[127] a utilisé une autre méthode de classification qui se base sur des images provenant de caméras thermiques et à lumière visible. Cette méthode classifie les véhicules en six types tels que le type SUV- Sport Utility Vehicles, berline, camion. Elle peut être utilisée dans des systèmes de surveillance intelligents en temps réel. Dans la première étape, les auteurs ont commencé par soustraire l'image de premier plan de l'image d'arrière-plan de référence correspondante (et appliquer un filtrage de l'arrière-plan gaussien) et décomposer l'image en régions qui correspondent aux objets. Dans la deuxième étape, ils ont extrait les zones des phares et de la calandre à partir des images thermiques et à lumière visible. Dans l'étape d'extraction des caractéristiques, ils ont extrait les caractéristiques de type contraste, texture, couleur, forme, homogénéité, entropie et énergie. La quatrième étape a consisté à comparer les caractéristiques de différents objets pour les classer. Les auteurs ont évalué leurs classifieurs en utilisant 6671 images visuelles et 4005 images thermiques comme ensemble d'entraînement et 767 images visuelles de véhicules et 447 images thermiques de véhicules comme ensemble de test. Ils ont construit des classifieurs qui sont basés sur des seuils de texture et de rapport largeur/hauteur de la calandre. Ils ont obtenu une exactitude de 92,7 % pour le classifieur d'images visibles et de 65,8 % pour le classifieur d'images thermiques.

Au cours des dernières années, l'apprentissage profond et plus particulièrement les CNNs ont été exploités pour traiter la tâche de classification des catégories de véhicules mais avec différentes approches et bases de données [146]. Les CNNs ont prouvé leurs performances dans la classification des images. [147], ont proposé une méthode basée sur une CNN semi-supervisée avec des filtres laplaciens pour les noyaux. Les auteurs ont construit un ensemble de données des véhicules appelé BIT Vehicle Dataset qui comprend 9 850 images de véhicules pour tester la méthode proposée. La proportion d'images de lumière nocturne dans l'ensemble de la base de données est d'environ 10%. Les images contiennent des changements dans les conditions d'éclairage, l'échelle, la couleur des véhicules, et de l'angle de vue. Tous les véhicules de l'ensemble de données sont divisés en six catégories : Bus, Microbus, Minivan, Sedan, SUV, et Truck. Pour chaque type de véhicule, 200 échantillons sont sélectionnés aléatoirement pour l'apprentissage des paramètres du Softmax et 200 échantillons comme échantillons de test. Afin de donner une meilleure estimation de la performance de la généralisation, les résultats rapportés de l'ensemble de données sont les moyennes de plusieurs expériences indépendantes. En utilisant cette base de données, la méthode a atteint une exactitude de 88,11 %. Les auteurs ont testé également leur méthode sur une autre base de données publique qui comprend 3 618 images de lumière du jour et 1 306 images de lumière de nuit. Les véhicules présents sur les images se répartissent en cinq catégories : Camion, Minivan, Bus, Voiture de tourisme, et Berline (y compris les véhicules utilitaires sport (SUV)). La méthode a atteint une exactitude de 96,1 % dans les conditions de jour et de 89,4 % dans les conditions de nuit. [133] a utilisé un système CNN constitué de deux étapes. La première étape consiste en l'augmentation des données afin d'atténuer le problème des ensembles de données déséquilibrés. La deuxième est la création du modèle de CNN. Les auteurs ont construit une base de données afin d'évaluer leur modèle. Cette base de données comprend 2400 échantillons répartis en quatre catégories (600 images dans chaque catégorie) : Bus scolaire, Ambulance, Police, et CTM (Compagnie Marocaine de Transport). Ils ont calculé donc les métriques de Precision, Recall et Accuracy. Les résultats obtenus pour les catégories Bus scolaire, Ambulance, Police, et CTM sont : 88%, 90%, 88%, 90% respectivement, pour la métrique Precision, et 83% en Recall pour toutes les catégories. Ce système fait partie d'une application intégrée qui permettra une gestion automatisée des feux de circulation basée sur la détection automatique du type de véhicule. Dans [146], les auteurs ont également utilisé les CNN avec des images de l'ensemble de données BIT-Vehicle avec une faible résolution/vue frontale. La

distribution des classes de la base de données BIT-Vehicle n'est pas uniforme. Pour équilibrer les classes, les auteurs ont créé un sous-ensemble de 476 échantillons sélectionnés au hasard pour chaque classe, soit un total de 2 856. Ensuite, les échantillons ont été répartis en trois ensembles : l'ensemble de training, de test et de validation contenant des proportions respectives de 65%, 30% et 5%. Les résultats montrent que les classes SUV et Sedan contiennent la plupart des erreurs de classification, car ces types de véhicules ont des apparences très similaires. La même observation est faite dans les travaux de Dong et al. [147], mais dans ce cas, c'est la classe "SUV" qui présente les taux d'exactitude les plus faibles. Les résultats du modèle dans l'ensemble de test ont atteint une exactitude de 93,90 %. En outre, [8] a développé un modèle utilisant CNN, mais sur des images de vue arrière. Les images de la vue arrière du véhicule sont détectées à partir des images vidéo du trafic. L'image est normalisée à la taille fixe de 32*32 pixels, puis par l'opération de suppression de la moyenne, l'image normalisée est envoyée dans le réseau entraîné. En tant que dernière couche du réseau neuronal, le classifieur softmax peut fournir la probabilité que l'image d'entrée appartient à un certain type de véhicule. La base de données utilisée est collectée à partir des images vidéo de la circulation, fournies par le département de la police de la circulation de Jinan. Les échantillons obtenus comprennent trois types de véhicules. Le nombre total d'images de berlines est de 1200, de microbus de 1030 et de SUV de 535. Pour équilibrer les classes, les auteurs ont augmenté les images SUV à 1070 en utilisant l'opération Gaussian Blur. Comparé à la méthode HOG-SVM, les résultats expérimentaux montrent que l'algorithme est plus performant et atteint une exactitude de 97,88 %. Ils montrent aussi que le SUV a le taux de reconnaissance le plus élevé dans les deux méthodes. En raison de leur profil similaire, le Microbus et le SUV obtiennent des taux de classification faibles.

Dans la littérature, les chercheurs ont proposé différentes solutions pour prioriser les ambulances. Ils ont utilisé des technologies comme Bluetooth et ZigBee afin de détecter l'approche des ambulances. Nous nous trouvons aucun travail impliquant la détection des ambulances (ou autre véhicule d'urgence) à partir des images.

4.3. Le système proposé

Cette section donne un aperçu de notre proposition de système de classification des véhicules indépendamment de la vue, qui est résumé à la Figure 64. Le système comporte deux phases. La première phase consiste à reconnaître la vue du véhicule suivant la vue reconnue, la deuxième phase consiste à reconnaitre la catégorie de

véhicule soit en vue de face ou soit en vue d'arrière. Nous limitons notre système à reconnaitre 4 types de véhicules : autobus, voiture, moto et camion.

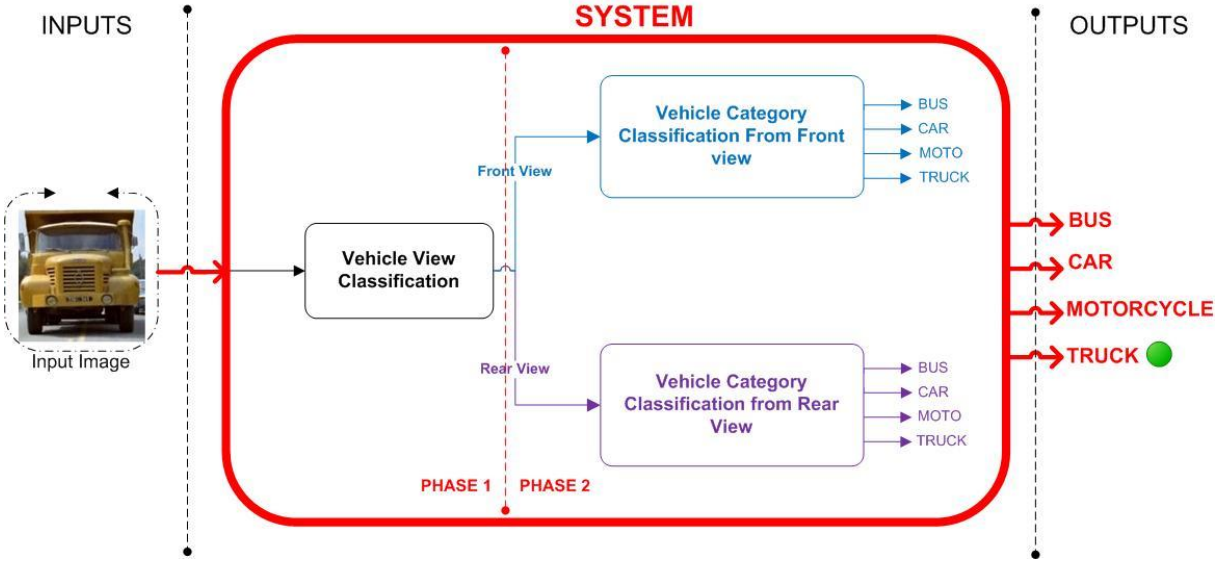


Figure 64. Système de classification des catégories de véhicules indépendant de la vue

Dans la suite, nous présentons chaque phase.

- Classification des vues de véhicules :

Cette étape a été traitée en détails dans le chapitre précédent.

- Classification des catégories de véhicules :

Cette étape vise à adapter un modèle classant les catégories de véhicules en vue de face et un autre pour la vue arrière. Nous avons donc utilisé ici deux bases de données. Chaque base de données contient les images d'une vue avec quatre classes dans lesquelles chaque classe contient les images d'une catégorie de véhicule. Comme expliqué précédemment, la construction d'un modèle nécessite deux étapes (Figure 65) : l'extraction des caractéristiques et la classification. Nous extrayons les caractéristiques des données de formation à l'aide de trois descripteurs : HOG, LBP et filtre de Gabor. Ensuite, nous formons un classifieur en utilisant ces caractéristiques pour comprendre comment les variables d'entrée données sont liées à la classe, ici nous avons utilisé les classifieurs SVM et k-NN. Nous appelons le processus de construction d'un modèle : étape de formation, d'apprentissage ou de modélisation.

La section suivante présente toutes les méthodes que nous avons utilisées dans cette étape.

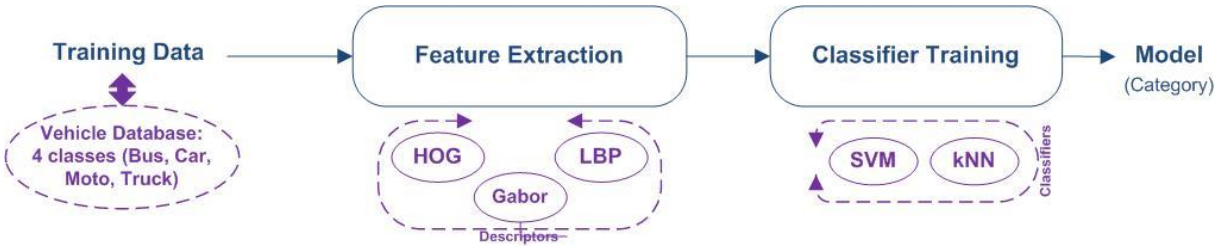


Figure 65. Schéma de construction d'un modèle de classification des catégories de véhicules

4.4. Méthodologie

Pour résoudre les problèmes de classification des catégories de véhicules et de vues, nous avons utilisé les deux approches : l'apprentissage automatique et l'apprentissage profond. Particulièrement, nous avons utilisé les classifieurs (SVM et k-NN) combinés avec les descripteurs (HOG, LBP, et Gabor) et aussi les réseaux de neurones convolutifs (CNNs).

Les classifieurs (SVM, k-NN) et les descripteurs (HOG, LBP, et Gabor) sont tous définit en détail dans le Chapitre 1. De plus, nous avons présenté dans le Chapitre 2 une description détaillée sur les CNNs en montrant le concept de construction de caractéristiques de niveau supérieur à partir de caractéristiques de niveau inférieur.

Dans la définition des SVM (Chapitre 1), nous nous sommes limités au problème de classification binaire. Dans cette section, nous présenterons le cas du problème de classification multi-classes.

- Support Vector Machine (multi-classes) :

La classification SVM est essentiellement une technique de classification binaire avec des étiquettes de catégorie ayant seulement deux valeurs, 1 et -1 respectivement. Elle est modifiée pour traiter des tâches multi-classes dans des situations réelles. Deux méthodes sont proposées pour cette adaptation, à savoir les techniques « One Against One » (1A1) et « One Against All » (1AA) [148]. L'approche « 1AA » représente l'approche SVM multi-classe la plus ancienne et la plus courante [149] et implique la division d'un ensemble de données de N classes en deux classes. En revanche, l'approche « 1A1 » implique la construction d'une machine pour chaque paire de classes, ce qui donne N(N-1)/2 machines. Lorsqu'elle est appliquée à un point de test, chaque classification donne un vote à la classe gagnante et le point est étiqueté avec la classe ayant reçu le plus de votes. Cette approche peut être encore modifiée pour pondérer le processus de vote. Selon la théorie de l'apprentissage automatique, il est

admis que l'approche 1AA présente plus d'inconvénients que l'approche 1A1 ; ses performances peuvent être compromises en raison d'ensembles de données d'apprentissage déséquilibrés. Cependant, l'approche 1A1 nécessite plus de calculs car les résultats de toutes les paires de SVM doivent être calculés [148].

3.1. Expérimentations & Résultats

Diverses expériences ont été effectuées pour évaluer les performances de toutes les méthodes présentées ci-dessus. Dans cette section, nous présentons 4 expériences avec les résultats que nous avons obtenus.

3.1.1. Base de données et matériel

En pratique, plusieurs facteurs rendent la classification des catégories de véhicules très difficile. Parmi eux : le manque de disponibilité des bases de données ; aucun des ensembles de données existants n'est adapté à notre objectif. Nous ne trouvons que quelques bases de données/vue de haut ou seulement des bases de données pour les voitures/vues latérales et arrière. Ainsi, nous avons collecté, à partir de nombreux sites web, des milliers d'images contenant des bus (Annexe 1/ Figure 97 et Figure 101), des voitures (Annexe 1/ Figure 98 et Figure 102), des motos (Annexe 1/ Figure 99 et Figure 103) et des camions (Annexe 1/ Figure 100 et Figure 104) à partir de vues avant et arrière. Nous avons construit trois ensembles de données. Le premier est destiné à la classification des vues (Chapitre 2 / Expérience 2). Les deux autres ensembles de données sont utilisés pour la classification des catégories de véhicules : un ensemble de données pour les images des véhicules avant (2182 images), l'autre pour les images des véhicules arrière (2000 images). Chacun de ces deux ensembles de données contient 4 classes : chaque classe pour une catégorie (500 images pour chaque classe).

Toutes les images ont été redimensionnées à une taille fixe de 150∗150 pixels. À partir de chaque base de données, deux ensembles sont créés avec une répartition aléatoire : un ensemble d'entraînement et un ensemble de test. Quatre-vingt pour cent des images ont été incluses dans l'ensemble d'entraînement (ou d'apprentissage). Les 20 % restants ont été inclus dans l'ensemble de test. Ces images sont sélectionnées dans des conditions météorologiques différentes.

Pour le problème de la détection des ambulances, nous avons rassemblé un ensemble de données d'images d'ambulances provenant de la base de données ImageNet [15], qui contient 900 images. Cette base de données comprend deux classes d'images, la première classe contient les images positives (images des ambulances sous différentes

vues, Échantillons de la classe des Ambulances), et l'autre classe contient des images dites négatives ne contenant pas d'ambulances.

Nous avons exécuté nos algorithmes sur un ThinkPad Lenovo avec un processeur Intel R , CoreTM, i5 7ème génération de CPU @ 2.50GHz 2.71GHz, RAM 8Go. Nous avons implémenté les algorithmes sous Matlab.

3.1.2. Métriques d'évaluation

Pour évaluer la performance des modèles construits, nous avons calculé de nombreuses mesures en utilisant les matrices de confusion extraites pour chaque modèle. Dans ces matrices, les lignes et les colonnes indiquent respectivement les classes réelles et prédites. Les valeurs dans la diagonale représentent les taux de classification corrects tandis que celles en dehors de la diagonale représentent les erreurs de classification.

Dans ce travail, nous avons deux problèmes : la classification binaire (arrière/avant) et la classification multiclasse (illustrée à la Figure 66) avec 4 classes dans l'ordre Bus, Voiture, Moto et Camion (Bus, Car, Moto, and Truck). Ainsi, pour une matrice de confusion avec plus de deux classes, nous avons calculé les métriques (True Positives TP, False Positives FP, False Negatives FN) de chaque classe comme suit :

- Le TP de chaque classe est la valeur correspondante dans la diagonale principale. Par exemple, la TP de la classe Moto est TP_M.

- Le FN de chaque classe est la somme des valeurs de la ligne correspondante, à l'exclusion de l'élément diagonal principal TP. Par exemple, le FN de la classe Camion est :

$$FN_T = ER_{TB} + ER_{TC} + ER_{TM} \qquad \text{Équation 49}$$

- TP+FN=100%

- Le FP pour chaque classe est la somme des valeurs de la colonne correspondante, à l'exclusion de TP. Par exemple, le FP de la classe Bus est :

$$FP_B = ER_{CB} + ER_{MB} + ER_{TB}$$ Équation 50

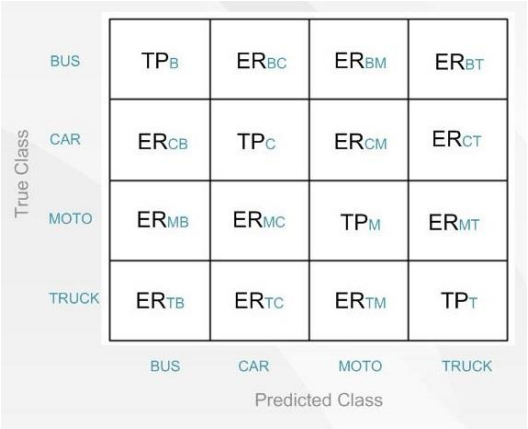


Figure 66. Matrice de confusion multi-classes (4 classes)

3.1.3. Résultats et Discussion

Dans cette section, nous présentons les différentes expériences menées afin de construire un système robuste pour classifier les catégories des véhicules indépendamment de la vue arrière/avant. En fait, nous avons mené 3 expériences. La première expérience est consacrée à la reconnaissance de la vue d'un véhicule (arrière/face) en comparant plusieurs approches. De même dans la seconde expérience, des méthodes de reconnaissance sont étudiées et comparées pour classer les catégories des véhicules selon chaque vue.

En sélectionnant les meilleurs modèles des 2 expériences précédentes, nous construisons, dans une 3ème expérience, notre système où les catégories ont été classées indépendamment de la vue.

Dans toutes nos expériences, nous avons suivi des configurations particulières de nos modèles utilisés. Dans la phase d'apprentissage, chaque modèle est formé sur 3 sous-ensembles de données d'entraînement en utilisant la méthode de rééchantillonnage-validation croisée 4-folds. Au total, nous avons généré 12 modèles pour chaque méthode, ce qui a permis d'obtenir différentes erreurs de prédiction sur les données de test. Et nous avons ensuite calculé la performance moyenne. Ce processus permet d'obtenir des prédictions plus stables que celles de n'importe quel modèle de membre individuel.

Concernant les classifieurs, nous avons utilisé k-NN et le SVM. Comme expliqué précédemment dans le chapitre 1, pour classer un échantillon, le k-NN commence par

trouver tous les k échantillons d'entraînement les plus proches, puis prédit la classe par un vote majoritaire [143]. Donc ici, dans les expériences, nous avons simplement choisi k=1 et la distance métrique=euclidienne. De plus, nous avons utilisé le SVM avec un noyau linéaire dans la classification des vues. Et pour la classification par catégorie, nous avons utilisé l'approche multi-classe « 1A1 ».

Quant aux descripteurs d'images, nous avons utilisé le HOG. Les caractéristiques du HOG ont été extraites des images des véhicules avant et arrière, comme le montre la Figure 3. Les entrées sont des images couleur de taille$150 * 150$. Tout d'abord, nous avons converti les images couleur en images grises. Chaque image RGB à trois canaux d'entrée est convertie en une image à un seul canal, la formule de transformation est la suivante :

$$Gry = 0{,}3.R + 0{,}59.G + 0{,}11.B$$

Équation 51

Notre HOG utilise une taille de cellule de $8 * 8$. Donc le nombre de cellules par image est de $18 * 18 = 324$ cellules. Chacune avec un descripteur de dimension 36. Donc la dimension de vecteur caractéristique HOG de toute l'image est de 324x36=11664.

Pour les modèles CNN, l'image d'entrée d'une couche de convolution a une dimension $D * D * c$, où "D" est la hauteur et la largeur, et "c" est le nombre de canaux. Pour les images RGB, $c = 3$ [150]. Dans nos expériences, nous avons D=150 et $c = 3$. Comme le montre la Figure 67, l'image passe par 6 couches de convolution mélangées avec 3 couches de pooling, puis 4 couches entièrement connectées. L'empilement de plusieurs couches de convolution avant l'application d'une couche pooling permet aux couches de convolution de développer des caractéristiques plus complexes avant que l'opération destructive de pooling ne soit effectuée [50].

Toutes les couches de convolution utilisent la fonction d'activation ReLU. Nous ajoutons la couche Dropout à la fin pour régulariser le réseau. Dans le réseau de classification, nous avons utilisé une couche entièrement connectée suivie d'un classifieur Softmax. La taille de la couche de sortie est identique au nombre de classes que nous attendons.

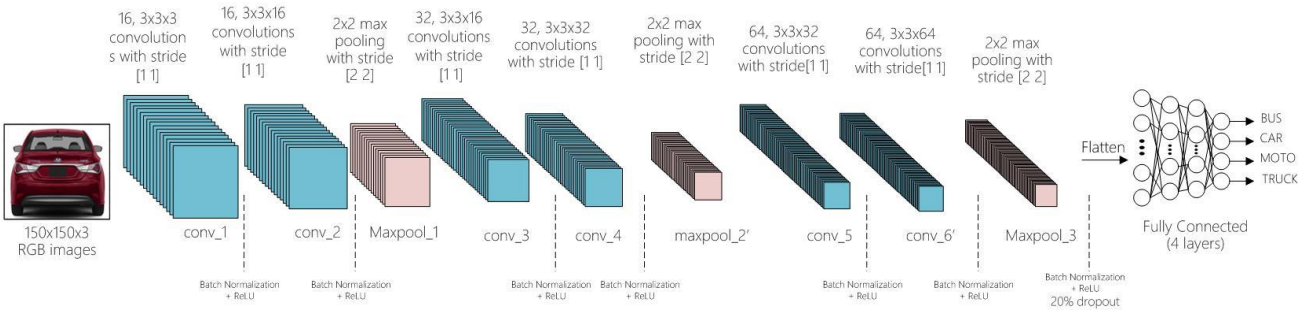


Figure 67.Architecture de CNN utilisée pour la classification des catégories

Les CNN apprennent automatiquement les valeurs de filtre (poids) en entraînant le réseau à l'aide d'une technique appelée rétropropagation, tout en continuant à avoir une erreur de classification minimale. Dans les premières couches, le réseau essaie de reconnaître certains aspects des images, tels que les bords, les formes (carrés, triangles, cercles) et les couleurs. De plus en plus de couches apprennent des motifs complexes jusqu'à ce que tous ces motifs puissent aider le réseau à apprendre la classification de l'entrée [151].

a- Expérience 1 (Classification des vues de véhicules)

Cette expérience a été entièrement décrite dans le Chapitre 2. Nous avons classé les vues des quatre catégories de véhicules (Bus, Car, Moto, Truck). Nous rappelons ici les principaux résultats.

Nous avons trouvé que le CNN a obtenu de meilleurs résultats (exactitude de 94.29%). HOG+kNN et HOG+SVM ont obtenu la même exactitude (~93%). LBP+kNN et LBP+SVM ont obtenu les exactitudes 86.06% et 62.79%, respectivement.

Cette classification est plus difficile, surtout pour les motos, car les deux vues (arrière et face) ont presque la même forme.

b- Expérience 2 (Classification des catégories de véhicules) :

- Vue de face

Dans cette expérience, nous avons testé nos algorithmes avec l'ensemble de données de la vue de face. L'évaluation a été basée sur les métriques que nous avons présentées auparavant. Les résultats sont affichés sur la Figure 68 et le Tableau 8.

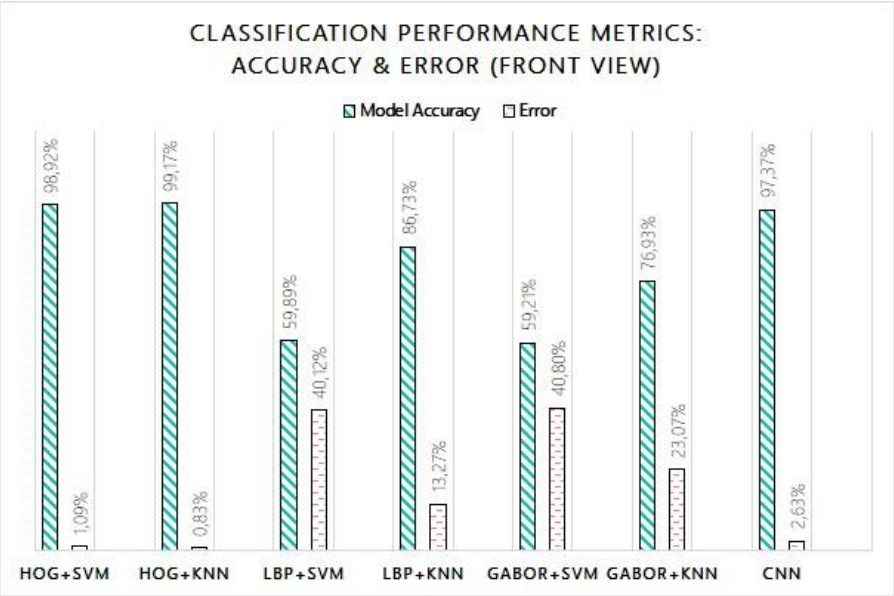


Figure 68. L'exactitude et l'erreur obtenues pour chaque modèle (Vue de face)

La Figure 68 présente la précision et l'erreur globales de chaque modèle. Le Tableau 8 détaille le pourcentage des exactitudes et des erreurs FP et FN obtenus pour chaque classe et chaque modèle. Nous présentons les résultats de la classification des types de véhicules sur la base des données de test. Plusieurs observations peuvent être tirées de ce tableau. Premièrement, la combinaison de HOG et k-NN est beaucoup plus performante que les autres combinaisons. Elle surpasse LBP et CNN. [134] a également constaté que la exactitude des caractéristiques du HOG est supérieure à celle du réseau neuronal convolutif (de 2,3 %). Le HOG est une grille dense ; il est utilisé en tant que caractéristiques de bas niveau, il peut donc extraire des informations plus riches des images [143]. Deuxièmement, le filtre de Gabor est le moins efficace parmi tous les descripteurs que nous avons testés. Gabor et LBP sont moins efficaces avec le SVM, mais lorsqu'ils sont combinés avec k-NN, ils donnent de meilleurs résultats. Tous les descripteurs sont très performants en combinaison avec le classifieur k-NN.

La Figure 68 nous permet de conclure que : HOG+kNN, HOG+SVM, CNN et LBP+kNN sont les meilleurs modèles, par ordre décroissant.

Tableau 8. Métriques de classification pour chaque catégorie (Vue de face)

		HOG +SVM	HOG +kNN	LBP +SVM	LBP +kNN	Gabor +SVM	Gabor +kNN	CNN
Classe BUS	TP (%)	95.835	97.40	39.8475	71.355	70.05	63.5425	91.494

	FN (%)	4.17	2.60	60.16	28.65	29.9575	36.4625	8.5049
	FP (%)	0	0.26	7.99	7.6705	50.695	16.6825	0.402466
Classe CAR	TP (%)	99.8225	99.8225	95.1775	92.8575	75	82.32	99.2275
	FN (%)	0.1775	0.1775	4.8175	7.1425	25	17.6775	0.772466
	FP (%)	0	0.257	127.04	16.89.5	53.36	33.9885	0.7741
Classe MOTO	TP (%)	100	100	78.75	94.585	41.2525	75	99.72166
	FN (%)	0	0	21.2525	5.42	58.7525	25	0.27833
	FP (%)	0	0	3.02	4.115	7.83	9.005	0.0866
Classe TRUCK	TP (%)	100	99.485	25.775	88.145	50.515	86.8575	99.0558
	FN (%)	0	0.515	74.4	11.8575	49.48	13.1475	0.9441
	FP (%)	4.3475	2.7825	22.405	22.85	51.305	32.615	9.23663

Pour chaque catégorie, nous avons calculé la TP, la FN et la FP comme indiqué dans le Tableau 8. Nous pouvons voir que tous les modèles (sauf Gabor+SVM et LBP+SVM) ont bien classé les trois classes MOTO, CAR et TRUCK. La classe MOTO était évidemment le plus facile à classer correctement.

Les résultats expérimentaux montrent que HOG+SVM a produit des erreurs FN_{BUS} = 4,17%, FN_{CAR} = 0,1775% et FP_{TRUCK} = 4,3475%. On observe que FN_{BUS} + FN_{CAR} = FP_{TRUCK}, ce qui signifie que 4,17% de bus et 0,1775% de voitures sont classés comme

des camions. Même observation pour le modèle HOG+kNN, 2,6% des bus et 0,1775% des voitures sont classés comme des camions, mais aussi 0,515% des camions sont classés comme des bus (0,26%) et des voitures (0,257%). Ces classifications erronées sont dues à la similitude de l'apparence de la vue de face de ces catégories.

- Vue arrière

Les mêmes algorithmes sont, cette fois-ci, testés sur des données de véhicules vue arrière. La Figure 69 montre les performances de classification de chaque modèle construit. Comme le montre cette figure, les modèles HOG+SVM, HOG+kNN et CNN ont atteint des exactitudes de classification plus élevées, respectivement de 99,58 %, 99,47 % et 97,43 % (ensemble de tests). Les mesures de chaque classe sont détaillées dans le Tableau 9. Idem pour la vue de face, HOG a obtenu les meilleurs résultats et k-NN surpasse SVM. Par exemple, pour la classe BUS, lorsque LBP est combiné avec SVM, il fournit une exactitude de 9,09% alors que lorsqu'il est combiné avec k-NN, il fournit une exactitude de 93,1825%. Cependant, lorsque k-NN et SVM sont combinés avec HOG, ils fonctionnent de manière légèrement similaire.

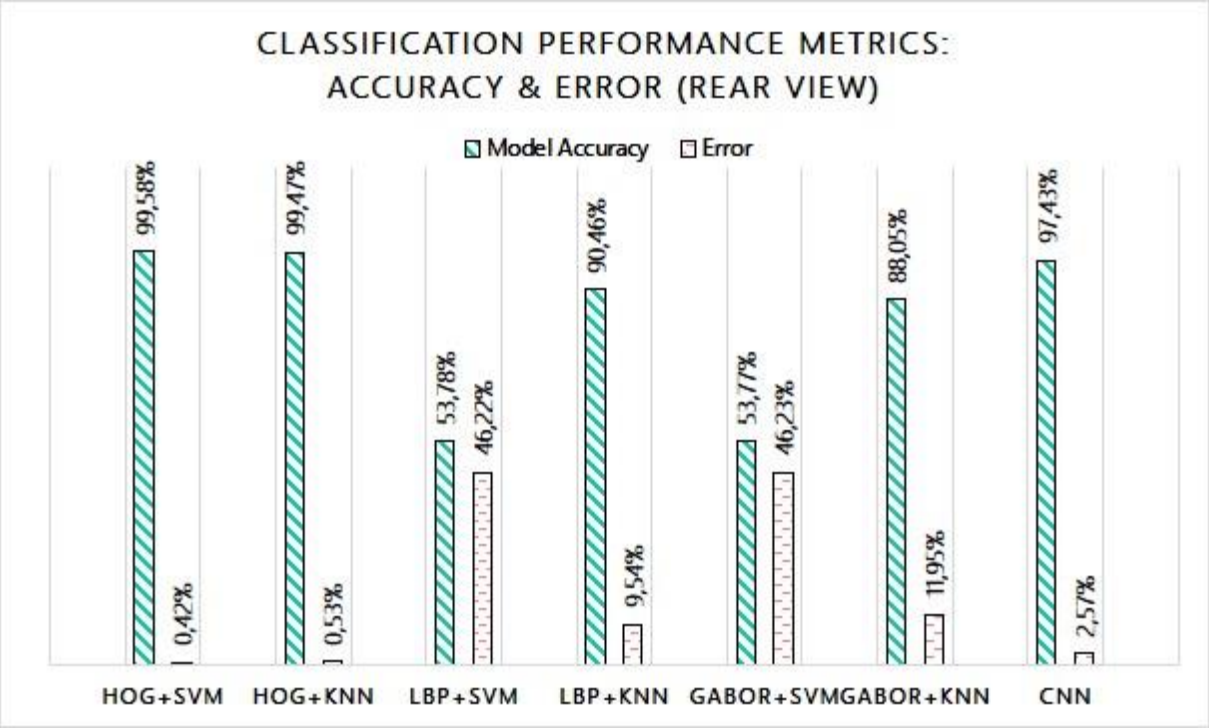


Figure 69. L'exactitude et l'erreur obtenues pour chaque modèle (vue arrière)

Le Tableau 9 présente les TP, FN et FP par classe.

Tableau 9. Métriques de classification pour chaque catégorie (Vue arrière)

		HOG +SVM	HOG +kNN	LBP +SVM	LBP +kNN	Gabor +SVM	Gabor +kNN	CNN
Classe BUS	TP(%)	99.4325	98.295	9.09	93.1825	17.0425	82.9525	96.0216
	FN(%)	0.5675	1.705	90.9075	6.8175	82.955	17.045	3.9775
	FP(%)	0.2875	0.425	4.3175	11.195	8.975	6.2675	4.04716
Classe CAR	TP(%)	99.7125	100	95.1125	93.675	85.9225	86.21	96.8375
	FN(%)	0.2875	0	4.8875	6.325	14.08	13.8	3.16232
	FP(%)	0	0	135.01	10.495	97.22	21.445	1.923326
Classe MOTO	TP(%)	99.5825	100	63.335	96.25	50.4175	86.665	99.5825
	FN(%)	0.4175	0	36.6625	3.75	49.585	13.3325	0.41746
	FP(%)	0.425	0	15.1725	3.58	61.2725	13.545	0.47916
Classe TRUCK	TP(%)	99.5975	99.5975	47.5825	90.7275	61.6925	96.3725	97.3125
	FN(%)	0.425	0.425	52.4225	9.2925	38.3075	3.6275	2.687166
	FP(%)	0.985	1.705	30.38	4.6725	17.46	6.5475	3.7948

En résumé, dans cette 1ère expérience avec les deux vues avant/arrière, il est facile de voir que la catégorie la plus difficile à classer était le BUS, surtout de face. En effet, la plupart des bus contiennent les mêmes détails et informations que les camions de face par rapport à ceux de derrière. Dans les deux expériences, le descripteur HOG était la clé de cette classification. Il a permis de mieux caractériser la forme et l'apparence des véhicules. Le HOG est considéré comme l'un des descripteurs de caractéristiques les plus précis pour les problèmes de classification visuelle [152].

CNN a surpassé les autres méthodes dans l'expérience de classification des vues de véhicules. Cependant, elle a été surpassée par HOG avec k-NN et SVM dans l'expérience de classification des catégories de véhicules. Ceci est dû à l'impact de la disponibilité des données. Dans la première expérience, nous avons utilisé une base de données plus large que dans la seconde. La Figure 70 illustre, en général, cet impact sur les performances des méthodes traditionnelles d'apprentissage automatique et des

réseaux de neurones. Les méthodes traditionnelles atteignent de meilleurs résultats avec de petites bases de données. Elles ne sont pas influencées par l'augmentation de la quantité de données (elles se stabilisent à un certain point). Cela est dû au fait que, par exemple dans les SVM, seuls les vecteurs de support interviennent dans la définition de l'hyperplan optimal. Par conséquent, même si nous augmentons les données, seul un petit sous-ensemble de points de données est nécessaire pour le calcul de la solution, les autres échantillons ne participent pas du tout à sa définition. En d'autres termes, l'ajout d'échantillons à l'ensemble d'apprentissage qui ne sont pas des vecteurs de support n'a aucune influence sur la solution finale. L'élargissement de l'ensemble d'apprentissage a donc moins d'influence que pour un modèle de Deep Learning, où tous les points participent à la solution. Plus la quantité de données fournies est importante, plus les performances des algorithmes de DL sont élevées. En raison de cette relation, nous devons associer l'apprentissage profond à de grands ensembles de données. Les modèles DL peuvent donc surpasser toutes les autres méthodes classiques [153]. Ils n'ont pas de limites et sont même allés jusqu'à dépasser les performances humaines dans des domaines tels que la vision par ordinateur.

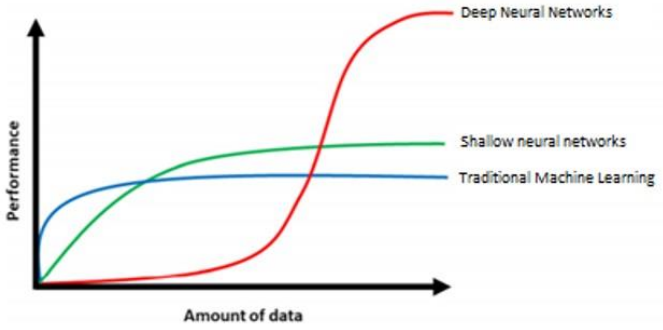


Figure 70. Impact de la disponibilité des données sur les algorithmes [153]

Il était difficile de comparer nos résultats avec ceux d'autres travaux sur la reconnaissance des catégories de véhicules, car aucun ensemble d'images de référence ne contient les catégories de leurs vues avant et arrière. Cependant, nous relevons dans le Tableau 10 quelques études de la littérature auxquelles nous comparons nos résultats. [141] a atteint une exactitude de 96,26%. [133] a atteint une exactitude de 93,90 % en utilisant les CNN sur les images de vue frontale des véhicules. Dans [8], le modèle CNN a atteint une exactitude de 97,88% sur les images de vue arrière. [134] ont atteint une exactitude de 92,6% sur l'ensemble des données des véhicules BIT. [147] ont utilisé le même ensemble de données et ont atteint 96,1 % dans des conditions de jour et 89,4 % dans des conditions de nuit.

Tableau 10. Comparaison des résultats de la classification des catégories de véhicules avec les travaux existants

Étude	Vue	Approche	Performance
[141]	Vue de profil	SVM and Random Forest	96.26%
[133]	Avant	CNN	93.90%
[8]	Arrière	CNN	97.88%
[134]	Avant	HOG-SVM amélioré	92,60%
[147]	Avant	CNN semi-supervisé	96.1% / jour 89.4% / nuit
Proposée	Avant et arrière	HOG, LBP, Gabor, SVM, k-NN, CNN	99.58% / arrière 99.17% / avant

c- Expérience 3 (système global) :

Après avoir étudié les deux problèmes de classification des vues et des catégories, nous avons choisi les meilleurs modèles pour créer notre système (Figure 71) afin de déterminer à quelle catégorie appartient chaque véhicule quel que soit sa prise de vue.

Les résultats des deux premières expériences ont démontré que le modèle CNN est le plus performant pour reconnaître les vues de véhicules, et le modèle HOG+kNN est le plus précis pour classer les catégories de véhicules de face, et le modèle HOG+SVM pour la vue arrière.

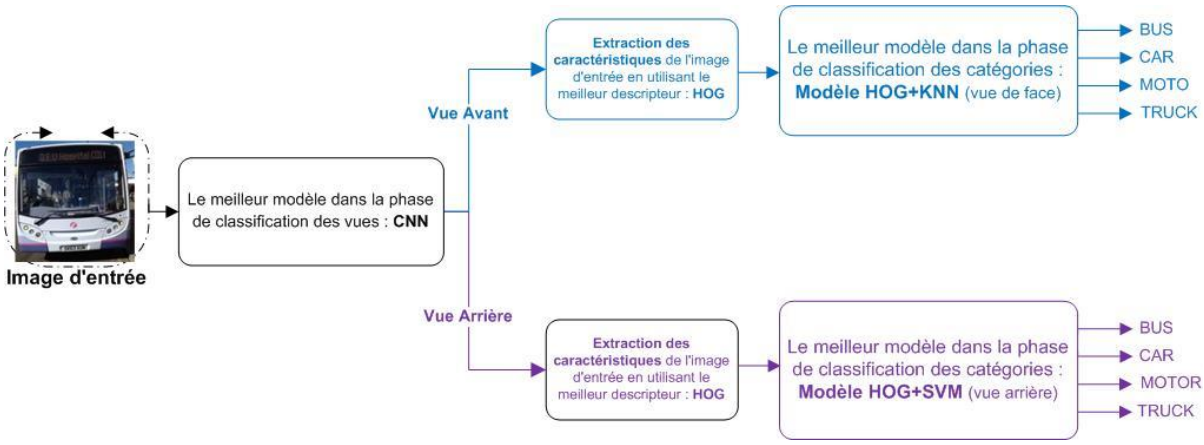


Figure 71. Schéma du système

Tableau 11. Mesure des performances du système

	Exactitude globale		Erreur		Temps d'exécution/Image (Second)
Système	95.7746%	Avant: 98.29%	4.22%	Avant: 1.709%	12.46
		Arrière: 92.5%		Arrière: 7.5%	

Pour évaluer la performance de notre système, nous l'avons testé avec une liste de 209 images (différentes de celles utilisées dans les expériences précédentes). Cette évaluation a été basée sur de nombreux paramètres : exactitude globale, exactitude pour chaque vue, durée d'exécution pour chaque image. Les résultats de l'évaluation sont présentés dans le Tableau 11. Comme indiqué, le système a atteint une exactitude de 95,7746 % et une durée d'exécution de 12,46 secondes par image.

d- Expérience 4 / Cas spécial : Détection des ambulances

Le Tableau 12 présente les durées d'exécution et les taux de détection moyens de chaque modèle lors de la détection des ambulances. Comme le montre ce tableau, les caractéristiques LBP Gabor Magnitude et Gabor Phase nécessitent moins de temps pour être extraites que les caractéristiques HOG. Les histogrammes ont besoin de plus de temps pour être calculés. Nous avons observé que LBP+SVM est le modèle le plus rapide.

Tableau 12. Durée d'exécution et taux de détection moyens de chaque modèle

			Données d'apprentissage			**Données de test**		
Méthodes de détection des véhicules		Temps d'exécution/ Image (Second)	Détections correctes	Nombre d'extra-détections (moyenne)	Nombre de détections manquées (moyenne)	Détections correctes	Nombre d'extra-détections (moyenne)	Nombre de détections manquées (moyenne)
Combinaison HOG+SVM		0.049230	100	0	0	97.78	13.8	0
Combinaison HOG+KNN	k=1 Distance= 'euclicean'	0.187373	100	0	0 %	97.22	0	3.1 %
Combinaison LBP+SVM		0.015779	99.86	0	0.2	100	0	0
Combinaison LBP+KNN	k=1 Distance= 'euclicean'	0.023226	100	0	0	100	0	0
Combinaison Gabor-Magnitude+SVM		0.021879	100	0	0	100	0	0
Combinaison Gabor-Magnitude +KNN	k=1 Distance= 'euclicean'	0.028630	100	0	0	100	0	0
Combinaison Gabor-Phase SVM		0.021372	93.75	25.9	2.5	98.33	0	1.9
Combinbaison Gabor-Phase KNN	k=1 Distance= 'euclicean'	0.029105	100	0	0	97.78	0	2.5

Comme on peut le voir, les descripteurs LBP et Gabor Magnitude sont beaucoup plus performants que les autres descripteurs. Les caractéristiques de texture comme LBP sont robustes aux changements d'illumination et aux ombres. Tous les descripteurs fonctionnent mieux en combinaison avec le classifieur SVM.

Sur la base des résultats obtenus avec les données de test, l'ordre des meilleurs modèles en fonction de l'exactitude est le suivant : LBP+SVM, LBP+KNN, Gabor Magnitude +SVM, et Gabor Magnitude+ KNN.

Nous avons observé que HOG+SVM produit des détections faussement positives, causées par les changements d'illumination et les ombres. Gabor Phase+SVM, Gabor Phase+kNN, et HOG+kNN produisent des détections faussement négatives dues au camouflage (lorsque l'arrière-plan et le premier plan partagent des couleurs similaires).

3.2. Conclusion :

Dans ce chapitre, nous avons proposé un système de classification des catégories de véhicules (bus, voiture, moto et camion) indépendamment de la vue, en utilisant les algorithmes traditionnels d'apprentissage automatique et les CNN. Les caractéristiques ont d'abord été extraites des images en utilisant les descripteurs HOG, LBP et Gabor. Les caractéristiques obtenues sont ensuite entrées dans l'étape de classification ; ici, nous avons utilisé les classifieurs SVM et k-NN. Le système utilise une approche en deux phases. La première phase est utilisée pour reconnaître la vue du véhicule. La seconde reconnaît les catégories. Pour évaluer les modèles construits, nous avons utilisé trois bases de données que nous avons construites nous-mêmes. La première base de données comprend 4000 images, la seconde contient 2182 images de véhicules/vue de face et la troisième contient 2000 images de véhicules/vue arrière. Les résultats ont montré que CNN offre la plus grande exactitude dans l'étape de classification des vues. Mais à l'étape de classification des catégories de véhicules, le descripteur HOG était meilleur que Gabor et LBP. Il caractérise bien les orientations de vue du véhicule, le rendant plus résistant aux variations géométriques et d'éclairage. En fait, dans cette classification, il était facile de voir que la catégorie Moto était la plus facile à classer correctement avec une exactitude de 100% pour les deux vues. Cependant, les bus et les camions ont été mal classés en raison de leur des formes similaires. HOG a atteint une exactitude de 99,58% avec le SVM pour la vue arrière et 99,17% avec k-NN pour la vue de face. Enfin, le système global a atteint une exactitude de 95,7746%. Nous pouvons remarquer que, dans la classification, deux facteurs principaux affectent la performance du système : les caractéristiques sélectionnées et la disponibilité des données. Les résultats que nous avons obtenus montrent que le système peut être utilisé avec succès pour de nombreuses applications.

De plus, nous avons présenté une étude comparative des méthodes de détection des ambulances. Nous avons comparé les modèles construits à l'aide des classifieurs : SVM et kNN avec les descripteurs HOG, LBP et le filtre de Gabor. D'après les résultats de la comparaison, nous avons conclu que les descripteurs LBP et Gabor Magnitude fournissent les meilleurs résultats avec une exactitude de 100%. Le LBP avec le classifieur SVM est le plus rapide avec un temps d'exécution par image d'environ 0.015779s. En fait, les résultats obtenus montrent que le système de détection des ambulances peut être exploité avec succès pour les applications déjà mentionnées. Nous visons à intégrer l'étape de détection des ambulances dans le système global afin de les autoriser si elles effectuent des dépassements interdits.

"Every company these days is basically in the data business and they're going to need AI to civilize and digest big data and make sense out of it—big data without AI is a big headache".

Kevin Kelly, Co-founder of Wired

4. **Chapitre 4 :** Détection des dépassements interdits

Sommaire

4.1. Introduction

Les changements de voie dangereux, les dépassements illégaux et la conduite sur une mauvaise voie représentent un pourcentage élevé du total des accidents qui se produisent sur la route, juste après les excès de vitesse [154]. Il existe plusieurs scénarios de changement de voie dangereux, le plus courant étant le dépassement d'un véhicule en franchissant une ligne continue. Les accidents correspondants sont souvent assez graves en raison de la nature frontale des collisions. Le franchissement d'une ligne continue est une infraction au code de la route dans de nombreuses juridictions à travers le monde [7].

Dans de nombreux contextes de surveillance, l'intervention humaine n'est devenue plus suffisante pour détecter les infractions sur la route ; une grande partie des vidéos de transport est simplement stockée sans être examinée à cause de grands volumes de séquences vidéo. La détection automatique des infractions dans des scénarios de trafic présente un intérêt évident.

Récemment, la plupart des villes du monde ont commencé à intégrer diverses technologies dans le domaine des transports afin d'accroître le confort et la sécurité. Ces technologies sont appelées les Systèmes de transport intelligents (Intelligent Transportation Systems, ITS). Les ITS comprennent deux parties principales : les systèmes basés sur les véhicules et les systèmes basés sur l'infrastructure. Les ITS basés sur les véhicules sont des systèmes d'automatisation des véhicules, des systèmes de sécurité active, et les systèmes avancés d'aide à la conduite (ADAS). Les ITS basés sur les infrastructures sont les péages, le suivi et la surveillance du trafic, le contrôle du respect du code de la route, etc. [4].

Notre système de détection des dépassements de véhicules peut être classé dans les deux parties des ITS. En effet, si on utilise une caméra fixe (caméra de surveillance), le système sera considéré comme un système ITS basé sur l'infrastructure dont l'objectif sera le contrôle et l'application automatiques du code de la route au niveau des dépassements interdits. Il vise à donner des contraventions aux conducteurs qui franchissent la ligne continue. Si on utilise une caméra mobile (Dash Cam), le système sera considéré comme un système ITS basé sur le véhicule. Il peut être intégré dans les voitures autonomes et les ADAS. Il sera utilisé pour avertir le conducteur pendant sa conduite qu'une autre voiture est en train de le dépasser. La mise en œuvre de ces

technologies intelligentes aide la société à faire appliquer strictement le code de la route, et aide aussi les conducteurs à reconnaître les situations autour de leur propre véhicule. Par conséquent, le taux d'accidents peut être considérablement réduit.

Dans la littérature, il n'y a pas beaucoup de travaux qui tentent de résoudre ce problème à l'exception de presque trois études qui sont toutes basées sur le même principe. Cependant, aucune étude précédente n'a utilisé comme les méthodes que nous avons développées pour résoudre ce problème, donc nous pensons que c'est le premier travail qui aborde sérieusement les dépassements interdits.

En plus de la détection automatique des dépassements interdits, le système peut être utilisé aussi dans toutes les situations où les conducteurs peuvent utiliser la mauvaise direction de la circulation. Par exemple, il peut être installé dans des rues à sens unique, dans des parkings automatiques, etc.

Dans notre système, nous avons utilisé des techniques efficaces de traitement d'image et de vision par ordinateur afin de détecter et classifier les véhicules en mouvement dans des séquences vidéo acquises par des caméras statiques ou mobiles. Dans cette étape de la détection, nous avons utilisé le modèle Faster R-CNN et les techniques de la détection et du suivi d'objets multiples basé sur le mouvement. Les Faster R-CNN prédisent les boîtes englobantes dans une image en utilisant des « ancres » pendant le training, (les ancres ou «anchors, en anglais» sont des boîtes superposées sur l'image à différents échelles et rapports d'aspect). En outre, la détection du mouvement dans un flux vidéo consiste à identifier un mouvement physique réalisé dans la vidéo d'une caméra dans un environnement 3D. Cette tâche est d'autant plus difficile dans des cas complexes tels que des changements soudains d'illumination, des environnements dynamiques ou avec une caméra mobile [155].

Ce chapitre présente notre système et décrit les principales phases qui ont été utilisées pour son développement. Dans la première section, nous présenterons les travaux précédents sur le sujet des dépassements interdits des véhicules. Dans la Section 2, nous décrierons notre système en présentant les deux solutions proposées. Ensuite, nous présenterons en détail les méthodes utilisées dans chaque étape du système dans la Section 3. Dans la Section 4, nous présenterons les résultats d'expériences menées dans diverses scènes du monde réel. Les résultats expérimentaux sont fournis avec une brève discussion. La dernière section présentera les conclusions finales.

4.2. Etat d'art

Dans la littérature existante, il est rare de trouver de travaux qui tentent de résoudre le problème de détection des dépassements interdits des véhicules à l'exception de presque trois études. Le premier [156] présente une méthode de reconnaissance et de suivi du véhicule en infraction. L'arrière-plan statique est modélisé par un modèle gaussien mixte, et l'emplacement de la ligne de la voie est détecté par la transformation de Hough. Les informations sur les véhicules en mouvement peuvent être obtenues par la méthode de différence entre l'arrière-plan et l'image courante. Une fois qu'un véhicule est détecté, le centre de la tache mobile est utilisé pour vérifier si le véhicule change de voie illégalement. En fonction de la distance entre le centre du véhicule (ou la zone de mouvement) et les coordonnées de la ligne de la voie, le véhicule suspect peut être détecté. Si la distance entre le centre du véhicule et la coordonnée la plus proche de la ligne de la voie est inférieure à cinq pixels. Une méthode de décalage moyen est utilisée pour suivre le véhicule suspect, et la caméra de proximité est utilisée pour photographier la plaque d'immatriculation en fonction du centre de la fenêtre de suivi. Des tests réels sur route montrent que l'exactitude atteint 80 %. L'approche suivie dans [156] peut être influencée par le problème des ombres qui peuvent être incluses comme faisant partie des objets. Ainsi, le centre de la zone en mouvement (véhicule + ombre) sera considéré comme le centre du véhicule. Et donc cela va influencer sur la distance entre le véhicule et la ligne et donc donner des fausses détections ou des détections manquées (cela dépend de l'orientation de l'ombre du véhicule). Il est donc nécessaire de commencer par enlever l'ombre du véhicule. Un autre système mentionné dans la littérature est connu sous le nom de Police Eyes [154]. Police Eyes fonctionne en détectant les blobs en mouvement et leur intersection avec la région de violation. Les étapes clés du système Police Eyes sont les suivantes. (1) Initialisation du système : Un opérateur spécifie une région de violation et la zone de traitement sur une image initiale indiquée en cliquant manuellement sur des points de l'image. (2) Acquisition d'images à partir de caméras IP : Les images sont acquises en continu à partir de deux caméras IP. Une image à basse résolution provenant d'une caméra est utilisée pour détecter les blobs et identifier les violations. Une image à haute résolution provenant de la deuxième caméra est utilisée pour identifier le véhicule. Les images et les clips vidéo peuvent être utilisés ultérieurement à des fins de preuve et pour rejeter des cas non triviaux tels que l'évitement de dangers. (3) Mise à jour du modèle de fond et soustraction du fond : Le modèle d'arrière-plan est initialisé en utilisant une seule image, puis mis à jour pour chaque nouvelle image. Un modèle de mélange gaussien est utilisé

pour chaque pixel de l'image. Le nombre de composantes gaussiennes est constamment adapté par pixel. La différence de l'image courante par rapport au modèle d'arrière-plan produit une image de premier plan. (4) Détection des ombres : Les pixels d'ombre doivent être supprimés de l'image de premier plan pour éviter les fausses détections de violation. Les pixels d'ombre sont identifiés en utilisant une combinaison de corrélation croisée normalisée entre la région de premier plan et les pixels d'arrière-plan correspondants, ainsi que les distances vectorielles RGB entre les pixels de premier plan et les pixels d'arrière-plan sous-jacents. (5) Extraction de blobs : Les blobs du premier-plan sont extraites par l'analyse des composantes connectées après avoir effectué des opérations morphologiques sur l'image du premier-plan pour éliminer les blobs bruyants. Le profil de base est extrait pour chaque blob restante. La base d'un blob est identifiée comme l'ensemble des pixels les plus bas du contour externe du blob. (6) Analyse des violations : L'analyse de la région d'intersection du profil de base de chaque blob avec la zone de violation est utilisée pour détecter les violations, en spécifiant des contraintes sur la longueur de l'intersection du profil de base d'un blob avec la largeur de la zone de violation au point de violation, et avec la longueur du profil de base du blob lui-même. Étant donné que L_B est la longueur de l'arc du profil de base d'une goutte, L_I est le nombre de pixels du profil de base coupant la zone de violation, L_r est la largeur de la zone de violation à la rième rangée de l'image, où r est la coordonnée *y* du centre de gravité du fond de la goutte (affinée et rognée), et L_i ; $i = 1 \ldots N$ est la longueur des *N* parties du profil de base d'une goutte résultant de l'élimination de la partie qui coupe la zone de violation, les auteurs considèrent que le blob est impliqué dans un franchissement illégal de ligne pleine si $L_I > \frac{1}{3} L_r$ et si au moins un des cas suivants est vrai : *Cas 1* : La totalité de Blob se trouve à l'intérieur de la zone de violation, c'est-à-dire que $N = 0$. *Cas 2* : Étant donné qu'une seule partie du Blob se trouve à l'extérieur de la zone de violation, c'est-à-dire $N = 1$, si $L_I > \frac{1}{3} L_1$, elle est considérée comme une violation. *Cas 3* : Étant donné qu'il y a deux parties ou plus du Blob en dehors de la zone de violation, c'est-à-dire $N \geq 2$, si $L_I + min(L_1, .., L_N) > \frac{1}{3} \sum_{i=1}^{N} L_i$, ils considèrent qu'il y a violation. Cette approche consistant à utiliser le profil de base pour détecter les infractions présente certaines limites. Des détections erronées peuvent se produire lorsque la zone de violation est une bande mince et que le Blob extraite est petite, en raison de la division du blob résultant de la soustraction du fond et de la détection des ombres. Si la zone de violation est grande, les violations causées par les motos peuvent être manquées

lorsqu'elles se produisent près de la caméra, car les profils de base des motos sont très courts par rapport à la largeur de la zone de violation. Afin de détecter les infractions causées par les motos, la contrainte sur la longueur de l'intersection du profil de base du blob avec la zone d'infraction peut être relâchée si nécessaire, en fonction du type de région d'infraction, afin de réduire le nombre de détections d'infractions manquées. La méthode basée sur l'intersection du blob avec la ligne n'est pas très efficace. Par exemple, dans le cas d'une dashcam ou d'une caméra (installée sur le côté de la route), le blob peut apparaître comme s'il croisait la ligne sans passer sur l'autre voie. En outre, le blob peut être plus large que le véhicule, notamment avec les ombres. Ainsi ces deux raisons peuvent également être à l'origine de fausses alarmes. Dans le troisième travail [157], les auteurs ont utilisé les techniques de traitement d'images et d'apprentissage automatique pour détecter les anomalies dans les mouvements des véhicules. Ces anomalies comprennent l'arrêt et le déplacement en sens inverse. La détection du flux de véhicules comprend trois phases : la collecte de données, l'extraction de caractéristiques et l'apprentissage. Les données sont collectées sur une autoroute publique. Le trafic montant est divisé en plusieurs voies dédiées ; chaque voie est équipée de ses propres caméras. Pour chaque véhicule, plusieurs images consécutives sont enregistrées pour l'estimation du flux optique. Les mouvements des véhicules sont enregistrés par plusieurs caméras afin d'éviter les occlusions. L'occlusion est une situation indésirable lorsque la région d'intérêt est bloquée ou masquée par un autre objet. Les informations sur le flux optique ainsi que la date, l'heure et la plaque d'immatriculation sont utilisées pour générer une base de données de véhicules (pour le training et le test). Après l'acquisition de plusieurs images consécutives, l'algorithme de flux optique est exécuté sur cet ensemble d'images. Pour ce faire, la méthode pyramidale Lucas-Kanade dans Open CV est utilisée. Les endroits très détaillés, comme les bords de la plaque d'immatriculation et la calandre, sont sélectionnés pour le suivi. Après l'exécution du flux optique, la sélection des caractéristiques est effectuée en deux phases. Dans la première étape, une opération de filtrage est appliquée pour éliminer les vecteurs de mouvement inutiles. Comme toutes les parties d'un véhicule se déplacent ensemble, les vecteurs de mouvement de ces parties doivent être cohérents. Cette information étant connue, les valeurs aberrantes sont éliminées. Dans un deuxième temps, les amplitudes des vecteurs de mouvement sont traitées pour éliminer les mouvements non pertinents autres que ceux du véhicule concerné. Les informations sur le flux optique sont combinées pour générer un seul vecteur de mouvement montrant la direction du flux. Après la phase d'acquisition et d'estimation du flux optique, les auteurs ont utilisé

l'apprentissage non supervisé. La direction du trafic le long de la route est déterminée pendant cette phase d'apprentissage, Pour évaluer la performance du schéma de détection proposé, un logiciel de détection de flux de véhicules est mis en œuvre. La plateforme expérimentale est construite sur OpenCV, C++ et Java. Chaque véhicule de l'ensemble de test est comparé aux résultats du training pour trouver des similitudes. La méthode d'apprentissage automatique kNN est utilisée comme classifieur. Les expériences sont réalisées sur un ordinateur avec un processeur de 1,73 GHz et 8 Go de RAM. kNN est utilisé pour trouver la direction du trafic. Puisque la plupart des véhicules se déplacent dans la bonne direction, la majorité des véhicules se regrouperont pour former un cluster représentant la direction avant. D'autre part, le mouvement inverse et d'autres mouvements non pertinents formeront des clusters plus petits. La méthode proposée est évaluée sur une autoroute publique et les résultats de détection sont prometteurs.

4.3. Description du système proposé

Comme expliqué précédemment, nous voulons développer un système de vision par ordinateur qui vise à détecter automatiquement les dépassements interdits. Ce système peut être intégré dans des systèmes de surveillance du trafic dans le cas où la caméra est statique, et lorsqu'elle est mobile, il peut être intégré dans des véhicules autonomes. Dans ce but, nous proposons deux solutions pour détecter les véhicules en dépassement interdit.

4.3.1. Première solution de détection du dépassement interdit

La première solution est basée sur trois étapes principales : la détection de la ligne continue, la détection du véhicule et la classification des véhicules avant/arrière qui détermine s'il s'agit d'un dépassement interdit ou non. Chacune de ces étapes sera présentée en détail. Le schéma général de la première solution est donné dans la Figure 72.

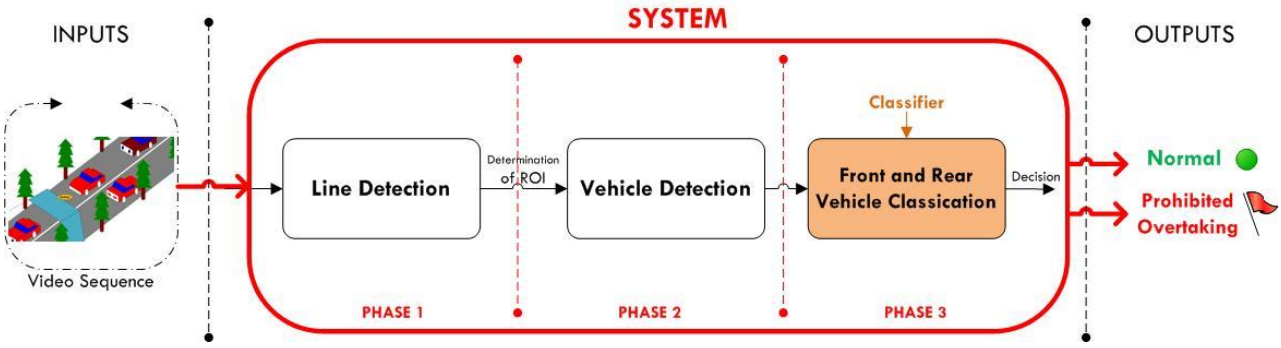


Figure 72. Système global pour la détection des dépassements interdits (solution 1)

Si on a une séquence vidéo en entrée (acquise par caméra statique ou mobile), on commence par détecter la ligne continue afin de déterminer la région d'intérêt (Region Of Interest, ROI) qui est ici la voie interdite aux véhicules de l'autre voie. On détecte ensuite les véhicules dans cette ROI afin de chercher l'existence du véhicule intru dans une voie. Dans la première solution, nous avons trouvé le véhicule intru en se basant sur la détermination de la vue avant ou arrière du véhicule. Par exemple, si nous traitons les dépassements des véhicules de la voie droite de la route. Donc la voie interdite pour les véhicules de cette voie est la voie gauche, comme l'illustre la Figure 73. Donc les vues normales des véhicules pour les deux voies sont : la voie droite possède les vues arrière, et la voie gauche possède les vues d'avant. Donc, nous détectons chaque véhicule qui existe dans la ROI, et nous classons les vues : si nous trouvons, comme résultat, la vue arrière donc il s'agit d'un dépassement interdit (Figure 74), sinon la situation est normale.

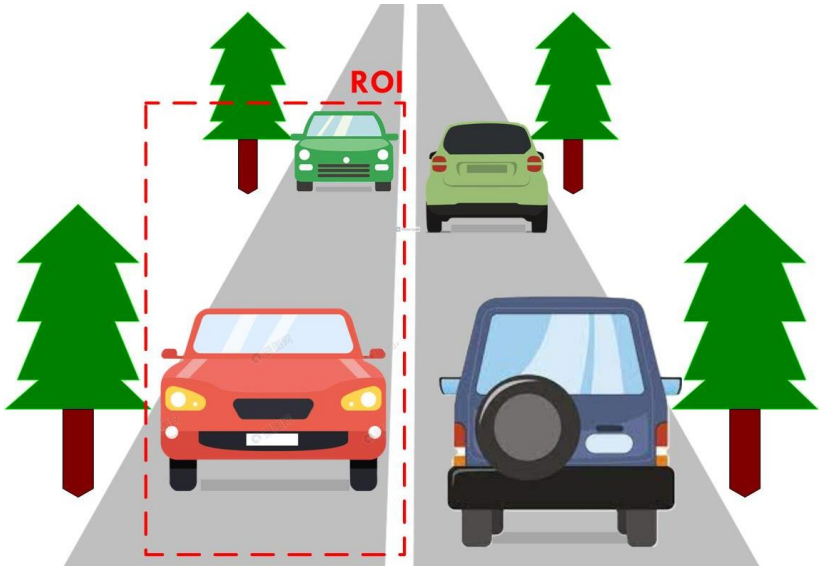


Figure 73. La région d'intérêt pour les véhicules de la voie droite

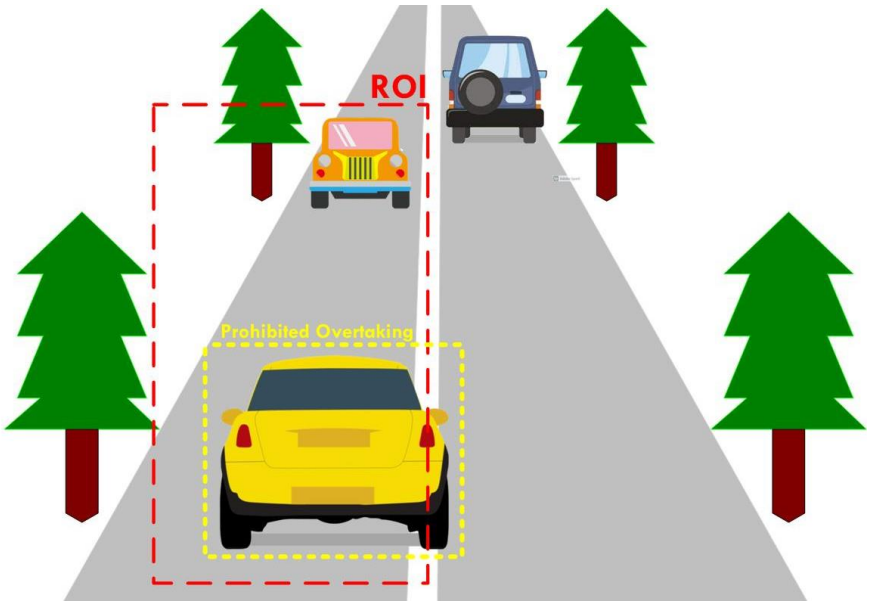


Figure 74. Détection d'un véhicule intru dans la voie gauche ; dépassement interdit d'un véhicule appartenant à la voie droite

Chaque étape de cette première solution (Figure 72) est expliquée en détail comme suit :

1) Détection de la ligne continue (Line Detection) : nous détectons d'abord la ligne en utilisant la technique de la transformée de Hough. Nous extrayons ensuite les coordonnées de la ligne afin de déterminer automatiquement la région d'intérêt ROI, qui est la zone interdite aux véhicules d'une voie spécifique (la zone gauche ou droite de la ligne). Si le système ne parvient pas à détecter la ligne continue (en raison du mauvais état de la route), il donne à l'opérateur la possibilité de déterminer la ROI manuellement, surtout lorsque la caméra est fixe (la ROI peut être déterminée juste après l'installation de la caméra de surveillance puisque la scène et la ligne restent toujours fixes).

2) Détection de véhicules : Dans cette étape, nous avons utilisé les techniques de détection de véhicules dans une séquence vidéo telles que le détecteur Faster R-CNN, et les techniques de détection d'objets en mouvement et suivi basé sur le mouvement. La détection des véhicules est l'étape principale du système global car des résultats corrects de détection des véhicules donnent une meilleure détection des dépassements interdits.

3) Classification des vues avant/arrière: Dans cette dernière étape de la première solution, nous avons construit le classifieur en utilisant les algorithmes d'apprentissage automatique, d'apprentissage profond et d'apprentissage par transfert, comme déjà présenté dans le Chapitre 2. Nous avons donc appliqué ici le meilleur modèle construit dans les expériences du Chapitre 2.

4.3.2. Deuxième solution de détection du dépassement interdit

La deuxième solution (Figure 75) est également composée de trois étapes principales : détection de la ligne, détection du véhicule, mais la dernière étape est l'identification du sens du mouvement du véhicule. C'est cette étape qui nous donne la décision finale ici ; s'il s'agit d'un dépassement interdit ou non. Nous allons présenter cette étape en détail.

Ainsi, pour cette solution, le système est essentiellement basé sur l'analyse du mouvement des véhicules afin de détecter les véhicules circulant dans la mauvaise direction.

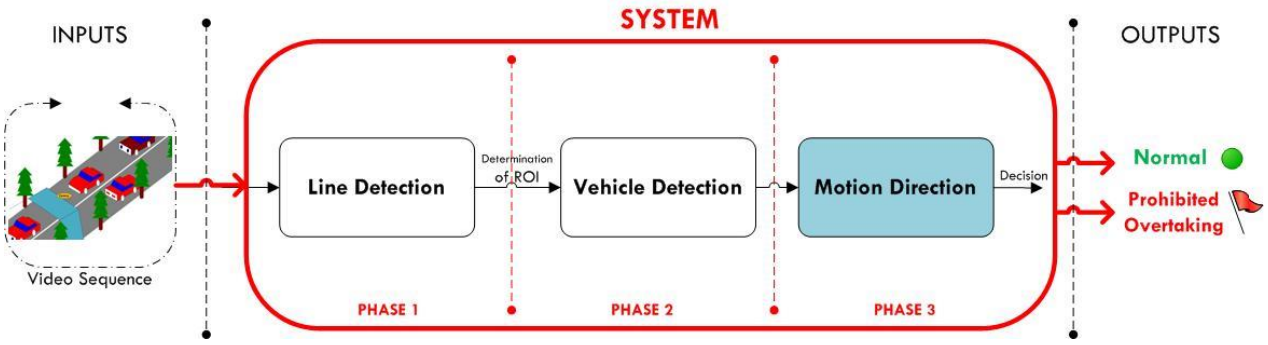


Figure 75. Système global pour la détection des dépassements interdits (solution 2)

Comme le montre la Figure 75, l'étape finale consiste à identifier le sens du mouvement du véhicule dans une voie. Pour ce faire, nous pouvons calculer la différence entre deux positions successives de la boîte englobante (Bounding Box).

Tout d'abord, une boîte englobante (Bbox) est un carré entourant les objets d'une image. Les coordonnées de la boîte englobante sont les suivantes (ligne, colonne, largeur, hauteur). Nous représentons ces coordonnées du Bbox par rapport à l'image dans la Figure 76.

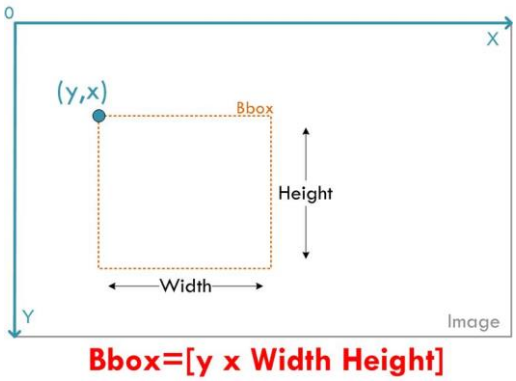


Figure 76. Coordonnées d'un Bounding Box sur une image

Nous considérons le même exemple de la première solution, c'est-à-dire que la ROI est la voie de gauche. La Figure 77 montre le mouvement du véhicule sur la ROI dans les deux cas (cas normal et cas de dépassement interdit). Elle présente le déplacement de la boîte englobante aux instants t1 et t2 (avec t2=t1+dt).

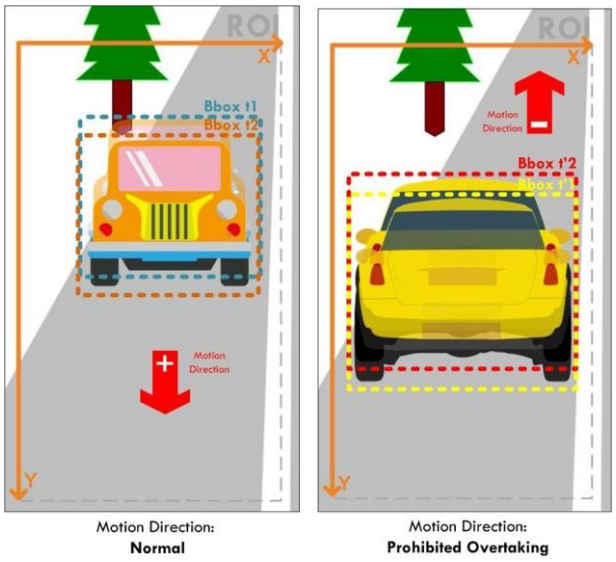


Figure 77. Le mouvement du véhicule sur le ROI dans le cas normal et dans le cas d'un dépassement interdit

La Figure 78 montre les coordonnées des deux Bbox et calcule leur différence afin de déterminer le sens du mouvement.

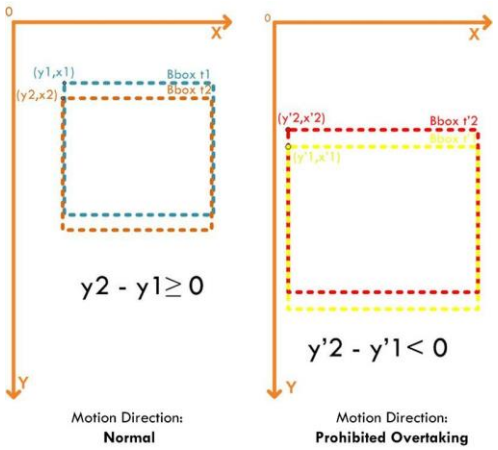


Figure 78. Sens du mouvement dans le cas normal et dans le cas d'un dépassement interdit

Si la caméra est mobile (installée dans le véhicule), l'angle de vue de la caméra sera latéral comme illustré à la Figure 79 et donc le processus de calcul du sens du mouvement change (voir Figure 80). Ici, le mouvement devient un peu horizontal (plutôt que vertical) alors qu'avant il était vertical. C'est-à-dire que le mouvement varie considérablement le long de x plutôt que de y (voir Figure 80).

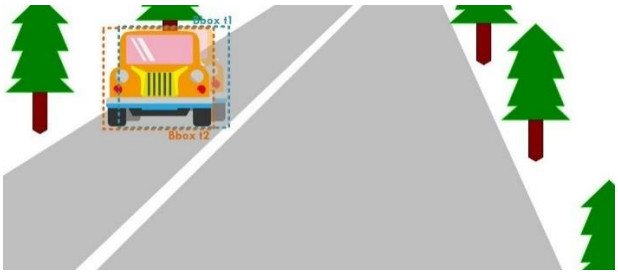


Figure 79. L'angle de vue de la caméra mobile

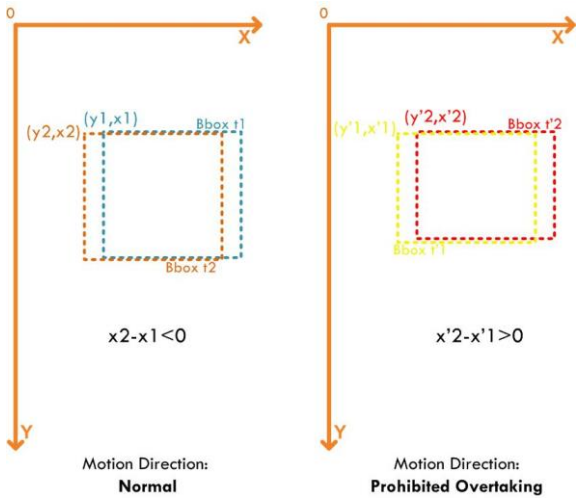


Figure 80. Cas Caméra mobile : Sens du mouvement dans le cas normal et dans le cas d'un dépassement interdit

Pourquoi avons-nous identifié la direction du mouvement en utilisant les déplacements du premier point (en haut à gauche) de chaque Bbox ? En fait, ce qui nous a guidé pour choisir ce point est le scénario présenté dans la Figure 81 – elle illustre l'état du mouvement du Bbox lorsque le véhicule quitte la ROI.

Tout d'abord, nous ne pouvons pas choisir les points R1 et R2, ou S1 et S2, car ils restent fixes dans cette période de sortie du véhicule de la ROI. Cependant, il est possible d'utiliser C1 et C2, en fait au début de nos expériences nous avons utilisé C1 et C2 les déplacements du centre pour identifier le sens du mouvement, mais nous avons constaté que cela donne des erreurs parce que la distance est plus étroite par rapport à la distance entre les points en haut (voir Figure 81). Les points optimaux sont donc P1 et P2, qui sont déjà calculés par défaut (dans les coordonnées Bbox). Il est également possible d'utiliser les points Q1 et Q2 mais ils nécessitent plus de calculs et donc plus de temps.

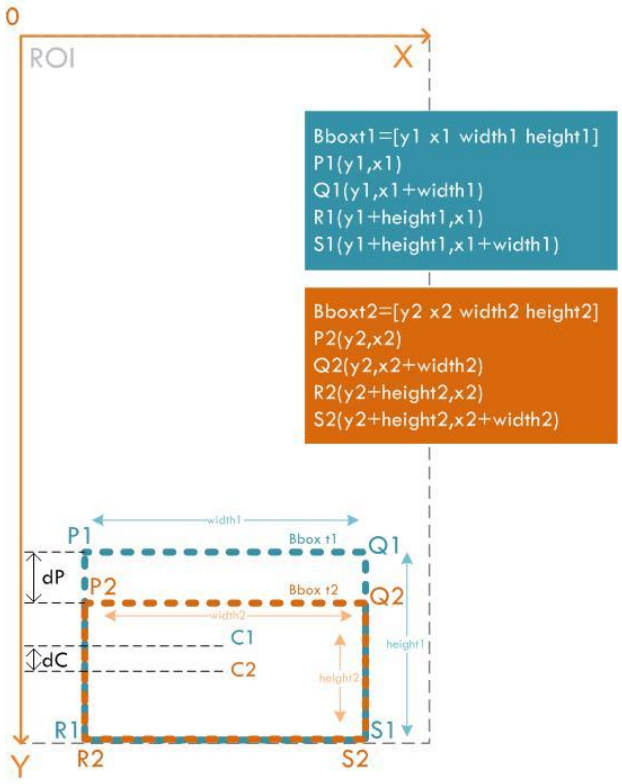


Figure 81. Scénario de sortie du véhicule de la région d'intérêt

4.4. Méthodologie

4.4.1. Hough transform

La transformée de Hough : La transformée de Hough est une technique d'extraction de caractéristiques utilisée en analyse d'images, en vision par ordinateur et en traitement numérique des images [158]. Le but de cette technique est de trouver des instances imparfaites d'objets dans une certaine classe de formes par une procédure de vote. Cette procédure de vote s'effectue dans un espace de paramètres, à partir duquel les objets candidats sont obtenus comme des maxima locaux dans un espace dit d'accumulation qui est explicitement construit par l'algorithme de calcul de la transformée de Hough.

La transformée de Hough classique (inventée en 1959 par Paul Hough [159] [160]) s'intéressait à l'identification des lignes dans l'image, mais plus tard, la transformée de Hough a été étendue à l'identification des positions de formes le plus souvent des cercles ou des ellipses (c'est la transformée généralisée de Hough [161] développée par Richard Duda et Peter Hart en 1972).

Par exemple, en prenant comme représentation des lignes droites (l'équation), toute ligne droite est complètement spécifiée par la valeur des paramètres (a, b).

$$y = ax + b \qquad \text{Équation 52}$$

L'inconvénient de cette approche est que les lignes verticales posent un problème ; la pente tend vers l'infini lorsque la ligne est verticale. En 1972, Duda et Hart ont proposé un autre type de représentation :

$$\rho = x.cos\theta + y.sin\theta \qquad \text{Équation 53}$$

La droite est complètement spécifiée par le couple (ρ, θ) [7].

Tout d'abord, un détecteur de contours (comme filtre de Canny, Sobel…) peut être utilisé comme étape de prétraitement pour identifier tous les points de contour de l'image.

Chacun des points des contours identifiés (x, y) va alors permettre une projection dans un plan (le plan transformé) des coordonnées polaires de toutes les droites passant par ce point. Les équations des droites passant par chacun de ces points (x, y) sont alors représentées par (ρ, θ).

Prenons comme exemple sur la Figure 82. Un point unique dans le plan image peut être représenté par l'intersection d'une collection de deux ou plusieurs lignes droites passant par ce point, qui correspondent à une collection de points tombant sur une courbe unique dans l'espace des paramètres. Inversement, une ligne droite dans le plan image correspond à un point d'intersection de plusieurs courbes dans le plan des paramètres [7].

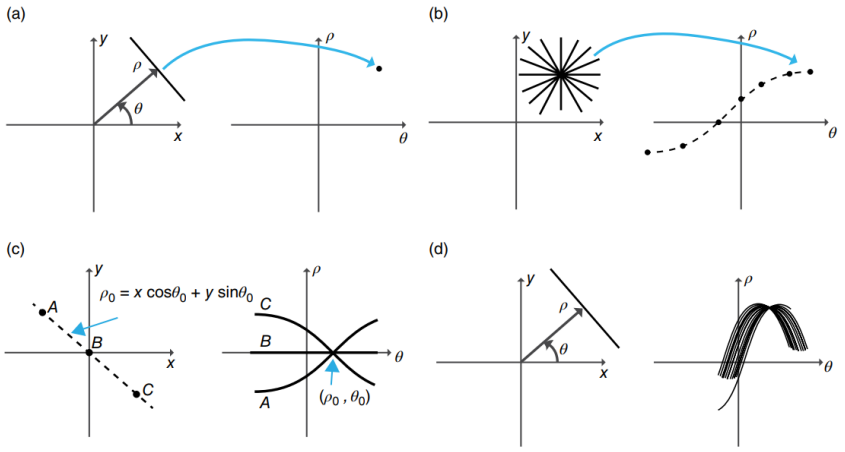


Figure 82. Transformation de Hough : (a) transformation du plan de l'image - plan des paramètres ; (b) transformation en un seul point ; (c) transformation en trois points ; et (d) transformation en ligne droite [7]

On associe donc à chaque point une densité correspondant au nombre de courbes qui le traversent. Plus la densité est élevée, plus les points de contour sont situés sur cette ligne. La ligne est donc très probablement un segment dans l'image.

4.4.2. Détection et suivi des véhicules dans un flux vidéo

Le choix de la technique appliquée pour la détection des véhicules dépend du type de caméra statique ou mobile. Une séquence vidéo capturée par une caméra statique a des caractéristiques différentes de celle capturée par une caméra mobile. Dans le flux vidéo d'une caméra fixe, les éléments statiques gardent leurs positions et leurs apparences à l'exception d'événements externes tels que des changements d'éclairage ou le déplacement d'un objet statique par un humain. L'apparence, la forme et la position des éléments mobiles varient en fonction de leurs mouvements, occultations et poses. En revanche, dans le flux vidéo d'une caméra mobile, tous les éléments, qu'ils soient statiques ou non, se comportent comme des éléments en mouvement : ils changent de position et de forme dans les images, ils peuvent être occultés par d'autres éléments, ils peuvent apparaître et disparaître du champ de vision de la caméra et leurs apparences varient en fonction de leurs positions et de leurs poses par rapport à la caméra [155].

Une scène filmée par une caméra fixe est composée de deux types d'éléments : des objets statiques et des objets/sujets en mouvement. Dans les séquences vidéo, on remarque que les éléments mobiles se déplacent devant la partie statique de la scène. Cette disposition en profondeur dans l'espace est définie dans le langage

cinématographique par les termes d'arrière-plan et de premier plan (ou fond) [155]. Plusieurs techniques reposent sur le fait que la partie statique de la scène reste inchangée pendant la prise de vue et que les changements observés proviennent d'un objet en mouvement. Il est possible donc d'extraire l'objet en mouvement, en appliquant une soustraction entre l'image courante et une image de la scène (qui ne contient pas d'objet en mouvement). Les techniques de ce type sont regroupées sous le terme de soustraction d'arrière-plan. Le résultat de la soustraction effectuée entre l'image courante et le modèle d'arrière-plan est un masque contenant les sujets en mouvement, appelé masque de premier plan [155].

La détection est plus complexe lorsque la caméra est mobile. L'environnement observé par ce type de caméra apparaît en mouvement et il est plus difficile de distinguer les objets qui sont en mouvement des autres qui composent la partie statique de la scène.

La détection d'un objet en mouvement est l'une des étapes importantes des applications de vision par ordinateur. En particulier, la détection des véhicules est l'étape principale de notre système de détection de dépassements interdits. Dans ce travail, nous avons utilisé plusieurs approches pour détecter les véhicules à savoir le détecteur Faster R-CNN, et les techniques de détection d'objets en mouvement et suivi basé sur le mouvement.

a- R-CNN, Fast R-CNN, Faster R-CNN, Mask R-CNN

Pour la caméra mobile, nous avons utilisé le Deep Learning et plus précisément le Faster R-CNN. Ce dernier est appliqué sur les images du flux vidéo afin de détecter les véhicules.

La détection d'objets est le processus qui consiste à trouver et à classer des objets dans une image. Une approche d'apprentissage profond, les régions avec réseaux neuronaux convolutifs (R-CNN), combine des propositions de régions rectangulaires avec des caractéristiques de réseaux neuronaux convolutifs. R-CNN est un algorithme de détection en deux étapes. La première étape identifie un sous-ensemble de régions dans une image qui pourrait contenir un objet. La deuxième étape classifie l'objet dans chaque région [162].

Les modèles de détection d'objets par régions avec des CNN sont basés sur les trois processus suivants : (1) Trouver des régions dans l'image qui pourraient contenir un objet. Ces régions sont appelées propositions de régions. (2) Extraire les

caractéristiques CNN des propositions de régions. (3) Classifier les objets à l'aide des caractéristiques extraites [162].

Plus récemment, plusieurs versions de R-CNN ont été développées, à savoir Fast R-CNN, Faster R-CNN, Mask R-CNN.

- R-CNN : Étant donné une image d'entrée, le R-CNN commence par appliquer un mécanisme appelé recherche sélective pour générer d'abord des propositions de régions (Figure 83), où chaque région est un rectangle qui peut représenter la limite d'un objet dans l'image. Selon le scénario, il peut y avoir jusqu'à deux mille régions. Ensuite, chaque région est soumise à un réseau neuronal convolutif pour produire des caractéristiques. Pour les caractéristiques de chaque région, un classifieur de machine à vecteur de support est utilisé pour déterminer quel type d'objet est contenu dans le ROI [163].

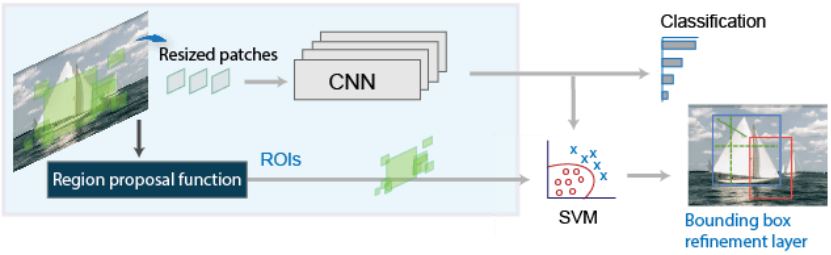


Figure 83. R-CNN [162]

- Fast R-CNN [164]: Comme dans le détecteur R-CNN, le détecteur Fast R-CNN (Figure 84) utilise également la recherche sélective pour générer ses propositions de régions [165]. Contrairement au R-CNN original qui calcule indépendamment les caractéristiques du réseau neuronal sur chacune des deux mille régions d'intérêt, le Fast R-CNN exécute le réseau neuronal une fois sur l'ensemble de l'image. À la fin du réseau se trouve une nouvelle méthode appelée ROI Pooling, qui met en commun les caractéristiques CNN correspondant à chaque proposition de région. Le Fast R-CNN est plus efficace que le R-CNN, car dans le détecteur Fast R-CNN, les calculs pour les régions qui se chevauchent sont partagés [162].

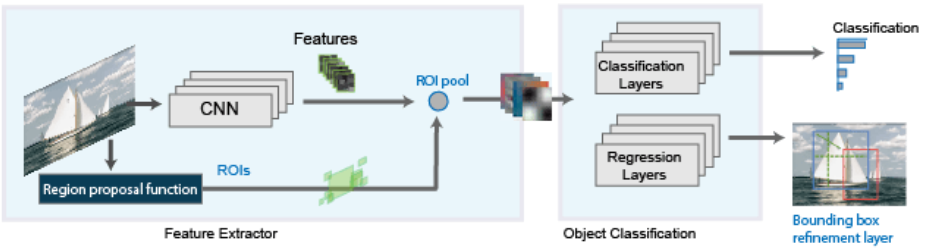


Figure 84. Fast R-CNN

- Faster R-CNN : Comme le montre Figure 85, le détecteur Faster R-CNN [166] ajoute un réseau de proposition de région (RPN, Region Proposal Network) pour générer des propositions de région directement dans le réseau au lieu d'utiliser un algorithme de la recherche sélective. Le RPN utilise des boîtes d'ancrage pour la détection des objets. La génération de propositions de régions dans le réseau est plus rapide et mieux adaptée aux données.

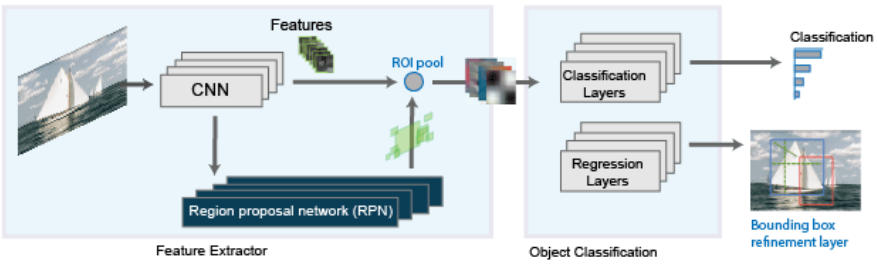


Figure 85. Faster R-CNN

- Mask R-CNN : Alors que les versions précédentes de R-CNN se concentraient sur la détection d'objets, Mask R-CNN ajoute la segmentation des instances. Mask R-CNN a également remplacé ROIPooling par une nouvelle méthode appelée ROIAlign, qui peut représenter des fractions de pixel [167] [168].

Dans ce travail, nous avons utilisé la version Faster R-CNN. Dans la section suivante, nous présenterons une brève description de cette version.

b- Faster R-CNN

Le Faster R-CNN, est composé de deux modules. Le premier module est un réseau profond entièrement convolutif (Fully Convolutional Network, FCN) qui propose des régions, et le second module est le détecteur Fast R-CNN [169] qui utilise les régions proposées. L'ensemble du système constitue un réseau unique et unifié pour la détection d'objets (Figure 86) [166].

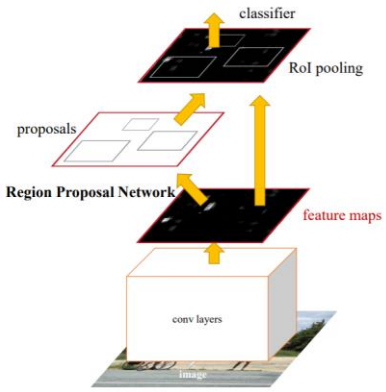


Figure 86. Faster R-CNN [166].

Le réseau de proposition de région (RPN) indique au module Faster R-CNN où regarder.

- Réseau de proposition de région (RPN) :

Un RPN prend une image (de n'importe quelle taille) en entrée et produit un ensemble de propositions d'objets rectangulaires, chacune avec un « objectness score ». Ce processus est modélisé avec un Fully Convolutional Network [170]. Comme l'objectif ultime est de partager les calculs avec un réseau de détection d'objets Fast R-CNN, les deux réseaux partagent un ensemble commun de couches convolutives [166].

Pour générer des propositions de régions, un petit réseau est glissé sur la carte de caractéristiques (feature map) produite par la dernière couche convolutionnelle partagée. Ce petit réseau prend en entrée une fenêtre spatiale n × n de la carte de caractéristiques convolutives d'entrée. Chaque fenêtre glissante est convertie en une caractéristique de dimension inférieure. Cette caractéristique est introduite dans deux couches jumelles entièrement connectées - une couche de box-régression (reg) et une couche de box-classification (cls). le mini-réseau est illustré dans la Figure 87. Comme le mini-réseau fonctionne selon le principe de la fenêtre glissante, les couches entièrement connectées sont partagées entre tous les emplacements spatiaux. Cette architecture est naturellement mise en œuvre avec une couche convolutive n×n suivie de deux couches convolutives sœurs 1 × 1 (pour reg et cls, respectivement) [166].

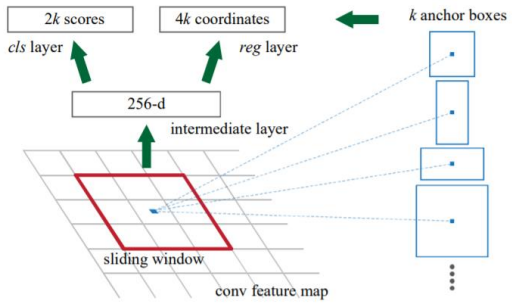


Figure 87. Réseau de propositions régionales [166].

- Ancres :

À chaque emplacement de la fenêtre coulissante, le modèle prédit simultanément plusieurs propositions de régions, où le nombre de propositions maximales possibles pour chaque emplacement est noté k. Ainsi, la couche reg a 4k sorties codant les coordonnées de k boîtes, et la couche cls produit 2k scores qui estiment la probabilité d'objet ou non pour chaque proposition. Les k propositions sont paramétrées par rapport à k boîtes de référence appelées ancres. Une ancre est centrée sur la fenêtre de glissement en question, et est associée à une échelle et un rapport d'aspect. Le modèle utilise par défaut 3 échelles et 3 rapports d'aspect, ce qui donne k = 9 ancres à chaque position de glissement. Pour une carte de caractéristiques convolutives de W×H emplacements, il y a W×H×k ancres au total [166].

c- La détection et le suivi d'objets multiples

La détection d'objets en mouvement et le suivi basé sur le mouvement sont des éléments importants de nombreuses applications de vision par ordinateur, notamment la reconnaissance d'activités, la surveillance du trafic et la sécurité automobile.

Le suivi est le processus de localisation d'un objet en mouvement ou de plusieurs objets au fil du temps dans un flux vidéo. Il associe les détections d'un objet sur plusieurs images. Le suivi de plusieurs objets nécessite la détection, la prédiction et l'association de données. La détection consiste à détecter les objets d'intérêt dans une image de la vidéo. La prédiction consiste à prédire l'emplacement des objets dans l'image suivante. L'association de données consiste à utiliser les emplacements prédits pour associer les détections entre les images et former des trajectoires [171].

- Détection :

Le choix de l'approche utilisée pour la détection des objets d'intérêt dépend du fait que la caméra soit fixe ou non.

- Détection d'objets à l'aide d'une caméra fixe :

Pour détecter des objets en mouvement avec une caméra fixe, on peut effectuer une soustraction d'arrière-plan. L'approche de la soustraction d'arrière-plan est efficace mais nécessite que la caméra soit immobile [171].

- Détection d'objets à l'aide d'une caméra mobile :

Pour détecter des objets en mouvement avec une caméra mobile, on peut utiliser une approche de détection par fenêtre glissante. Cette approche fonctionne généralement plus lentement que l'approche par soustraction d'arrière-plan [171].

➢ Prédiction

Pour suivre un objet dans le temps, on doit prédire son emplacement dans l'image suivante. La méthode de prédiction la plus simple consiste à supposer que l'objet sera proche de sa dernière position connue (la détection précédente sert de prédiction suivante). Cette méthode est particulièrement efficace pour les fréquences d'images élevées. Cependant, cette méthode de prédiction peut échouer lorsque les objets se déplacent à des vitesses variables ou lorsque la fréquence d'images est faible par rapport à la vitesse de l'objet en mouvement [171].

Une méthode de prédiction plus sophistiquée consiste à utiliser le mouvement précédemment observé de l'objet. Le filtre de Kalman prédit le prochain emplacement d'un objet, en supposant qu'il se déplace selon un modèle de mouvement, comme une vitesse constante ou une accélération constante [171]. Par définition, le filtre de Kalman est un algorithme qui estime l'état d'un système à partir de données mesurées. Il a été principalement développé par l'ingénieur hongrois Rudolf Kalman, qui a donné son nom au filtre. L'algorithme du filtre est un processus en deux étapes : la première étape prédit l'état du système, et la deuxième étape utilise des mesures incomplètes ou bruitées pour affiner l'estimation de l'état du système. Il existe aujourd'hui plusieurs variantes du filtre de Kalman original. Ces filtres sont largement utilisés pour les applications qui reposent sur l'estimation, notamment la vision par ordinateur [172].

➢ Association de données :

L'association de données est le processus d'association des détections correspondant au même objet physique sur plusieurs images. L'historique temporel d'un objet particulier est constitué de plusieurs détections et est appelé une trajectoire. La représentation d'une trajectoire peut inclure l'historique complet des emplacements précédents de

l'objet. Elle peut également se composer uniquement de la dernière position connue de l'objet et de sa vitesse actuelle [171].

Pour notre travail, nous avons effectué une détection automatique et un suivi basé sur le mouvement d'objets (camera fixe) en utilisant l'algorithme de soustraction d'arrière-plan basé sur des modèles de mélange gaussien. Chaque pixel de l'arrière-plan est modélisé avec un filtre de Kalman de sorte qu'il fonctionne bien dans les changements d'illumination. Le modèle de mélange gaussien est un excellent modèle utilisé pour déterminer si un pixel appartient ou non à l'arrière-plan. La détection d'un objet en mouvement est assez complexe en raison de la présence de bruit dans la scène [173]. La plupart des opérations morphologiques sont utilisées pour la réduction du bruit. Elles sont appliquées au masque de premier plan résultant. Enfin, l'analyse des blobs détecte les groupes de pixels connectés, qui sont susceptibles de correspondre à des objets en mouvement [174]. Pour le suivi basé sur le mouvement, l'association des détections au même objet est basée uniquement sur le mouvement. Le mouvement de chaque position est estimé par le filtre de Kalman. Ce filtre est utilisé pour prédire l'emplacement de la position dans chaque image et déterminer la probabilité que chaque détection soit attribuée à chaque position estimée [174].

Dans une image donnée, certaines détections peuvent être affectées aux positions prédites, tandis que d'autres détections et positions peuvent rester non affectées. Les positions attribuées sont mises à jour à l'aide des détections correspondantes. Les positions non attribuées sont marquées comme invisibles. Une détection non attribuée commence un nouveau suivi. Chaque position compte le nombre d'images consécutives où elle est restée non attribuée. Si le nombre dépasse un seuil spécifié, l'exemple suppose que l'objet a quitté le champ de vision et supprime la piste [174].

4.5. Expérimentations & Résultats

4.5.1. Base de données & matériel

Les vidéos ont été téléchargées à partir d'internet et d'autres filmées à l'aide d'une caméra (13MP) pouvant traiter 30 images par seconde pour une image de 1920x1080p couleur. Nous avons construit une base de données de 200 vidéos. Toutes ces vidéos ont été prises dans des conditions météorologiques et d'éclairage très variées, dans un environnement complexe, ou avec une caméra très éloignée, certaines autres sont de mauvaise qualité. Un échantillon est présenté dans la Figure 88.

Figure 88. Echantillon de la base de données vidéo (environnement complexe, caméra éloignée et mauvais éclairage)

Les spécifications de la machine utilisée pour développer notre système sont les suivantes : Lenovo ThinkPad avec un processeur Intel Core i5 7ème génération CPU @ 2.50 GHz 271 GHz, RAM 8Go. Par ailleurs, les algorithmes sont implémentés dans MATLAB version R2019a.

Nous avons calculé les détections correctes en tant que vrais positifs (TP) et vrais négatifs (TN), les fausses détections en tant que détections manquées (faux négatifs (FN)) et extra-détections (faux positifs (FP)). Ces valeurs ont été calculées manuellement.

4.5.2. Résultats et Discussion

Comme nous l'avons déjà mentionné, nous avons proposé deux solutions pour résoudre le problème de la détection des dépassements interdits des véhicules. De plus, pour chaque solution il y'a deux scénarios, l'un pour le cas où la caméra est fixe, et l'autre pour le cas où la caméra est mobile.

Dans nos expériences, nous avons commencé par la détection de la ligne pour les deux solutions. Pour ce faire, l'opérateur "Canny" est utilisé pour extraire les bords de l'image de fond. Une transformation de Hough est utilisée pour détecter les lignes, ainsi les coordonnées des lignes peuvent être obtenues. Figure 89 et Figure 90 montrent quelques résultats de la détection des lignes continues.

Comme nous l'avons déjà expliqué, pour la caméra fixe, nous avons utilisé la méthode de la soustraction de l'arrière-plan basée sur des modèles de mélange gaussien pour détecter les véhicules dans une vidéo. La Figure 91 montre un résultat de cette méthode. Pour la caméra mobile, nous avons utilisé le Faster R-CNN pour la détection des véhicules sur les images du flux vidéo. La Figure 92 affiche un exemple de résultats de

cette technique. Ainsi, si la violation est détectée sur n'importe quelle image, et que la détection est correcte, elle est considérée comme une contravention. Figure 93 et Figure 94 sont les résultats finaux de la détection d'un dépassement interdit en appliquant la solution 1. Pour la solution 2, Figure 95 et Figure 96 présentent quelques résultats de la détection d'un dépassement interdit. Tous ces résultats montrent que notre système est robuste et fiable.

Figure 89. Détection de la ligne continue (caméra fixe)

Figure 90. Détection de la ligne continue (caméra mobile)

Figure 91. Détection de véhicules (Soustraction de l'arrière-plan basée sur le modèle de mélange gaussian)

Figure 92. Détection de véhicules (Faster RCNN)

Figure 93. Résultat final (Solution 1- Caméra fixe): Situation normale

Figure 94. Résultat final (Solution 1- Caméra mobile): Dépassement interdit

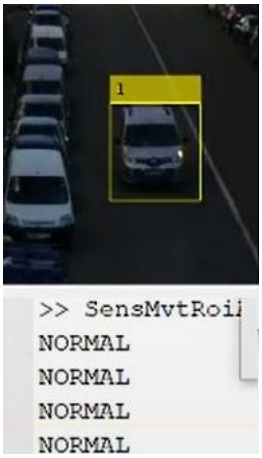


Figure 95. Résultat final (Solution 2- Caméra fixe): Situation normale

Figure 96. Résultat final (Solution 2- Caméra mobile): Dépassement interdit

Les résultats des expériences de la solution 1 et 2 sont présentés sur le Tableau 13 et Tableau 14, respectivement.

Tableau 13. Résultats obtenus pour la solution 1

Méthode de détection	Méthode de décision	Taux de détection des véhicules		Taux de classification des vues		Taux de détection des Dépassements Interdits	
		Détections Correctes	Fausses Détections	Classifications Correctes	Fausses Classifications	Détections Correctes	Fausses Détections
Soustraction arrière-plan basée sur MMG (Caméra fixe)	Classification Front/Rear	94,74%	5,26%	94,44%	5,55%	94,44%	5,55%
Faster RCNN (Caméra Mobile)	Classification Front/Rear	83,53%	16,463%	88,97%	11,029%	88,97%	11,029%

Tableau 14. Résultats obtenus pour la solution 2

Méthode de détection	Méthode de décision	Taux de détection des véhicules		Sens du mouvement		Taux de détection des Dépassements Interdits	
		Détections Correctes	Fausses Détections	Correct	Incorrect	Détections Correctes	Fausses Détections
Soustraction arrière-plan basée sur MMG (Caméra fixe)	Identification de la direction du mouvement	95,83%	4,16%	100%	0%	100%	0%
Faster RCNN	Identification de la	84,71%	15,286%	94,73%	5,26%	94,73%	5,26%

(Caméra Mobile)	direction du mouvement						

Discussion :

Le Tableau 13 et le Tableau 14 montre que les deux solutions donnent de bons résultats. Cependant, on peut observer qu'il y'a un nombre considérable de fausses détections lorsque la caméra est mobile (Dash Cam). Il est très difficile de détecter les dépassements des autres véhicules avec dash cam car les véhicules passant près de la caméra occultent fortement la vue en raison de la faible hauteur de la caméra.

Les résultats expérimentaux montrent que la solution 2 (Direction du mouvement) présente des taux de détection de dépassements interdits de véhicules plus élevés que la solution 1 (classification des vues). Un autre avantage de la solution 2 est que si la caméra ne capte qu'une partie de la voiture, le système donne toujours de bons résultats. Donc ce problème n'influence pas sur la détection du dépassement. Par contre, dans le cas de la solution 1, nous devons avoir l'image du véhicule presque complète pour avoir une bonne classification des vues.

La solution 2 n'est pas influencée par le problème des ombres, car même si l'ombre fait partie de l'objet en mouvement, elle n'influencera pas sur la direction du mouvement. Par contre, si le système peut être influencé par ce problème lorsqu'il est basé sur la distance entre le centre de la Bbox et les coordonnées de la ligne de la voie, comme dans le travail de [156], ou sur la classification des vues comme dans la solution 1.

Cependant, la solution 1 présente également certains avantages. Nous avons remarqué que la solution 1 est plus rapide que la solution 2. De plus, s'il y a plusieurs véhicules sur la ROI, le système traite facilement et rapidement tous les véhicules en parallèle. En revanche, dans la solution 2 (direction du mouvement), chaque véhicule doit être traité séparément.

Nous avons constaté que les extra-détections sont plus nombreuses que les détections manquées. Les fausses alarmes sont généralement tolérables car elles peuvent être facilement rejetées lors d'un examen visuel. En effet, la police doit toujours examiner les alarmes du système pour vérifier si la violation détectée est punissable.

L'étape de détection de véhicules est le cœur du système. Si le système a manqué un véhicule en dépassement interdit, il manque automatiquement ce dépassement. Par conséquent, si le véhicule est détecté, la détection dépend uniquement du résultat du classifieur de vue (pour la première solution). Dans la deuxième solution, si le véhicule est détecté, la détection dépend uniquement du calcul du sens de mouvement. Cette méthode est sûre, elle n'a manqué aucun dépassement interdit si la caméra est fixe. Cependant, il existe des erreurs pour la caméra mobile. La détection est plus complexe lorsque la caméra est en mouvement. Il est plus difficile de distinguer les objets qui sont en mouvement des autres qui constituent la partie statique de la scène.

En fait, la méthode de la soustraction de l'arrière-plan basée sur MMG peut détecter tous les objets passant sur la route. Pour cela, nous avons réduit la région d'intérêt en nous concentrant uniquement sur la chaussée et en filtrant les très petites Bboxes.

On peut conclure que la détection des dépassements interdits est un problème très difficile pour plusieurs raisons. Premièrement, les variations internes de la classe de véhicules sont extrêmement nombreuses en raison des changements d'angle de vue, de catégories, de couleur, de taille et de la forme. Deuxièmement, l'environnement de conduite est très complexe et dynamique. Les vidéos capturées dans des environnements urbains naturels contiennent souvent un grand nombre de parasites. En outre les changements brusques des conditions d'éclairage, les conditions météorologiques, la réflexion des véhicules sur d'autres véhicules, l'éclat métallique, les ombres fortes, la similarité des parties de la voiture avec la route en termes de couleur et de texture, la division et la fusion des blobs, la réflexion des voitures sur les routes mouillées et les phares des voitures sont parmi les défis importants rendant l'environnement plus complexe. Troisièmement, malheureusement, les infractions étant par définition sont des événements rares ; il est difficile d'obtenir un nombre suffisamment important d'échantillons pour construire la base de données. En outre, l'absence de bases de données communes rend difficile la comparaison entre les méthodes. Quatrièmement, le champ de vision de la caméra mobile est très limité par les véhicules qui passent tout près de la caméra. Si la caméra fixe est très éloignée, la scène devient plus complexe ; la ligne devient peu visible, et les véhicules deviennent petits, ce qui rend la classification de la vue plus difficile. Cinquièmement, l'utilisation d'un CPU pour nos expériences rend le processus d'apprentissage fastidieux et long (le temps d'apprentissage des modèles varie généralement de plusieurs jours à plusieurs semaines de calcul), ce qui limite considérablement les possibilités de test et de sélection

des paramètres. Cela nous force également à être limités en termes de nombre de données.

Dans le monde réel, pour rendre le système plus rapide, il est nécessaire d'utiliser une caméra à basse résolution utilisée uniquement pour détecter le scénario du dépassement interdit et pour filmer un court clip vidéo (pour aider à vérifier l'infraction détectée, et aussi pour servir de preuve s'il s'agit bien d'une contravention). Une deuxième caméra à haute résolution doit être ajoutée pour prendre une capture du véhicule en infraction afin de reconnaître sa plaque d'immatriculation et d'en extraire le numéro.

4.6. Conclusion

La détection des infractions au code de la route nécessite un grand nombre de policiers à plusieurs endroits. Dans ce rapport, nous avons proposé un système qui détecte automatiquement les dépassements interdits sur la route commis par les conducteurs pour aider la police routière à faire respecter le code de la route. Nous avons utilisé des techniques efficaces de traitement d'image et de vision par ordinateur afin de détecter et classifier les véhicules en mouvement dans des séquences vidéo acquises par des caméras statiques ou mobiles. La détection et la classification des véhicules sur la route est un problème difficile. Les routes sont des environnements dynamiques, avec des arrière-plans et des éclairages toujours changeants. Les tailles et les emplacements des véhicules dans le plan de l'image sont divers. Il existe une grande variabilité dans la forme, la taille, la couleur et l'apparence des véhicules que l'on trouve dans des scénarios de conduite.

Nous avons proposé deux solutions pour réaliser notre système. Si on a une séquence vidéo en entrée (acquise par caméra statique ou mobile), on commence par détecter la ligne continue afin de déterminer la région d'intérêt (Region Of Interest, ROI) qui est ici la voie interdite aux véhicules de l'autre voie. On détecte ensuite les véhicules dans cette ROI afin de chercher l'existence du véhicule intru. Dans cette étape, nous avons utilisé le modèle Faster R-CNN et les techniques de suivi d'objets multiples basé sur le mouvement. Dans la première solution, nous avons trouvé le véhicule intru en se basant sur la détermination de la vue avant ou arrière du véhicule. Dans la deuxième solution, le système est basé sur l'analyse du mouvement et l'identification du sens du mouvement des véhicules afin de détecter les véhicules circulant dans la mauvaise direction. Un grand nombre de tests sont effectués sur le système à partir de vidéos réelles tournées

dans différentes circonstances, notamment en cas de lumière faible, de lumière forte, et d'autres scènes complexes. Dans tous ces cas, le système est testé, et l'exactitude est relativement élevée. Il peut être donc intégré au système de circulation. Une fois le système déployé, il s'avérera bénéfique pour la société.

5.Conclusion générale

Avec l'augmentation du nombre de véhicules sur les routes, les infractions au code de la route deviennent fréquentes. L'une des infractions les plus courantes est les dépassements interdits. Aujourd'hui, l'intervention humaine est devenue insuffisante pour détecter ces infractions. La mise en œuvre d'une technologie de détection automatique des infractions est donc une nécessité évidente. C'est dans cette perspective que s'inscrit l'objectif principal de cette thèse. Il s'agit de développer un système de vision par ordinateur visant à détecter automatiquement les dépassements interdits des véhicules sur la route. Un tel système est très utile pour faire respecter le code de la route, identifier et sanctionner les contrevenants. Une fois que les conducteurs seront conscients de l'existence d'un système automatique qui garantit qu'ils ne peuvent pas s'en tirer en cas d'infraction, cela conduira à une amélioration significative de l'efficacité du réseau routier en diminuant le taux d'accidents de la route.

Notre système est basé sur l'apprentissage automatique, l'apprentissage profond et les techniques de traitement des images et des vidéos. Vu notre objectif de détecter les dépassements interdits, nous avons étudié plusieurs solutions pour détecter les véhicules en vues de face et d'arrière.

Dans un premier chapitre, nous avons étudié plusieurs solutions pour détecter les véhicules en vues de face et d'arrière en utilisant dans un premier temps les différents descripteurs d'images les plus connus tels que HOG (Histogram of Oriented Gradient), SURF (Speeded Up Robust Features), filtre de Gabor et LBP (Local Binary Patterns) associés à différents classifieurs d'apprentissage automatique comme SVM (Support Vector Machine), kNN (k-Nearest Neighbors) et les arbres de décision. Dans un second temps, nous avons testé la combinaison des descripteurs précédents pour profiter

mutuellement de leur pouvoir de description des images. Finalement, nous avons introduit des techniques de normalisation de l'illumination et d'élimination des ombres pour améliorer la détection des véhicules dans les conditions de changement d'éclairage. La performance de tous les modèles construits est évaluée avec plusieurs bases de données standards ainsi que sur notre base de données collectée. Les résultats obtenus montrent que notre idée d'introduction de ces dernières techniques améliore les taux de détection. Dans un second chapitre, nous proposons de classifier les vues des véhicules. Ce traitement s'avère utile pour notre système de détection des dépassements interdits. Pour ce faire, nous proposons une approche fondée sur l'apprentissage profond avec deux variantes : l'apprentissage par transfert et l'apprentissage par scratch. En fait, l'apprentissage par transfert est actuellement une tendance majeure dans les solutions d'apprentissage profond. Nous avons expliqué comment elle intègre des connaissances préalables dans le développement d'un nouveau modèle afin qu'il n'apprenne pas uniquement à partir des données du problème à résoudre. Dans nos expériences, nous avons employé le fameux modèle pré-entrainé AlexNet. Pour la deuxième, nous avons conçu une architecture CNN que nous l'avons utilisé pour entrainer notre modèle de classification des vues. Dans les deux cas, nous avons fait varier la partie classification des modèles en utilisant des méthodes de types Réseaux de neurones ou SVM. Les meilleurs résultats sont obtenus par un modèle AlexNet associé à une combinaison entre NN et SVM comme classifieur. Dans le troisième chapitre, la problématique de la classification des catégories des véhicules à partir des images est traitée. Nous proposons ainsi un système de classification indépendant de l'angle de vue des véhicules. A l'instar des problématiques précédentes, nous nous sommes appuyés sur les deux approches : approche descripteur-classifieur, et approche apprentissage profond. En plus, nous avons également développé un algorithme permettant de distinguer les véhicules d'urgence tels que les ambulances.

Ayant traité les problématiques de détection de véhicules, classification des vues et classification des catégories des véhicules dans les chapitres précédents, nous tentons dans le dernier chapitre de répondre à notre principale problématique qui est la détection des dépassements interdits dans des séquences d'images. Ainsi, nous proposons un système à trois phases. La première phase consiste à localiser la région d'intérêt du dépassement en s'appuyant sur la détection des lignes sur les routes. La deuxième phase se consacre à la détection des véhicules dans la région d'intérêt localisée précédemment en utilisant la meilleure technique parmi celles vues auparavant. Dans la dernière phase, nous proposons deux manières pour classer la

situation du véhicule détecté : est-ce qu'il est en état d'infraction ou non ? La première s'appuie sur une méthode de classification de la vue du véhicule détecté : vue de face ou vue d'arrière. Si le classifieur fournit une vue arrière le véhicule est en infraction sinon il est en état normal. La seconde manière utilise la direction du mouvement du véhicule pour décider l'état d'infraction ou non.

Les deux solutions ont été appliquées sur une base de données de séquences vidéo collectées pour deux situations selon que la caméra est fixe ou mobile. Pour le cas d'une caméra fixe, nous obtenons le meilleur taux de détection en voisinant les 100%. Tandis que dans le cas mobile, le meilleur taux atteint environ 94,7%. Ces deux meilleurs scores sont obtenus par la deuxième solution utilisant la direction du mouvement.

En plus de la détection automatique des dépassements interdits, le système peut être utilisé aussi dans toutes les situations où les conducteurs peuvent utiliser la mauvaise direction de la circulation. Par exemple, il peut être installé dans des rues à sens unique, dans des parkings automatiques, etc.

Nos futurs travaux seront consacrés à l'amélioration du système en améliorant les détecteurs de véhicules et de voies et en les rendant plus insensibles aux conditions de luminosité. La base de données devrait être continuellement mise à jour avec les données de nouvelles voitures, afin de maintenir le système à jour. En outre, nous pourrions proposer une autre solution basée sur la trajectoire du véhicule en détectant les anomalies qui signifient les franchissements de la ligne continue.

"Our intelligence is what makes us human, and AI is an extension of that quality".

Yann LeCun. Professor, New York University

& Director of AI Research, Facebook

5.1. ANNEXE

Quelques exemples de nos ensembles de données contenant quatre types : Bus, voiture, moto et camion.

Figure 97. Échantillons de la classe de bus (Vue de face)

Figure 98. Échantillons de la classe de voiture (Vue de face)

Figure 99. Échantillons de la classe de moto (Vue de face)

Figure 100. Échantillons de la classe des camions (Vue de face)

Figure 101. Échantillons de la classe de bus (Vue arrière)

Figure 102. Échantillons de la classe de voiture (Vue arrière)

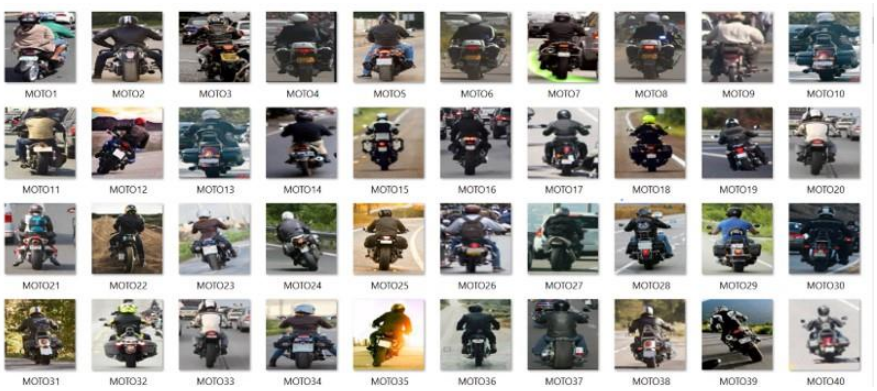


Figure 103. Échantillons de la classe de moto (Vue arrière)

Figure 104. Échantillons de la classe des camions (Vue arrière)

Figure 105. Échantillons de la classe des Ambulances

Références

[1] "Road traffic injuries," 7 February 2020. [Online]. Available: https://www.who.int/news-room/fact-sheets/detail/road-traffic-injuries.

[2] N. Mansouri, "Approche automatique à base de traitement d'images pour l'analyse comportementale de piétons âgés lors de la traversée d'une rue," 2017.

[3] "Un nouveau rapport de l'OMS épingle l'insuffisance des progrès pour améliorer la sécurité routière dans le monde," 7 décembre 2018. [Online]. Available: https://www.who.int/fr/news/item/07-12-2018-new-who-report-highlights-insufficient-progress-to-tackle-lack-of-safety-on-the-world's-roads.

[4] L. Meng, Evaluation of Intelligent Road Transport Systems Methods and Results, 2016.

[5] M. Aly, "Real time detection of lane markers in urban streets," *Intelligent Vehicles Symposium IEEE,* pp. 7-12, 2008.

[6] "Accidents de la route, Principaux repères sur les accidents de la route," [Online]. Available: https://www.who.int/fr/news-room/fact-sheets/detail/road-traffic-injuries.

[7] R. P. Loce, R. Bala and M. Trivedi, Computer Vision and Imaging in Intelligent Transportation Systems, IEEE Press, 2017.

[8] Y. Chen, W. Zhu, D. Yao and L. Zhang, "Vehicle type classification based on convolutional neural network," *Chinese Automation Congress (CAC),* pp. 1898-1901, 2017.

[9] Z. Sun, G. Bebis and R. Miller, "On-road vehicle detection: A review," *IEEE Trans. Pattern Anal. Mach. Intell.,* vol. 28, no. 5, pp. 694-711, May 2006.

[10] M. Gessica, R. Meneses and L. Nogueira, "Vehicle Shape Recognition Using SVM and ASM," 2015.

[11] V. Vinoharan, A. Ramanan and S. R. Kodituwakku, "A wheel-based side-view car detection using snake algorithm," *6th International Conference on Information and Automation for Sustainability (ICIAFS),* p. 185–189, 2012.

[12] M. Kafai and B. Bhanu, "Dynamic bayesian networks for vehicle classification in video," *IEEE Transactions on Industrial Informatics,* vol. 8, no. 1, p. 100–109, 2012.

[13] "Détection d'objet," Juillet 2019. [Online]. Available: https://fr.wikipedia.org/wiki/D%C3%A9tection_d%27objet.

[14] S. Asiri, "Machine learning classifiers," Jun 2018. [Online]. Available: https://towardsdatascience.com/machine-learning-classifiers-a5cc4e1b0623.

[15] B. Cai, F. Tan, Y. Lu and D. Zhang, "Knowledge Template Based Multiperspective Car Recognition Algorithm," *International Journal of Information Engineering and Electronic Business,* vol. 2, no. 2, pp. 38-45, 2010.

[16] H. Wang, Y. Yu, Y. Cai, X. Chen, L. Long Chen and Q. Liu, "A Comparative Study of State-of-the-Art Deep Learning Algorithms for Vehicle Detection," *IEEE Intelligent Transportation Systems Magazine,* 2019.

[17] S. Sivaraman and M. Trivedi, "Vehicle detection by independent parts for urban driver assistance," *IEEE Trans. Intell. Transp. Syst.,* 2013.

[18] D. Sung Pae, I. Hwan Choi, T. Koo Kang and M. Taeg Li, "Vehicle detection framework for challenging lighting driving environment based on feature fusion method using adaptive neuro-fuzzy inference system," *International Journal of Advanced Robotic Systems,* 2018.

[19] B. Tian, Y. Li, B. Li and D. Wen, "Rear-view vehicle detection and tracking by combining multiple parts for complex urban surveillance," *IEEE Trans. Intell. Transp. Syst.,* vol. 15, no. 2, pp. 597-606, , April 2014.

[20] Y. Wang, X. Ban, H. Wang, D. Wu, H. Wang, S. Yang, S. Liu and J. Lai, "Detection and classification of moving vehicle from video using multiple spatio-temporal features, recent advances in video coding and security," *IEEE Access 7, 80287–80299,* 2019.

[21] R. Velazquez-Pupo, A. Sierra-Romero, D. Torres-Roman, Y. V. Shkvarko, J. Santiago-Paz, D. Gómez-Gutiérrez, D. Robles-Valdez, F. Hermosillo-Reynoso and M. Romero-Delgado, "Vehicle detection with occlusion handling, tracking, and OC-SVM classification: a high performance vision-based system," *Sensors 18,* 2018.

[22] N. Seenouvong, U. Watchareeruetai and C. Nuthong, "Vehicle detection and classification system based on virtual detection zone," *International Joint Conference on Computer Science and Software Engineering (JCSSE),* 2016.

[23] S. Banu and P. Vasuki, "Video based vehicle detection using morphological operation and hog feature extraction," *ARPN J. Eng. Appl. Sci.,* vol. 10, no. 4, p. 1866–1871, 2015.

[24] T. Mita, T. Kaneko and O. Hori, "Joint Haar-like features for face detection," *Proc. 10th IEEE Int. Conf. Computer Vision,* p. 1619, 2005.

[25] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," *Proc. Int. Conf. Comp. Vis. Patt. Recog,* vol. 1, no. 12, p. 886–893, 2005.

[26] G. Zhang, X. Huang, S. Z. Li, Y. Wang and X. Wu, "Boosting local binary pattern (LBP)-based face recognition," *Proc. Chinese Conf. Advances Biometric Person Authentication,* p. 179–186, 2004.

[27] P. Viola and M. Jones, "Rapid object detection using a boosted cascade of simple features," *Proceeding of International Conference on Computer Vision and Pattern Recognition,* pp. 511-518, 2001.

[28] P. Viola and M. Jones, "Robust real-time face detection," *International Journal of Computer Vision,* vol. 57, no. 2, pp. 137-154, 2004.

[29] W. Xuezhi and Z. Yuhui, "An improved algorithm based on AdaBoost for vehicle recognition," December 2010.

[30] H. Wang and H. Ahang, "Hybrid Method of Vehicle Detection based on Computer Vision for intelligent Transportation System," *International Journal of Multimedia and Ubiquitous Engineering,* vol. 9, no. 6, pp. 105-118, 2014.

[31] A. Takeuchi, S. Mita and D. Mcallester, "On-road vehicle tracking using deformable object model and particle filter with integrated likelihoods," *Proc. IEEE Intelligent Vehicles Symp,* p. 1014–1021, 2010.

[32] I. El Jaafari, M. El Ansari, L. Koutti, A. Ellahyani and S. Charfi, "A novel approach for on-road vehicle detection and tracking," *Int. J. Adv. Comput. Sci. Appl.,* vol. 7, no. 1, pp. 594-601, 2016.

[33] Z. Sun, G. Bebis and R. Miller, "Evaluationary Gabor filter optimization with application to vehicle detection," *Proceedings of the 3rd IEEE international conference on data mining,* 2003.

[34] Z. Sun, G. Bebis and R. Miller, "Monocular Pre-crash Vehicle Detection: Features and Classifiers," *IEEE Trans. Image Process,* vol. 15, no. 7, p. 2019–2034, 2006.

[35] D. Lowe, "Distinctive image features from scale-invariant keypoints," *IJCV,* vol. 2, no. 60, p. 91– 110, 2004.

[36] K. Mikolajczyk and C. Schmid, "A performance evaluation of local descriptors," *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition,* pp. 525-531, 2003.

[37] M. A. Manzoor and Y. Morgan, "Vehicle Make and Model classification system using bag of SIFT features," *IEEE 7th Annual Computing and Communication Workshop and Conference (CCWC),* 2017.

[38] A. Psyllos, C. N. Anagnostopoulos and E. Kayafas, "M-SIFT: A new method for Vehicle Logo Recognition," *2012 IEEE International Conference on Vehicular Electronics and Safety (ICVES 2012),* 2012.

[39] H. Bay, A. Ess, T. Tuytelaars and L. V. Gool, "Speeded-Up Robust Features (SURF)," *Comput. Vis. Image Understand,* vol. 110, no. 3.

[40] J. W. Hsieh, L. C. Chen and D. Y. Chen, "Symmetrical SURF and Its Applications to Vehicle Detection and Vehicle Make and Model Recognition," *IEEE Transactions On Intelligent Transportation Systems,* vol. 15, no. 1, 2014.

[41] B. F. Momin and S. M. Kumbhare, "Vehicle detection in video surveillance system using Symmetrical SURF," *IEEE International Conference on Electrical, Computer and Communication Technologies (ICECCT),* 2015.

[42] S. Shujuan, X. Zhize, W. Xingang, H. Guan, W. Wenqi and X. De, "Real-time Vehicle Detection using Haar-SURF Mixed Features and Gentle AdaBoost Classifier," *27th Chinese Control and Decision Conference (CCDC), IEEE,* 2015.

[43] M. Vargas, S. L. Toral, J. M. Milla and F. Barrero, "A Shadow Removal Algorithm for Vehicle Detection based on Reflectance Ratio and Edge Density," *13th International IEEE, Annual Conference on Intelligent Transportation Systems,* pp. 19-22, September 2010.

[44] C. Yuan, C. Yang and Z. Xu, "Simple Vehicle Detection with Shadow Removal at Intersection," *Second International Conference on Multimedia and Information Technology,* pp. 188-191, 2010.

[45] N. Almoussa, "Variational retinex and shadow removal," 2009.

[46] R. Kimmel, M. Elad, D. Shaked, R. Keshet and I. Sobel, "A Variational Framework to Retinex," *Int'l J. Computer Vision,* vol. 52, no. 1, pp. 7-23, 2003.

[47] Q.-s. Wu, X.-l. Luo, H. Li and P.-z. Liu, "An Improved Multi-Scale Retinex Algorithm for Vehicle Shadow Elimination Based on Variational Kimmel," *7th International Conference on Ubiquitous Intelligence & Computing and 7th International Conference on Autonomic & Trusted Computing,* 2010.

[48] M. S. Alluhaidan, M. Alsafasfeh, I. Abdel-Qader and O. Abudayyeh, "Retinex-Based Framework for Visibility Enhancement During Inclement Weather with Tracking and Estimating Distance of Vehicles," *IEEE Jordan International Joint Conference on Electrical Engineering and Information Technology (JEEIT),* 2019.

[49] M. Kaur, J. Kaur and J. Kaur, "Survey of Contrast Enhancement Techniques based on Histogram Equalization," *International Journal of Advanced Computer Science and Applications,* vol. 2, no. 7, 2011.

[50] A. Rosebrock, Deep Learning for Computer Vision with Python, PYIMAGESEARCH, 2017.

[51] F. Chollet, Deep Learning with Python-Manning Publications, Manning Publications Co, 2018.

[52] S. Lin, C. Zhao and X. Qi, "Comparative Analysis of Several Feature Extraction Methods in Vehicle Brand Recognition," *10th International Conference on Sensing Technology (ICST),* 2016.

[53] "Histogramme de gradient orienté," [Online]. Available: https://fr.wikipedia.org/wiki/Histogramme_de_gradient_orient%C3%A9.

[54] G. Penghua and H. Yanping, "Vehicle Type Classification based on Improved HOG SVM," *Proceeding 3rd International Conference on Mechatronics Engineering and Information Technology ICMEIT,* 2019.

[55] H. Bay, T. Tuytelaars and L. V. Gool, "SURF: Speeded Up Robust Features," *9th European Conference on Computer Vision,* pp. 7-13 , May 2006.

[56] H. Bay, A. Ess, T. Tuytelaars and L. V. Gool, "SURF: Speeded Up Robust Features," *Computer Vision and Image Understanding,* vol. 110, no. 3, pp. 346-359, 2008.

[57] R. Scherer, Computer Vision Methods For Fast Image Classification and Retrieval, Springer, 2020.

[58] C. Evans, "Notes on the opensurf library. Technical Report CSTR-09-001," University of Bristol, 2009.

[59] A. I. Awad and M. Hassaballah, Image Feature Detectors and Descriptors, Foundations and Applications, Switzerland: Springer, 2016.

[60] "Motif binaire local," 2018. [Online]. Available: https://fr.wikipedia.org/wiki/Motif_binaire_local.

[61] "Local binary patterns," 2020. [Online]. Available: https://en.wikipedia.org/wiki/Local_binary_patterns.

[62] A. Calmettes and G. Cedille, "VISION PAR ORDINATEUR – FILTRES DE GABOR".

[63] J. R. Movellan, "Tutorial on Gabor Filters".

[64] A. Nigam and P. Gupta, "Finger-Knuckle-Print ROI Extraction Using Curvature Gabor Filter for Human Authentication," *Proceedings of the 11th Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications,* vol. 3, pp. 364-371, 2016.

[65] "Filtre de Gabor," 2020. [Online]. Available: https://fr.wikipedia.org/wiki/Filtre_de_Gabor.

[66] P. Kim, MatLab Deep Learning with Machine Learning, Neural Networks and Artificial Intelligence, Apress, 2017.

[67] M. Paluszek and S. Thomas, MATLAB Machine Learning, Apress, 2017.

[68] "Apprentissage automatique," [Online]. Available: https://fr.wikipedia.org/wiki/Apprentissage_automatique#:~:text=Selon%20les%20informations%20disponibles%20durant,agit%20d'un%20apprentissage%20supervis%C3%A9..

[69] V. Vapnik and C. Cortes, "“Support-Vector Networks," *Machine Learning,* vol. 20, no. 3, p. 273–297, 1995.

[70] "Support Vector Machine algorithm," [Online]. Available: https://www.javatpoint.com/machine-learning-support-vector-machine-algorithm.

[71] "Machine à vecteurs de support," 2020. [Online]. Available: https://fr.wikipedia.org/wiki/Machine_%C3%A0_vecteurs_de_support.

[72] M. A. Maloof, Machine Learning and Data Mining for Computer Security, Springer, 2006.

[73] "Hyperplan," 2018. [Online]. Available: https://fr.wikipedia.org/wiki/Hyperplan.

[74] "k-Nearest Neighbor (kNN) algorithm," [Online]. Available: https://www.javatpoint.com/k-nearest-neighbor-algorithm-for-machine-learning.

[75] "Similarité Cosinus," 2020. [Online]. Available: https://fr.wikipedia.org/wiki/Similarit%C3%A9_cosinus.

[76] "Généralités sur les mesures de similarité," [Online]. Available: http://pageperso.lif.univ-mrs.fr/~andreea.dragut/enseignementWebMining/r/tp4.html.

[77] "Decision Tree Classification Algorithm," [Online]. Available: https://www.javatpoint.com/machine-learning-decision-tree-classification-algorithm.

[78] J. Y. Zhu, W. S. Zheng and J. H. Lai, "Logarithm gradient histogram: A general illumination invariant descriptor for face recognition," *Proc. 10th IEEE Int. Conf. Autom. Face Gesture Recognit. (AFGR),* pp. 1-8, April 2013.

[79] Y. Zhang, J. Tian, X. He and X. Yang, "MQI Based Face Recognition Under Uneven Illumination," *Proc. ICB,* 2007.

[80] S. Thapar and S. Garg, "Study and Implementation of Various Morphology Based Image Contrast Enhancement Techniques," *International Journal of Computing Business Research,* pp. 2229- 6166, 2012.

[81] S. L. Zhang, H. H. Zhong, Y. J. Kuang and J. W. Mo, "Illumination Robust Face Recognition Based on Image Fusion of Local Contrast Enhancement and Adaptive Smoothing," *International Conference on Information Science and Technology,* 2016.

[82] "Adjusting Intensity Values to a specified Range," [Online]. Available: http://matlab.izmiran.ru/help/toolbox/images/enhanc17.html#12121.

[83] J. Arróspide, L. Salgado and M. Nieto, "Video analysis based vehicle detection and tracking using an MCMC sampling framework," *EURASIP Journal on Advances in Signal Processing,,* 2012.

[84] "The Caltech Database," 2011. [Online]. Available: http://www.vision.caltech.edu/html-files/archive.html.

[85] R. Fergus, P. Perona and A. Zisserman, *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition,* 2003.

[86] "The TU Graz-02 Database," 2011. [Online]. Available: http://www.emt.tugraz.at/~pinz/data/GRAZ_02/.

[87] A. Opelt and A. Pinz, *Proceedings of the 14th Scandinavian Conference on Image Analysis,* 2005.

[88] "The GTI-UPM Vehicle Image Database," [Online]. Available: http://www.gti.ssr.upm.es/data.

[89] "Vrai Positif," 2018. [Online]. Available: https://fr.wikipedia.org/wiki/Vrai_positif#:~:text=Dans%20un%20test%20de%20classification,%C3%A9tait%20en%20r%C3%A9alit%C3%A9%20positif)%20et.

[90] N. Dalal and B. Triggs, "Histograms of Oriented Gradients for Human Detection," *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition,* 2005.

[91] T. Bouwmans, C. Silva, C. Marghes, M. Zitouni, H. Bhaskar and C. Frelicot, "On the role and the importance of features for background modeling and foreground detection," *CoRR,* pp. 1-131, Nov 2016.

[92] L. Suhaoa, L. Jinzhao, L. Guoquan, B. Tong, W. Huiqian and P. Yu, "Vehicle type detection based on deep learning in traffic scene," *8th International Congress of Information and Communication Technology (ICICT-2018),* p. 564–572, 2018.

[93] O. Khalifa, M. Balfaqih, S. Bashir and A. Albagoul, "Automatic car identification system," *Proc. 2nd. Int. Conf. on Mathematical Applications in Engineering,* 2012.

[94] M. N. Roecker, M. C. Yandre, L. A. Joo and H. M. Gustavo, "Automatic Vehicle type Classification with Convolutional Neural Networks," *25th International Conference on Systems, Signals and Image Processing (IWSSIP) ,* pp. 1-5, 2018.

[95] S. Kul, S. Eken and A. Sayar, "A concise review on vehicle detection and classification," *International Conference on Engineering and Technology (ICET),* 2017.

[96] F. Chollet, Deep Learning with Python, Manning Publications Co, 2018.

[97] Z. Huo, Y. Xia and B. Zhang, "Vehicle type classification and attribute prediction using multi-task RCNN," *Proc. of 9th international congress on image and signal processing, biomedical engineering and informatics (CISP-BMEI),* pp. 564-569, 2016.

[98] J. Castañón, "10 Machine Learning Methods that Every Data Scientist Should Know," May 2019. [Online]. Available: https://towardsdatascience.com/10-machine-learning-methods-that-every-data-scientist-should-know-3cc96e0eeee9.

[99] I. Oztel, G. Yolcu and C. Oz, "Performance Comparison of Transfer Learning and Training from Scratch Approaches for Deep Facial Expression Recognition," *4th Int. Conf. on Computer Science and Engineering (UBMK),* pp. 1-6, 2019.

[100] A. Krizhevsky, I. Sutskever and G. E. Hinton, "ImageNet Classification with Deep Convolutional Neural Networks," *Advances In Neural Information Processing Systems,* p. 19, 2012.

[101] D. Santos and P. L. Correia, "Car recognition based on back lights and rear view features," *Proc. of the IEEE Workshop Image Analysis for Multimedia Interactive Services,* pp. 137-140, 2009.

[102] M. A. Manzoor and Y. Morgan, "Vehicle Make and Model Classification System using Bag of SIFT Features," *7th IEEE Annual Conf. on Computing and Communication Workshop and Conference (CCWC),* pp. 572-577, 2017.

[103] I. Oztel, G. Yolcu and C. Oz, "Performance Comparison of Transfer Learning and Training from Scratch Approaches for Deep Facial Expression Recognition," *4th International Conference on Computer Science and Engineering (UBMK),* Sept 2019.

[104] E. Ribeiro, M. Häfner, G. Wimmer, T. Tamaki, J. W. Tischendorf, S. Yoshida, S. Tanaka and A. Uhl, "Exploring texture Transfer Learning for Colonic Polyp Classification via Convolutional Neural Networks," *IEEE 14th International Symposium on Biomedical Imaging (ISBI 2017),* April 2017.

[105] K. R. Weiss and T. M. Khoshgoftaar, "Comparing Transfer Learning and Traditional Learning Under Domain Class Imbalance," *16th IEEE International Conference on Machine Learning and Applications (ICMLA),* pp. 337-343, 2017.

[106] S. Albawi, T. A. Mohammed and S. Al-Zawi, "Understanding of a convolutional neural network," *International Conference on Engineering and Technology (ICET),* pp. 1-6, 2017.

[107] I. Goodfellow, Y. Bengio and A. Courville, Deep Learning, MIT Press, 2016.

[108] W. S. McCulloch and W. Pitts, "A Logical Calculus of the Ideas Immanent in Nervous Activity," *Bulletin of Mathematical Biophysics,* vol. 5, pp. 115-133, 1943.

[109] F. Rosenblatt, "The Perceptron: A Probabilistic Model for Information Storage and Organization in The Brain," *Psychological Review,* p. 65–386, 1958.

[110] F. Rosenblatt, Principles of Neurodynamics: Perceptrons and the Theory of Brain Mechanisms, Spartan, 1962.

[111] M. Minsky and S. Papert. , Perceptrons, Cambridge: MIT Press, 1969.

[112] P. J. Werbos, "Beyond Regression: New Tools for Prediction and Analysis in the Behavioral Sciences. PhD thesis. Harvard University," 1974.

[113] D. E. Rumelhart,, G. E. Hinton and R. J. Williams., Neurocomputing: Foundations of Research, J. A. Anderson and E. Rosenfeld, Eds., Cambridge: MIT Press, 1988.

[114] Y. Lecun, L. Bottou, G. B. Orr and K. R. Müller, "Efficient BackProp," *Neural Networks: Tricks of the Trade, This Book is an Outgrowth of a 1996 NIPS Workshop,* p. 9–50, 1998.

[115] C. Eugenio, "Neural network architectures," [Online].

[116] Y. Lecun, L. Bottou, Y. Bengio and P. Haffner, "Yann Lecun et al. “Gradient-based learning applied to document recognition," *Proceedings of the IEEE,* p. 2278–2324, 1998.

[117] K. Phil, Matlab Deep Learning: with Machine learning, neural network and artificial intelligence, Apress, 2017.

[118] M. Salvaris, D. Dean and W. H. Tok, Deep Learning with Azure, Apress, 2018.

[119] K. Jarrett, K. Kavukcuoglu, M. A. Ranzato and Y. LeCun, "What is the best multi stage architecture for object recognition?," *IEEE 12th International Conference on Computer Vision,* 2009.

[120] S. Ioffe and C. Szegedy, "Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift," C*oRR abs/1502.03167,* 2015.

[121] S. Mathew, D. Danielle and H. T. Wee , Deep Learning with Azure, Apress, 2018.

[122] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Computation,* vol. 9, no. 8, pp. 1735-1780, 1997.

[123] I. J. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville and Y. Bengio, "Generative Adversarial Networks," *Advances in Neural Information Processing Systems ,* 2014.

[124] D. Sarkar, R. Bali, T. Ghosh and N. Panwar, Hands on Transfer learning with python, ISBN: 978-78883-130-7..

[125] "Caradisiac Website," [Online]. Available: https://www.caradisiac.com/photo/.

[126] "The GTI-UPM, Vehicle Image Database," [Online]. Available: http://www.gti.ssr.upm.es/data.

[127] Y. Nam and Y.-C. Nam, "Vehicle classification based on images from visible light and thermal cameras," *EURASIP Journal on Image and Video Processing,* 2018.

[128] S. Baghdadi and N. Aboutabit, "Front and Rear Vehicle Classification," *Advanced Intelligent Systems for Sustainable Development (AI2SD'2019),* vol. 4, no. 142, 2019.

[129] Y. Zhang, J. P. Allem, J. B. Unger and C. T. Boley, "Automated Identification of Hookahs (Waterpipes) on Instagram: An Application in Feature Extraction Using Convolutional Neural Network and Support Vector Machine Classification".

[130] A. Khvostikov, K. Aderghal, J. Benois-Pineau , A. Krylov and G. Catheline, "3D CNN-based classification using sMRI and MD-DTI images for Alzheimer's disease studies, arXiv preprint arXiv: 1801.05968," 2018.

[131] S. Rohan, S. Debaruna and J. Sudan, "Transfer learning a comparative analysis," 2018.

[132] K. Phi, Matlab Deep Learning: with Machine Learning, neural network and artificial intelligence, Apress, 2017.

[133] B. Hicham, A. Ahmed and M. Mohammed, "Vehicle Type Classification Using Convolutional Neural Network," *IEEE 5th International Congress on Information Science and Technology (CiSt),* pp. 313-316, 2018.

[134] G. Penghua and H. Yanping, "Vehicle Type Classification based on Improved HOG SVM," *Proceedings of the 3rd International Conference on Mechatronics Engineering and Information Technology ICMEIT,* 2019.

[135] L. Jie, Z. Jun, G. Tong and J. Shaobo, "Vehicle Type Classification using Hierarchical Classifiers," *Journal of Physics: Conference Series,* vol. 1069, 2018.

[136] S. Djahel, N. Smith, S. Wang and J. Murphy, "Reducing emergency services response time in smart cities: an advanced adaptive and fuzzy approach," *IEEE 1st International Smart Cities Conference ISC2 2015,* 2015.

[137] L. Sumia and V. Ranga, "Intelligent traffic management system for prioritizing emergency vehicles in a smart city.," *International Journal of Engineering,* vol. 2, no. 31, 2018.

[138] T. Swathi and B. V. Mallu, "Emergency vehicle recognition system," *Int. J. Eng. Trends Technol,* p. 987–990, 2013.

[139] K. Nellore and G. P. Hancke, "Traffic management for emergency vehicle priority based on," *visual sensing. Sensors,* vol. 11, no. 16, p. 1–28, 2016.

[140] C. O'Keeffe, J. Nicholl, J. Turner and S. Goodacre, "Role of ambulance response times in the survival of patients with out-of-hospital cardiac arrest," *Emergency Medicine Journal,* 2011.

[141] Z. Chen, T. Ellis and S. A. Velastin, "Vehicle type categorization: A comparison of classification schemes," *14th IEEE Annual Conference on Intelligent Transportation Systems,* pp. 74-79, 5-7 Oct 2011.

[142] Y. C. Wang, C. C. Han, C. T. Hsieh and K. C. Fan, "Vehicle type classification from surveillance videos on urban roads," *Ubi-Media Computing and Workshops (UMEDIA) 7th International Conference,* pp. 266-270, 2014.

[143] Z. Bailing and W. Yifan, "Vehicle type and make recognition by combined features and rotation forest ensemble," *International Journal of Pattern Recognition and Artificial Intelligence,* vol. 26, no. 3, 2012.

[144] D. Kleyko, R. Hostettler, W. Birk and E. Osipov, "Comparison of Machine Learning Techniques for Vehicle Classification Using Road Side Sensors," *IEEE 18th Int Conf. Intell. Transp. Syst,* pp. 572-577, 2015 .

[145] A. S. Lai, G. K. Fung and N. C. Yung, "Vehicle Type Classification from Visual Based Dimension Estimation.," *Intelligent Transportation Systems. Proceedings 2001 IEEE,* pp. 201-206, 25-29 Aug 2001.

[146] M. N. Roecker, M. C. Yandre, L. A. Joo and H. M. Gustavo, "Automatic Vehicle type Classification with Convolutional Neural Networks," *25th International Conference on Systems, Signals and Image Processing (IWSSIP),* pp. 1-5, 2018.

[147] Z. Dong, Y. Wu, M. Pei and Y. Jia, "Vehicle type classification using a semisupervised convolutional neural network," *IEEE Transactions on Intelligent Transportation Systems ,* vol. 16, no. 4, pp. 2247-2256, August 2015.

[148] C. W. Hsu and C. J. Lin, "A Comparison of Methods for Multi-class Support Vector Machines," *IEEE Transactions on Neural Networks,* vol. 2, no. 13, pp. 415-25, 2002.

[149] F. Melgani and L. Bruzzone, "Classification of hyperspectral remote sensing images with Support Vector Machines," *IEEE Transactions on Geoscience and Remote Sensing,* p. 1778 – 1790, 2004.

[150] F. Özyurt, , T. Tuncer, E. Avci, M. Koç and İ. Serhatlioğlu, "A Novel Liver Image Classification Method Using Perceptual Hash-Based Convolutional Neural Network," *Arabian Journal for Science and Engineering,* 2019.

[151] M. Salvaris, D. Dean and W. H. Tok, Deep Learning with Azure, Apress, 2018.

[152] X. Le, J. Jun, Y. Sakong and S. Dejan, "Detection and Classification of Vehicle Types from Moving Backgrounds. RiTA," 2017.

[153] S. Peter, C. Jaron, D. Anthony, V. Maria, R. Caroline, P. Nikos and G. Benoit, "Demystification of AI-driven medical image interpretation: past, present and future".

[154] R. Marikhu, J. Moonrinta, M. Ekpanyapong, M. Dailey and S. Siddhichai, "Police eyes: real world automated detection of traffic violations," *in Proceedings of the 2013 10th International Conference on Electrical Engineering/Electronics, Computer Telecommunications and Information Technology (ECTI-CON),* 15–17 May 2013.

[155] M.-N. Chapel, "Détection d'objets en mouvement à l'aide d'une caméra mobile. Vision par ordinateur et reconnaissance de formes.," 2017.

[156] J. Zhang, T. Gao and Z. g. Liu, "Traffic Video based Cross Road Violation Detection," *International Conference on Measuring Technology and Mechatronics Automation,* 2009.

[157] S. S. Sarikana, and M. A. Ozbayoglu, "Anomaly Detection in Vehicle Traffic with Image Processing and Machine Learning," *Complex Adaptive Systems Conference with Theme: Cyber Physical Systems and Deep Learning, CAS 2018,* 2018.

[158] L. Shapiro and G. Stockman, Computer Vision, Prentice-Hall, 2001.

[159] P. Hough, "Method and means for recognizing complex patterns," *U.S, Patent,* 18 Dec 1962.

[160] P. Hough, "Machine Analysis of Bubble Chamber Pictures," *Proc. Int. Conf. High Energy Accelerators and Instrumentation,* p. 1959.

[161] R. O. Duda and P. E. Hart, "Use of the Hough Transformation to Detect Lines and Curves in Pictures," *Comm. ACM,* vol. 15, p. 11–15, January, 1972.

[162] "R-CNN, Fast R-CNN, and Faster R-CNN," [Online]. Available: https://ch.mathworks.com/help/vision/ug/getting-started-with-r-cnn-fast-r-cnn-and-faster-r-cnn.html.

[163] G. Rohith, "R-CNN, Fast R-CNN, Faster R-CNN, YOLO - Object Detection Algorithms," 9 July 2018. [Online].

[164] R. Girshick, "Fast r-cnn," *Proceedings of the IEEE International Conference on Computer Vision,* 2015.

[165] B. Richa, "What is region of interest pooling?," 10 September 2018. [Online].

[166] R. Shaoqing, K. He, R. Girshick and J. Sun, "Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks," *Advances in Neural Information Processing Systems,* vol. 28, 2015.

[167] F. Umer , "From R-CNN to Mask R-CNN," 15 February 2018. [Online].

[168] W. Lilian, "Object Detection for Dummies Part 3: R-CNN Family," 31 December 2017. [Online].

[169] R. Girshick, "Fast R-CNN," *IEEE International Conference on Computer Vision (ICCV),* 2015.

[170] J. Long, E. Shelhamer and T. Darrell, "Fully convolutional networks for semantic segmentation," *IEEE Conference on Computer Vision and Pattern Recognition (CVPR),* 2015.

[171] "multiple object tracking," [Online]. Available: https://ch.mathworks.com/help/vision/ug/multiple-object-tracking.html.

[172] "Kalman filter," [Online]. Available: https://ch.mathworks.com/discovery/kalman-filter.html.

[173] N. Aslam and V. Sharma, "Foreground detection of moving object using Gaussian mixture model," *International Conference on Communication and Signal Processing (ICCSP),* pp. 1071-1074, 2017.

[174] "motion based multiple object tracking," [Online]. Available: https://ch.mathworks.com/help/vision/ug/motion-based-multiple-object-tracking.html.

[175] H. Wang and H. Zhang, "A hybrid method of vehicle detection based on computer vision for intelligent transportation system," *International Journal of Multimedia and Ubiquitous Engineering,* vol. 9, no. 6 , p. 105–118, 2014.

[176] G. Subbammal, M. Sharon Nisha and J. A. Jevin, "A new rear view vehicle detection and tracking for driverless vehicle assistance system," *International Journal of Advance Research In Science And Engineering,* 2015.

[177] Y. Ke and R. Sukthankar, "PCA-SIFT: a more distinctive representation for local image descriptors," *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition ,* pp. 506-513, 2004.

[178] R. Rani, R. Kumar and A. Prakash Singh, "A Comparative Study of Object Recognition Techniques," *7th International Conference on Intelligent Systems, Modelling and Simulation (ISMS),* 2016.

[179] M. Paluszek and S. Thomas, MATLAB Machine Learning, Apress, 2017.

[180] R. Kimmel, M. Elad, D. Shaked, R. Keshet and I. Sobel, "A Variational Framework to Retinex," *Int'l J. Computer Vision,,* vol. 52, no. 1, pp. 7-23, 2003.

[181] A. Psyllos and C. N. Anagnostopoulos, "Vehicle Logo Recognition Using a SIFT-Based Enhanced Matching Scheme," *IEEE Transactions On Intelligent Transportation Systems, Vol. 11, No. 2, June 2010,* pp. 322-328, 2010.

[182] S. Baghdadi and N. Aboutabit, "Front and rear vehicle classification," *Advanced Intelligent Systems for Sustainable Development (AI2SD2019),* 2019.

[183] S. Baghdadi and N. Aboutabit, "Illumination Correction in a Comparative Analysis of Feature selection for Rear-View Vehicle Detection," *International Journal of Machine Learning and Computing (IJMLC),* vol. 9, no. 6, pp. 712-720, 2019.

[184] Y. Lecun. [Online]. Available: http://yann.lecun.com.

[185] "Convolutional neural network, History section," [Online].

[186] F. Chabot, M. Chaouch, J. Rabarisoa, T. Chateau and C. Teulière, "Détection de pose de véhicule pour la reconnaissance de marque et modèle.," 2015.

[187] B. Selbes and M. Sert, "Multimodal vehicle type classification using convolutional neural network and statistical representations of MFCC," *14th IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS),* pp. 1-6, 2017.

[188] A. Krizhevsky, I. Sutskever and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," *NIPS,* 2012.

[189] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke and A. Rabinovich, "Going deeper with convolutions," *Computer Vision and Pattern Recognition (CVPR),* June 2015.

[190] S. Djahel, M. Salehie, I. Tal and P. Jamshidi, "Adaptive traffic management for secure and efficient emergency services in smart cities," *IEEE International Conference on Pervasive Computing and Communications Workshops (PERCOM Workshops),* pp. 340-343, 2013.

[191] R. Sundar, S. Hebbar and V. Golla, "Implementing intelligent traffic control system for congestion control, ambulance clearance, and stolen vehicle detection," *IEEE Sens. J.,* vol. 15, no. 2, p. 1109–1113, 2015.

[192] B. Fazenda, H. Atmoko, F. Gu, L. Guan and A. Ball, "Acoustic based safety emergency vehicle detection for intelligent transport systems," *ICCAS-SICE,* p. 4250–4255, 2009.

[193] V. Srinivasan, Y. P. Rajesh, S. Yuvaraj and M. Manigandan, "Smart traffic control with ambulance detection," *IOP Conference Series: Materials Science and Engineering,* vol. 402, no. 1, 2018.

[194] [Online]. Available: http://iitlab.bit.edu.cn/mcislab/vehicledb/.

[195] G. Penghua and H. Yanping, "Vehicle Type Classification based on Improved HOG SVM," *Proceeding 3rd International Conference on Mechatronics Engineering and Information Technology ICMEIT,* 2019.

[196] M. A. Manzoor and Y. Morgan, "Vehicle Make and Model Classification System using Bag of SIFT Features," *7th IEEE Annual Conference on Computing and Communication Workshop and Conference (CCWC),* pp. 572-577, 02 March 2017.

[197] B. Hicham, A. Ahmed and M. Mohammed, "Vehicle Type Classification Using Convolutional Neural Network," *IEEE 5th International Congress on Information Science and Technology (CiSt),* pp. 313-316, 2018.