Conception et développement d'un système de segmentation d'images pour la vision embarquée des véhicules autonomes

Stéphanie ROULLAND

17 novembre 2024

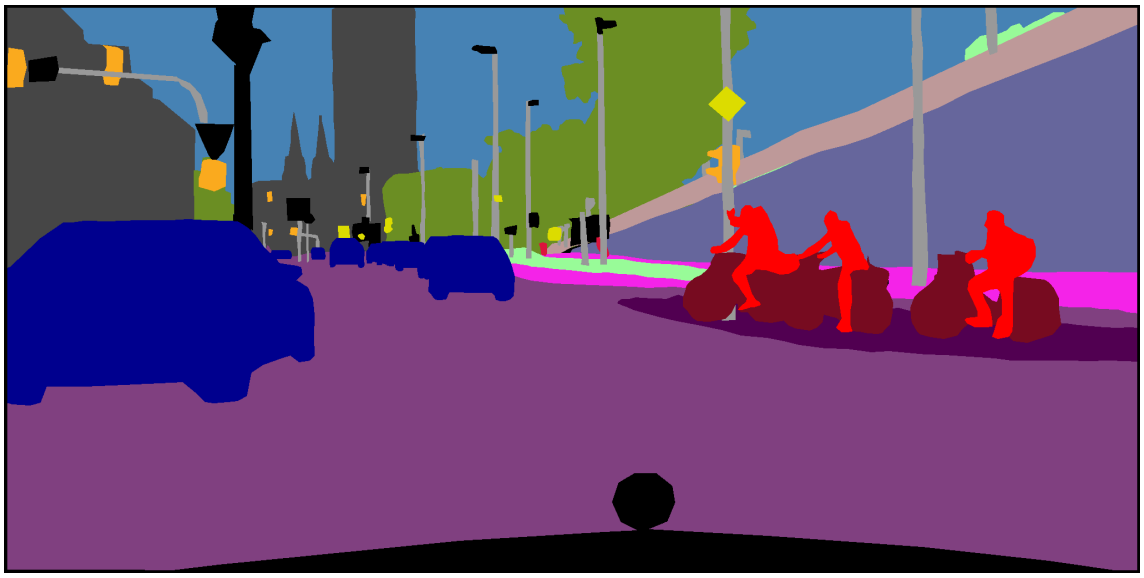


Table des matières

1 Introduction

Dans un contexte où les véhicules autonomes représentent un enjeu majeur pour l'avenir de la mobilité, la vision par ordinateur joue un rôle central dans leur capacité à percevoir et interpréter leur environnement. Future Vision Transport, entreprise spécialisée dans la conception de systèmes embarqués de vision par ordinateur pour véhicules autonomes, développe des solutions innovantes pour répondre à ces défis technologiques.

1.1 Contexte du projet

Au sein du système embarqué de vision par ordinateur de Future Vision Transport, la chaîne de traitement se compose de quatre maillons essentiels : l'acquisition des images en temps réel, leur traitement, leur segmentation et enfin le système de décision. La segmentation d'images constitue une étape critique de cette chaîne, permettant d'identifier et de classifier les différents éléments de la scène urbaine (routes, véhicules, piétons, etc.).

1.2 Rôle de la segmentation dans le système

Positionnée entre le bloc de traitement des images et le système de décision, la segmentation sémantique permet une compréhension fine de l'environnement du véhicule. Elle reçoit en entrée les images prétraitées et fournit au système de décision une carte détaillée des différents éléments de la scène, essentielle pour la prise de décision en temps réel du véhicule autonome.

1.3 Objectifs du projet

Ce projet vise trois objectifs principaux :

1. L'entraînement d'un modèle de segmentation performant sur les 8 catégories principales du dataset Cityscapes, en utilisant le framework Keras, garantissant ainsi la compatibilité avec l'infrastructure existante.
2. La conception et le déploiement d'une API de prédiction permettant l'intégration fluide du modèle dans la chaîne de traitement.
3. Le développement d'une application web de visualisation pour tester et valider les performances du système.

1.4 Structure de la note technique

Cette note technique s'articule autour de plusieurs sections clés : un état de l'art des techniques de segmentation d'images, une présentation détaillée du modèle retenu et de son architecture, une analyse des résultats obtenus avec différentes approches d'augmentation des données, et enfin une discussion des perspectives d'amélioration du système.

2 État de l'art en segmentation d'images

2.1 Introduction aux techniques de segmentation

La segmentation d'images constitue une étape fondamentale dans la compréhension de scènes pour les véhicules autonomes. Elle permet d'identifier et de délimiter précisément les différents éléments présents dans l'environnement urbain. Dans ce domaine, on distingue trois types principaux de segmentation :

- **La segmentation sémantique** : assigne une classe à chaque pixel de l'image (route, voiture, piéton, etc.).
- **La segmentation d'instance** : différencie les objets individuels au sein d'une même classe (distingue différentes voitures entre elles).
- **La segmentation panoptique** : combine les deux approches précédentes.

Dans le contexte des véhicules autonome, la segmentation sémantique revêt une importance particulière car elle permet une compréhension globale de la scène, essentielle pour la navigation et la prise de décision en temps réel.

2.2 Évolution des approches de segmentation

2.2.1 Méthodes traditionnelles

L'histoire de la segmentation d'images trouve ses racines dans des techniques classiques de traitement d'images. Ces approches fondamentales comprennent :

- **Le seuillage** : séparation des objets basée sur l'intensité des pixels.
- **La croissance de régions** : regroupement de pixels selon des critères de similarité.
- **Les contours actifs** : évolution des courbes pour délimiter les objets.
- **Les watershed** : segmentation basée sur la topographie de l'intensité de l'image.

Bien que ces méthodes aient posé les bases théoriques essentielles de la segmentation, leurs limitations sont devenues évidentes face à la complexité des scènes réelles, particulièrement en environnement urbain. Le manque de robustesse face aux variations d'éclairage, aux occlusions et à la diversité des objets a motivé la recherche de solutions plus avancées.

2.2.2 Avènement des réseaux de neurones convolutifs

L'introduction des réseaux de neurones convolutifs (CNN) a marqué un tournant décisif dans le domaine de la vision par ordinateur. Les Fully Convolutional Networks (FCN) ont particulièrement révolutionné l'approche de la segmentation sémantique en introduisant la première architecture entièrement convolutive.

Cette innovation majeure a permis de traiter les images de bout en bout, en combinant l'apprentissage automatique des caractéristiques pertinentes avec une segmentation précise au niveau du pixel. L'architecture FCN a ouvert la voie à de nombreuses évolutions, établissant un nouveau paradigme dans le traitement des images.

2.3 Architecture VGG16 et son adaptation pour la segmentation

2.3.1 Architecture originale de VGG16

Développé par le Visual Geometry Group de l'Université d'Oxford, VGG16 s'est imposé comme une architecture de référence en vision par ordinateur grâce à sa simplicité et son efficacité remarquables.

Le réseau s'articule autour de 13 couches convolutives suivies de 3 couches fully connected, utilisant exclusivement des filtres de convolution 3x3. Cette organisation permet une extraction progressive des caractéristiques, avec une profondeur croissante des features maps. À mesure que l'on avance dans le réseau, les champs réceptifs s'élargissent graduellement, permettant de capturer des motifs de plus en plus complexes et abstraits. La simplicité architecturale de VGG16, combinée à sa capacité à extraire des caractéristiques hiérarchiques pertinentes, en fait un choix particulièrement intéressant pour la segmentation d'images.

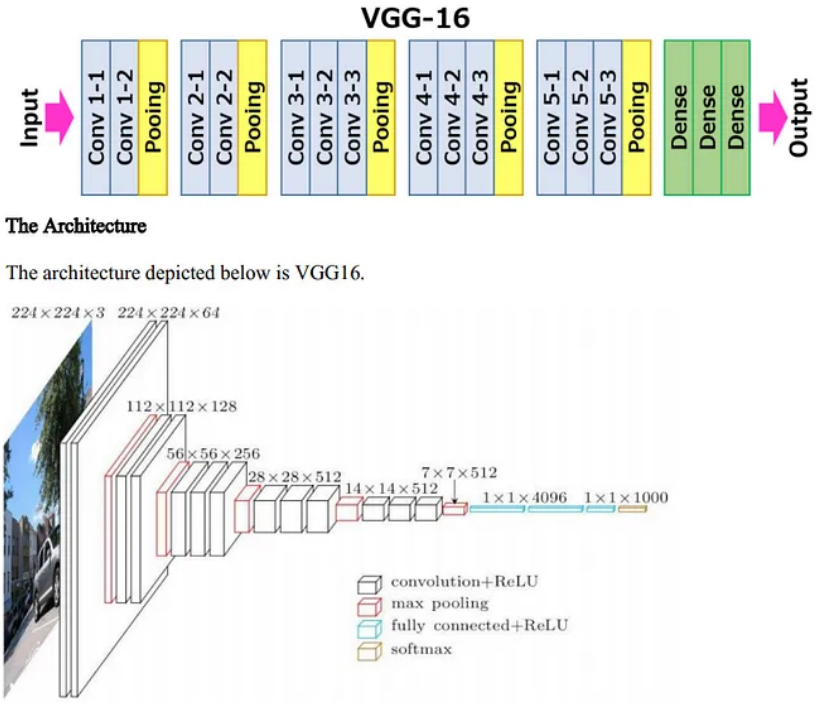


FIGURE 1 – Architecture de VGG16

2.3.2 Adaptation pour la segmentation sémantique

La transformation de VGG16 pour la tâche de segmentation sémantique nécessite plusieurs modifications architecturales stratégiques tout en préservant ses points forts fondamentaux. L'adaptation principale consiste à convertir les couches fully connected en couches convolutives 1x1, une modification qui permet de préserver l'information spatiale cruciale tout au long du réseau. Cette conversion s'accompagne de l'ajout de couches de déconvolution, essentielles pour retrouver la résolution d'origine de l'image.

L'introduction de connexions skip permet également de conserver les détails spatiaux fins qui risqueraient d'être perdus dans les couches profondes du réseau. Ces modifications permettent de bénéficier des poids préentraînés sur ImageNet tout en adaptant le réseau à la tâche spécifique de segmentation.

2.4 Justification du choix de VGG16

2.4.1 Avantages techniques

Sur le plan technique, VGG16 présente des atouts majeurs qui justifient son choix pour notre projet. Sa robustesse, démontrée sur de nombreuses tâches de vision par ordinateur, s'accompagne d'excellentes performances de généralisation et d'une grande stabilité lors de l'entraînement. L'architecture offre un équilibre optimal entre profondeur du réseau et efficacité computationnelle, permettant des temps d'inférence raisonnables tout en maintenant une précision élevée. La gestion efficace de la mémoire et la facilité d'implémentation sous Keras en font une solution particulièrement adaptée aux contraintes de développement.

2.4.2 Adéquation avec le projet

Dans le contexte spécifique de notre projet de segmentation pour véhicules autonomes, VGG16 répond parfaitement aux exigences établies. Son intégration transparente dans l'écosystème Keras existant facilite le développement et la maintenance du système. L'architecture s'adapte naturellement aux huit catégories principales de Cityscapes, permettant une segmentation efficace des éléments critiques des scènes urbaines. La simplicité de son architecture facilite les futures optimisations et évolutions du système, tout en maintenant un niveau de performance élevé.

2.4.3 Considérations pratiques

D'un point de vue pratique, le choix de VGG16 s'appuie sur un écosystème mature et bien documenté. La disponibilité d'une documentation extensive et le support d'une communauté active facilitent grandement le développement et la résolution des problèmes potentiels. Le processus de déploiement est simplifié par l'existence de nombreuses optimisations documentées et la possibilité d'exporter facilement le modèle vers différentes plateformes. Ces aspects pratiques s'avèrent particulièrement précieux pour l'intégration dans une API, répondant ainsi parfaitement aux besoins spécifiques de notre projet tout en facilitant les futures évolutions du système.

3 Présentation du modèle et des catégories de segmentation

3.1 Choix du modèle

Le choix d'un modèle de segmentation adapté aux besoins spécifiques des véhicules autonomes nécessite une analyse approfondie des différentes architectures disponibles. Dans notre contexte, l'utilisation de VGG16 préentraîné sur ImageNet s'est imposée comme une solution particulièrement pertinente pour le jeu de données Cityscapes. Ce choix repose notamment sur la capacité du modèle à extraire efficacement les caractéristiques visuelles essentielles des scènes urbaines, tout en maintenant un équilibre optimal entre performance et complexité computationnelle.

L'architecture VGG16, bien qu'initialement conçue pour la classification d'images, présente des caractéristiques qui la rendent particulièrement adaptée à la segmentation d'images urbaines. Sa profondeur modérée et son architecture régulière facilitent son adaptation à notre tâche spécifique, tout en bénéficiant de la puissance des poids préentraînés sur le vaste jeu de données ImageNet.

3.2 Architecture du modèle

3.2.1 Structure générale du réseau

L'architecture de notre modèle de segmentation repose sur une adaptation méticuleuse de VGG16. La structure du réseau peut être décomposée en deux parties principales : l'encodeur, basé sur l'architecture VGG16 originale, et le décodeur, spécifiquement conçu pour la tâche de segmentation. L'encodeur est constitué de blocs de convolution successifs, chacun augmentant progressivement la profondeur des feature maps tout en réduisant leur dimension spatiale. Cette organisation permet une extraction hiérarchique des caractéristiques, des motifs simples aux concepts plus abstraits.

3.2.2 Encodeur VGG16

L'encodeur VGG16 se compose de cinq blocs de convolution principaux. Chaque bloc comprend des couches de convolution avec des filtres 3x3, suivies d'une activation ReLU et d'une opération de max pooling. Cette structure progressive permet d'extraire des caractéristiques de plus en plus complexes :

- **Bloc 1** : Capture des caractéristiques de bas niveau (contours, textures) ;
- **Bloc 2** : Détection de motifs simples ;
- **Bloc 3** : Identification de formes plus complexes ;
- **Bloc 4** : Reconnaissance de parties d'objets ;
- **Bloc 5** : Compréhension de structures complexes.

3.2.3 Adaptation pour la segmentation

La transformation de VGG16 en un modèle de segmentation nécessite plusieurs modifications architecturales essentielles ; Les couches fully connected originales sont remplacées par des couches convolutives 1x1, permettant de préserver l'information spatiale tout en réduisant significativement le nombre de paramètres. Le décodeur est ensuite construit en utilisant des couches de déconvolution successives, permettant de restaurer progressivement la résolution spatiale de l'image d'entrée.

L'ajout de connexions skip entre l'encodeur et le décodeur constitue une amélioration cruciale. Ces connexions permettent de combiner les informations de localisation précise des premières couches avec les caractéristiques sémantiques plus riches des couches profondes, aboutissant à une segmentation plus précise des contours des objets.

3.3 Catégories de segmentation et mapping des classes

3.3.1 Organisation des classes Cityscapes

Le jeu de données Cityscapes propose initialement une annotation fine avec 32 classes différentes. Cependant, pour notre application dans le contexte des véhicules autonomes, nous avons effectué un regroupement en 8 catégories principales :

- **Routes et trottoirs** : surfaces de circulation pour véhicules et piétons ;
- **Bâtiments** : structures construites et infrastructures urbaines ;
- **Végétation** : arbres, buissons et espaces verts ;
- **Ciel** : fond de scène
- **Véhicules** : voitures, camions, bus, motos, vélos ;

— **Piétons** : personnes en mouvement ou statiques ;
— **Panneaux et feux** : signalisation routière ;
— **Autres** : autres objets urbains.

3.3.2 Justification du regroupement

Ce regroupement en 8 catégories principales répond à plusieurs objectifs stratégiques. Premièrement, il permet de réduire la complexité du problème tout en maintenant une segmentation pertinente pour la prise de décision du véhicule autonome. Les classes choisies représentent les éléments les plus critiques pour la navigation urbaine sûre, permettant une distinction claire entre les zones circulables, les obstacles fixes et mobiles, et les éléments de signalisation.

La réduction du nombre de classes présente également des avantages pratiques significatifs. Elle permet d'améliorer la stabilité de l'entraînement et de réduire le temps nécessaire à la convergence du modèle. De plus, ce regroupement facilite l'intégration avec le système de décision en fournissant une représentation plus concise mais suffisamment riche de l'environnement.

3.3.3 Impact sur l'apprentissage

Le regroupement des classes influence positivement le processus d'apprentissage du modèle. La réduction du nombre de catégories permet d'augmenter le nombre d'exemples par classe, conduisant à un apprentissage plus robuste. Cette approche atténue également le problème du déséquilibre des classes, fréquent dans les données de scènes urbaines, où certaines catégories sont naturellement sous-représentées.

4 Préparation des données et techniques d'augmentation

4.1 Dataset Cityscapes

Le dataset Cityscapes constitue une ressource précieuse pour notre projet de segmentation d'images pour véhicules autonomes. Ce jeu de données se distingue par sa richesse et sa pertinence, capturant la complexité des environnements urbains à travers des images haute résolution (2048x1024 pixels) prises dans 50 villes différentes. Les images présentent une grande diversité de scènes urbaines, incluant différentes conditions météorologiques, moments de la journée et configurations de trafic. Chaque image est accompagnée d'annotations pixelliques précises, essentielles pour l'entraînement supervisé de notre modèle de segmentation.

4.2 Prétraitement des données

Le prétraitement des données constitue une étape cruciale pour optimiser l'entraînement de notre modèle VGG16. La première étape consiste à redimensionner les images à une résolution plus adaptée au traitement en temps réel (224x224 pixels), compatible avec l'architecture VGG16. Cette réduction de taille permet d'accélérer l'entraînement et l'inférence tout en conservant suffisamment de détails pour une segmentation précise.

La normalisation des images joue également un rôle fondamental dans notre pipeline de prétraitement. Les valeurs des pixels sont normalisées dans l'intervalle [0,1] et standardisées selon les statistiques du jeu de données ImageNet sur lequel VGG16 a été préentraîné Cette standardisation assure une meilleure compatibilité avec les poids préentraînés et favorise une convergence plus stable lors de l'entraînement.

4.3 Techniques d'augmentation des données

L'augmentation des données s'avère essentielle pour améliorer la robustesse et la généralisation de notre modèle. Dans ce projet, nous avons expérimenté deux approches différentes d'augmentation des données : une utilisant la bibliothèque Albumentations et une autre basée sur TensorFlow. Pour chacune de ces approches, nous avons développé et testé une version basique puis une version étendue avec des transformations plus sophistiquées.

4.3.1 Première approche : Albumentations

La première approche testée utilisé la bibliothèque Albumentations, spécialisée dans l'augmentation d'images. Dans sa version basique, nous avons implémenté les transformations géométriques fondamentales :

- **Retournement horizontal** avec une probabilité de 50% ;
- **Retournement vertical** avec une probabilité de 20% ;
- **Rotation aléatoire** dans une plage de ±10 degrés ;
- **Mise à l'échelle aléatoire** avec une variation de ±20%.

La version étendue de cette approche enrichit ces transformations avec des modifications plus complexes :

- **Ajustements de luminosité et de contraste** ;
- **Application de flou gaussien** avec des noyaux de taille variable (3 à 7 pixels) ;
- **Modifications de teinte et de saturation** ;
- **Transformations élastiques** pour simuler des déformations non linéaires ;
- **Distorsions en grille** pour une variabilité géométrique accrue.

4.3.2 Seconde approche : TensorFlow

La seconde approche testée utilise les fonctionnalités natives de TensorFlow. La version basique comprend :

- **Retournement horizontal** avec une probabilité de 50% ;
- **Retournement vertical** avec une probabilité de 20% ;
- **Rotation par multiples de 90 degrés** ;
- **Mise à l'échelle dynamique** entre 80% et 120% de la taille originale.

La version étendue de cette approche ajoute des transformations photométriques :

- **Ajustement de luminosité** avec une variation maximale de 20% ;
- **Modification du contraste** dans une plage de 80% à 120% ;
- **Ajustement de la saturation** des couleurs (variation de ±20%) ;

— **Modification de la teinte** avec une variation maximale de 10%.

4.3.3 Gestion des masques de segmentation

Pour les deux approches, un aspect crucial est la préservation de la cohérence entre les images transformées et leurs masques de segmentation correspondants. Chaque transformation appliquée à une image est simultanément appliquée à son masque de segmentation, maintenant ainsi l'alignement spatial essentiel pour l'entraînement du modèle.

Ces deux approches alternatives d'augmentation de données ont été testées séparément, chacune avec sa version basique et étendue, permettant d'évaluer leur impact respectif sur les performances du modèle. Cette expérimentation nous permet d'identifier la stratégie la plus efficace pour notre cas d'usage spécifique de segmentation de scènes urbaines.

5 Entraînement du modèle et évaluation initiale

5.1 Configuration d'entraînement

L'entraînement d'un modèle de segmentation efficace nécessite une configuration minutieuse des paramètres d'apprentissage. Dans notre projet, nous avons utilisé Keras comme framework d'entraînement.

Le processus d'entraînement s'étend sur 20 époques, un nombre déterminé pour permettre une convergence satisfaisante du modèle tout en évitant le surapprentissage. La taille de batch est fixée à 8 images, un choix qui permet d'optimiser l'utilisation de la mémoire GPU disponible tout en maintenant une stabilité satisfaisante de l'apprentissage.

L'optimiseur Adam a été sélectionné pour ses bonnes performances générales et sa facilité d'utilisation, ne nécessitant pas de réglage fin de multiples hyperparamètres. La fonction de perte utilisée est la *categorical crossentropy*, particulièrement adaptée aux problèmes de segmentations multiclasse.

5.2 Métriques d'évaluation

L'évaluation précise des performances du modèle de segmentation repose sur plusieurs métriques complémentaires, chacune apportant un éclairage différent sur la qualité de la segmentation :

- **Coefficient de Dice (Dice Coefficient)** : Cette métrique, également connue sous le nom de F1-score, mesure le degré de chevauchement entre les prédictions et les annotations réelles. Elle est particulièrement pertinente pour évaluer la qualité de la segmentation car elle prend en compte à la fois la précision et le rappel.
- **Dice Loss** : dérivée du coefficient de Dice, cette fonction de perte permet d'optimiser directement la similarité entre les masques prédits et réels. Elle est particulièrement adaptée aux problèmes de segmentation où les classes peuvent être déséquilibrées.
- **Total Loss** : Cette métrique combine différentes fonctions de perte pour une évaluation globale de la performance du modèle. Elle offre une vue d'ensemble de la qualité de l'apprentissage.
- **Jaccard Score** : Également connu sous le nom d'Intersection over Union (IoU), il mesure le rapport entre l'intersection et l'union des zones prédites et réelles. Cette métrique est particulièrement stricte car elle pénalise toute divergence entre la prédiction et la réalité.
- **IoU (Intersection over Union)** : Similaire au Jaccard Score, cette métrique est calculée pour chaque classe individuellement, permettant une évaluation fine de la performance du modèle sur chaque catégorie d'objets.

5.3 Résultats initiaux

5.3.1 Évolution des performances

L'analyse de l'évolution des métriques au cours des 20 époques d'entraînement révèle une amélioration progressive et significative des performances du modèle. Partant d'une situation initiale modeste, le modèle montre une progression constante jusqu'à atteindre des performances satisfaisantes.

La **précision** (accuracy) a connu une amélioration remarquable, passant de 28.93% lors de la première époque à 79.69% à la fin de l'entraînement. Cette progression de plus de 50 points de poucentage démontre la capacité du modèle à apprendre efficacement les patterns de segmentation.

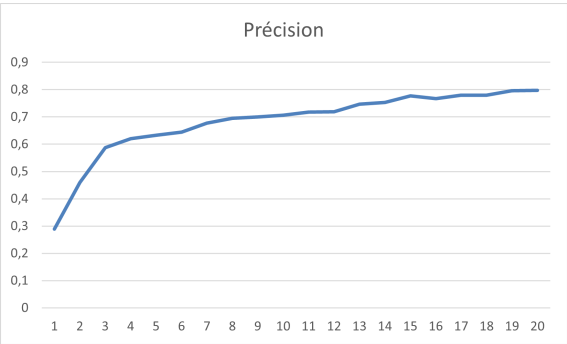


Figure 2 – Précision

Le **coefficient de Dice**, une métrique clé pour la segmentation, suit une trajectoire similaire, progressant de 18.53% à 71.57%. Cette évolution est particulièrement encourageante car elle indique une amélioration continue de la qualité de la segmentation, avec une meilleure correspondance entre les prédictions et les masques réels.

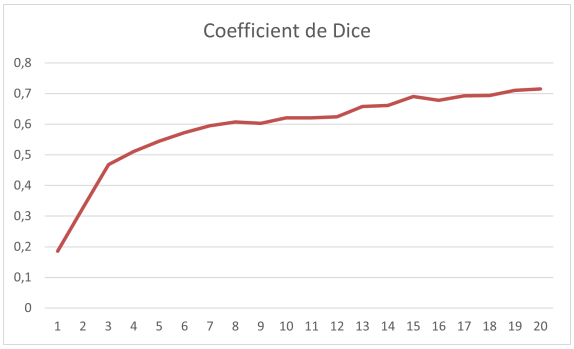


FIGURE 3 – Coefficient de Dice

Le **score de Jaccard (IoU)** montre également une progression significative, passant de 10.49% à 55.82%. Cette métrique, plus stricte que le coefficient de Dice, confirme l'amélioration réelle des capacités de segmentation du modèle.

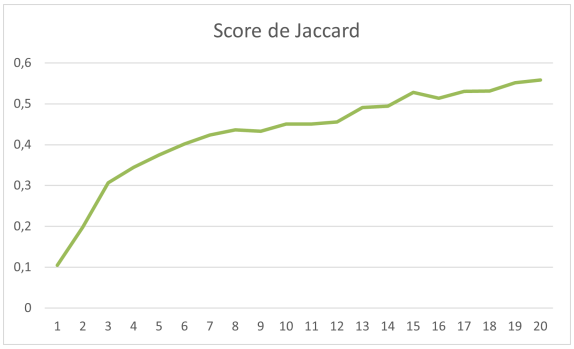
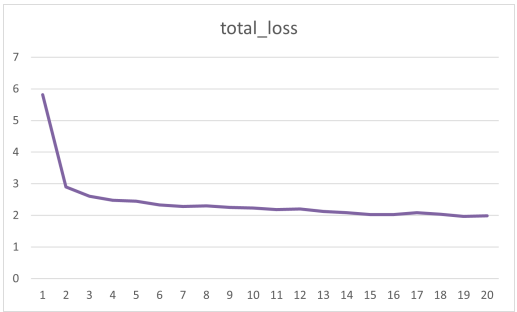


FIGURE 4 – Score de Jaccard

5.3.2 Analyse des pertes

L'évolution des différentes fonctions de perte témoigne de l'apprentissage progressif du modèle.

La **perte totale (total loss)** a diminué de manière significative, passant de 5.82 à 1.99, indiquant une convergence progressive du modèle. La **dice loss** suit une tendance similaire s'améliorant de 0.81 à 0.28, ce qui confirme l'amélioration de la qualité de segmentation.



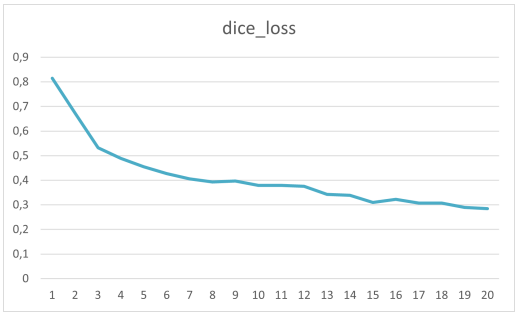


FIGURE 5 – Perte totale et Dice loss

La **perte principale (loss)** montre une réduction drastique, passant de 13.92 à 0.61, particulièrement marquée durant les premières époques. Cette diminution rapide initiale suivie d'une stabilisation progressive suggère un apprentissage efficace des caractéristiques principales des images.

FIGURE 6 – Perte principale

5.3.3 Analyse des résultats

L'analyse des résultats révèle plusieurs aspects intéressants du processus d'apprentissage. Les améliorations les plus significatives sont observées durant les premières époques, avec un gain particulièrement marqué entre les époques 1 et 5. Cette phase correspond à l'apprentissage des caractéristiques fondamentales de segmentation.

À partir de l'époque 15, on observe une certaine stabilisation des métriques, avec des améliorations plus modestes mais continues. Les dernières époques montrent des fluctuations plus faibles, suggérant que le modèle approche d'un point de convergence. La progression finale reste positive, indiquant que le modèle pourrait potentiellement bénéficier d'époques supplémentaires d'entraînement, bien que les gains attendus seraient probablement marginaux.

Le temps d'entraînement moyen par époque se situe autour de 670 secondes (soit environ 17-18 secondes par step), ce qui représente un compromis acceptable entre la qualité des résultats et les contraintes de temps de développement.

Ces résultats initiaux, obtenus sans augmentation de données, constituent une base solide pour notre système de segmentation. Ils suggèrent également plusieurs pistes d'amélioration, notamment l'introduction de techniques d'augmentation de données pour potentiellement améliorer encore les performances du modèle.

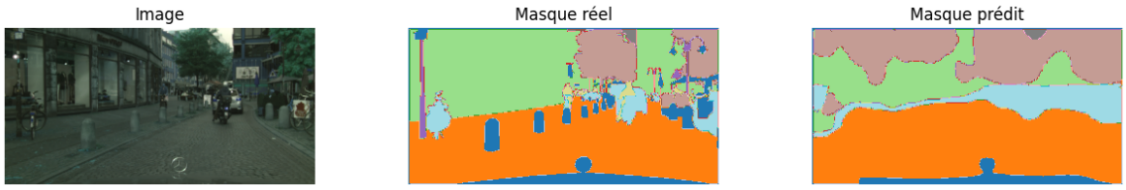


FIGURE 7 – Résultats sans data augmentation

6 Impact des techniques d'augmentation de données sur les résultats

6.1 Comparaison des approches d'augmentation

Notre expérimentation a porté sur deux approches d'augmentation de données : Albumentations et TensorFlow. L'approche utilisant Albumentations s'est révélée nettement supérieure, tant en termes de métriques quantitatives que de qualité visuelle des masques générés.

6.1.1 Résultats avec Albumentations

L'augmentation basique avec Albumentations a permis d'atteindre des performances significativement améliorées, avec une précision finale de 88.94% et un coefficient de Dice de 83.92%. Cette amélioration substantielle par rapport au modèle sans augmentation démontre l'efficacité des transformations géométriques simples.

La version étendue d'Albumentations a poussé ces résultats encore plus loin, atteignant une précision remarquable de 94.24% et un coefficient de Dice de 88.77%. Le score de Jaccard (IoU) a également progressé jusqu'à 79.82%, démontrant une segmentation plus précise des différentes classes.

L'évolution des métriques révèle une progression particulièrement intéressante :

— La perte (loss) a diminué de manière significative, passant de 0.75 à 0.22 ;
— Le dice loss s'est réduit à 0.11, indiquant une meilleure correspondance entre prédictions et vérité terrain ;
— Les gains sont particulièrement visibles dans les dernières époques, suggérant une meilleure capacité de généralisation.

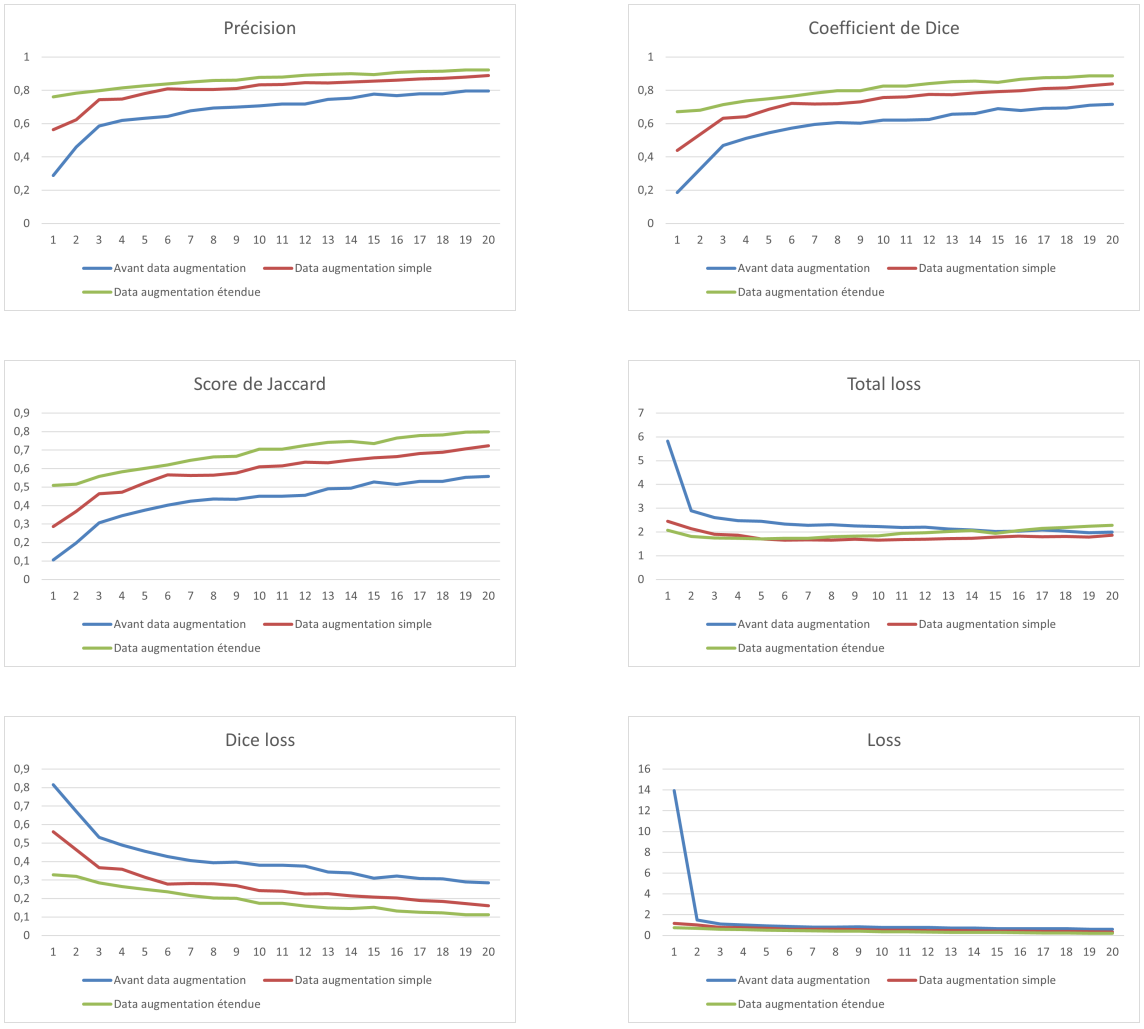


FIGURE 8 – Visualisation des différentes métriques avec l'approche Albumentations

6.1.2 Limitations de l'approche TensorFlow

L'implémentation basée sur TensorFlow, bien que théoriquement similaire, n'a pas produit les résultats escomptés. Les métriques sont restées significativement inférieures, avec une précision maximale de 58.53% et un coefficient de Dice ne dépassant pas 44.80%. La qualité visuelle masques générés s'est également révélée insuffisante pour une utilisation pratique.

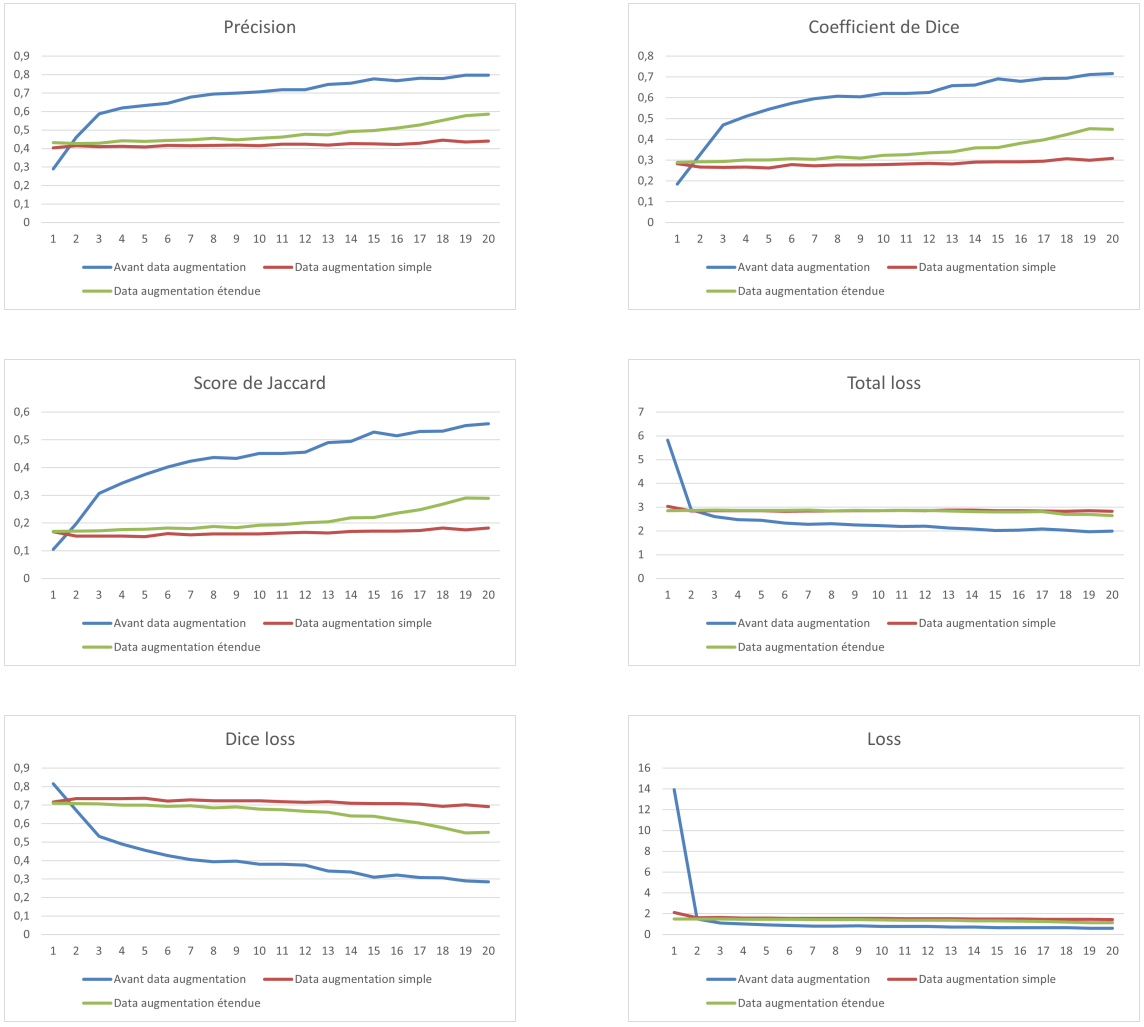


FIGURE 9 – Visualisation des différentes métriques avec l'approche TensorFlow

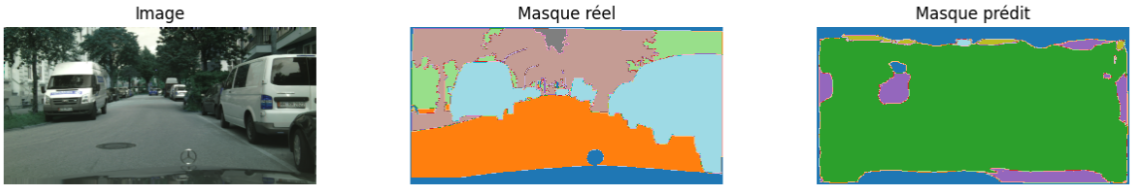


FIGURE 10 – Résultats avec l'approche TensorFlow simple

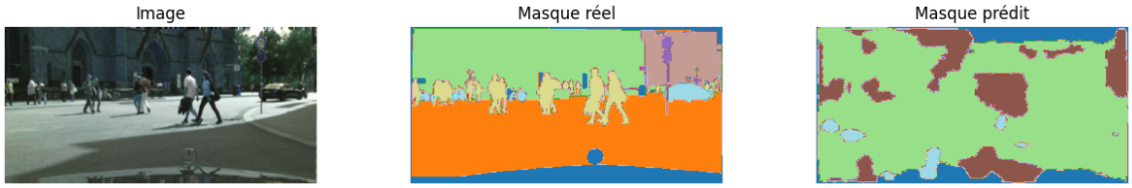


FIGURE 11 – Résultats avec l'approche TensorFlow étendue

6.2 Analyse des améliorations

Les améliorations les plus notables avec Albumentations concernent la robustesse du modèle face aux variations des scènes urbaines. Les transformations photométriques de la version étendue ont particulièrement contribué à une meilleure gestion des conditions d'éclairage variables et des différentes textures urbaines.

L'augmentation des données a également permis de réduire significativement le surapprentissage, comme en témoigne la progression régulière des métriques tout au long de l'entraînement. La version étendue d'Albumentations, avec ses transformations plus sophistiquées, a notamment permis d'améliorer la détection des détails fins et la gestion des cas complexes comme les occlusions partielles.

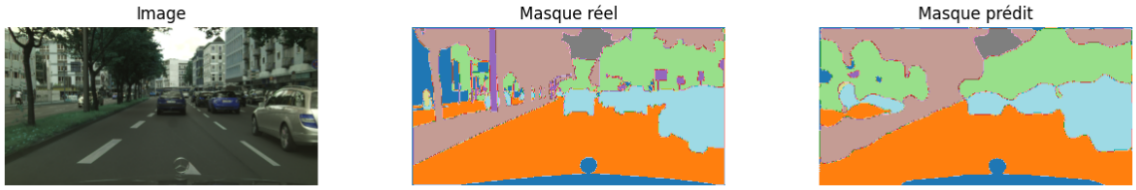


FIGURE 12 – Résultats avec l'approche Albumentation simple

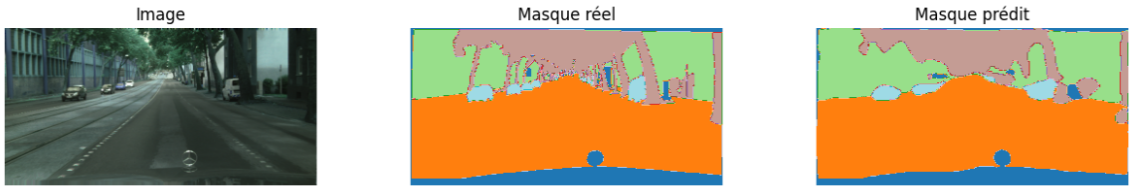


FIGURE 13 – Résultats avec l'approche Albumentation étendue

6.3 Implications pratiques

Ces résultats démontrent clairement la supériorité de l'approche Albumentations pour notre cas d'usage spécifique. Les gains de performance justifient pleinement l'intégration de ces techniques d'augmentation dans notre pipeline de traitement. Pour une implémentation en production, la version étendue d'Albumentations apparaît comme le choix optimal, offrant le meilleur compromis entre qualité de segmentation et robustesse.

7 Résultats et analyse des performances

7.1 Synthèse des résultats

Notre étude a permis d'évaluer différentes approches de segmentation d'images sur le dataset Cityscapes. Les résultats démontrent une progression significative des performances grâce à l'utilisation de techniques d'augmentation de données appropriées.

7.1.1 Modèle initial sans augmentation

Le modèle de base, entraîné sans augmentation de données, a montré des performances encourageantes mais limitées :

— Précision finale : 79.69% ;
— Coefficient de Dice : 71.57% ;
— Score de Jaccard (IoU) : 55.82% ;
— Perte finale : 0.61.

Ces résultats, bien qu'acceptables, laissaient une marge d'amélioration significative, particulièrement dans la gestion des cas complexes et des variations de scènes urbaines.

7.1.2 Impact des techniques d'augmentation

L'introduction des techniques d'augmentation a conduit à des améliorations notables, avec des résultats variant selon l'approche utilise :

Augmentation avec albumentations (version basique)
— Précision : 88.94% (+9.25 points) ;
— Coefficient de Dice : 83.92% (+12.35 points) ;
— Score de Jaccard : 72.31% (+16.49 points) ;
— Perte : 0.32 (amélioration de 47%).

Augmentation avec albumentations (version étendue)
— Précision : 92.24% (+12.55 points) ;
— Coefficient de Dice : 88.77% (+17.20 points) ;
— Score de Jaccard : 79.82% (+24.00 points) ;
— Perte : 0.22 (amélioration de 64 %).

7.2 Analyse comparative des performances

L'approche utilisant Albumentations s'est révélée particulièrement efficace, avec des gains substantiels sur toutes les métriques. La version étendue, intégrant des transformations photométriques sophistiquées, a permis d'obtenir les meilleurs résultats, démontrant l'importance d'une stratégie d'augmentation diversifiée.

7.2.1 Evolution des métriques

L'analyse de l'évolution des métriques au cours de l'entraînement révèle plusieurs points intéressants :

1. **Stabilité de l'apprentissage** : La convergence s'est montrée plus stable et plus rapide avec l'augmentation de données, particulièrement avec la version étendue d'Albumentations. La diminution régulière de la perte principale, passant de 0.75 à 0.22, et l'amélioration continue de la précision, passant de 76.14% à 92.24%, témoignent d'un apprentissage maîtrisé.
2. **Progression du coefficient de Dice** : L'amélioration continue du coefficient de Dice, particulièrement marquée dans les dernières époques, infique une meilleure capacité du modèle à délimiter précisément les différentes classes.
3. **Réduction du surapprentissage** : L'écart entre les performances sur les ensembles d'entraînement et de validation s'est réduit avec l'augmentation de données, suggérant une meilleure généralisation du modèle.

7.3 Forces et limitations

7.3.1 Points forts

Les principales forces du modèle final incluent :
— Une excellente précision générale sur les scènes urbaines standard ;

— Une robustesse accrue face aux variations d'éclairage et de perspective ;
— Une segmentation précise des classes principales (route, bâtiments, véhicules, etc.).

7.3.2 Limitations identifiées

Malgré ces résultats positifs, certaines limitations persistent :
— La segmentation des objets de petite taille reste perfectible ;
— Les performances peuvent varier selon les conditions d'éclairage extrêmes ;
— Certaines classes minoritaires présentent des résultats légèrement inférieurs

7.4 Impact sur le projet global

Ces résultats démontrent la viabilité de notre approche pour le système de vision par ordinateur de Future Vision Transport. La qualité de la segmentation obtenue, particulièrement avec l'augmentation de données étendue, fournit une base solide pour l'analyse de scènes urbaines dans le contexte des véhicules autonomes. La robustesse accrue du modèle face aux variations des conditions réelles représente un atout majeur pour son déploiement pratique.

8 Conclusion et pistes d'amélioration

8.1 Résumé des contributions

Ce projet de segmentation d'images pour Future Vision Transport a permis d'établir une base solide pour la compréhension de scènes urbaines dans le contexte des véhicules autonomes. L'utilisation de VGG16, combinée à des techniques d'augmentation de données appropriées, a démontré des résultats prometteurs. L'approche retenue, particulièrement avec l'augmentation de données via Albumentations dans sa version étendue, a permis d'atteindre des performances remarquables avec une précision de 92.24% et un coefficient de Dice de 88.77%.

La progression significative des performances entre le modèle initial et le modèle final témoigne de l'importance cruciale des techniques d'augmentation de données dans l'amélioration de la robustesse et de la précision du système.

8.2 Pistes d'amélioration

8.2.1 Enrichissement des données

Pour améliorer davantage les performances du modèle, plusieurs pistes peuvent être explorées :
- L'intégration d'autres jeux de données urbains comme Mapillary Vistas ou BDD100K pour diversifier les situations d'apprentissage.
- Le développement de techniques d'augmentation spécifiques pour les classes minoritaires ou difficiles à segmenter.

8.2.2 Optimisations architecturales

Des améliorations architecturales pourraient être envisagées :
- L'expérimentation avec des architectures plus récentes comme DeepLabV3+ ou PSPNet, qui pourraient offrir de meilleures performances sur les objets de petite taille.
- L'implémentation d'une approche d'ensemble combinant plusieurs modèles pour améliorer la robustesse des prédictions.
- L'exploration de techniques de distillation de modèle pour optimiser les performances tout en maintenant une efficacité computationnelle acceptable.

8.2.3 Améliorations méthodologiques

Plusieurs axes d'amélioration méthodologique peuvent être explorées :
- Le développement d'une stratégie d'apprentissage focalisée sur les cas difficiles identifiés pendant l'évaluation.
- L'implémentation d'un système de détection et de gestion des cas aberrants pour améliorer la fiabilité globale.
- L'introduction de techniques d'apprentissage semi-supervisé pour tirer parti de données non annnotées.

8.3 Perspectives

Ce projet constitue une première étape importante dans le développement du système de vision par ordinateur de Future Vision Transport. Les résultats obtenus démontrent la viabilité de l'approche choisie et ouvrent la voie à des améliorations futures.

L'attention particulière portée à la qualité de la segmentation et à la robustesse du modèle face aux variations des conditions réelles constitue une base solide pour les développements futurs. La poursuite de ces travaux, en intégrant les pistes d'amélioration identifiées, permettra d'accroître encore la fiabilité et la précision du système de vision par ordinateur, contribuant ainsi à la sécurité et à l'efficacité des véhicules autonomes de demain.