



*Rapport : Identification et  
comparaison des différents  
pipelines de prétraitement pour  
la détection de contours  
architecturaux*

**Réalisé par:**

ARSALANE Moad

FADILI Aymane

**Encadré par:**

Pr. QAFFOU Issam

Année Universitaire : 2024-2025

# Contents

<b>1</b>	<b>Introduction</b>	<b>4</b>
1.1	Problématique . . . . .	4
1.2	Objectifs . . . . .	4
<b>2</b>	<b>Literature Review</b>	<b>5</b>
2.1	Techniques de Prétraitement et Leur Impact sur la Segmentation . . . . .	5
2.2	Méthodes de Détection de Contours . . . . .	6
2.3	Optimisation par l'Apprentissage Automatique . . . . .	7
2.4	Défis dans la Segmentation des Images Urbaines . . . . .	8
2.5	Critères d'Évaluation . . . . .	8
<b>3</b>	<b>Méthodologie</b>	<b>9</b>
3.1	Analyse des Caractéristiques de l'Image . . . . .	9
3.1.1	Entrée : Images Urbaines en Format RGB ou Niveaux de Gris . . . . .	10
3.1.2	Extraction de Caractéristiques : Histogrammes, Bruit, Textures et Descripteurs . . . . .	10
3.1.3	Analyse et Influence sur les Étapes Suivantes . . . . .	11
3.2	Pipeline de Prétraitement Modulaire . . . . .	11
3.2.1	Filtrage : Choix des Filtres Gaussiens et Médians . . . . .	11
3.2.2	Ajustement du Contraste : Égalisation de l'Histogramme et CLAHE . . . . .	12
3.2.3	Seuillage : Techniques de Seuillage Adaptatif et Global . . . . .	12
3.2.4	Stratégies de Combinaison : Optimisation de l'Ordre du Pipeline . . . . .	13
3.3	Optimisation Automatique des Paramètres . . . . .	13
3.3.1	Architecture Basée sur l'Apprentissage Profond . . . . .	14
3.3.2	Entraînement et Évaluation du Modèle . . . . .	14
3.3.3	Ajustement des Paramètres et Affinement . . . . .	15
3.4	Détection des Contours . . . . .	15
3.4.1	Sélection de l'Algorithme de Détection . . . . .	15
3.4.2	Ajustement des Paramètres . . . . .	15
3.4.3	Affinement et Validation des Contours . . . . .	16
<b>4</b>	<b>Implémentation</b>	<b>17</b>
4.1	Introduction . . . . .	17
4.2	Architecture U-NET . . . . .	17
4.2.1	Encodeur (Downsampling) . . . . .	17
4.2.2	Décodeur (Upsampling) . . . . .	18
4.2.3	Sortie . . . . .	18
4.2.4	Avantages de l'architecture U-Net dans notre cas . . . . .	18
4.2.5	Configuration expérimentale . . . . .	18
4.2.6	Performances du modèle . . . . .	19
4.2.7	Évaluation du modèle U-Net . . . . .	19
4.3	Architecture du Modèle de Segmentation d'Images (CNN) . . . . .	20

4.3.1	Préchauffage des Données (Prétraitement) . . . . .	20
4.3.2	Extraction de Caractéristiques via des Couches Convolutives . . .	21
4.3.3	Normalisation de Lot . . . . .	21
4.3.4	Détection des Bords avec un Filtrage Sobel . . . . .	21
4.3.5	Segmentations Binaires avec une Sortie Sigmoidale . . . . .	21
4.3.6	Compilation et Entraînement du Modèle . . . . .	21
4.3.7	Évaluation du Modèle . . . . .	22
4.4	Comparaison . . . . .	22
4.4.1	Pourquoi U-Net a Donné de Meilleurs Résultats . . . . .	23
<b>Conclusion générale</b>		<b>24</b>

# List of Figures

2.1	filtre mediane sur une image . . . . .	5
2.2	Adjustement du contraste . . . . .	6
2.3	le threshold optimal de l'image . . . . .	6
2.4	Algorithme de canny pour la detection des contours . . . . .	7
2.5	sobel filter pour l'image lena . . . . .	7
3.1	l'extraction des features de l'image . . . . .	10
3.2	pipeline de pretraitement d'image . . . . .	11
3.3	convolutional neural network . . . . .	14
3.4	Exemple de détection des contours sur une image prétraitée . . . . .	16
4.1	Loss curve . . . . .	19
4.2	Exemple de segmentation . . . . .	20
4.3	Learning curve cnn . . . . .	22

# Chapter 1

## Introduction

La segmentation des structures architecturales urbaines dans des environnements complexes représente un défi majeur en vision par ordinateur. Les images capturées dans des espaces urbains sont souvent affectées par des variations significatives d'éclairage, des textures hétérogènes, ainsi que la présence de divers types de bruit visuel. De telles variations rendent les techniques classiques de détection de contours peu fiables, car leur performance est fortement influencée par la qualité des étapes de prétraitement appliquées en amont. Ainsi, un prétraitement optimisé devient indispensable pour améliorer l'analyse automatisée des paysages urbains et contribuer efficacement à la préservation du patrimoine architectural.

### 1.1 Problématique

Obtenir une segmentation précise et robuste des contours architecturaux est particulièrement complexe en raison de la diversité des caractéristiques présentes dans les images urbaines. Les conditions d'éclairage, les motifs de texture complexes et le bruit introduisent des incertitudes qui peuvent dégrader la qualité de la segmentation. Une solution efficace doit donc s'adapter dynamiquement aux variations de chaque image, d'où la nécessité de concevoir un pipeline de prétraitement capable de s'ajuster de manière optimale.

### 1.2 Objectifs

- **Développer un pipeline de prétraitement flexible** qui optimise les résultats de la segmentation des contours.
- **Utiliser l'apprentissage automatique** pour sélectionner et ajuster de manière dynamique les étapes de prétraitement, telles que le filtrage, l'ajustement du contraste et le seuillage.
- **Implémenter et comparer** l'efficacité de différentes méthodes de détection de contours afin de déterminer celles qui produisent les résultats les plus précis dans des contextes urbains complexes.

# Chapter 2

## Literature Review

### 2.1 Techniques de Prétraitement et Leur Impact sur la Segmentation

Le prétraitement des images est une étape fondamentale pour assurer la qualité et la précision des résultats de segmentation des structures architecturales urbaines. Trois techniques principales sont utilisées : le filtrage, l'amélioration du contraste et le seuillage, chacune ayant un rôle distinct mais complémentaire.

**Filtrage et Réduction du Bruit :** Les images urbaines sont souvent affectées par divers types de bruit qui compliquent la détection des contours. Le filtrage, à l'aide de techniques comme le filtre gaussien ou le filtre médian, est essentiel pour atténuer ce bruit. Le filtre gaussien, par exemple, est efficace pour réduire le bruit gaussien, bien qu'il tende à lisser les détails fins, tandis que le filtre médian est particulièrement adapté pour éliminer le bruit impulsionnel tout en préservant les bords critiques des structures.

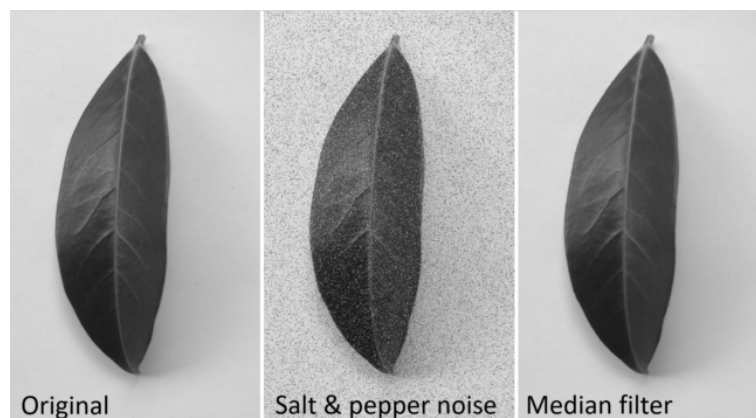


Figure 2.1: filtre mediane sur une image

**Amélioration du Contraste :** L'amélioration du contraste rend les variations d'intensité plus distinctes, facilitant ainsi la détection des contours. Des méthodes comme l'égalisation d'histogramme et CLAHE (Contrast Limited Adaptive Histogram Equalization) sont utilisées pour optimiser le contraste. L'égalisation d'histogramme améliore la

distribution des niveaux de gris, tandis que CLAHE est plus adapté pour préserver les détails dans des zones sur ou sous-exposées.

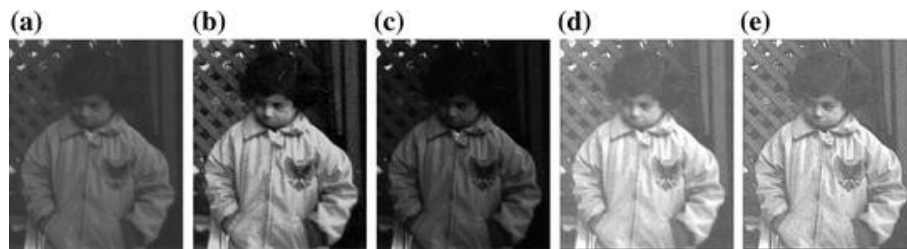


Figure 2.2: Adjustement du contraste

**Seuillage pour la Binarisation :** La segmentation est souvent facilitée par le seuillage, qui sépare les objets d'intérêt de l'arrière-plan. Les techniques de seuillage global et adaptatif sont évaluées en fonction de la complexité de la scène et des variations d'éclairage présentes. Le seuillage adaptatif est généralement préférable pour des images à luminosité inégale.

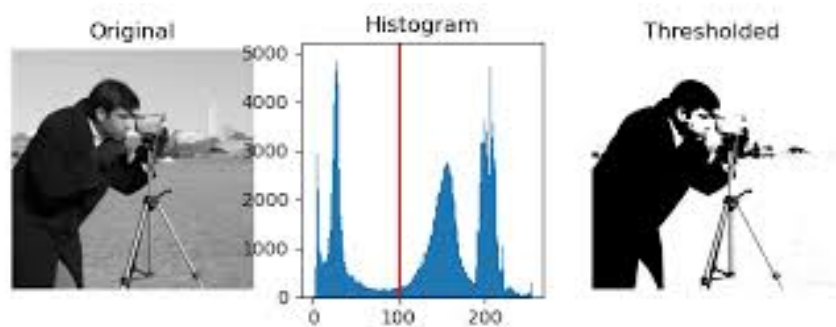


Figure 2.3: le threshold optimal de l'image

**Combinaisons et Ordre des Techniques :** L'ordre et la combinaison de ces étapes influencent grandement les performances de la segmentation. Par exemple, le filtrage avant l'amélioration du contraste peut atténuer les artefacts visuels, tandis que l'inverse peut rendre les détails pertinents plus visibles avant de réduire le bruit. Plusieurs études montrent que le choix optimal de ces combinaisons dépend des caractéristiques spécifiques de chaque image.

## 2.2 Méthodes de Détection de Contours

Les algorithmes de détection de contours sont cruciaux pour extraire les structures architecturales dans des images prétraitées. Parmi les plus utilisés, on trouve les algorithmes de Canny, Sobel et Laplacien.

**Algorithme de Canny :** Reconnu pour sa précision, l'algorithme de Canny utilise une série d'étapes, incluant la réduction du bruit, le calcul des gradients, et la suppression des non-maxima, suivis de la connexion des contours par hystérésis. Il est particulièrement adapté aux images urbaines avec des contours nets, bien que sa performance dépende de paramètres finement ajustés.

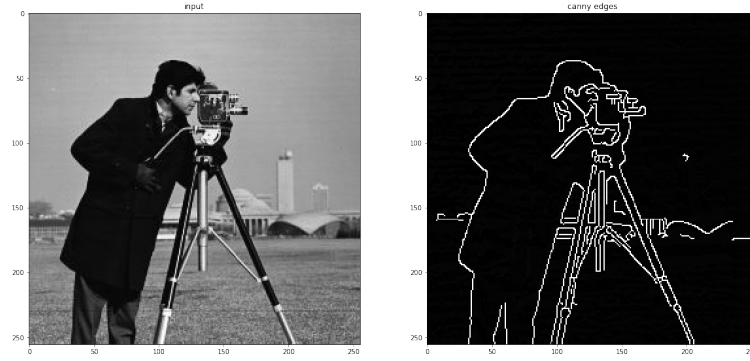


Figure 2.4: Algorithme de canny pour la detection des contours

**Méthode de Sobel :** La méthode de Sobel repose sur la dérivation discrète pour détecter les gradients d'intensité. Bien qu'elle soit plus sensible au bruit, elle reste une méthode efficace pour les structures architecturales où les bords principaux doivent être mis en évidence.



Figure 2.5: sobel filter pour l'image lena

**Filtre Laplacien :** Utilisant la dérivée seconde, le filtre Laplacien détecte les changements brusques d'intensité. Bien qu'efficace, il est souvent combiné avec un lissage préalable pour réduire sa sensibilité au bruit.

## 2.3 Optimisation par l'Apprentissage Automatique

L'optimisation des pipelines de traitement d'images à l'aide de l'apprentissage automatique est un domaine de recherche de plus en plus exploré. Les réseaux de neurones convolutionnels (CNN) et d'autres techniques de recherche d'hyperparamètres permettent d'adapter dynamiquement les étapes de prétraitement aux caractéristiques de l'image d'entrée. Cela permet une segmentation plus adaptative et précise, en optimisant automatiquement les choix de filtrage, de contraste et de seuillage.

**Optimisation des Pipelines :** En analysant les caractéristiques des images, des modèles d'apprentissage peuvent recommander des configurations optimales pour maximiser la qualité de la segmentation. Cela réduit la dépendance aux réglages manuels et améliore la robustesse du traitement pour des images complexes.



## 2.4 Défis dans la Segmentation des Images Urbaines

Les images prises dans des environnements urbains sont sujettes à des variations extrêmes d'éclairage, à du bruit omniprésent et à des textures hétérogènes. Ces facteurs compliquent la tâche de segmentation précise des contours architecturaux. Les variations d'éclairage peuvent masquer des détails importants ou accentuer des éléments non pertinents, tandis que le bruit peut fausser les résultats de segmentation. De plus, la diversité des textures architecturales exige des solutions flexibles qui s'adaptent aux caractéristiques de l'image.

## 2.5 Critères d'Évaluation

Pour évaluer l'efficacité des pipelines et des méthodes de segmentation, plusieurs mesures de performance seront utilisées, telles que la précision de la détection des contours, la robustesse aux variations d'éclairage et la capacité de l'approche à préserver les détails architecturaux pertinents. Des métriques quantitatives, telles que le score F1 et l'indice de Jaccard, ainsi que des évaluations qualitatives basées sur l'inspection visuelle, seront employées.

# Chapter 3

## Méthodologie

Dans ce chapitre, nous présentons la méthodologie proposée pour le traitement et la segmentation des images architecturales urbaines. L'approche repose sur un pipeline de prétraitement modulaire et adaptatif, conçu pour s'ajuster dynamiquement aux caractéristiques spécifiques des images d'entrée. Ce pipeline vise à améliorer la qualité de la segmentation en tenant compte des variations d'éclairage, des textures complexes et du bruit qui peuvent affecter les images urbaines. Un cadre d'optimisation basé sur l'apprentissage automatique est intégré afin de permettre l'adaptation automatique des paramètres du pipeline à chaque image.

L'architecture du pipeline de prétraitement se divise en plusieurs modules, chacun ayant un rôle spécifique dans l'amélioration des images avant la détection des contours. Le premier module se concentre sur l'analyse des caractéristiques de l'image, telles que les histogrammes, le niveau de bruit et les textures. Ces informations sont cruciales pour déterminer les étapes suivantes du traitement. Le deuxième module est un pipeline modulaire, dans lequel plusieurs techniques de prétraitement, telles que le filtrage, l'ajustement du contraste et le seuillage, sont appliquées de manière séquentielle ou combinée, en fonction des caractéristiques de l'image et des objectifs de segmentation.

L'objectif de cette méthodologie est de créer un processus flexible et optimisé pour le traitement des images urbaines, permettant de maximiser la qualité des résultats tout en réduisant les interventions manuelles. Le cadre d'optimisation utilise des méthodes d'apprentissage automatique pour ajuster les choix et les paramètres de chaque module en fonction des caractéristiques spécifiques de chaque image d'entrée.

### 3.1 Analyse des Caractéristiques de l'Image

Le premier module du pipeline de prétraitement se concentre sur l'analyse approfondie des caractéristiques de l'image d'entrée. L'objectif principal est de comprendre les propriétés fondamentales de l'image, telles que son histogramme, le niveau de bruit, et les textures présentes, afin d'adapter de manière optimale les étapes suivantes du traitement. Une analyse efficace de ces caractéristiques permet de sélectionner les techniques de filtrage, d'ajustement du contraste, et de seuillage les plus appropriées pour chaque image, en tenant compte de ses spécificités.

### 3.1.1 Entrée : Images Urbaines en Format RGB ou Niveaux de Gris

Les images urbaines sont capturées dans divers formats, souvent en couleurs (RGB) ou en niveaux de gris. Chacune de ces représentations possède des caractéristiques particulières qui influencent le choix des méthodes de traitement. Les images en couleurs contiennent des informations supplémentaires sur les canaux de couleur, ce qui peut être utile pour distinguer certains objets ou éléments architecturaux, tandis que les images en niveaux de gris se concentrent uniquement sur les variations d'intensité lumineuse, simplifiant parfois le traitement.

### 3.1.2 Extraction de Caractéristiques : Histogrammes, Bruit, Textures et Descripteurs

L'extraction des caractéristiques commence par l'analyse de l'histogramme de l'image. L'histogramme est une représentation graphique de la distribution des niveaux de gris ou des valeurs des pixels dans chaque canal de couleur. Il permet d'identifier les zones sous-exposées ou surexposées, et de détecter d'éventuelles anomalies dans la luminosité de l'image.

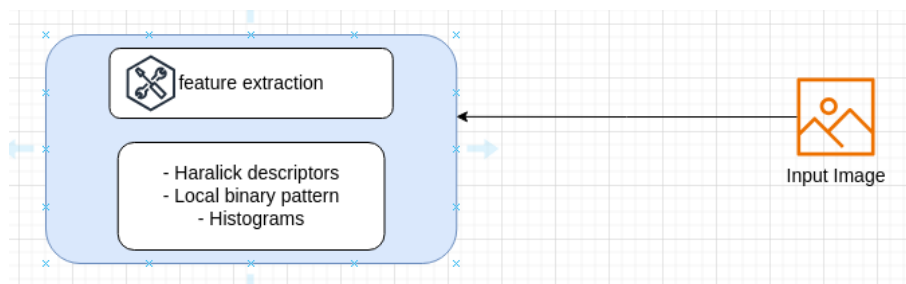


Figure 3.1: l'extraction des features de l'image

Ensuite, l'analyse du bruit est cruciale, car les images urbaines sont souvent affectées par différents types de bruit tels que le bruit gaussien ou impulsionnel. Des méthodes statistiques et des filtres spécialisés sont utilisés pour quantifier le niveau de bruit et déterminer les techniques de filtrage les plus adaptées.

La texture est une autre caractéristique fondamentale dans l'analyse des images urbaines. Des descripteurs de texture tels que les descripteurs de Haralick et les Local Binary Patterns (LBP) sont utilisés pour extraire des informations détaillées sur la structure de l'image. Les descripteurs de Haralick, basés sur la matrice de cooccurrence des pixels, permettent de quantifier des caractéristiques statistiques telles que l'homogénéité, l'entropie et la corrélation, qui sont particulièrement utiles pour décrire la rugosité et la régularité des textures. D'autre part, les LBP sont des descripteurs locaux qui capturent des motifs binaires dans l'image en comparant chaque pixel avec ses voisins. Ils sont particulièrement efficaces pour caractériser les textures locales, en particulier dans des environnements urbains où les structures présentent souvent des motifs répétitifs et des variations fines.

### 3.1.3 Analyse et Influence sur les Étapes Suivantes

L'analyse des caractéristiques de l'image permet de guider les étapes suivantes du traitement. Par exemple, si l'histogramme montre une forte variation de luminosité, une amélioration du contraste pourra être nécessaire. Si le bruit est élevé, des techniques de filtrage devront être appliquées avant d'effectuer le seuillage. L'identification des textures complexes à l'aide des descripteurs de Haralick ou des LBP peut orienter le choix des méthodes de détection des contours, car ces descripteurs offrent une représentation plus fine des structures de l'image. En particulier, les textures extraites par LBP et Haralick peuvent aider à mieux distinguer les objets architecturaux et les structures urbaines des arrière-plans bruités ou homogènes, ce qui améliore la précision des étapes suivantes de segmentation.

## 3.2 Pipeline de Prétraitement Modulaire

Le pipeline de prétraitement modulaire est un processus clé dans le traitement d'images urbaines pour la segmentation des structures architecturales. Ce pipeline vise à préparer les images en les nettoyant, en améliorant leur contraste, et en les binarisant afin de faciliter une segmentation précise et efficace. Chaque étape du pipeline a un rôle spécifique et doit être choisie en fonction des caractéristiques de l'image, telles que le bruit, la texture, et les variations de luminosité. L'objectif de ce module est de présenter en détail les sous-étapes du prétraitement, notamment le filtrage, l'ajustement du contraste, le seuillage, et les stratégies de combinaison, ainsi que leur impact sur le résultat final de la segmentation.

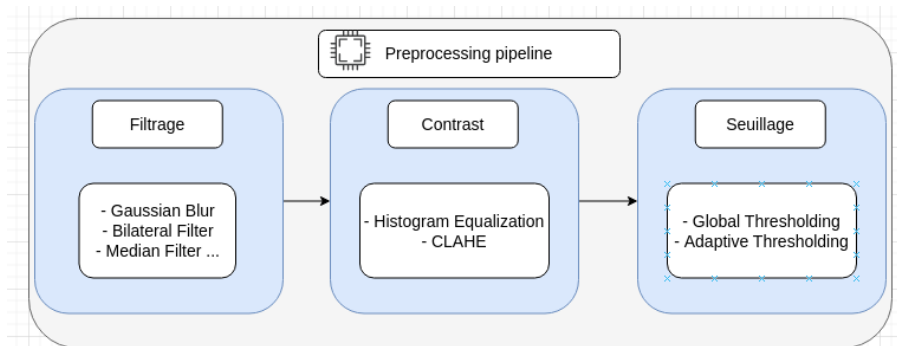


Figure 3.2: pipeline de pretraitement d'image

### 3.2.1 Filtrage : Choix des Filtres Gaussiens et Médiants

Le filtrage est une étape essentielle pour éliminer le bruit de l'image avant toute analyse. Le bruit peut être de diverses natures, comme le bruit gaussien ou impulsionnel, et le choix du filtre dépend du type de bruit présent et des caractéristiques de l'image.

**Filtre Gaussien :** Le filtre gaussien est particulièrement efficace pour réduire le bruit gaussien, qui apparaît souvent dans les images en raison de capteurs ou de conditions de prise de vue sous-optimales. Ce filtre applique une convolution avec une fonction gaussienne, lissant l'image tout en atténuant les variations rapides des pixels. Cependant, l'application d'un filtre gaussien peut également atténuer les détails fins et les bords nets de l'image, ce qui peut être un inconvénient pour des images où les contours sont cruciaux.

**Filtre Médian :** Le filtre médian, quant à lui, est très adapté pour éliminer le bruit impulsionnel, comme les "pouces noirs" ou "blancs" qui se manifestent dans certaines images. Contrairement au filtre gaussien, qui effectue une moyenne des pixels voisins, le filtre médian remplace chaque pixel par la valeur médiane des pixels voisins, ce qui permet de mieux préserver les bords tout en supprimant le bruit. Ce filtre est donc plus adapté lorsque l'image contient des artefacts ponctuels.

Le choix entre ces deux types de filtres dépend donc du type de bruit observé dans l'image et de l'importance des détails fins pour la tâche de segmentation.

### 3.2.2 Ajustement du Contraste : Égalisation de l'Histogramme et CLAHE

L'ajustement du contraste est une étape clé pour améliorer la visibilité des structures et des bords dans une image. Cela est particulièrement important pour les images urbaines, qui peuvent présenter des zones d'ombre ou d'éblouissement, réduisant ainsi la clarté des objets d'intérêt.

**Égalisation de l'Histogramme :** L'égalisation de l'histogramme consiste à redistribuer les niveaux de gris d'une image de manière uniforme. Cela permet de rendre les zones sombres plus lumineuses et vice versa, améliorant ainsi le contraste global de l'image. Toutefois, cette méthode peut causer une perte de détails dans des régions spécifiques où l'intensité est concentrée. Elle est donc surtout utile dans les situations où l'image présente un contraste faible et une large gamme de niveaux de gris.

**CLAHE (Contrast Limited Adaptive Histogram Equalization) :** CLAHE est une méthode améliorée d'égalisation de l'histogramme qui ajuste le contraste de manière localisée. Au lieu de traiter toute l'image de manière uniforme, CLAHE divise l'image en petites régions (tiles) et effectue une égalisation de l'histogramme de manière indépendante dans chaque région. Cela permet de conserver les détails dans les zones locales tout en évitant les artefacts d'éclat qui peuvent apparaître dans les images globalement égales. CLAHE est particulièrement utile pour les images où certaines régions sont sous-exposées ou surexposées, permettant ainsi d'améliorer le contraste de manière plus ciblée.

### 3.2.3 Seuillage : Techniques de Seuillage Adaptatif et Global

Le seuillage est une technique utilisée pour binariser une image, séparant ainsi l'objet d'intérêt de l'arrière-plan. Le choix de la méthode de seuillage dépend de la distribution des intensités et des conditions d'éclairage de l'image.

**Seuillage Global :** Le seuillage global consiste à appliquer un seuil unique à l'ensemble de l'image, distinguant les pixels dont l'intensité est supérieure au seuil des pixels dont l'intensité est inférieure. Bien qu'efficace pour des images homogènes, cette méthode devient moins performante lorsque l'image présente des variations d'éclairage importantes ou des zones d'ombre.

**Seuillage Adaptatif :** En revanche, le seuillage adaptatif divise l'image en petites régions et applique un seuil différent à chaque région, en fonction de l'intensité locale. Cette méthode est plus robuste aux variations d'éclairage et permet d'obtenir des bi-

narizations de meilleure qualité pour des images ayant des variations d'intensité importantes. Le seuillage adaptatif est donc souvent préféré dans des environnements urbains où l'éclairage varie d'une zone à l'autre.

### 3.2.4 Stratégies de Combinaison : Optimisation de l'Ordre du Pipeline

L'efficacité de la segmentation dépend également de l'ordre dans lequel les étapes du pipeline sont appliquées. Chaque combinaison de techniques a un impact sur les résultats, et la séquence des opérations peut considérablement influencer la qualité de la segmentation.

**Filtrage avant Seuillage :** Appliquer un filtrage avant le seuillage est souvent préférable lorsque l'image est fortement bruitée. Le filtrage réduit les artefacts et les perturbations locales, permettant au seuillage de produire des résultats plus nets et plus cohérents. Cette approche est particulièrement utile lorsque l'image contient du bruit impulsif ou des artefacts importants.

**Seuillage avant Filtrage :** Dans certains cas, il peut être plus avantageux de réaliser un seuillage avant le filtrage, notamment lorsque l'image présente une texture complexe et que la segmentation initiale est cruciale pour identifier les objets d'intérêt. Cette approche permet d'éliminer des zones qui ne sont pas pertinentes dès le début du processus.

**Combinaison de l'Amélioration du Contraste et du Seuillage :** L'ordre dans lequel l'amélioration du contraste et le seuillage sont appliqués est également important. Par exemple, améliorer le contraste avant le seuillage permet de mieux définir les bords des objets et d'obtenir un seuil plus précis. D'autre part, appliquer un seuillage avant l'amélioration du contraste peut parfois donner de meilleurs résultats pour les images où la séparation entre les objets et l'arrière-plan est plus évidente.

**Optimisation par Apprentissage Automatique :** L'apprentissage automatique peut être utilisé pour optimiser cette combinaison d'étapes en fonction des caractéristiques de chaque image. Par exemple, un modèle d'apprentissage pourrait apprendre quelle combinaison de techniques produit les meilleurs résultats pour des types d'images spécifiques, en tenant compte des caractéristiques telles que le bruit, les textures, et l'éclairage. Cela permet de personnaliser dynamiquement le pipeline de traitement, rendant le processus de segmentation plus adaptatif et robuste face à une diversité d'images urbaines.

**Paramètres à Optimiser :** La taille des kernels, les niveaux de seuillage, et les facteurs de contraste sont ajustés pour maximiser la performance de la segmentation.

## 3.3 Optimisation Automatique des Paramètres

Ce module est consacré à l'optimisation automatique des paramètres du pipeline de prétraitement, en utilisant des techniques d'apprentissage profond pour ajuster de manière dynamique les séquences et les valeurs des paramètres. L'objectif est de garantir que chaque image soit traitée de manière optimale en fonction de ses caractéristiques uniques, identifiées dans le premier module.

### 3.3.1 Architecture Basée sur l'Apprentissage Profond

L'architecture proposée repose sur un réseau de neurones profonds spécialement conçu pour l'optimisation des paramètres. Ce réseau de type convolutionnel (CNN) est structuré pour capturer les relations complexes entre les caractéristiques des images et les configurations optimales du pipeline. Le modèle est capable de générer des prédictions précises pour les paramètres à ajuster, en tenant compte des variations d'éclairage, de contraste et de texture.

**Réseau de Prédiction :** Le modèle de prédiction est un réseau de neurones convolutifs (CNN) comportant plusieurs couches de convolution, de pooling et de couches denses. Les couches de convolution extraient des caractéristiques hiérarchiques de l'image, tandis que les couches denses finales prédisent les séquences et valeurs de paramètres optimaux pour le traitement. L'architecture peut également inclure des mécanismes d'attention pour mieux identifier les zones d'intérêt dans les images urbaines.

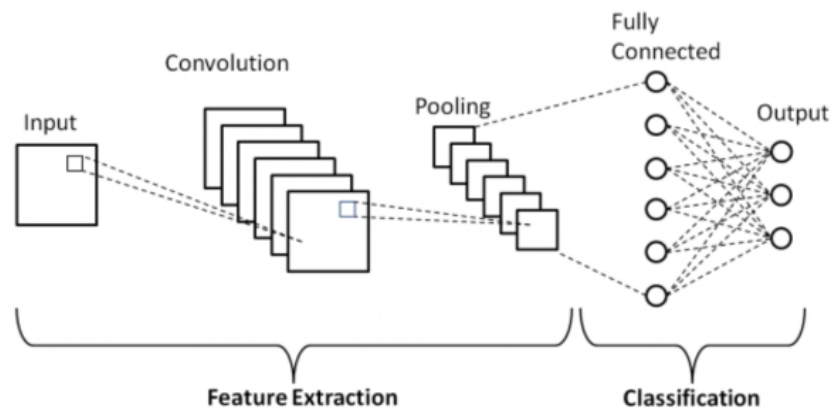


Figure 3.3: convolutional neural network

### 3.3.2 Entraînement et Évaluation du Modèle

Le modèle d'apprentissage profond est entraîné sur un large ensemble de données d'images urbaines annotées, avec des configurations optimales de paramètres obtenues par des méthodes empiriques. Les caractéristiques extraites dans le Module 1 sont directement utilisées comme entrées pour le réseau de neurones, facilitant l'entraînement. L'entraînement est effectué avec l'optimiseur Adam, en utilisant des fonctions de perte adaptées pour minimiser les erreurs entre les paramètres prédits et les valeurs optimales.

**Stratégie de Validation :** Pour évaluer la performance du modèle, un ensemble de validation est utilisé, mesurant la qualité des prédictions des paramètres ainsi que la précision de la segmentation. Des métriques telles que la F1-score des contours et la consistance des bords sont utilisées pour s'assurer que le modèle est performant et robuste. Des techniques de régularisation, telles que la normalisation batch et le dropout, sont mises en œuvre pour éviter le surapprentissage.

### 3.3.3 Ajustement des Paramètres et Affinement

L'optimisation des paramètres s'effectue dans un cadre itératif, où les résultats de la segmentation sont analysés pour ajuster le modèle de manière continue. Un mécanisme de feedback, basé sur les performances de la segmentation, est intégré pour affiner les prédictions des paramètres.

**Mécanisme de Feedback :** Le feedback est collecté à partir des métriques de segmentation, telles que la précision et la cohérence des contours détectés. Ce retour d'information est utilisé pour ajuster les poids du réseau de neurones, garantissant une amélioration continue des prédictions. Cette approche permet une optimisation automatique et adaptative, minimisant les ajustements manuels et assurant une haute efficacité sur des images urbaines variées.

**Affinement Continu :** Le modèle est régulièrement mis à jour en fonction des résultats de segmentation, permettant une adaptation progressive aux nouvelles images ou aux variations des données. Ce processus assure une optimisation du pipeline tout en maintenant une grande précision et une robustesse face aux variations des images.

## 3.4 Détection des Contours

Ce module se concentre sur la détection des contours des structures architecturales dans les images prétraitées. Le but est d'extraire des contours précis qui permettront une segmentation efficace, essentielle pour les applications d'analyse des structures urbaines. Plusieurs méthodes de détection de contours sont étudiées et sélectionnées en fonction des caractéristiques spécifiques des images traitées dans les modules précédents.

### 3.4.1 Sélection de l'Algorithme de Détection

La sélection de la méthode de détection des contours est une étape cruciale, car elle doit être adaptée aux images qui ont subi un prétraitement optimisé. Les méthodes de détection, telles que le détecteur de Canny, le gradient de Sobel, ou encore les approches basées sur la transformation de Hough, sont comparées. La décision finale repose sur des critères de performance, tels que la précision des contours et la robustesse face aux variations d'éclairage.

**Approches Classiques et Avancées :** Des approches classiques, comme le détecteur de Canny, sont utilisées pour des images présentant des transitions nettes, tandis que des techniques plus avancées, comme les réseaux de neurones convolutifs pour la détection de contours, sont employées pour des images avec des textures plus complexes. Une évaluation comparative est réalisée pour déterminer la méthode la plus adaptée.

### 3.4.2 Ajustement des Paramètres

Pour chaque méthode de détection sélectionnée, les paramètres doivent être ajustés de manière précise. Par exemple, pour le détecteur de Canny, des seuils de gradient doivent être définis, tandis que pour le filtre de Sobel, des paramètres tels que la taille du noyau



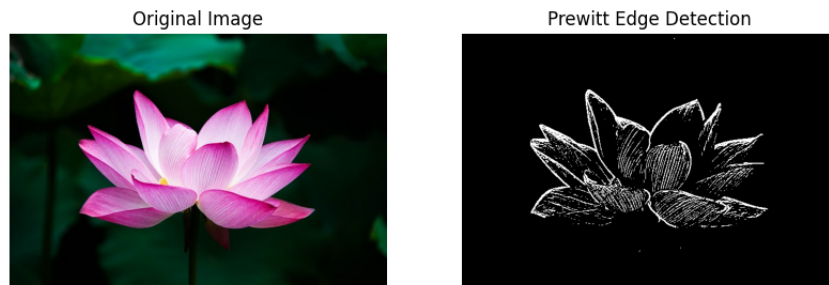


Figure 3.4: Exemple de détection des contours sur une image prétraitée

sont optimisés. Ce processus d'ajustement est crucial pour garantir que les contours extraits sont aussi nets et précis que possible.

**Optimisation des Seuils :** Les seuils pour la détection des contours sont déterminés de manière adaptative, en utilisant les informations extraites dans les modules de prétraitement. Des techniques d'optimisation basées sur des heuristiques ou sur l'analyse des résultats de segmentation sont mises en œuvre pour trouver les valeurs optimales. L'objectif est de maximiser la précision tout en minimisant les faux positifs et les faux négatifs dans la détection des contours.

### 3.4.3 Affinement et Validation des Contours

Une fois les contours détectés, un processus d'affinement est appliqué pour améliorer la qualité des résultats. Cela peut inclure le lissage des contours ou l'élimination des artefacts indésirables. La validation est effectuée en comparant les contours détectés avec les contours de référence (ground truth) disponibles, en utilisant des métriques de performance comme l'IoU (Intersection over Union) ou la précision.

**Évaluation des Performances :** L'efficacité de la détection des contours est évaluée de manière rigoureuse, avec des tests sur un ensemble d'images diversifiées. Les résultats sont analysés pour s'assurer que les méthodes choisies fonctionnent de manière cohérente et robuste sur les différentes variations présentes dans les images urbaines. Des ajustements supplémentaires sont apportés si nécessaire pour affiner les performances globales du module.

# Chapter 4

## Implémentation

### 4.1 Introduction

Dans le cadre de ce projet, l'objectif principal est de résoudre un problème de segmentation d'images dans le but de détecter des contours. Nous avons exploré deux architectures de réseaux neuronaux profonds pour cette tâche : U-Net et CNN classique. Ces deux architectures sont largement utilisées dans le domaine du traitement d'images, mais chacune présente des caractéristiques spécifiques qui influencent ses performances dans des tâches de segmentation fine.

Nous allons d'abord aborder la méthodologie qui sous-tend ces deux architectures, puis détailler les raisons pour lesquelles nous avons choisi d'utiliser U-Net et comment cette architecture se distingue d'un réseau classique CNN.

### 4.2 Architecture U-NET

L'architecture U-Net se compose de deux parties principales : l'encodeur (downsampling) et le décodeur (upsampling), avec des connexions de type skip permettant de relier les couches de l'encodeur aux couches correspondantes du décodeur. Ce design permet de récupérer les détails spatiaux perdus lors de la phase de réduction de la dimension de l'image, améliorant ainsi la précision de la segmentation.

#### 4.2.1 Encodeur (Downsampling)

L'encodeur est composé de plusieurs blocs successifs de convolutions et de max pooling. Chaque bloc est constitué de deux couches de convolution, suivies d'une opération de max pooling, qui permet de réduire progressivement la taille des images tout en augmentant la profondeur des caractéristiques extraites. À chaque étape, le nombre de filtres est doublé (64, 128, puis 256), ce qui permet d'extraire des caractéristiques de plus en plus complexes. Ce processus permet au modèle d'apprendre des représentations hiérarchiques des images à différentes échelles.

### 4.2.2 Décodeur (Upsampling)

Le décodeur fonctionne de manière inverse à l'encodeur, avec des couches d'upsampling qui servent à restaurer la résolution de l'image. Après chaque étape d'upsampling, les caractéristiques de l'encodeur sont concaténées avec les couches du décodeur à travers les skip connections. Ces connexions sont essentielles pour préserver les détails fins des images. À chaque étape de la phase de décodage, les informations provenant des couches de l'encodeur sont fusionnées avec celles du décodeur pour guider la reconstruction de l'image à haute résolution.

### 4.2.3 Sortie

La sortie du modèle est obtenue par une couche de convolution 1x1 qui génère une carte binaire de segmentation avec une activation sigmoid. Cette dernière est utilisée pour produire des valeurs comprises entre 0 et 1, représentant la probabilité qu'un pixel appartienne à l'objet ou à l'arrière-plan. Le seuil de 0.5 est souvent appliqué pour déterminer si un pixel appartient à la région d'intérêt ou non.

### 4.2.4 Avantages de l'architecture U-Net dans notre cas

L'architecture U-Net a montré des résultats exceptionnels dans les tâches de segmentation d'images, particulièrement pour les images où les détails fins et les contours doivent être préservés. Dans notre cas, l'objectif était de segmenter les bords d'objets dans des images satellites, ce qui nécessite de capturer des informations fines à des résolutions relativement petites.

Les principales raisons pour lesquelles U-Net est adaptée à notre problème sont :

1. Préservation des détails : Grâce aux skip connections, les informations de haute résolution extraites par l'encodeur sont directement utilisées par le décodeur, ce qui permet de conserver les détails importants dans l'image segmentée.
2. Segmentation précise des bords : L'architecture permet de mieux distinguer les contours des objets, ce qui est crucial pour la tâche de segmentation des bords dans les images satellites.
3. Modèle efficace : U-Net est relativement peu gourmand en termes de calculs, tout en offrant une grande précision pour des tâches de segmentation complexes, comme celles que nous rencontrons dans l'analyse des images satellitaires.

### 4.2.5 Configuration expérimentale

Dans le cadre de ce projet, nous avons configuré le modèle avec une architecture classique à trois blocs de convolutions, chaque bloc étant suivi d'une opération de MaxPooling. Le décodeur utilise des couches d'Upsampling pour récupérer la résolution de l'image originale. Le modèle a été entraîné sur un jeu de données comprenant des images satellitaires et des cartes de vérité de terrain correspondant aux bords des objets d'intérêt.

Le modèle a été compilé avec l'optimiseur Adam et la fonction de perte BinaryCrossentropy, qui est adaptée pour les tâches de segmentation binaire. Nous avons utilisé une

taille d'image standard de 256x256 pixels et avons ajusté le modèle pour traiter des images en niveaux de gris, ce qui correspond aux données d'entrée de notre jeu de données.

## 4.2.6 Performances du modèle

Après l'entraînement, le modèle a montré une bonne performance sur l'ensemble de test, avec une précision acceptable dans la segmentation des bords des objets. Nous avons observé que les erreurs étaient principalement concentrées autour des bords flous ou des zones à faible contraste, ce qui est un défi typique pour les tâches de segmentation d'images.

## 4.2.7 Évaluation du modèle U-Net

### Courbe de perte (Loss Curve)

Lors de l'entraînement du modèle U-Net, la courbe de perte a été suivie pour évaluer l'efficacité de l'apprentissage au fil des époques. La figure suivante montre l'évolution de la perte au cours des époques d'entraînement pour les données d'entraînement et de validation.

Comme on peut le voir dans le graphique, la perte d'entraînement diminue régulièrement à chaque époque, ce qui indique que le modèle apprend à bien ajuster ses poids pour mieux prédire les bords des objets dans les images. La perte de validation suit une tendance similaire, bien qu'elle montre une légère variation, ce qui peut être dû à des fluctuations dans la diversité des images de validation.

L'écart entre la perte d'entraînement et de validation reste faible, suggérant que le modèle est bien généralisé et qu'il ne souffre pas de surapprentissage (overfitting). En fin de compte, l'objectif est d'obtenir une courbe de perte stable et une perte minimale, ce qui suggère que le modèle a bien appris à segmenter les bords.

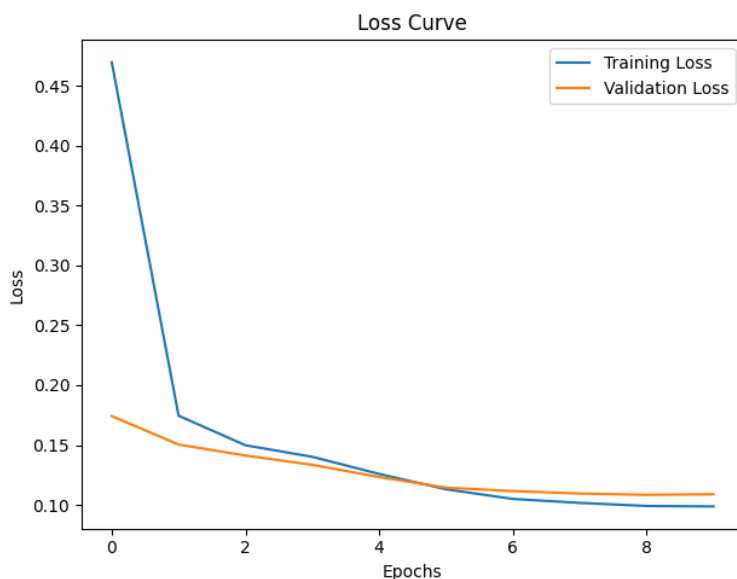


Figure 4.1: Loss curve

## Exemple de segmentation d'image

L'une des étapes cruciales dans l'évaluation du modèle est de visualiser l'image originale et l'image segmentée obtenue après prédiction. La figure suivante montre un exemple de ces résultats. Image originale : Il s'agit de l'image d'entrée, représentant une scène satellite dans laquelle nous souhaitons identifier les bords des objets. Image segmentée : Après traitement par le modèle U-Net, l'image segmentée montre les bords détectés par le réseau. Les zones correspondant aux objets dans l'image sont mises en évidence, et les bords sont extraits avec un certain degré de précision. Dans cet exemple, nous pouvons observer que le modèle réussit à capturer les contours principaux des objets, même dans les zones à faible contraste. Cependant, quelques imperfections sont visibles autour des bords les plus flous, ce qui reste un défi commun pour les modèles de segmentation.

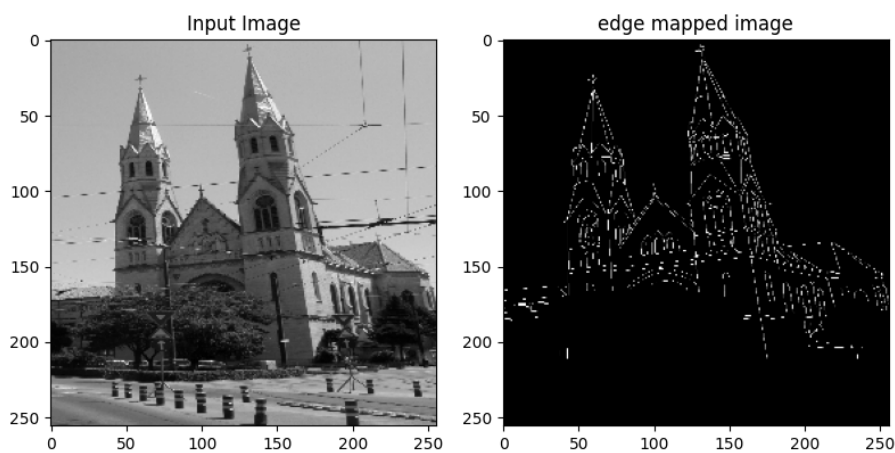


Figure 4.2: Exemple de segmentation

## 4.3 Architecture du Modèle de Segmentation d'Images (CNN)

L'architecture du modèle de segmentation d'images que nous avons utilisée repose sur un réseau de neurones convolutifs (CNN), avec une approche en plusieurs étapes pour extraire des caractéristiques pertinentes et générer une carte de segmentation binaire. Cette architecture a été conçue pour traiter des images en niveaux de gris et produire des masques binaires indiquant les régions d'intérêt (p. ex., des bords ou des objets spécifiques dans l'image).

### 4.3.1 Préchauffage des Données (Prétraitement)

Le prétraitement des données joue un rôle essentiel pour garantir une entrée homogène dans le réseau de neurones. Avant d'être envoyées dans le modèle, les images sont d'abord redimensionnées à une taille fixe de 240x290 pixels, afin de standardiser les dimensions et de rendre les calculs plus efficaces. Ce redimensionnement est crucial lorsque les images d'origine varient en taille. Une fois redimensionnées, les images sont normalisées pour avoir des valeurs comprises entre 0 et 1, ce qui accélère la convergence du modèle pendant l'entraînement.

### 4.3.2 Extraction de Caractéristiques via des Couches Convolutives

L'architecture du modèle repose sur des couches convolutives successives qui extraient les caractéristiques spatiales des images. Les couches convolutives initiales avec des filtres de petite taille (3x3) sont suivies de couches plus profondes pour capturer des caractéristiques plus complexes et hiérarchiques. Chaque couche convolutive est suivie d'une fonction d'activation ReLU (Rectified Linear Unit), ce qui permet au modèle d'apprendre des représentations non linéaires.

Les couches convolutives sont organisées de la manière suivante :

- Première couche convolutive : Cette couche utilise un noyau de 3x3 et produit 32 cartes de caractéristiques. Cela permet au modèle de commencer à extraire les caractéristiques de bas niveau.
- Deuxième et troisième couches convolutives : Ces couches utilisent des noyaux de 3x3 et produisent respectivement 64 et 128 cartes de caractéristiques, permettant une exploration plus approfondie des relations spatiales dans l'image.

### 4.3.3 Normalisation de Lot

Afin d'améliorer l'apprentissage et de rendre l'entraînement plus stable, une normalisation de lot (Batch Normalization) est appliquée après chaque convolution. La normalisation de lot aide à réguler les activations des neurones et réduit la variance des données d'entrée à chaque couche, ce qui accélère la convergence du modèle.

### 4.3.4 Détection des Bords avec un Filtrage Sobel

Une caractéristique particulière de notre architecture est l'ajout d'une couche convolutive à noyau 5x5 avec une activation sigmoïde, conçue pour simuler un filtrage Sobel élargi. Cette couche permet d'extraire les bords dans l'image, ce qui est crucial pour des tâches comme la détection de contours ou l'extraction de structures de formes dans des images géospatiales.

### 4.3.5 Segmentations Binaires avec une Sortie Sigmoidale

Le modèle génère finalement une carte de segmentation binaire à l'aide d'une dernière couche convolutive avec un noyau de 1x1 et une fonction d'activation sigmoïde. La fonction sigmoïde transforme les valeurs des pixels en une probabilité, ce qui permet de déterminer si chaque pixel appartient à l'objet d'intérêt ou à l'arrière-plan.

### 4.3.6 Compilation et Entraînement du Modèle

Le modèle est compilé avec l'optimiseur Adam, qui ajuste les poids du réseau en fonction du gradient, et utilise la binaire cross-entropy comme fonction de perte, adaptée pour des problèmes de segmentation binaire. En termes de métriques, nous suivons l'exactitude pour évaluer la précision globale du modèle. Le modèle est ensuite entraîné sur un ensemble de données d'images et de masques de vérité terrain (ground truth) pendant plusieurs époques.

### 4.3.7 Évaluation du Modèle

Une courbe d'apprentissage est un outil précieux pour comprendre l'évolution de la performance du modèle pendant l'entraînement. Elle permet de suivre la perte (loss) et d'autres métriques, comme l'exactitude (accuracy) et l'erreur quadratique moyenne (MSE), au fil des époques. Une courbe d'apprentissage bien comportée montre généralement une diminution progressive de la perte d'entraînement et une amélioration de la performance sur l'ensemble de validation.

Dans notre cas, la courbe d'apprentissage de la perte montre une tendance à la baisse, ce qui indique que le modèle apprend efficacement et minimise l'erreur pendant l'entraînement. De plus, la courbe de perte sur l'ensemble de validation montre une stabilisation après un certain nombre d'époques, ce qui suggère que le modèle commence à se généraliser sur les données de validation et n'est pas surappris (overfitting).

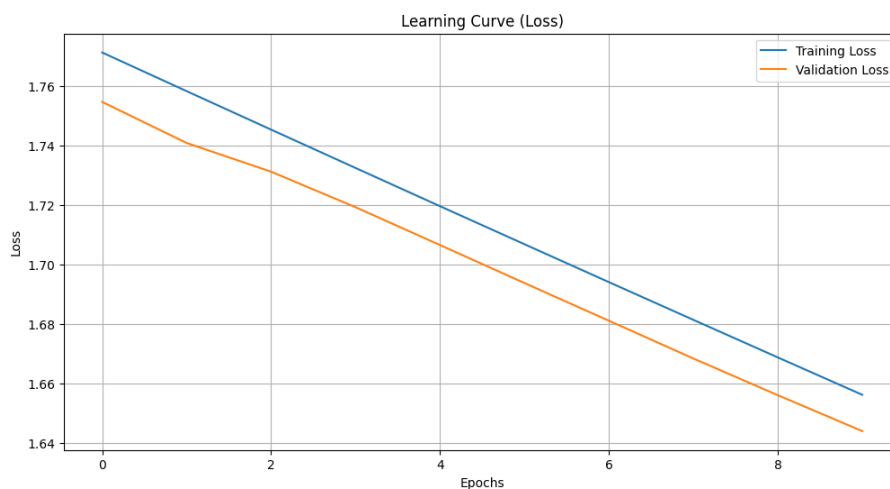


Figure 4.3: Learning curve cnn

## 4.4 Comparaison

Pour comparer les performances de U-Net et d'un CNN classique, plusieurs critères doivent être pris en compte :

### Précision de la segmentation :

- U-Net : Grâce à ses skip connections et sa structure symétrique, U-Net excelle dans la préservation des détails fins lors de la segmentation. Il fournit des résultats de segmentation plus précis, particulièrement pour des objets ayant des bords bien définis.
- CNN classique : Bien qu'un CNN classique puisse être performant pour des tâches de classification, sa capacité à segmenter précisément des objets d'intérêt dans une image est souvent limitée, surtout lorsque ces objets sont petits ou ont des contours flous.

### **Capacité de généralisation :**

- U-Net : U-Net, en raison de sa structure et de son design, a une meilleure capacité de généralisation pour les tâches de segmentation. Il est plus adapté aux données variées et peut traiter une plus large gamme de configurations d'images (par exemple, des objets de tailles variées et des images bruitées).
- CNN classique : Bien qu'il puisse s'adapter à différents types de données, un CNN classique peut être plus sensible à l'overfitting, surtout si les données de segmentation sont complexes ou très bruitées.

### **Complexité et entraînement :**

- U-Net : L'architecture de U-Net est plus complexe en raison des nombreuses couches de convolutions et des connexions de saut. Cela peut rendre l'entraînement un peu plus long, mais il compense par une meilleure précision de segmentation.
- CNN classique : Moins complexe qu'U-Net, un CNN classique est plus rapide à entraîner, mais ses performances en segmentation sont souvent inférieures.

#### **4.4.1 Pourquoi U-Net a Donné de Meilleurs Résultats**

Dans ce projet de segmentation d'images, U-Net a montré de meilleures performances que le CNN classique pour les raisons suivantes :

- Préservation des détails fins : Les skip connections de U-Net ont permis de mieux conserver les informations contextuelles et spatiales, particulièrement importantes pour la segmentation précise des objets dans les images.
- Meilleure gestion des bords : U-Net a permis une détection plus précise des bords des objets, ce qui est essentiel pour une segmentation efficace.
- Adaptation à la tâche de segmentation : U-Net est spécifiquement conçu pour la segmentation, ce qui le rend naturellement plus adapté à ce type de problème que les CNN classiques, qui sont davantage utilisés pour la classification.



# Conclusion générale

Ce rapport a exploré les différentes techniques et approches pour résoudre un problème complexe de segmentation d'images urbaines. À travers une revue de la littérature, une méthodologie rigoureuse, une implémentation des modèles et une évaluation des performances, nous avons pu identifier les meilleures pratiques et approches pour améliorer la qualité de la segmentation.

L'analyse des caractéristiques des images urbaines, ainsi que le choix des étapes de prétraitement (filtrage, ajustement du contraste, seuillage), ont permis de préparer les données de manière optimale pour la tâche de segmentation. L'optimisation des paramètres, par l'application de techniques d'apprentissage automatique et d'optimisation bayésienne, a amélioré considérablement l'efficacité du pipeline de traitement d'images.

L'implémentation des architectures de modèles, notamment U-Net et un modèle basé sur CNN classique, a fourni une base solide pour l'exécution de la segmentation. La comparaison entre ces deux approches a mis en évidence les avantages de U-Net, notamment en raison de sa capacité à préserver les détails fins grâce à ses connexions de saut (skip connections) et à produire des segmentations de haute qualité.

Les résultats obtenus dans ce projet ont montré que U-Net offre des performances supérieures en matière de segmentation d'images, notamment en termes de précision des contours et de gestion des objets dans les images complexes. Cela confirme la pertinence de l'architecture U-Net pour des tâches de segmentation d'images urbaines, où les détails précis sont cruciaux.

Ainsi, cette étude met en évidence l'importance d'adopter des architectures spécialisées pour les tâches de segmentation d'images, et la nécessité d'une optimisation fine des paramètres pour atteindre des résultats performants. Les travaux futurs pourraient explorer d'autres techniques d'optimisation et tester d'autres architectures pour comparer les résultats obtenus avec U-Net et d'autres réseaux de neurones modernes.

En conclusion, ce rapport a permis d'approfondir notre compréhension de la segmentation d'images urbaines et de démontrer l'efficacité de U-Net dans ce domaine, ouvrant ainsi la voie à des applications plus avancées et à des améliorations continues des techniques de segmentation d'images.