REPUBLIQUE ALGERIENNE DEMOCRATIQUE ET POPULAIRE

MINISTERE DE L'ENSEIGNEMENT SUPERIEUR ET DE LA RECHERCHE SCIENTIFIQUE

Université de Mohamed El-Bachir El-Ibrahimi - Bordj Bou Arreridj

Faculté *des Sciences et de la technologie*

Département d'Electronique

Mémoire

Présenté pour obtenir

LE DIPLOME DE MASTER

FILIERE : **Télécommunications.**

Spécialité : Systèmes des télécommunications.

Par

- **Melle Batli Siham**
- **Melle Bengana Ghofrane**

Intitulé

Super résolution d'image par apprentissage profond

Évalué le : ……

Par la commission d'évaluation composée de* :

Nom & Prénom	*Grade*	*Qualité*	*Etablissement*
M.TALBI M.Lamine	*MCA*	*Président*	*Univ-BBA*
Mme. SID AHMED Soumia	*MCB*	*Encadreur*	*Univ-BBA*
Mme. MESSALI Zoubeida	*Prof*	*Co-encadreur*	*Univ-BBA*
M.A.BENGUEDOUDJ AbdeAllah		*Examinateur*	*Univ-BBA*

Année Universitaire 2022/2023

REMERCIMENTS

Avant toute chose, je remercie ***«Allah »****, qui nous aide et nous donne la volonté, le courage et la patience durant ces années d'étude.*

Je souhaite adresser mes remerciements les plus chères aux personnes qui ont contribués à l'élaboration et l'avancement de mon sujet d'étude, ainsi que l'ensemble de la famille universitaire de Bordj-Bou-Arreridj et plus précisément ceux du département d'Electronique.

Je remercie mes encadrantes ***Dr.SID AHMED Soumia****, et* ***Prof MESSALI Zoubeida*** *qui ont toujours montré leurs intérêts et leur suivi de ce travail et qui ont été disponibles tout au long de mon travail, sans oublier leurs remarques pertinentes et leurs conseils et patiences durant l'étude de ce sujet pour aboutir à un travail de qualité. Je remercie en particulier,* ***Dr.Boudechiche****.*

Je désir aussi remercier les enseignants de l'université Mohamed Elbachir El Ibrahimi, qui nous ont fourni les outils nécessaires à la réussite de nos études universitaires pendant mon cursus.

Enfin je tiens à exprimer toute ma gratitude à l'ensemble de ma famille et proches pour leur soutien tout au long de ces années.

Dédicace

Siham :

Je dédie ce mémoire

Mon cher Père, qui a été sauvé par la grâce de Dieu.

Ma chère mère, dont Dieu a prolongé l'âge.

Tous mes frères, sœurs et amis qui m'ont soutenu tout au long de ce travail.

غفران:

إلى من اشتاق إليه بكل جوارحي

إلى من رحل عن عالمنا ومازال دوي نصائحه يوجهني

إلى من علمني كيف أقف بكل ثبات فوق الأرض

الرجل الأبرز في حياتي

جدي رحمه الله

إلى التي فارقتني بجسدها ولكن روحها مازالت ترفرف في سماء حياتي

إلى رمز التفاني والإخلاص التي أفنت عمرها في تربيتي وخدمتي، التي لم تبخل بمساعدتي يوما

إلى نبع المحبة والإيثار والكرم

جدتي حبيبة قلبي رحمها الله.

Table des Matières

Liste des Figures

Liste des Tableaux

Liste des Acronymes

SR Super Résolution

HR High Resolution (Haute résolution)

LR Low Resolution (Basse Résolution)

IA Intelligence Artificielle

ML Machine Learning (Apprentissage automatique)

DL Deep Learning (Apprentissage Profond)

CNN Convolutional Neural Network (Réseau de neurones convolutifs)

PSNR Peak Signal to Noise Ratio (Rapport signal sur bruit de crête)

MSE Mean Squared Error (Erreur quadratique moyenne)

SSIM Indice de similarité structurelle

SRCNN Super-Resolution Convolution neural network (Réseau de neurones à super-résolution à convolution)

EDSR Enhanced deep super-resolution network (Réseau de super-résolution profonde amélioré)

CAR Content adaptive resample (Rééchantillonnage adaptatif du contenu)

Introduction générale

La super résolution (SR) fait référence au processus d'augmentation de la résolution d'une image au-delà de sa taille ou de sa qualité d'origine. Cela se fait généralement à l'aide d'algorithmes qui interpolent entre les pixels de l'image d'origine pour produire une version à plus haute résolution. Cependant, les techniques d'interpolation traditionnelles donnent souvent des images floues ou pixélisées.

Pour résoudre ce problème, les chercheurs ont développé des techniques d'apprentissage profond, qui se sont révélées très prometteuses pour produire des images super résolues de haute qualité. L'apprentissage profond (Deep Learning) implique la formation de réseaux de neurones sur de grands ensembles de données pour apprendre des modèles et des relations complexes au sein des données. En appliquant ces réseaux neurones entraînés à des images basse résolution, il est possible de générer des images haute résolution avec beaucoup plus de détails et de clarté.

Avec le développement rapide des techniques d'apprentissage profond au cours des dernières années [1], les modèles basés sur l'apprentissage profond ont été activement explorés et appliqués dans différents domaines tels que la tomodensitométrie (Computed Tomography CT [2]), les images biomédicales, les images radar à synthèse d'ouverture (Synthetic aperture radar SAR [3]) et le contrôle non destructif (NDC) [4]. Pour résoudre les tâches de SR, une gamme d'approches d'apprentissage profond a été développée, allant de la méthode des premiers réseaux de neurones convolutionels (Super Resolution Convolutional Neurol Network CNN : SRCNN) à la méthode plus récente des réseaux de neurones convolutionnels profonds comme le rééchantillonnage adaptatif du contenu (Content Adapative Resampling CAR). Comme nous le verrons dans notre étude de simulation.

Notre travail est structuré en trois chapitres :

Le premier chapitre abordera les concepts fondamentaux de la super-résolution d'image. Nous passerons en revue les méthodes conventionnelles d'interpolation, puis nous nous concentrerons sur les algorithmes de super-résolution basés sur l'apprentissage profond (Deep Learning).

Le deuxième chapitre présentera les concepts fondamentaux de l'apprentissage profond ainsi que les différentes couches d'un réseau de Deep Learning. Une description détaillée sera consacrée aux réseaux de neurones convolutifs (CNN).

Le troisième chapitre présentera notre étude comparative ainsi que la discussion des résultats obtenus. Plus précisément, nous allons implémenter les réseaux DL de super-résolution suivants : SRCNN et Content Adaptive Resampler (CAR). La génération des images dégradées est accomplie en sous échantillonnant les images originales. Il est important de noter que le bruit additif n'a pas été pris en compte dans notre étude.

Pour chaque algorithme que nous avons mis en œuvre, nous disposons des images originales haute résolution (HR) en tant qu'images de référence, des images générées en basse résolution (LR) et enfin des images estimées en super-résolution (SR). Nous évaluons les performances des différents algorithmes SR en utilisant deux métriques : le rapport signal sur bruit maximum (peak signal-to-noise ratio PSNR) et l'indice de similarité structurelle (structural similarity index SSIM).

Chapitre 1

Super résolution des images

Résumé

Dans ce Chapitre, nous présenterons les principaux concepts de la super résolution d'image (ISR). Une description des méthodes conventionnelles d'interpolation sera présentée avant la description de trois algorithmes SR basés sur l'apprentissage profond. Deux mesures seront définies pour l'évaluation des performances.

Sommaire

1.1. Introduction

Lors de l'utilisation de photos numériques dans des applications, il est toujours recommandé d'utiliser des photos haute résolution , car elles peuvent contenir des détails plus utiles pour certaines applications que leurs équivalents basse résolution. La précision du diagnostic d' un médecin peut dépendre d' une image haute résolution, les objets peuvent être clairement distinguables sur une image satellite haute résolution, ou un algorithme de reconnaissance de forme peut fonctionner plus efficacement lorsqu'il reçoit des échantillons d'image haute résolution.

La super-résolution, est un terme désignant un ensemble de techniques d'imagerie visant à transformer une image ou une vidéo à faible résolution en quelque chose avec une plus haute résolution.

Dans ce chapitre, nous allons décrire le principe de la super résolution (SR) des images ainsi que méthodes conventionnelles d'interpolation et les algorithmes basés sur l'apprentissage profond. Nous mettrons en évidences les avantages de l'apprentissage profond en SR.

1.2. Principe de la Super Résolution

La super résolution est une technique de traitement d'image qui permet d'obtenir une image de haute résolution à partir d'une ou plusieurs images de basse résolution. Cette technique est largement utilisée dans des domaines tels que l'imagerie médicale) la capture d'images IRM haute résolution peut s'avérer délicate en termes de durée d'acquisition, de couverture spatiale et de rapport- Signal sur bruit (SNR). La super résolution aide à résoudre Ce problème en générant une IRM haute résolution à partir d'images IRM autrement à faible résolution) , la vidéo-surveillance(pour détecter, identifier et effectuer une reconnaissance faciale sur des images basse résolution obtenues à partir de caméras de sécurité) , la photographie numérique, etc.

La super résolution peuvent être réalisées par différentes manières, mais la plupart des méthodes utilisent des algorithmes de traitement d'image pour estimer les détails manquants ou perdus dans les images de basse résolution. Les techniques les plus courantes incluent la super-résolution basée sur l'interpolation, la super-résolution basée sur l'apprentissage profond (Deep Learning).

1.3 Modélisation Mathématique

La super résolution d'image peuvent être modélisées mathématiquement comme un problème inverse, dans lequel on cherche à estimer une image haute résolution à partir d'une image basse résolution et de certaines informations supplémentaires.

Plus précisément, supposons que l'on dispose d'une image haute résolution $\boldsymbol{x}$, qui a été sous-échantillonnée et flouée pour donner une image basse résolution $\boldsymbol{y}$. Le processus de sous-échantillonnage et de floutage peut être modélisé mathématiquement à l'aide d'un opérateur de convolution H, qui représente le flou et la réduction de la résolution.

Le modèle mathématique peut donc être écrit sous la forme suivante :

$$\boldsymbol{y}=\boldsymbol{H} * \boldsymbol{x}+\mathbf{n} \qquad \mathbf{1.1}$$

Où $\boldsymbol{y}$ est l'image basse résolution, $\boldsymbol{x}$ est l'image haute résolution, $\boldsymbol{n}$ est le bruit ajouté et * est l'opérateur de convolution.

Le but de la super résolution d'image est alors de trouver une estimation de l'image haute résolution x à partir de l'image basse résolution y et de l'opérateur de convolution avec $\boldsymbol{H}$. Cela peut être réalisé en utilisant des algorithmes de traitement d'image qui estiment la fonction de transfert inverse de $\boldsymbol{H}$, appelée opérateur de déconvolution, afin de récupérer l'image haute résolution x.

Dans certains cas, des informations supplémentaires peuvent être disponibles pour aider à résoudre le problème inverse. Par exemple, des connaissances sur la structure spatiale de l'image, sur les propriétés du bruit ou sur les caractéristiques des objets dans l'image peuvent être utilisées pour améliorer la qualité de l'estimation de l'image haute résolution. Ces informations supplémentaires peuvent être incorporées dans le modèle mathématique sous la forme de termes de régularisation ou de contraintes. [5]

On peut écrire le modèle de dégradation par l'équation :

$$\boldsymbol{G_k = DM_k\, B_k\, f + \varepsilon_K} \qquad \mathbf{1.2}$$

Où $\boldsymbol{G_K}$ est l'ensemble de k images LR, $\boldsymbol{D}$ est l'opérateur du sous-échantillonnage, $\boldsymbol{M_K}$ désigne l'opérateur de décalage (translation, rotation), $\boldsymbol{B_K}$ est la matrice du flou, $\boldsymbol{f}$ est l'image HR et $\boldsymbol{\varepsilon_K}$ représente le bruit du système.

Figure 1.1 résume la génération d'image LR à partir d'une image HR.

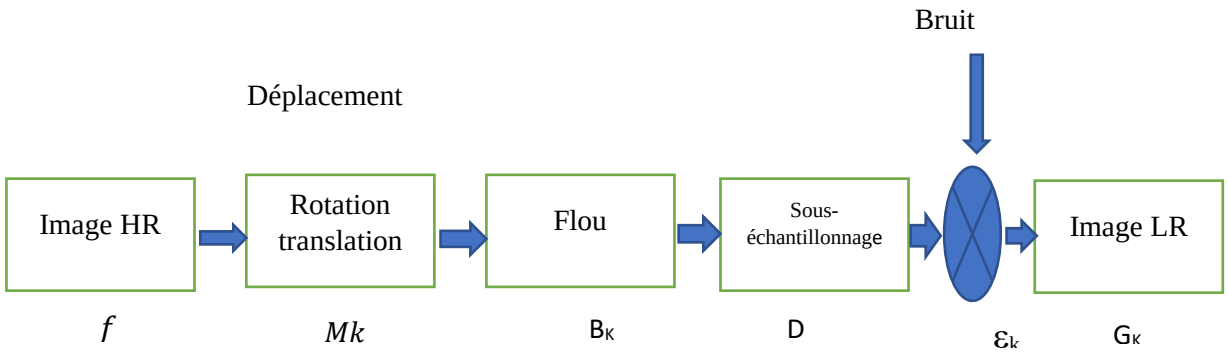


Figure 1.1: Modèle d'observation qui relie les images LR aux images HR.

1.4 Méthodes et Techniques de la Super Résolution

Les méthodes de la super résolution basées sur l'interpolation consistent à estimer une image haute résolution à partir d'une ou plusieurs images basse résolution. Ces méthodes sont basées sur l'hypothèse que les images haute résolution et basse résolution sont liées par une fonction d'interpolation.

L'approche de la super-résolution d'image peut être divisée en deux classes :

1. Méthodes basées sur l'interpolation telles que les interpolations bilinéaire, bicubique et Nearest.
2. Méthodes basées sur l'apprentissage et l'apprentissage profond (Machine Learning et Deep Learning). [6]

La classification des différentes méthodes de la super résolution est illustrée dans la Figure 1.2

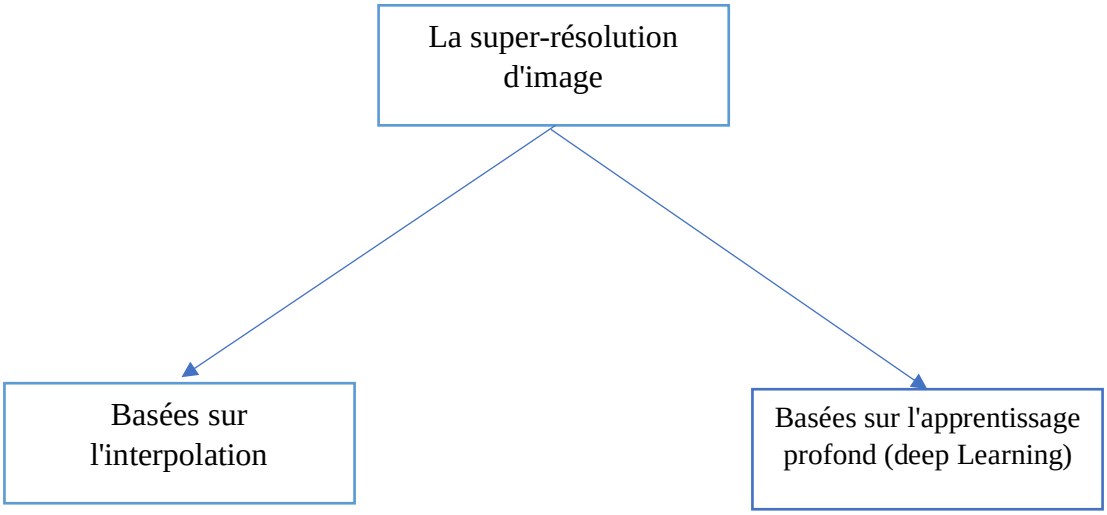


Figure 1.2: Classification de l'approche de la super-résolution d'image.

1.5. Méthodes basées sur l'interpolation

Les méthodes de la super résolution basées sur l'interpolation consistent à estimer une image haute résolution à partir d'une ou plusieurs images basse résolution. Ces méthodes sont basées sur l'hypothèse que les images haute résolution et basse résolution sont liées par une fonction d'interpolation.

IL existe plusieurs méthodes interpolation de super résolution, notamment :

1. L'interpolation bilinéaire

2. L'interpolation bicubique

3. L'interpolation par le voisin le plus proche (nearest neighbor)

1.5.1. L'Interpolation Bilinéaire

L'interpolation bilinéaire est une méthode d'interpolation en deux dimensions utilisée pour estimer des valeurs de points dans une grille régulière à partir de valeurs connues aux points de la grille. Cette méthode suppose que la variation de la fonction entre les points de grille est linéaire dans chaque direction.

Plus précisément, si nous avons une grille régulière de points avec des valeurs connues aux nœuds de la grille, et que nous voulons estimer la valeur de la fonction en un point qui ne

Se trouve pas sur la grille, l'interpolation bilinéaire utilise une combinaison linéaire des quatre points de la grille les plus proches pour estimer la valeur de la fonction en ce point.

Pour cela, on commence par identifier les quatre points les plus proches du point d'interpolation, puis on calcule une fonction linéaire de deux variables qui passe par ces quatre points. Enfin, on évalue cette fonction linéaire au point d'interpolation pour obtenir une estimation de la valeur de la fonction à ce point. [7]. La Figure 1.3 illustre le principe d'interpolation bilinéaire.

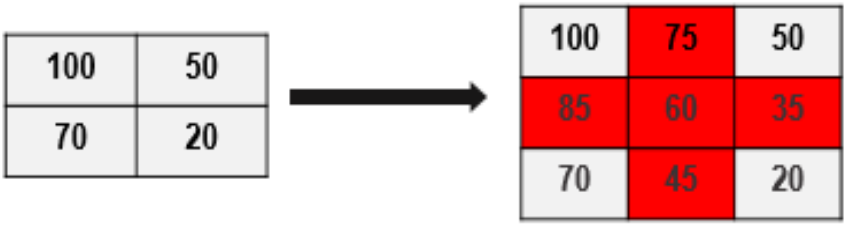


Figure 1.3:L'interpolation Bilinéaire.

1.5.2. L'Interpolation Bicubique

L'interpolation bicubique est une méthode d'interpolation de surface qui utilise une fonction cubique pour estimer les valeurs de pixels manquants dans une image. Cette méthode est souvent utilisée pour augmenter la résolution d'une image ou pour effectuer une interpolation plus précise que l'interpolation bilinéaire.

L'interpolation bicubique fonctionne en ajustant une fonction cubique à chaque ensemble de quatre pixels voisins dans l'image. Cette fonction cubique est ensuite utilisée pour estimer la valeur de tout point dans l'image qui n'est pas défini. Les fonctions cubiques sont ajustées de manière à ce que la surface de l'image interpolée soit lisse et continue.

L'interpolation bicubique est plus précise que l'interpolation bilinéaire, mais elle peut également être plus coûteuse en termes de calcul. En outre, l'interpolation bicubique peut introduire des artefacts dans l'image interpolée, en particulier si les données d'entrée sont bruitées. La Figure 1.4 montre le principe de l'interpolation bicubique.

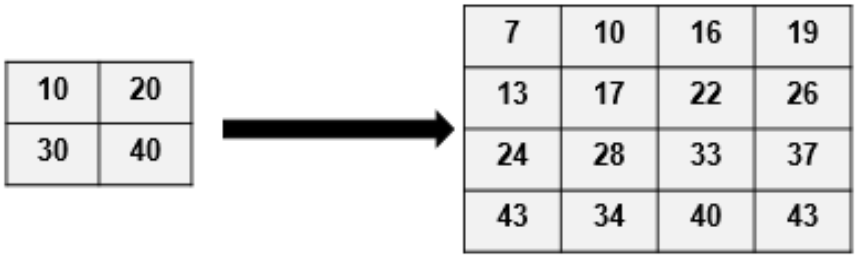


Figure 1.4:L'interpolation Bicubique [8]

1.5.3. L'Interpolation par le Voisin le Plus Proche (Nearest Neighbor)

L'interpolation "nearest-neighbor" (ou interpolation par plus proche voisin) est une méthode d'interpolation d'images qui consiste à remplacer chaque pixel manquant d'une image par la valeur du pixel le plus proche dans l'image originale.

Concrètement, cette méthode consiste à déterminer le pixel le plus proche de la position où se trouve le pixel manquant, puis à lui attribuer la même valeur que ce pixel. Cette méthode est simple et rapide à mettre en œuvre, mais elle peut produire des images avec une qualité inférieure à celle des autres méthodes d'interpolation.

En effet, cette méthode peut créer des images avec des bords abrupts et des contours marqués, qui peuvent donner une impression de pixellisation et de manque de fluidité dans les transitions.

Cependant, dans certains cas spécifiques, l'interpolation "nearest-neighbor" peut être préférable à d'autres méthodes d'interpolation, par exemple pour la segmentation d'images ou pour la classification d'images binaires. Dans ces cas, l'interpolation "nearest-neighbor" peut offrir des résultats plus précis et plus cohérents que les autres méthodes.

1.6. Méthodes basées sur l'Apprentissage Profond (Deep Learning)

Les méthodes basées sur l'apprentissage telles que la mise en correspondance des correctifs, l'apprentissage automatique, CNN et le réseau contradictoire génératif., Enhanced deep super-résolution network (EDSR), Content adaptive resample (CAR) et Super-Resolution Convolution neural network (SRCNN).

1.7.Algorithmes de Super Résolution (SR) Basés Sur des Réseaux Neurone Profonds

1.7.1. SRCNN : Super-Resolution Convolution neural network

SRCNN est une architecture CNN simple composée de trois couches : une pour l'extraction de patch, le mappage non linéaire et la reconstruction. La couche d'extraction de patch est utilisée pour extraire les patchs denses de l'entrée et pour les représenter à l'aide de filtres convolutifs. La couche de mappage non linéaire se compose de filtres convolutifs 1 × 1 utilisés pour modifier le nombre de canaux et ajouter de la non-linéarité. La couche de reconstruction finale reconstruit l'image haute résolution. La fonction de perte MSE est utilisée pour former le réseau, et le PSNR est utilisé pour évaluer les résultats. le SRCNN consiste en l'opération suivante, comme le montre la figure. [9]

1) Prétraitement (Preprocessing) : Image LR à l'échelle vers l'image HR souhaitée à l'aide d'une interpolation Bicubique.

2) Extraction de caractéristiques (**Feature extraction**) : extrait un ensemble de cartes de caractéristiques à partir de l'image LR haut de gamme.

3) Mappage non linéaire non linéaire (**Non –linear mapping)** : mappe les cartes de caractéristiques entre le patch LR et HR.

4) Reconstruction (**Reconstruction**) : produire l'image HR à partir de patchs HR.

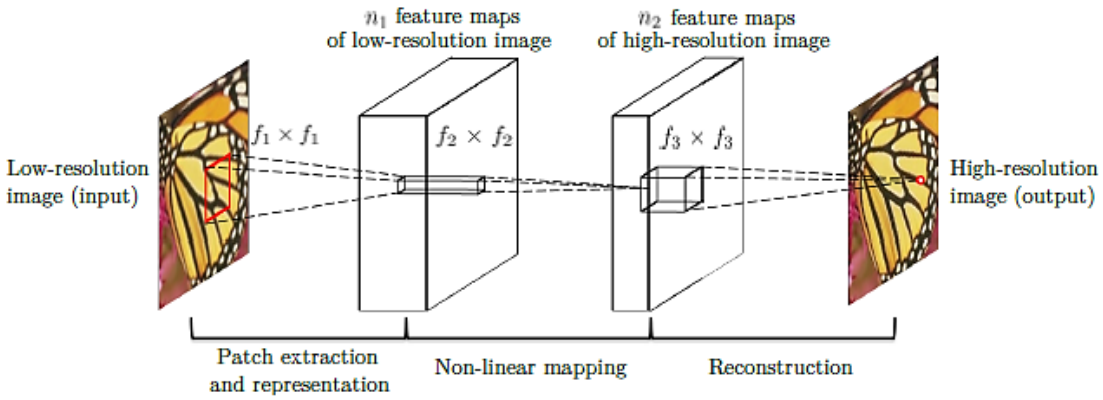


Figure 1.5:*Architecture du réseau de neurones convolutif à super résolution (SRCNN) [9]*

1.7.2. CAR : Content Adaptive Resampler

Le modèle CAR (Content Adaptive Resampler) est utilisé par le biais d'un modèle de super-résolution (SR) de pointe appelé EDSR (Enhanced Deep Super-Resolution) pour guider l'entraînement du modèle CAR proposé.

Dans cette méthode, nous proposons une méthode d'échelle d'image apprise basée sur (CAR), en tenant compte du processus d'agrandissement. Le réseau de resampling proposé génère des noyaux de resampling d'image adaptatifs au contenu qui sont appliqués à l'entrée HR originale pour générer des pixels sur l'image réduite.

De plus, un module de mise à l'échelle différentiable (SR) est utilisé pour agrandir le résultat LR en sa contrepartie HR sous-jacente. En rétro-propageant l'erreur de reconstruction jusqu'à l'entrée HR originale dans l'ensemble du cadre pour ajuster les paramètres du modèle, le cadre atteint une nouvelle performance de SR de pointe grâce aux resamplers d'image guidés par l'agrandissement qui préservent de manière adaptative les informations détaillées essentielles à l'agrandissement.

- **Model Architecture**

Il se compose de trois parties, le ResamplerNet, le module Downscaling et le SRNet. Le ResamplerNet est conçu pour estimer les noyaux de rééchantillonnage adaptatif du contenu et ses décalages correspondants pour chaque pixel de l’image réduite. Le SRNet, qui peut être n'importe quelle forme d'opérations de suréchantillonnage différentiables, est utilisé pour guider la formation des ResamplerNet en minimisant simplement l'erreur SR. La Figure 1.6 montre Architecture du réseau CAR. [10]

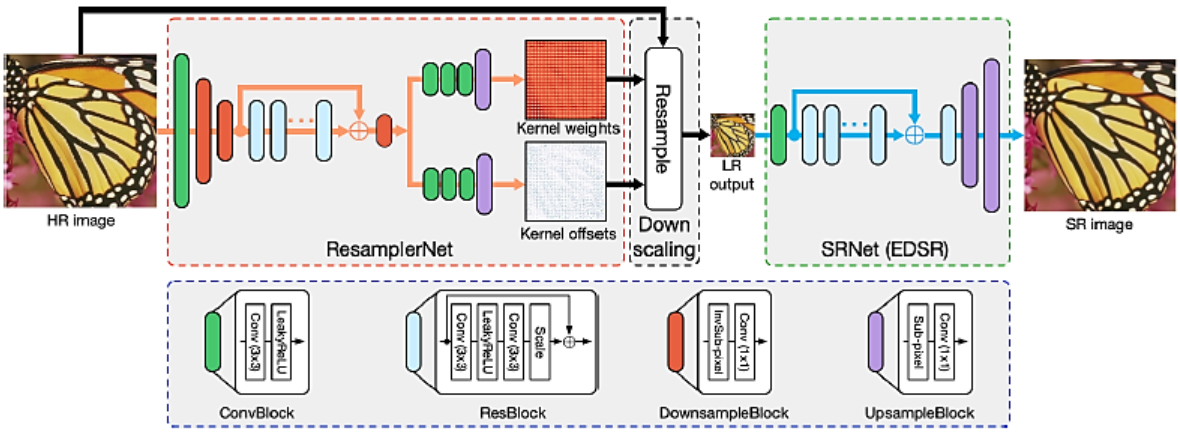


Figure 1.6: Architecture du réseau CAR [10]

Le ResamplerNet se compose de blocs de réduction d'échelle (downscaling), de blocs résiduels et de blocs de mise à l'échelle (upscaling).

- **Réduction d'échelle d'image (Image downscaling)**

Le rôle des blocs de réduction d'échelle avec les blocs résiduels est d'obtenir les blocs de mise à l'échelle des cartes d'entités constitués d'une étape d'interpolation basée sur un algorithme bicubique.

- **Mise à l'échelle de l'image (Image upscaling)**

Le modèle CAR proposé est formé en utilisant la rétro-propagation pour maximiser les performances SR. Le module d'upscaling d'image peut être de toute forme de réseaux SR, même les réseaux différentiables bilinéaires ou opérations de mise à l'échelle bicubique**.**

1.8. Évaluation des performances en termes de paramètres quantitatifs (PSNR, SSIM)

Parmi les mesures de qualité d'image couramment utilisées pour évaluer les performances de ces méthodes, on trouve le PSNR (Peak Signal to Noise Ratio) et le SSIM (Structural SIMilarity).

1.8.1. PSNR

Mesure la qualité de l'image reconstruite en comparant la différence entre chaque pixel de l'image originale et chaque pixel de l'image reconstruite, et en le comparant à l'amplitude maximale du signal (c'est-à-dire la valeur maximale de la plage dynamique de l'image). Le rapport signal/bruit utilisée pour déterminer la qualité des résultats. Il peut être calculé directement à partir du **MSE** en utilisant la formule ci-dessous, où est la valeur de pixel maximale possible (255 pour une image 8 bits). [11]

$$MSE=\frac{1}{MN}\sum_{i=1}^{M}\sum_{J=1}^{N}(u(i,j)-\hat{u}\ (i,j))^2, \quad \textbf{1.3}$$

$$PSNR=10.log_{10}\left(\frac{(2^k-1)^2}{MSE}\right). \quad \textbf{1.4}$$

$\boldsymbol{MSE}$ représente l'erreur quadratique moyenne entre deux images, u est l'image initiale et $\hat{\boldsymbol{u}}$ est l'image reconstruite de taille $M\ x\ N, i$ et j représentent respectivement la position des pixels de la ligne et de la colonne de l'image, k est le nombre de bits de chaque valeur d'échantillon.

1.8.2. SSIM

Evalue la similitude structurelle entre l'image originale et l'image reconstruite en mesurant les différences de luminance, de contraste et de structure entre les deux images. Le SSIM prend en compte les caractéristiques perceptuelles de l'image et est donc considéré comme une mesure plus précise de la qualité visuelle. [11]

$$SSIM = l(u,\hat{u}) \times c(u,\hat{u}) \times s(u,\hat{u}) \quad \textbf{1.5}$$

Tel que, $l(i,j)$ est la comparaison de luminance définie en fonction de la moyenne

Intensités $\mu_i$et μ_j des signaux j et i :

$$l(u,\hat{u}) = \frac{2\mu_i\ \mu_j+c_1}{\mu_i^2+\mu_j^2+c_1} \quad \textbf{1.6}$$

La comparaison de contraste $c(i,j)$ est fonction des écarts-types σ_i et σ_j prend la forme suivante :

$$c(i,j) = \frac{2\sigma_i\sigma_j + c_2}{{\sigma_i}^2 + {\sigma_j}^2 + c_2} \quad \textbf{1.7}$$

La comparaison de structure $s(i,j)$ est définie comme suit :

$$s(i,j) = \frac{\sigma_{ij} + c_3}{\sigma_i\,\sigma_j + c_3} \quad \textbf{1.8}$$

Où, μ_i et μ_j , représentent la valeur moyenne des images u et $\hat{u}$ respectivement. ${\sigma_i}^2$ Et ${\sigma_j}^2$ représentent respectivement la variance des images u et $\hat{u}$. σ_{ij} Est la covariance des images u et $\hat{u}$. c_1, c_2 et c_3 sont des constantes.

Concrètement, il est possible de choisir $c_i = k_i^2 D^2, i = 1$ et k_i, est une constante, telle que k_i<< 1 et D est la dynamique des valeurs des pixels ($D = 255$ correspond à une image numérique en niveaux de gris lorsque le nombre de bits/pixel est de 8).

1.9. Conclusion

Ce chapitre nous a donné un aperçu des différentes techniques utilisées en super-résolution. Nous avons exploré deux approches principales : les méthodes conventionnelles basées sur l'interpolation, telles que le Bicubique, le Bilinéaire et le Plus proche voisin, ainsi que les méthodes basées sur l'apprentissage profond, notamment le Super-Resolution Convolutional Neural Network (SRCNN) et le Content Adaptive Resampler (CAR). De plus, nous avons mis en évidence les principales métriques d'évaluation utilisées pour évaluer les performances des modèles de super-résolution.

Dans le chapitre suivant, nous approfondirons les détails de l'apprentissage profond.

Chapitre 2

Deep Learning

Résumé

Dans ce chapitre, nous présentons les principaux concepts de l'apprentissage automatique et de l'apprentissage profond aussi les réseaux neuronaux et réseaux de neurones convolutifs (CNN).

Sommaire

2.1. Introduction

Depuis 2006, le concept d'apprentissage profond est apparu comme un nouveau domaine d'étude pour l'apprentissage automatique. Les méthodes d' apprentissage profond qui ont été développées récemment ont déjà eu une influence sur le travail effectué pour traiter les signaux et les informations, y compris des parties de l'apprentissage automatique et de l'intelligence artificielle.

Dans ce chapitre, nous allons aborder les concepts de base de l'intelligence artificielle (AI), ainsi que Machine Learning et nous allons décris les différentes structures de Deep Learning.

2.2. L' Apprentissage Automatique

Souvent appelé Machine Learning ou ML en anglais, est une forme d' Intelligence Artificielle qui permet aux ordinateurs d' apprendre sans être explicitement programmés pour le faire .Néanmoins, pour que les ordinateurs apprennent et se développent, ils ont besoin de données qui peuvent être analysées et formées. En réalité, c'est la technologie qui permet d'exploiter pleinement le potentiel du Big Data .Les méthodes d'apprentissage automatique conventionnelles ont été limitées dans leur capacité à traiter les données naturelles sous leur forme brute. [12]

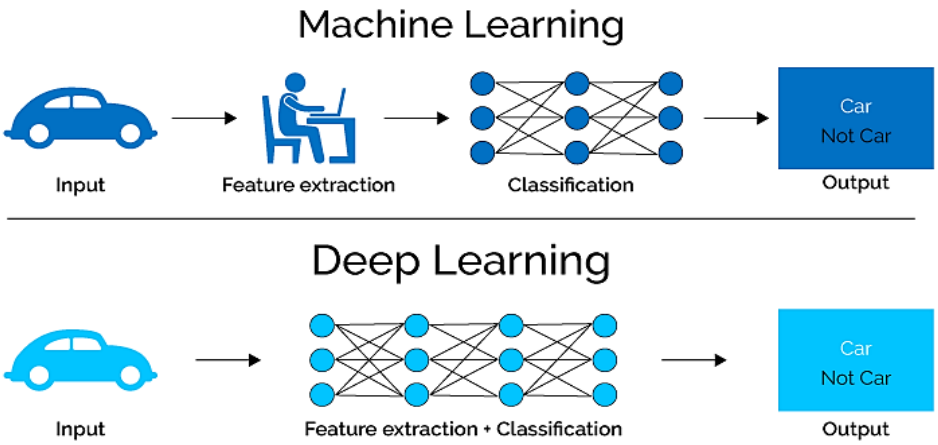


Figure 2. 1 : Difference entre le Machine Learning et le Deep Learning**.** [13]

2.2.1. L'Apprentissage Supervisé

L'apprentissage supervisé est le paradigme d'apprentissage le plus populaire en apprentissage machine et en apprentissage profond. Comme son nom l'indique, cela implique de superviser l'apprentissage automatique en lui montrant des exemples (données) de la tâche qu'il doit effectuer. [14] Les applications sont nombreuses : vision par ordinateur, régressions, classifications… La grande majorité des problèmes d'apprentissage automatique et d'apprentissage profond utilisent l'apprentissage supervisé.

2.2.2 L'Apprentissage Non Supervisé

L'apprentissage non supervisé est le mieux adapté lorsque le problème nécessite une quantité massive de données non étiquetées. Par exemple, les applications de médias sociaux, comme Twitter, Snap Chat, et ainsi de suite, ont toutes de grandes quantités de données non étiquetées.

2.3. L`Apprentissage Profond (Deep Learning)

Deep Learning est une approche d'apprentissage automatique, un sous-ensemble de l'intelligence artificielle, qui permet la construction d'un réseau de neurones pour permettre l'apprentissage automatique par lui- même. Par exemple, lorsqu'un ordinateur reconnaît du texte ou le visage d'une personne sur une image, il désassemble l'image et recherche des caractéristiques telles que la bouche, le visage, les cheveux, etc. « Avec les méthodes classiques ,la machine se contente de comparer des pixels. [15]

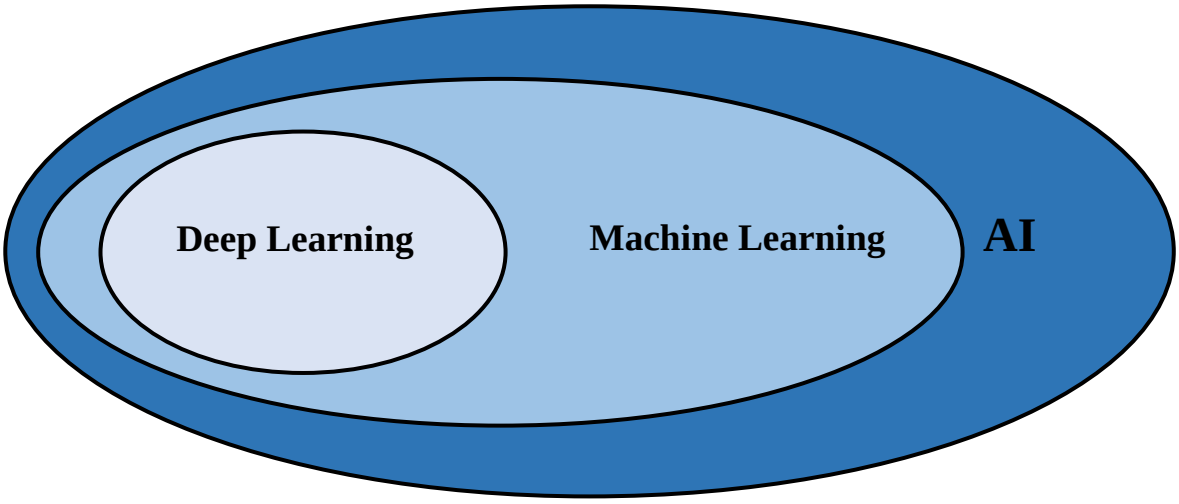


Figure 2.2 : Hiérarchie et position du Deep Learning par rapport au Machine Learning et IA.

2.4. Réseaux de Neurones

L'architecture d'un réseau de neurones est composée de différentes couches qui se connectent les unes aux autres pour traiter les informations. Voici une composition typique d'un réseau de neurones :

Couche d'entrée (Input Layer) : Cette couche reçoit les données d'entrée du réseau. Chaque neurone dans cette couche représente une caractéristique ou une variable d'entrée. Par exemple, dans le cas d'un réseau de neurones traitant des images, chaque neurone peut représenter un pixel.

Couches cachées (Hidden Layer) : Les couches cachées sont intermédiaires entre la couche d'entrée et la couche de sortie. Chaque couche cachée est composée de plusieurs neurones qui effectuent des calculs sur les entrées reçues et transmettent les résultats aux neurones de la couche suivante. Le nombre et la disposition des couches cachées dépendent de la complexité de la tâche à accomplir.

Couche de sortie (Output Layer) : Cette couche est responsable de la production des résultats du réseau de neurones. Chaque neurone dans cette couche représente une classe, une catégorie ou une valeur de sortie. Par exemple, dans le cas d'un réseau de neurones classifiant des images, chaque neurone peut représenter une classe spécifique (chien, chat, voiture, etc.). [16]

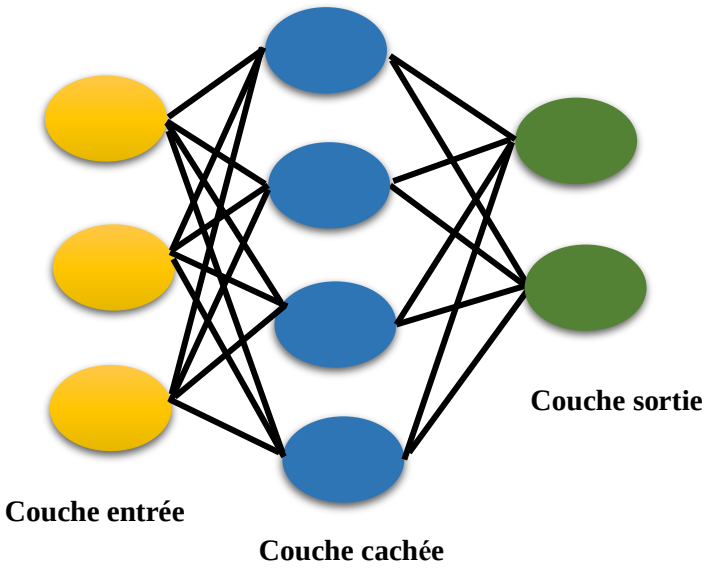


Figure 2.3 : L'architecture des réseaux neurones.

2.4.1. Perceptron

C'est un neurone binaire, c'est-à-dire dont la sortie est 0 ou 1. Pour calculer cette sortie, le neurone effectue une somme pondérée de ses entrées (chaque entrée a un poids)

$$Y = (W1X1 + W2X2 + W3X3) \quad \textbf{2.1}$$

Puis appliquez une fonction d'activation de seuil : si la valeur pondérée de somme dépasse une certaine valeur la sortie du neurone est 1, sinon elle est 0.Il ne peut donc pas faire seulement la classification, et alors la prédiction ne fait que la classification, alors la prédiction

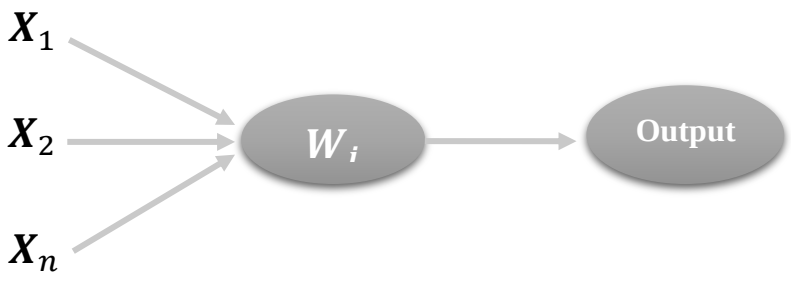


Figure 2.4 : Schéma du perceptron simple.

2.5. Vocabulaire des Réseaux Neurones

Nous avons présenté les généralités sur le Machine Learning afin d'introduire le Deep Learning. Cependant, il est primordial de connaître des concepts et vocabulaires de l'algorithme d'apprentissage pour une architecture d'un réseau de neurones, à travers ces vocabulaires, nous comprendrons ainsi la conception d'un apprentissage profond**.**

2.5.1. Fonction de Coût (Loss Function)

La fonction de coût est généralement dérivée de la fonction d'erreur. L'erreur quadratique moyenne (Mean Squared Error - MSE) est l'une des fonctions de coût couramment utilisées pour les problèmes de régression. Elle mesure la moyenne des carrés des différences entre les prédictions du modèle(Y_pred) et les valeurs réelles (Y) pour un ensemble de données d'entraînement.

La formule de l'erreur quadratique moyenne est la suivante [17]:

$$E = \frac{1}{2m}\sum_{i=1}^{m}(Y_{pred} - Y)^2 \qquad 2.2$$

Où m est le nombre d'exemples d'entraînement, Y_{pred} représente les prédictions du modèle pour chaque exemple, et Y sont les valeurs réelles correspondantes. La somme Σ est effectuée sur tous les exemples d'entraînement.

En réseau de neurones, la fonction coût est exprimée par la technique de Back propagation. [17]

2.5.2. La Descente de Gradient

La descente de gradient *(Gradient Descent en anglais)* est en effet un algorithme d'optimisation couramment utilisé dans l'apprentissage automatique pour minimiser la fonction de coût et trouver les valeurs optimales des poids synaptiques d'un modèle. Son objectif est de trouver le minimum global ou local de la fonction de coût en ajustant itérativement les poids du modèle.

L'algorithme de descente de gradient se base sur le calcul du gradient de la fonction de coût par rapport aux poids du modèle. Le gradient est une mesure de la pente de la fonction de coût et indique la direction dans laquelle les poids doivent être ajustés pour minimiser la fonction de coût**.** [18]

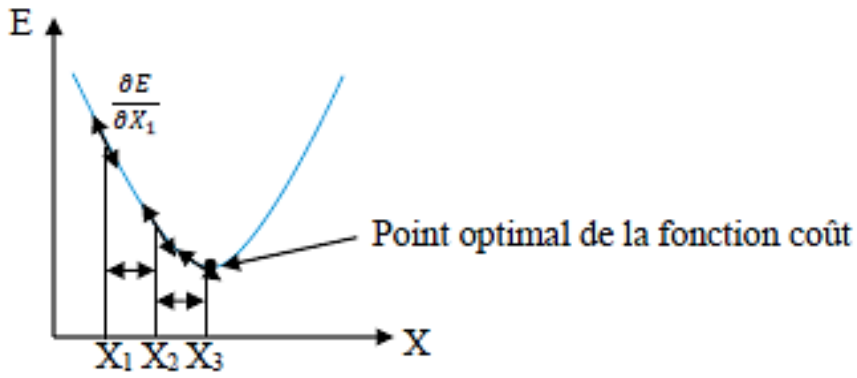


Figure 2.5 : Optimisation de la fonction coût par descente de gradient. [18].

2.5.3. Backpropagation

La mise à jour des poids et des biais dans un réseau de neurones se fait grâce à la Backpropagation. Après avoir calculé la sortie prédite et l'erreur , le gradient d'erreur est propagé en sens inverse à travers les couches du réseau en utilisant la chaîne de gradient. Cela permet de calculer les gradients des poids et des biais à chaque couche. En utilisant ces gradients, les poids et les biais sont itérativement mis à jour à l'aide d'algorithmes d'optimisation tels que la descente de gradient. Ce processus vise à minimiser la fonction de coût et à améliorer les performances du modèle en ajustant les paramètres du réseau de neurones.

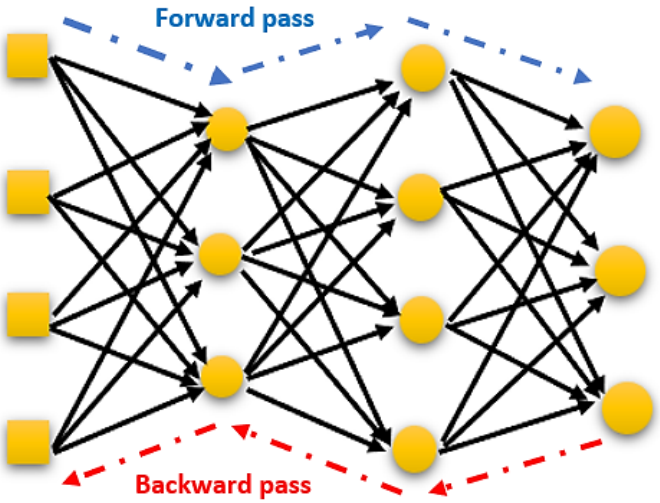


Figure 2.6 : Illustration de Backpropagation.

2.5.4. Pas d'Apprentissage (Learning Rate)

Le choix d'un pas d'apprentissage approprié est crucial pour éviter le sur-apprentissage (overfitting) dans votre étude visant à enlever le bruit des images. Lorsque le pas d'apprentissage est trop élevé, le modèle peut rapidement converger vers une solution, mais il risque de ne pas capturer correctement les caractéristiques pertinentes des données et de modéliser également le bruit présent dans les données d'entraînement. Cela peut entraîner des résultats de prédiction non corrects lorsqu'on applique le modèle à de nouvelles données.

Pour éviter le sur-apprentissage, il est recommandé d'utiliser des techniques de régularisation telles que la régularisation L1 ou L2, qui ajoutent une pénalité aux paramètres du

modèle pour limiter leur complexité. Ces techniques favorisent des modèles plus simples et réduisent ainsi les risques de sur-apprentissage.

2.5.5. Batches

Lors de l'entraînement d'un réseau neurone, les données d'entraînement sont généralement divisées en mini-batches. Chaque mini-batch contient un nombre spécifié d'échantillons, appelé batch size. Ces mini-batches sont ensuite utilisés pour mettre à jour les paramètres du modèle lors de la Backpropagation. La figure suivante montre un bloc de données à apprendre dans lequel un échantillon de données correspond à un batch [17]:

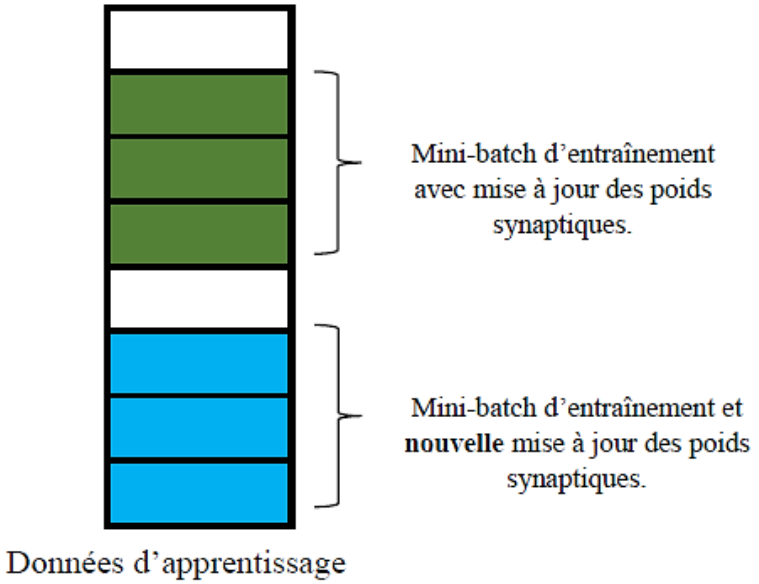


Figure 2.7 : Exemple de mini-batch de données d'apprentissage.

2.5.6. Epochs

Dans l'apprentissage d'un réseau neurone, un epoch représente un cycle complet de propagation avant (feed-forward) et de rétro propagation (back propagation) à travers le réseau. Cela implique le traitement de l'ensemble des données d'entraînement.

Supposons que vous disposez de 500 images dans votre ensemble d'entraînement et que vous choisissez un batch size (taille du lot) de 25 images. Dans ce cas, chaque epoch consistera en 20 itérations (500 images / 25 images par lot). Lors de chaque itération, vous passerez un lot de 25 images dans le réseau, effectuerez la propagation avant pour obtenir les prédictions,

calculerez la fonction de perte et utiliserez la rétro propagation pour mettre à jour les paramètres du réseau.

Après avoir parcouru les 20 itérations (ou mini-batchs) correspondant à un epoch complet, vous aurez effectué 20 mises à jour des paramètres du réseau en utilisant l'algorithme d'optimisation choisi (par exemple, la descente de gradient stochastique). [17]

2.6. Réseaux de Neurones Convolutifs (Convolutional Neural Network CNN)

Les réseaux de neurones convolutifs se démarquent des autres types de réseaux neuronaux en raison de leurs performances supérieures lorsqu'ils sont utilisés avec des entrées telles que des images, de la parole ou des signaux audio. Ils comprennent plusieurs types de couches clés, notamment :

- Couches de convolutions
- couches de pooling (Layer of POOLING)
- couches entièrement connectées (FULLY CONNECTED Layer)

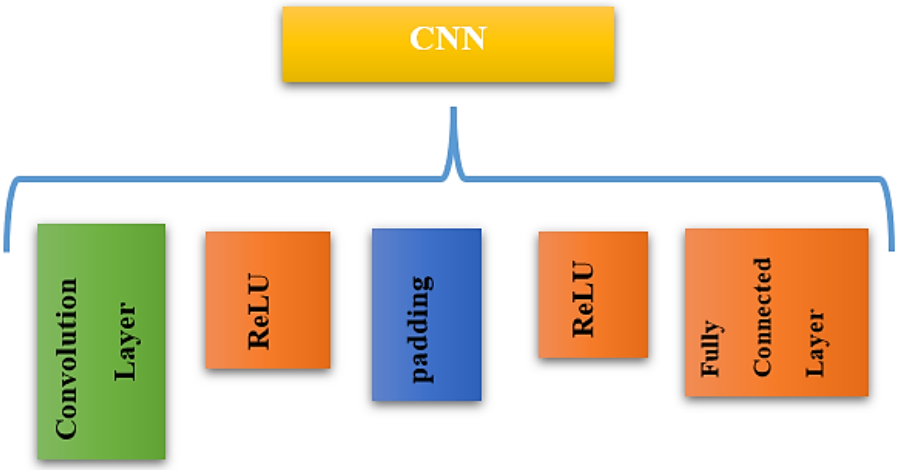


Figure 2.8 : L'architecture principale de CNN.

2.6.1. Couches de convolutions

L'entrée sera une image en couleur, qui est composée d'une matrice de pixels en 3D. Cela signifie que l'entrée aura trois dimensions : hauteur, largeur et profondeur, qui correspondent aux composantes RVB d'une image. qui se déplace à travers les champs récepteurs de l'image afin d'extraire les caractéristiques les plus pertinentes. Ce processus est connu sous le nom de convolution. [19]

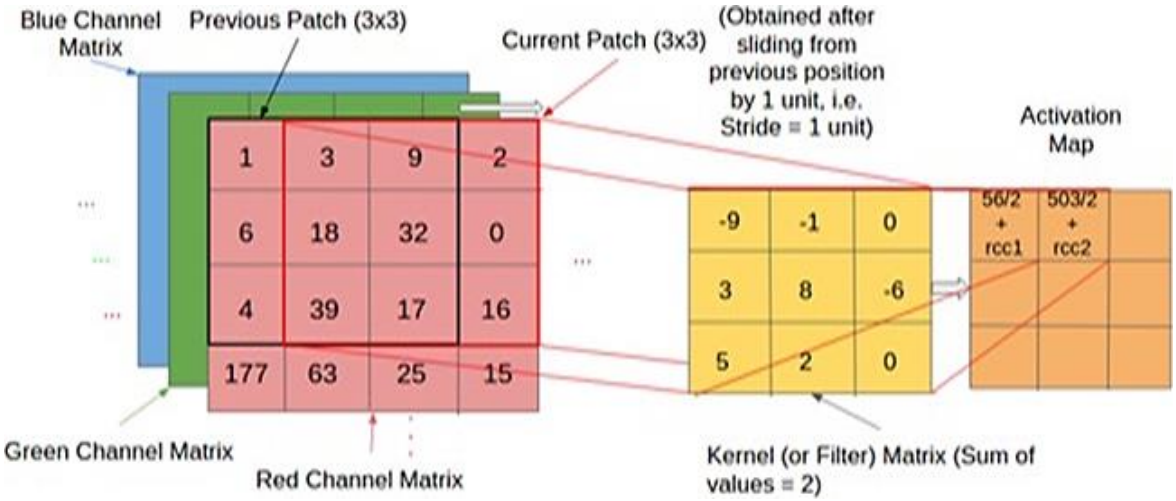


Figure 2.9 : Illustration d'une opération de Convolution sur une région d'image couleur**.**

2.6.2. Détecteur de Caractéristiques (Feature Detector)

Est un tableau bidimensionnel (2D) de poids qui représente une partie de l'image. Bien qu'ils puissent varier en taille, la taille du filtre est généralement une matrice de $3x3$, ce qui détermine également la taille du champ récepteur.

Le padding (remplissage) est utilisé pour gérer les dimensions de l'image lors de la convolution dans un réseau de neurones convolutifs (CNN). Voici les différents types de padding :

- **Valid padding (remplissage valide) :** C'est également connu sous le nom de "no padding" (aucun remplissage). Dans ce cas, si les dimensions ne s'alignent pas, la dernière convolution est abandonnée.
- **Same padding (remplissage identique) :** Ce type de padding garantit que la couche de sortie à la même taille que la couche d'entrée. Cela est généralement réalisé en ajoutant

des zéros autour des bords de l'entrée avant d'appliquer la convolution.

- Full padding (remplissage complet) : Ce type de padding augmente la taille de la sortie en ajoutant des zéros à la bordure de l'entrée.

Un CNN ajoute un ajustement Rectified Linear Unit (ReLU) à la carte des caractéristiques(feature map) après chaque opération de convolution, ce qui introduit de la non-linéarité dans le modèle. La fonction ReLU est une fonction d'activation couramment utilisée dans les réseaux de neurones qui permet de passer les valeurs positives sans les modifier et de mettre à zéro les valeurs négatives. Cela permet d'introduire une non-linéarité dans le modèle et d'améliorer sa capacité à apprendre des motifs complexes à partir des données.

2.6.3. Couches de pooling (Layer of POOLING)

Sont utilisées pour réduire les dimensions des cartes d'entités. Ainsi, il réduit le nombre de paramètres à apprendre et la quantité de calcul effectuée dans le réseau.

- **Max Pooling**

Est une opération de regroupement qui sélectionne le maximum d'éléments dans la région de la carte des entités couverte par le filtre. Ainsi, la sortie après la couche de regroupement maximal serait une carte d'entités contenant les caractéristiques les plus importantes de la carte d'entités précédente. [20]

- **Average Pooling**

Est une opération de pooling qui calcule la valeur moyenne des patchs d'une carte d'entités et l'utilise pour créer une carte d'entités sous-échantillonnée (regroupée).Il est généralement utilisé après une couche convolutive. [21]

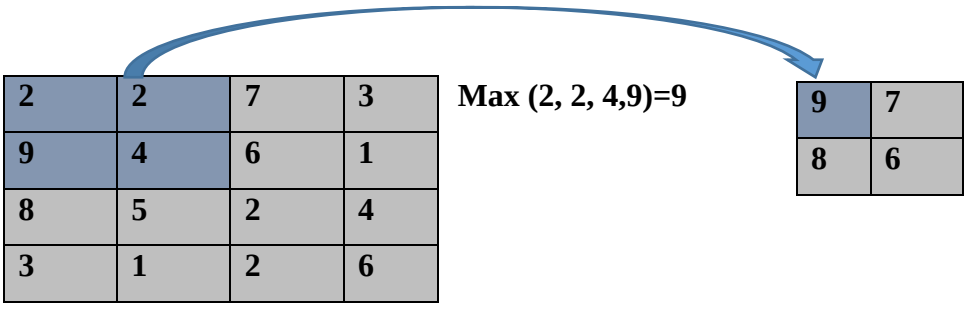


Figure 2.10 : Illustration de l'opération de Max Pooling.

2.6.4. Couches entièrement connectées (FULLY CONNECTED Layer)

Dans les couches partiellement connectées, les valeurs des pixels de l'image d'entrée ne sont pas directement connectées à la couche de sortie, comme cela a été mentionné précédemment. En revanche, chaque nœud de la couche de sortie est directement connecté à un nœud de la couche précédente dans la couche entièrement connectée. Cette couche effectue des tâches de classification en se basant sur les caractéristiques extraites par les couches précédentes et leurs différents filtres.

Alors que les couches de convolution et de regroupement utilisent généralement des fonctions ReLU pour catégoriser les entrées, les couches entièrement connectées utilisent généralement une fonction d'activation soft max pour produire une probabilité de 0 à 1.

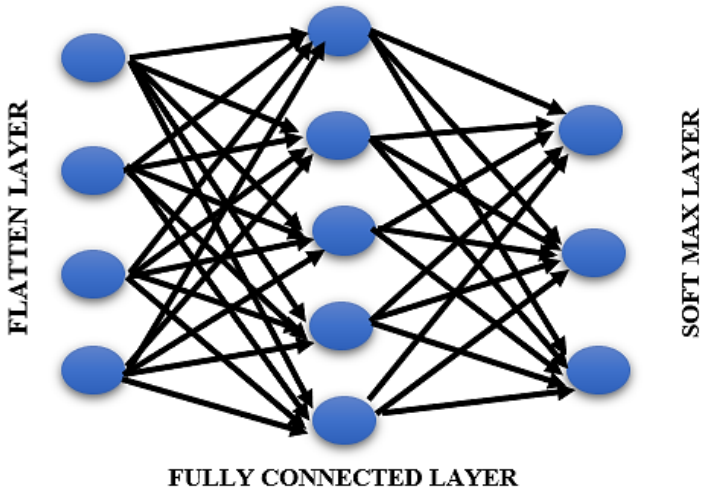


Figure 2.11 : FULLY CONNECTED Layer.

2.7. Principe de Fonctionnement des Réseaux de Neurones Convolutionnels

Les éléments de base d'un CNN sont des couches convolutives. Ces couches Appliquent un ensemble de filtres (également appelés noyaux) à l'image d'entrée, chacun d'entre eux extrayant une caractéristique ou un motif spécifique. La sortie de chaque filtre est une carte d'entités, qui met en évidence la présence de cette entité dans l'image d'entrée. En appliquant

plusieurs filtres, le CNN peut apprendre à reconnaître différents motifs et caractéristiques dans l'image.

Après les couches convolutionnelles, la sortie passe par une fonction d'activation non linéaire (généralement ReLU) pour introduire la non-linéarité dans le modèle. Ensuite, une couche de regroupement est appliquée pour sous-échantillonner les cartes d'entités de sortie et réduire les dimensions spatiales des données, tout en préservant les caractéristiques importantes. Cela permet de réduire le nombre de paramètres dans le modèle et d'éviter le surajustement.

Le processus d'application de couches convolutives et de regroupement de couches est répété plusieurs fois, créant un réseau neuronal profond capable d'apprendre des représentations complexes de l'image d'entrée.

Enfin, la sortie des couches convolutionnelles est aplatie et passée à travers une ou plusieurs couches entièrement connectées, qui effectuent la tâche finale de classification ou de régression.

2.8. Conclusion

L'apprentissage profond est un sous-ensemble de l'apprentissage automatique qui utilise des réseaux de neurones multicouches pour apprendre des représentations hiérarchiques des données. L'apprentissage profond a connu un succès remarquable dans un large éventail d'applications, notamment la vision par ordinateur, le traitement du langage naturel, la reconnaissance vocale et la robotique.

Dans ce chapitre on a présenté les notions importantes qui sont en relation avec l'apprentissage profond, on a parlé aussi sur les réseaux de neurones et réseau de neurones convolutifs (CNN).

Le prochain chapitre sera consacré aux résultats de notre travail, où nous effectuerons une étude comparative entre les méthodes conventionnelles et les méthodes basées sur l'apprentissage profond dans le domaine de la super-résolution.

Chapitre 3

Implementation d'algorithmes de super résolution d'image basés sur le Deep Learning

Résumé

Ce chapitre est consacré à la présentation et à la discussion de nos résultats de simulation. Nous mettrons en œuvre les algorithmes SR basés sur l'apprentissage profond : SRCNN et CAR. Une étude comparative quantitative est menée en termes de PSNR et SSIM

Sommaire

3.1. Introduction

Ce chapitre est entièrement dédié à la présentation de nos résultats de simulation. Plusieurs expériences ont été menées pour faire apparaitre l'intérêt d'utiliser les réseaux deep Learning en super résolution d'images. Nous avons commencé par l'implémentation des méthodes conventionnelles SR, à savoir : bicubique, bilinéaire et le plus proche voisin (nearest neighbor). Pour toutes les expériences menées, les images test sont considérées haute résolution HR et sont disponibles. Dans le cas des méthodes conventionnelles, nous commençons d'abord par la génération des images basse résolution (Low résolution LR). Rappelons pour mémoire que la dégradation d'image est introduite par le sous échantillonnage uniquement : le bruit et le flou ne sont pas considéré dans notre étude. Dans un souci de comparaison avec les travaux déjà existants, les valeurs du facteur de sous échantillonnage que nous avons pris sont: $\times 2 \quad et \quad \times 4$. Nous avons ensuite, appliqué les réseaux DL : SRCNN et CAR sur des images test tirées de différentes bases de données, plus précisemment, les bases usuelles dans la littérature . Une étude comparative quantitative et qualitative est effectuée pour chaque image test en termes de PSNR et SSIM en plus de la qualité visuelle des images obtenues.

3.2. Ensembles de données utilisés

Dans nos expériences de simulation, nous avons utilisé de nombreux ensembles de données, notamment : **LIVE1** qui contient 29 images de différents contenus , DIV2K [23] largement utilisée avec 1000 images de haute qualité.

Rappelons pour mémoire que, dans le domaine du deep learning, il est nécessaire de diviser l'ensemble de données en données d'entraînement, de test et de validation. Dans l'ensemble de données **DIV2K**, nous avons utilisé 800 images pour l'entraînement du réseau, 100 images pour la validation et les 100 images restantes ont été dédiées au test.

Nous avons également utilisé deux autres ensembles de données, Set5 [24] et Set14[25], pour le test. Le tableau 3.1 résume l'ensemble de données utilisées.

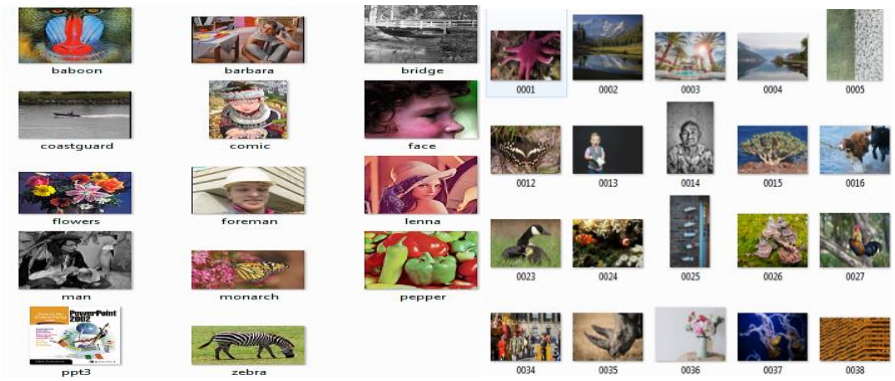
Tableau 3.1 : Données utilisées

Bases	Format	Number	Type	Résolution
DIV2K	PNG	1000	RGB	2.793 , 250
LIVE1	BMP	29	RGB	666 , 548
Set14	PNG	14	RGB	491.5 , 445.5
Set5	PNG	5	RGB	313 , 336

La Figure 3.1 illustre les images test des différents bases de données correspondante au tableau 3.1.

Set 5

LIVE1



Set14 **DIV2K**

Figure 3.1: Images haute résolution (HR) originaels.

3.3. Étapes d'implémentation

3.3.1. Software utilisé

Dans notre travail, nous avons exécuté nos codes sur la plateforme Google Colab en utilisant le langage Python. L'environnement de Colab est basé sur Jupyter Notebook, ce qui signifie que les utilisateurs peuvent écrire et exécuter du code Python par blocs dans un navigateur web, ce qui facilite le partage de projets, la collaboration et l'expérimentation rapide.

3.3.2. Étape de Simulation

3.3.2. A. Méthodes Conventionnelles

Sur la Figure 3.2, l'organigramme illustre les étapes d'implémentation des méthodes conventionnelles de super-résolution SR .

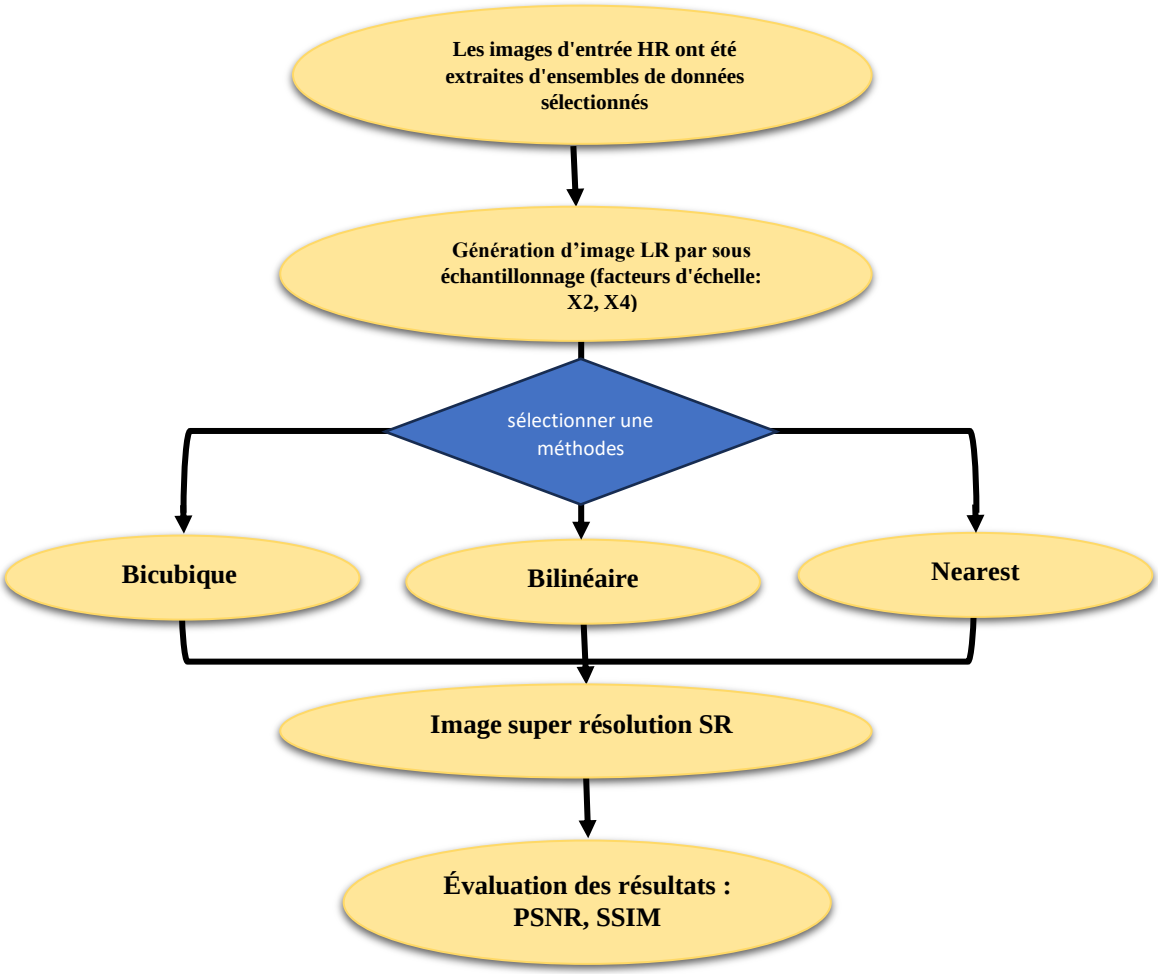


Figure 3.2: Étapes d'implémentation des méthodes conventionnelles SR.

Dans ce qui suit, nous détaillerons les étapes de simulation pour les algorithmes SR à base de Deep Learning que nous avons considéré dans notre étude.

3.3.2. B. SRCNN

Par analogie aux algorithmes SR conventionnels, le SRCNN (Super-Resolution Convolutional Neural Network) suit une approche similaire en commençant par générer des images de basse résolution (LR) à partir des images haute résolution (HR) en entrée. Son workflow comme c'est montré sur la Figure 3.3, se compose de trois parties principales: l'extraction et la représentation des patchs, la transformation non linéaire et la reconstruction de l'image.

1. **Extraction et représentation des patchs :** Cette étape correspond à la première couche du SRCNN. Elle extrait des patchs de l'image d'entrée à basse résolution. L'opération de cette couche consiste à appliquer une interpolation bicubique à l'image d'entrée pour obtenir une version à basse résolution, puis à extraire des patchs de taille 33x33 pixels.

   Les couches suivantes peuvent comporter des filtres avec des nombres spécifiques, tels que 64, 32 ou 1. Cependant, il est important de préciser que la configuration exacte du SRCNN peut varier en fonction de l'architecture spécifique utilisée.

2. **Transformation non linéaire :** Cette étape fait référence à la couche intermédiaire du SRCNN. Elle applique une transformation non linéaire aux vecteurs de caractéristiques afin de les mapper vers un autre ensemble de vecteurs de caractéristiques. La fonction ReLU est utilisée pour introduire la non-linéarité dans le réseau. L'opération de cette couche consiste à appliquer des filtres de taille 9x9 à chaque patch extrait, suivis d'une fonction d'activation ReLU.

3. **Reconstruction :** Cette étape combine les fonctionnalités à haute résolution pour générer l'image finale à haute résolution HR. Elle correspond à la dernière couche du SRCNN. L'opération de cette couche consiste à appliquer des filtres de taille 5x5 aux caractéristiques extraites par la couche intermédiaire, suivis d'une opération de reconstruction pour obtenir l'image finale à haute résolution.

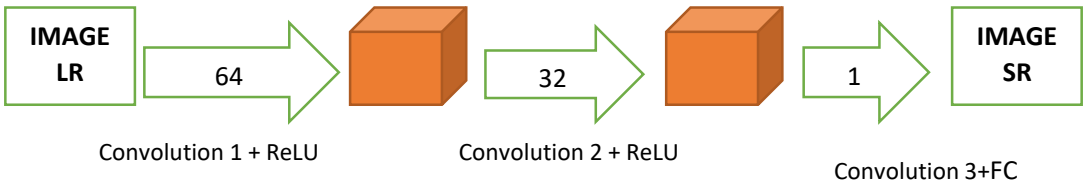


Figure 3.3 :workflow du SRCNN

3.3.2. C. CAR

Pour ce réseau, l'étape de génération des images LR est inclue automatiquement dans le réseau, en d'autres terme, les images d'entrée du réseau sont en haute résolution (HR) et ne subbissent aucun sous échantillonnage. Comme nous l'avons décrit dans le chapitre 1, l'architecture du modèle "content-adaptive resampled" (CAR) se compose d'un bloc ResamplerNet et d'un bloc SRNet. Le framework est composé de deux composants principaux:

ResamplerNet : Il est chargé de prédire les noyaux de rééchantillonnage adaptatifs en fonction de l'image HR d'entrée, puis d'appliquer ces noyaux de rééchantillonnage à l'image HR d'entrée pour produire l'image réduite en taille.

SRNet : Ce bloc du réseau est responsable de la reconstruction d'image SR à partir de l'image réduite à son entrée. La Figure 3.4 résume les blocs essentiels de CAR.Quant au nombre de filtres utilisés dans le bloc SRNet, il peut également varier selon la configuration spécifique du réseau. En général, des architectures profondes peuvent utiliser un nombre plus élevé de filtres pour capturer des caractéristiques complexes à différentes échelles. Des valeurs courantes pour le nombre de filtres dans les couches de convolution peuvent être de l'ordre de 64, 128, 256 ou plus, en fonction de la complexité et de la capacité du modèle.

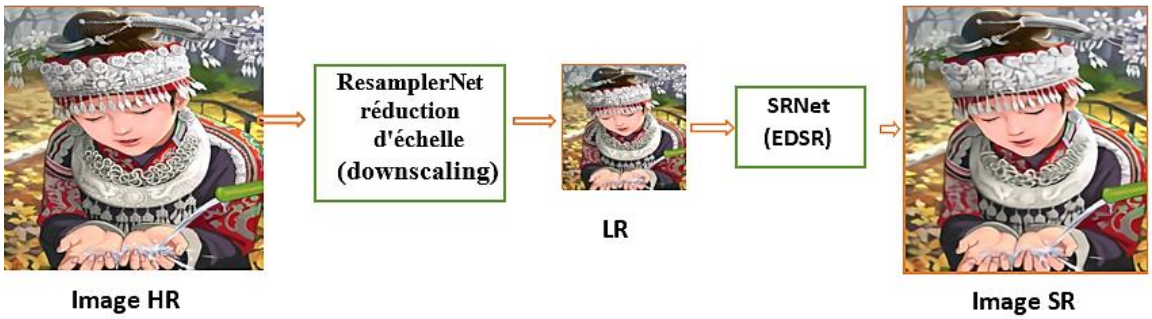


Figure 3.4: Architecture du réseau CAR

3.4. Résultat et Discussion

3.4.1. Méthodes Conventionnelles SR

Pour les algorithmes bicubique, bilinéaire et le plus proche voisin, nous commençons par générer une image LR basse résolution en utilisant une étape de sous-échantillonnage avec un facteur d'échelle de 2 ou 4.

Rappelons pour mémoire que seul le sous échantillonage est pris en considération en tant que facteur de détérioration. le flou et le bruit ne sont pas pris en compte dans notre étude.

Au cours de ce travail, nous avons mené plusieurs expériences en raison de la disponibilité des bases de données. Cependant, en raison de la contrainte du nombre de pages imposée, nous avons choisi de sélectionner une partie des résultats obtenus. Pour évaluer les critères PSNR et SSIM, nous avons calculé les valeurs moyennes après avoir déterminé celles de chaque image individuellement.

Le tableau 3.2 résume les valeurs moyennes de PSNR (dB) et de SSIM pour les données Set5, lors de l'application des méthodes conventionnelles. Les images LR et SR correspondantes sont illustrées sur les Figure 3.5_3.7.

Tableau 3.2: PSNR (dB), SSIM pour Set5, méthodes conventionnelle.

Méthodes	**Bicubique**				**Bilinéaire**				**Nearest neighbor**			
Facteurs	**X2**		**X4**		**X2**		**X4**		**X2**		**X4**	
Métriques	**PSNR**	**SSIM**	**PSNR**	**SSIM**	**PSNR**	**SSIM**	**PSNR**	**SSIM**	**PSNR**	**SSIM**	**PSNR**	**SSIM**
butterfly	32,13	0 ,91	30,12	0,72	32,03	0,88	30,47	0,72	32,72	0,87	31,55	0,65
baby	37.00	0.95	33.49	0.86	35.71	0.93	33.24	0.85	35.11	0.93	32.79	0.80

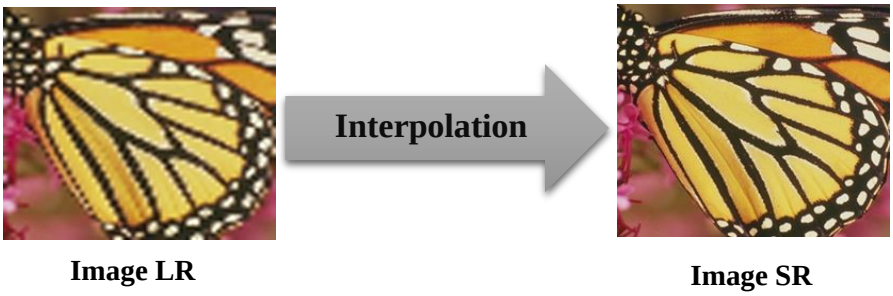


Figure 3.5: L'image HR après l'application de la méthode Bicubique sur l'image LR.

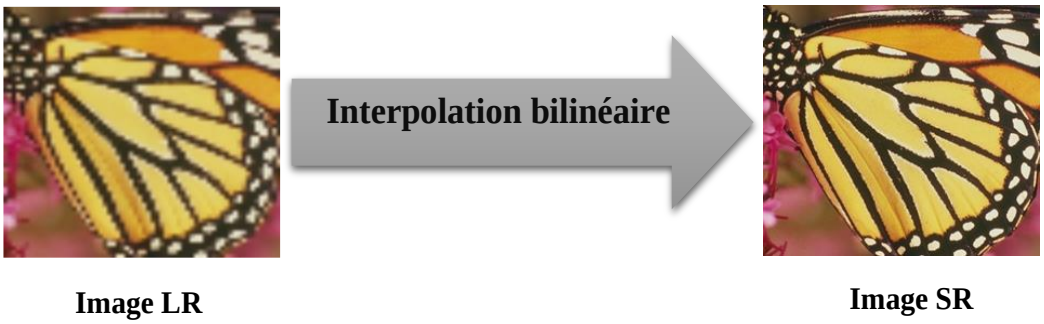


Figure 3.6: L'image HR après l'application de la méthode Bilinéaire sur l'image LR.

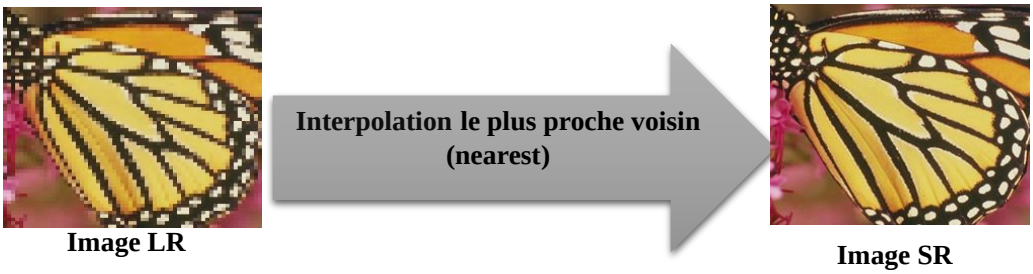


Figure 3.7: L'image HR après l'application de la méthode Nearest Neighbor sur l'image LR.

Nous remarquons quel les résultats d'interpolations sont très proches en terme de qualité visuelle. Bicubique surpasse légérement les deux autres méthodes, en terme de PSNR.

3.4.2. Implémentation des Algorithmes SR

3.4.2. A. SRCNN

Nous avons calculé le PSNR et le SSIM de chaque image SR obtenue. Puis, nous avons calculé la valeur moyenne. Les tableaux 3.3 résume les valeurs de PSNR et SSIM obtenus en appliquant la méthode SRCNN.

Tableau 3.3 : Les valeurs de PSNR et de SSIM des données Set5, Set14 et DIV2K (SRCNN).

- **Set5**

Nom	**X2**		**X4**	
	PSNR	**SSIM**	**PSNR**	**SSIM**
baby	38.46	0.9658	28.86	0.8620
brid	41.37	0.9873	26.53	0.8403
butterfly	32.80	0.9657	19.34	0.6710
head	35.72	0.8869	29.06	0.7717
woman	35.27	0.9704	22.92	0.8099

- **Set14**

Nom	**X2**		**X4**	
	PSNR	**SSIM**	**PSNR**	**SSIM**
bridge	25.70	0.7691	20.55	0.5847
foreman	28.52	0.8766	22.92	0.7203
face	29.12	0.8582	22.95	0.6658
man	30.45	0.8425	24.59	0.6083
Barbara	28.52	0.9167	19.11	0.6528
comic	35.70	0.8855	29.09	0.7708
coastguard	33.23	0.9372	23.09	0.7422
monarch	36.52	0.9693	24.21	0.8052
pepper	36.63	0.9298	26.15	0.8080
Lenna	30.97	0.8882	23.53	0.7157
zebra	37.73	0.9783	25.09	0.8697
flowers	37.02	0.9219	28.17	0.8234
ppt3	30.49	0.9761	21.86	0.8067
baboon	33.57	0.9431	22.10	0.7388

- **DIV2K**

Nom	**X2**		**X4**	
	PSNR	**SSIM**	**PSNR**	**SSIM**
0001	40.30	0.9730	27.74	0.8362
0002	28.50	0.8468	23.89	0.7102
0003	30.26	0.9113	22.40	0.7317
0004	35.40	0.9582	29.64	0.9034
0005	34.41	0.9545	24.65	0.7970

Le Tableau 3.4 résume les valeurs de PSNRmean et SSIMmean obtenus en appliquant SRCNN

Tableau 3.4 : PSNRmean (dB), SSIMmean Set5, et LIVE1 (SR) SRCNN

Algorithme	Données	SCAL	PSNRmean	SSIMmean
SRCNN	**DIV2K**	**X2**	34.49	0.9358
		X4	26.03	0.8115
	Set14	**X2**	32.44	0.9066
		X4	23.81	0.7366
	Set5	**X2**	36.72	0.9552
		X4	25.34	0.7910

L'image~~s~~ LR générées et l'image~~s~~ SR obtenue~~s~~ par SRCNN correspondantes aux tableaux 3.3 est illustrée sur la Figure 3.8.

Figure 3.8 : L' images SR (à droite) suite à l'application de SRCNN sur L'image LR (à gauche).

3.4.2. B. CAR

Pour chaque image SR obtenue, nous avons calculé les valeurs de PSNR et de SSIM, puis déterminé leur moyenne. Le tableau 3.5 résume les valeurs de PSNR et SSIM obtenus en appliquant CAR.

Tableau 3.5 :PSNR et SSIM, Set5,Set14,LIVE1 (CAR)

- **Set14**

Nom	**X2**		**X4**	
	PSNR	**SSIM**	**PSNR**	**SSIM**
bridge	30.34	0.88	26.39	0.69
foreman	40.05	0.97	35.85	0.94
face	36.30	0.89	33.78	0.82
man	32.80	0.91	28.53	0.79
Barbara	36.67	0.96	27.94	0.84
comic	31.88	0.96	25.51	0.82
coastguard	35.34	0.94	29.30	0.77
monarch	41.34	0.98	35.47	0.95
pepper	38.03	0.92	35.43	0.89
Lenna	38.29	0.94	34.22	0.88
zebra	35.94	0.95	30.61	0.85
flowers	36.74	0.96	30.33	0.86
ppt3	40.08	0.99	30.70	0.97
baboon	27.95	0.84	24.38	0.65

- **Set5**

Nom	**X2**		**X4**	
	PSNR	**SSIM**	**PSNR**	**SSIM**
baby	39.64	0.97	35.27	0.92
brid	44.51	0.99	37.96	0.96
butterfly	36.85	0.98	30.95	0.94
head	36.27	0.89	33.79	0.82
woman	37.50	0.97	32.84	0.94

- **LIVE1**

Nom	**X2**		**X4**	
	PSNR	**SSIM**	**PSNR**	**SSIM**
woman	31.95	0.92	28.01	0.79
house	35.76	0.93	31.52	0.83
statue	36.99	0.96	31.31	0.89
rapids	34.91	0.93	29.70	0.80
coinsinfountain	35.09	0.94	29.92	0.83
studentsculpture	30.22	0.92	25.01	0.73
churchandcapitol	34.23	0.95	28.29	0.86
lighthouse2	33.69	0.94	28.95	0.84
bikes	34.52	0.95	27.75	0.82
flowersonih35	27.70	0.92	23.06	0.75
paintedhouse	31.87	0.94	27.56	0.83
womanhat	37.43	0.94	33.34	0.86
sailing2	38.87	0.96	33.57	0.91
building2	28.75	0.91	23.67	0.73
ocean	37.02	0.94	32.04	0.84
buildings	32.65	0.94	26.11	0.81
lighthouse3	35.76	0.93	30.47	0.84
parrots	41.75	0.97	35.73	0.94
plane	37.90	0.95	33.19	0.90
manfishing	36.07	0.96	29.91	0.85
sailing4	35.18	0.94	30.45	0.83

carnivaldolls	36.44	0.97	30.42	0.88
caps	40.96	0.97	35.45	0.91
sailing1	33.55	0.93	28.38	0.79
stream	27.89	0.85	23.97	0.64
cemetry	32.92	0.93	27.43	0.78
monarch	41.34	0.98	35.47	0.95
dancers	31.02	0.94	25.90	0.83
sailing3	39.43	0.96	33.98	0.91

En général, on a trouvé le PSNR qui considéré comme acceptable supérieure à 30db,le SSIM élevé est proche le 1 Par rapport X4 le PSNR moins de 30db et le SSIM moins de 0.90.

Le Tableau 3.6 résume les valeurs PSNRmean et SSIMmean obtenus en appliquant CAR

Tableau 3.6: PSNRmean (dB), SSIMmean Set5, et LIVE1 (SR) algorithme CAR

Algorithme	Données	SCAL	PSNRmean	SSIMmean
CAR	LIVE1	X2	34,90	0,9446
		X4	29.71	0.8393
	Set14	X2	35.84	0.9394
		X4	30.61	0.8427
	Set5	X2	38.96	0.9643
		X4	34.17	0.9196

La Figure 3.8 illustre les images LR générées (à gauche) ainsi que les images SR (à droite) correspondantes obtenues par la méthode CAR, telles que présentées dans le tableau 3.3.

- **Set5 (X2)**

- **Set5 (X4)**




- **Set14 (X2)**





- **Set14 (X4)**






- **LIVE1 (X2)**






- **LIVE1 (X4)**


Figure 3.9 : L' images SR (à droite) suite à l'application de CAR sur L'image LR (à gauche).

3.5. Synthèse

Après avoir examiné tous les résultats obtenus à partir des algorithmes conventionnels ainsi que des algorithmes basés sur l'apprentissage profond, nous avons observé :

- Pour les algorithmes conventionnels, les meilleures valeurs de PSNR et SSIM sont obtenus par la méthode Bicubique, tel que PSNR Bicubique est supérieur par environ 1 dB et le SSIM par une valeur de 0.5 par rapport aux méthodes du Plus Proche Voisin et Bilinéaire. Cette remarque a tendance à se reproduire pour toutes les images.

- Lors de l'évaluation des algorithmes SRCNN et CAR, nous avons remarqué une amélioration notable du PSNR d'environ 2 dB avec CAR par rapport à SRCNN. Cette amélioration est soutenue par la qualité visuelle supérieure des images SR produites par CAR. CAR s'est avéré être le meilleur choix en termes de SSIM , surpassant ainsi les résultats obtenus par SRCNN.

- Les algorithmes CAR et SRCNN surpassent largement les méthodes conventionnelles en termes de PSNR, de SSIM et de qualité visuelle. Leurs performances sont nettement supérieures, offrant des résultats plus précis, plus cohérents et visuellement plus satisfaisants.

3.6. Conclusion

Ce chapitre, représente une description générale de la mise en oeuvre de notre travail. Nous avons effectué plusieurs expériences de simulation en comparant les méthodes conventionnelles de super-résolution (Bicubique, Bilinéaire et Plus Proche Voisin) avec les approches basées sur les réseaux neuronaux profonds (SRCNN et CAR). À chaque fois, nous avons généré des images à basse résolution (LR) à partir d'images haute résolution (HR) disponibles.

Dans l'ensemble des résultats obtenus, nous avons observé ce qui suit :

Les méthodes conventionnelles, en particulier la méthode Bicubique, ont donné les meilleures valeurs de PSNR et SSIM par rapport aux méthodes Bilinéaire et Plus Proche Voisin. Ces résultats se sont généralement reproduits pour toutes les images testées.

Les réseaux neuronaux profonds, notamment l'approche CAR, ont montré une amélioration significative du PSNR (environ 2 dB de plus) par rapport à l'approche SRCNN. Cette amélioration est également confirmée par la qualité visuelle des images super-résolues obtenues. De plus, l'approche CAR a également surpassé SRCNN en termes de SSIM.

Dans l'ensemble, les approches basées sur les réseaux neuronaux profonds (CAR et SRCNN) ont nettement surpassé les méthodes conventionnelles en termes de PSNR, de SSIM et de qualité visuelle des images super-résolues.

En conclusion, nos résultats démontrent clairement l'avantage des réseaux neuronaux profonds, en particulier l'approche CAR, par rapport aux méthodes conventionnelles de super-résolution d'images. Ces approches ont permis d'obtenir des images super-résolues de meilleure qualité, avec des valeurs de PSNR et de SSIM plus élevées, ouvrant ainsi de nouvelles perspectives pour les applications de super-résolution d'images.

Conclusion générale

La finalité de ce travail est de reconstruire des images haute résolution (HR) à partir d'images basse résolution (LR). Ce mémoire aborde le domaine de la super-résolution d'image en mettant l'accent sur les algorithmes basés sur l'apprentissage profond.

Les résultats obtenus ont clairement démontré que les algorithmes de super-résolution basés sur l'apprentissage profond surpassent les méthodes conventionnelles d'interpolation. Ces algorithmes ont considérablement amélioré la qualité des images détériorées, comme en témoignent les valeurs élevées de PSNR et de SSIM. Cette étude confirme donc l'efficacité et le potentiel des approches de super-résolution basées sur l'apprentissage profond pour résoudre le problème de la récupération de détails et d'amélioration de la résolution des images.

Les avancées récentes dans le domaine de l'apprentissage profond ont ouvert de nouvelles perspectives dans le domaine de la super-résolution d'image. Les algorithmes basés sur le deep Learning permettent de réaliser des améliorations significatives en termes de qualité des images, ouvrant ainsi de nombreuses opportunités dans des domaines tels que la reconnaissance faciale, la surveillance vidéo, la vision par ordinateur et bien d'autres. Il reste encore des défis à relever, notamment en termes de performances et de temps de calcul, mais les résultats obtenus jusqu'à présent sont prometteurs et suggèrent un avenir passionnant pour la super-résolution d'image grâce à l'utilisation de l'apprentissage profond.

Bibliographie

[1] Goodfellow, I., Bengio, Y., & Courville, A. (2016). Deep Learning. MIT Press.

[2] Yang, C. Y., Chen, Y. C., & Wang, Y. C. F. (2014). Image super-resolution via sparse representation. IEEE Transactions on Image Processing, 23(2), 485-495.

[3] Dong, C., Loy, C. C., He, K., & Tang, X. (2014). Learning a deep convolutional network for image super-resolution. European Conference on Computer Vision, 184-199.

[4] Ledig, C., Theis, L., Huszár, F., Caballero, J., Cunningham, A., Acosta, A., ... & Shi, W. (2017). Photo-realistic single image super-resolution using a generative adversarial network. In Proceedings of the IEEE conference on computer vision and pattern recognition, 4681-4690.

[5] Yahiaoui, Mohamed Laid ,Les techniques de super-résolution SR appliquées en imagerie astronomique et nucléaire.

[6] Saadallah, Wissem, Apprentissage profond pour la super résolution des images .

[7] Legrand, F. (no date) *Interpolation Bilinéaire*, *Interpolation*. Available at: https://www.f-legrand.fr/scidoc/docmml/image/niveaux/interpolation/interpolation.html (Accessed: 26 May 2023).

[8] *File:interpolation-bicubic.svg* (no date) *Wikimedia Commons*. Available at: https://commons.wikimedia.org/wiki/File:Interpolation-bicubic.svg (Accessed: 26 May 2023).

[9] Chao Dong, Chen Change Loy, Kaiming He, Xiaoou Tang, Image Super-Resolution Using Deep Convolutional Networks , 31 Jul 2015.

[10] Sun, W., & Chen, Z**.** (2018). Learned Image Downscaling for Upscaling using Content Adaptive Resampler. In Proceedings of IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW).

[11] Zhou Wang, , Alan Conrad Bovik, , Hamid Rahim Sheikh, and Eero P. Simoncelli, Senior, 'Image Quality Assessment: From Error Visibility to Structural Similarit', IEEE TRANSACTIONS ON IMAGE PROCESSING, VOL. 13, NO. 4, APRIL 2004.

[12] Mohamed Abd Elmoumen DJABALLAH,Système de prédiction de la consommation d'énergie basé Deep Learning.

[13] Sperling, E. (2018) *Deep learning spreads*, *Semiconductor Engineering*. Available at: https://semiengineering.com/deep-learning-spreads (Accessed: 26 May 2023).

[14] Zhihao Wang, J. C. (2021). Deep Learning for Image Super-Resolution: A Survey.

[15] Annie Passalacqua, 10 termes à connaître pour parler de l'IA à vos collègues, 18 avril

2018.

[16] Neural Networks and Deep Learning" by Michael Nielsen.

[17] TERAI Salim ,Algorithme Hybride de Débruitage Image/Vidéo à base de Réseaux de Neurones Convolutionnels (CNN).

[18] Ian Goodfellow, Yoshua Bengio, Aaron Courville. "Deep Learning". MIT Press, 2016.

[19] Sharma, P. (2020). A Comprehensive Tutorial to learn Convolutional Neural.

[20] A. Romero and A. Romero, “Assisting the training of deep neural networks with applications to computer vision Assisting the training of deep neural networks with applications to computer vision.”University of Barcelona. 2015.

[21] CNN: Introduction to pooling layer (2023) GeeksforGeeks. Available at: https://www.geeksforgeeks.org/cnn-introduction-to-pooling-layer/ (Accessed: 26 May 2023).

[22] *Papers with code - average pooling explained* (no date) *Explained | Papers With Code*. (Accessed: 26 May 2023).

[23] Eirikur Agustsson and Radu Timofte. Ntire 2017 challengeon single image super-resolution: Dataset and study. In IEEE Conference on Computer Vision and Pattern Recognition Workshops, pages 1122–1131, 2017.

[24] Marco Bevilacqua, Aline Roumy, Christine Guillemot, and Marie-Line Alberi Morel. Low-complexity single-image super-resolution based on nonnegative neighbor embedding.In British Machine Vision Conference, 2012.

[25] Roman Zeyde, Michael Elad, and Matan Protter. On single image scale-up using sparse-representations. In Proceedings of the 7th International Conference on Curves and Surfaces, pages 711–730, 2012.

Résumés

La super résolution d'image (SRI) est un ensemble de techniques de traitement d'image utilisées pour améliorer la qualité des images dégradées. La dégradation peut être causée par un flou, un sous échantillonnage ou un bruit additif. Ces dernières années, les méthodes d'apprentissage en profondeur ont fait de grands progrès dans la super-résolution d'image. Notre travail s'inscrit dans le cadre d'algorithmes de super résolution d'images à base d'apprentissage profond. Plus exactement, nous implémenterons tout d'abord les techniques conventionnelles d'interpolation sur des images dégradées générées par un sous échantillonnage. Nous implémenterons ensuite les algorithmes de super résolution à base d'apprentissage profond (Deep Learning DL). Deux réseaux DL seront pris en considération : le réseau de neurone convolutif (SRCNN) et le réseau DL Content Adaptive Resampler (CAR).

Les algorithmes seront appliqués sur des images de bases de données usuelles. Les métriques considérées pour l'étude des performances des algorithmes sont le rapport du signal maximum sur bruit (Peak signal ti noise ratio PSNR) et l'indice de structure de de similarité (Sturcture similarity index (SSIM). Les résultats obtenus ont bien montré l'intérêt de solliciter le deep learning en super résolution.

Mots clés : super résolution d'image, l'apprentissage en profondeur, réseau neuronal convolutif, sous échantillonnage.

ملخص

دقة الصورة الفائقة(SRI) هي مجموعة من تقنيات معالجة الصور المستخدمة لتحسين جودة الصور المتدهورة. يمكن أن يكون التدهور نتيجة للضبابية أو التخفيض في التحليل أو إضافة الضوضاء. في السنوات الأخيرة، حققت طرق التعلم العميق تقدمًا كبيرًا في تقنية تحسين جودة الصور. يندرج عملنا ضمن إطار خوارزميات تحسين جودة الصور باستخدام التعلم العميق. بالتحديد، سنقوم أولاً بتنفيذ تقنيات التكبير التقليدية على الصور المتدهورة التي تم إنشاؤها بواسطة التخفيض في التحليل. بعد ذلك، سنقوم بتنفيذ خوارزميات تحسين جودة الصور باستخدام التعلم العميق (Deep Learning DL). سنأخذ في الاعتبار اثنين من الشبكات العميقة وهما Convolutional Neural Network (SRCNN) و Content Adaptive Resampler (CAR).

سيتم تطبيق الخوارزميات على صور من قواعد البيانات الشائعة. وتعد المقاييس المعتبرة لدراسة أداء الخوارزميات هي نسبة الإشارة إلى الضوضاء القصوى (Peak signal to noise ratio PSNR) ومؤشر تشابه الهيكل (Structure similarity index (SSIM)). أظهرت النتائج المحصلة فوائد استدعاء التعلم العميق في تقنية تحسين جودة الصور بوضوح.

الكلمات الرئيسية: تحسين جودة الصورة، التعلم العميق، الشبكة العصبية العميقة، التخفيض في التحلي

Abstracts

Image super-resolution (ISR) is a set of image processing techniques used to enhance the quality of degraded images. Degradation can be caused by blur, undersampling, or additive noise. In recent years, deep learning methods have made significant advancements in image super-resolution. Our work falls within the framework of deep learning-based image super-resolution algorithms. Specifically, we will first implement conventional interpolation techniques on degraded images generated by undersampling. We will then implement deep learning-based super-resolution algorithms, namely the Convolutional Neural Network (SRCNN) and the Content Adaptive Resampler (CAR) deep learning networks.

The algorithms will be applied to images from standard databases. The metrics considered for evaluating the algorithm's performance are the Peak Signal-to-Noise Ratio (PSNR) and the Structural Similarity Index (SSIM). The obtained results have clearly demonstrated the benefits of utilizing deep learning in super-resolution.

Keywords : image super-resolution, deep learning, convolutional neural network, undersampling.