



الجمهورية الجزائرية الديمقراطية الشعبية

Republique Algerienne Democratique Et Populaire

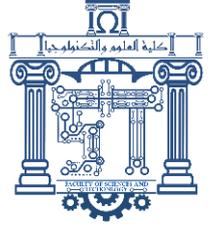
وزارة التعليم العالي والبحث العلمي

Ministère de l'Enseignement Supérieur et de la Recherche Scientifique

جامعة العربي التبسي - تبسة

Université Larbi Tebessi Faculté des sciences exactes
et sciences de la nature et de la vie

Département de Mathématiques et Informatique



MEMOIRE

Mémoire de fin d'étude en vue d'obtention d'un diplôme de Master en
Informatique

Domaine : Informatique

Spécialité : Systèmes et Multimédias

Apprentissage profond pour la super résolution des images

Présenté par :

Saadallah Wissem

Devant le jury :

Dr. Bendib Issam
Dr. Gherbi tahar

Président
Encadreur

Dr. DJeddai alaa
Dr. Zeggari ahmed

Examineur
Co-encadreur

Année universitaire : 2021/2022

Remerciements

Tout d'abord, Je remercie Allah qui m'a donné la force et la patience pour terminer ce travail dans le temps.

J'adresse aussi les sincères remerciements à : mes encadreurs, qui m'ont supporté dans les moments les plus difficiles pour terminer ce travail avec leur disponibilité et

leurs précieux conseils. Pour tous les enseignants du département de math et informatique et en particulier la

branche d'informatique. Pour tous les membres de la famille qui m'ont supporté moralement et pour leurs

prières sincères. Tous mes amis et camarades de classe.

J'exprime tout mon respect aux jurys qui feront l'honneur d'apprécier ce travail .

Dédicaces

Je dédie ce modeste travail à mes chers parents qui n'ont pas cessé de m'encourager durant toutes mes études et qui m'accompagnés dans toute ma vie, s'inquiétant énormément pour m'offrir une meilleure vie, Que Dieu les protège

A mes frères Aimen et Wassim,

Pour leurs soutiens moral et leurs conseils précieux tout au long de mes études.

A toute ma famille, A mes chers amis.

Pour leurs aides et supports dans les moments difficiles.

Enfin, je dédie ce travail à tous ceux qui me connaissent de près ou de loin.

Résumé

Lors de l'utilisation de photos numériques dans des applications, il est toujours recommandé d'utiliser des photos haute résolution, étant donné qu'elles peuvent contenir des détails critiques pour diverses applications par rapport à leurs équivalents basse résolution. Un médecin peut s'appuyer sur une image haute résolution pour établir un diagnostic correct, les objets peuvent être facilement distingués sur une image satellite haute résolution, ou un algorithme de reconnaissance de formes peut atteindre une plus grande efficacité lorsqu'il est fourni avec des échantillons d'image haute résolution.

Dans ce travail, nous proposons une méthode basée sur le deep learning pour convertir une image basse résolution en une image haute résolution. Cette méthode utilise un réseau neuronal convolutif pour trouver les similitudes entre les patches basse résolution et haute résolution d'une image et apprendre une cartographie entre eux. Le réseau est capable de produire une image haute résolution, en prenant une image basse résolution comme entrée.

Nous allons explorer les différentes techniques de la super résolution des images qui sont basées sur les réseaux de neurones profonds. Nous avons discuté plusieurs modèles différents selon l'ordonnement des couches et un jeu de paramètres visant à obtenir les meilleurs résultats possibles.

Les mots clés : super résolution, l'apprentissage en profondeur, réseau neuronal convolutif

Abstract :

When using digital photos in applications, it is always recommended to use high-resolution photos, since they may contain critical detail for various applications compared to their low-resolution counterparts. A doctor can rely on a high-resolution image to make a correct diagnosis, objects can be easily distinguished on a high-resolution satellite image, or a pattern recognition algorithm can achieve greater efficiency when provided with high resolution image samples.

In this work, we propose a method based on deep learning to convert a low-resolution image into a high-resolution image. This method uses a convolutional neural network to find similarities between low resolution and high-resolution patches of an image and learn a mapping between them. The network is capable of outputting a high-resolution image, taking a low-resolution image as input it can handle three color channels.

We will explore the different super resolution image techniques that are based on deep neural networks. We have discussed several different models depending on the ordering of the layers and a set of parameters aimed at obtaining the best possible results.

Keywords: super resolution, deep learning, convolutional neural network

ملخص:

عند استخدام الصور الرقمية في التطبيقات من المستحسن دائما استخدام صور عالية الدقة، حيث إنها قد تحتوي على تفاصيل مهمة لتطبيقات مختلفة مقارنة بنظيراتها منخفضة الدقة. يمكن للطبيب الاعتماد على صورة عالية الدقة لإجراء التشخيص الصحيح، ويمكن تمييز الكائنات بسهولة على صورة القمر الصناعي عالية الدقة، أو يمكن أن تحقق خوارزمية التعرف على الأنماط كفاءة أكبر عند تزويدها بعينات صور عالية الدقة.

في هذا العمل، نقترح طريقة تعتمد على التعلم العميق لتحويل صورة منخفضة الدقة إلى صورة عالية الدقة. تستخدم هذه الطريقة شبكة عصبية تلافيفية للعثور على أوجه التشابه بين التصحيحات منخفضة الدقة وعالية الدقة للصورة ومعرفة رسم الخرائط بينهما. الشبكة قادرة على إخراج صورة عالية الدقة، مع أخذ صورة منخفضة الدقة كمدخل يمكنها التعامل مع ثلاث قنوات ملونة.

الكلمات المفتاحية: الدقة الفائقة، التعلم العميق، الشبكة العصبية التلافيفية

Liste des abréviations

- ✚ **SR : Super Resolution**
- ✚ **HR : High Resolution**
- ✚ **LR : Low Resolution**
- ✚ **IA : Intelligence Artificielle**
- ✚ **ML : Machine Learning**
- ✚ **DL : Deep Learning.**
- ✚ **CNN : Convolutional Neural Network**
- ✚ **SRCNN : Super-Resolution Convolutional Neural Networ**
- ✚ **FRCNN : Fast Super-Resolution Convolutional Neural Network**
- ✚ **VDSR : Very Deep Super Résolution**
- ✚ **GAN : Generative Adversarial Networks**
- ✚ **VGG : Visual Geometry Group**
- ✚ **ANN : Artificial Neural Network**
- ✚ **RNN : Recurrent Neural Network**
- ✚ **Res-net: Residual neural network**
- ✚ **PSNR : Rapport signal sur bruit de crête**
- ✚ **MSE : Erreur quadratique moyenne**
- ✚ **SSIM : Indice de similarité structurelle**

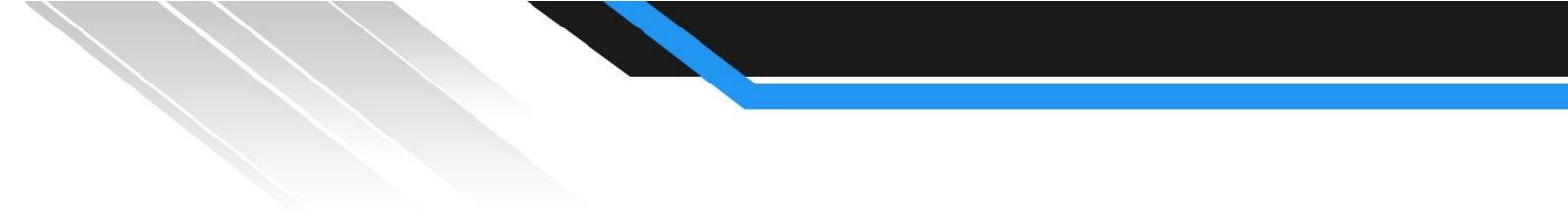


Sommaire

Sommaire

Liste des figures	/
Liste des abréviations	/
Introduction générale	01
<i>Chapitre I : Super résolution des images et traitement des images</i>	
1-Introduction	02
2-Définition de l'image	02
3- Définition de l'image numérique	02
4-Défèrent formats d'images	02
4-1 - Les images matricielles	03
4-2 - Les images vectorielles	03
5-Caractéristiques des images	04
5-1 – Pixel	04
5-2 - Dimension & Résolution	04
5-3 - Voisinage L'image	05
5-4 - Niveau de gris	05
5-5 – Contraste	06
5-6 - Luminance L'image	06
5-7 - Bruit	06
5-8 - Contour	07
6- Traitement d'image	07
6.1 C'est quoi le traitement d'images	07
6.2 Étapes fondamentales du traitement d'image	08
6.3 Applications du traitement d'images	09
6.4 Avantages du traitement d'image	10
7- Super-résolution	10
7-1 – Définition	10
7-2 - pourquoi la super résolution	11
7-3 - domaines d'application	11
7-4 - Nature du problème	11
7-5 - Méthodes et techniques de super-résolution	12
8- Conclusion	12

<i>Chapitre II : L'intégrations du deep Learning dans la super résolution des images</i>	
1-Introduction	14
2- Définition de l'apprentissage profond (deep Learning)	14
3- Historique de l'apprentissage profond	15
4- Domaine d'application de l'apprentissage profond	16
5- Pour quoi le deep Learning ?	17
6-Fonctionnement du deep Learning	18
7-Méthodes d'apprentissage profond	19
8-Un neurone artificiel, qu'est-ce que c'est ?	19
9- Types de réseaux d'apprentissage en profondeur	20
9-1- Réseau de neurones feedforward	20
9-2- Réseaux de neurones à fonction de base radiale	20
9-3- Perceptron multicouche	21
9-4- Réseau de neurones à convolution (CNN)	21
9-5- Réseau neuronal récurrent	22
9-6- Réseau neuronal modulaire	22
9-7- Modèles de séquence à séquence	23
10- l'intégration du deep Learning dans la super résolution des images	23
11- Jeux de données	28
12- Fonctions de perte	29
13- Métrique	30
14 – Conclusion	30
<i>Chapitre III : Mise en œuvre et exploitation d'un apprentissage profond pour la super-résolution des</i>	
1-Introduction	32
2- Environnement de mise en œuvre adopté	32
3- Mise en œuvre	33
4- Résultat obtenus	41
5- Discussion	44
6- Conclusion	48
Conclusion générale	49
Bibliographie	50



Liste des Figures



Liste des figures

<i>Figures</i>	<i>Page</i>
Figure 1 : Image matricielle (Bitmap)	03
Figure 2 : Image vectorielle	03
Figure 3 : Représentation de la lettre a sous la forme d'un groupe de pixels	04
Figure 4 : Représentation de dimension d'une image	04
Figure 5 : Voisinage à 4 et voisinage à 8	05
Figure 6 : Exemple de niveaux de gris	06
Figure 7 : Image avec et sans bruit	07
Figure 8 : Exemple de contour	07
Figure 9 : Schéma d'un système de traitement d'images	09
Figure II.1 : La relation entre l'intelligence artificielle, le ML et le deep Learning	14
Figure II.2: Difference entre Machine Learning & Deep Learning	18
Figure II.3 : Le processus d'auto-apprentissage pour l'identification d'un chat	18
Figure II.4 : Modèle d'un neurone biologique (à gauche) et son modèle mathématique (à droite)	19
Figure II.5 : Schéma d'un réseau de neurones feedforward	20
Figure II.6 : Perceptron multicouche	21
Figure II.7 : Schéma d'un Réseau de neurones à convolution (CNN)	22

Figure II.8 : Schéma d'un Réseau neuronal modulaire	22
Figure II.9 : Schéma d'un Réseau neuronal modulaire	22
Figure II.10 : Architecture du (SRCNN)	23
Figure II.11 : Architecture du réseau de neurones convolutif profond (VDSR)	24
Figure II.12 : Comparaison entre SRCNN et FSRCNN	25
Figure II.13 : Architecture du générateur et du réseau discriminateur	26
Figure II.14: The architecture of the proposed single-scale SR network (EDSR).	26
Figure II.15 : Architecture lapsrn.	27
Figure II.16 : Unité de sélection (SU) et architecture de réseau selnet	28
Figure III.1 : Environnement de Google Colab	32
Figure III.2 : Architecture du (SRCNN)	33
Figure III.3 : Perte d'entraînement/perte de validation	40
Figure III.4 : Comparaison entre les deux modèles	48



Introduction

générale

Introduction générale :

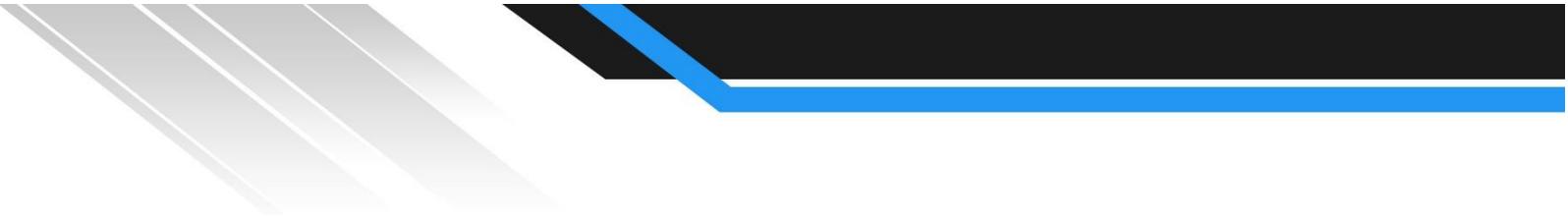
Au cours de la dernière décennie, le monde a connu une immense avancée technologique mondiale, à la fois dans le matériel et les logiciels. Les industries ont profité de la technologie de pointe pour produire des gadgets électroniques tels que des ordinateurs, des téléphones portables et bien d'autres à des prix abordables.

Les unités de fabrication de capteurs de caméra ont également évolué pour produire des caméras numériques haute résolution (HR) de bonne qualité.

Bien que des caméras numériques haute résolution soient disponibles, de nombreuses applications de vision par ordinateur avaient encore une forte demande d'imagerie à plus haute résolution qui dépassait très souvent les capacités de ces caméras numériques HR.

Le problème est que les images haute résolution ne sont pas toujours disponibles. En effet, la configuration de l'imagerie haute résolution s'est avérée coûteuse et n'est pas toujours réalisable en raison des limitations inhérentes à la technologie de fabrication des capteurs et des optiques.

Pour résoudre ce problème, des progrès significatifs ont été réalisés dans le domaine du traitement et l'amélioration des images. Pour faire avancer ce progrès, le concept de super-résolution est devenu très populaire au cours de la dernière décennie.



Chapitre 01

Super résolution des images et traitement des images



1. Introduction

Dans la vie d'aujourd'hui, l'image est une source importante permettant aux utilisateurs d'échanger et d'obtenir des informations abondantes sous forme de photos, de graphiques, de données bidimensionnelles (2D), de données tridimensionnelles (3D), de résonance magnétique (MR), etc. Le traitement d'image est utilisé pour obtenir des informations fiables dans l'image.

Classiquement, les technologies récentes de traitement d'images se concentrent sur l'amélioration, la restauration, la compression, l'extraction, la transformation et la super-résolution d'images dans les systèmes de vision industrielle. Avec l'évolution des technologies de l'information, la super-résolution a fait de grands progrès dans le traitement des images.

La super-résolution d'image (SR) est un processus qui consiste à reconstruire des images haute résolution (HR) à partir d'images basse résolution (LR). Il s'agit d'une classe importante de techniques de traitement d'images dans le domaine du traitement d'images et de la vision par ordinateur.

2. Définition de L'image

Une image est une représentation visuelle d'un objet, d'une personne, d'un animal ou d'un autre objet qui peut être capturée par l'œil humain grâce à diverses techniques telles que le dessin, la conception, la photographie, la vidéo et autres. [1]

3. Définition de L'image numérique

Image numérique désigne toute image (dessin, icône, photographie) acquise, créé, traitée et stockée sous forme binaire (suite de 0,1)

- acquise par des convertisseurs analogique-numérique situés dans des dispositifs comme les scanners, les appareils photo ou les caméscopes numériques, les cartes d'acquisition vidéo
- Créée directement par des programmes informatiques, grâce à une souris, des tablettes graphiques ou par de la modélisation 3D
- Traitée grâce à des outils graphiques, de façon à la transformer, à en modifier la taille, les couleurs, d'y ajouter ou d'en supprimer des éléments, d'y appliquer des filtres variés, etc.
- stockée sur un support informatique (clé USB, SSD, disque dur, CD-ROM...). [2]

4. Différent formats d'images

Il y a deux types d'images différentes :

4.1 Les images matricielles :

Image matricielle : appelée aussi image bitmap, c'est un ensemble des bits qui spécifie la couleur de chaque pixel dans un tableau rectangulaire de pixels. Mais lorsque vous effectuez un zoom avant, vous constatez une perte de qualité de l'image. Pour résoudre ce problème, vous devez prendre une photo de très grandes tailles, ce qui pose de nombreux problèmes [3].

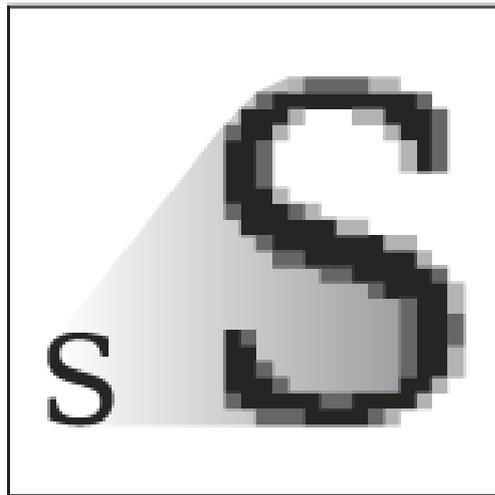


Figure 1 : image matricielle(bitmap) [3].

4.2 Les images vectorielles :

Comme toute image, l'image vectorielle utilise également la technique pixel, mais comme elle est calculée dynamiquement par le programme, sa position et sa couleur ne sont pas fixes. Par exemple, pour afficher une ligne, le programme spécifie le point de départ et le point d'accès suivi du chemin à suivre, puis calcule tous les pixels nécessaires pour afficher cette ligne. La même chose s'applique aux formes et aux couleurs plus complexes. [4]



Figure 2 : image vectorielle [4]

5. Caractéristiques des images

5.1 Pixel

Le pixel est l'abrégié du mot « Pictures élément », il détermine la base de l'image, car il est considéré comme l'unité de surface. Le pixel est présenté le niveau de gris, et il est lié avec certaines couleurs et aussi une intensité. Le Pixel est le plus petit composant pouvant être manipulé pour l'affichage ou l'impression.

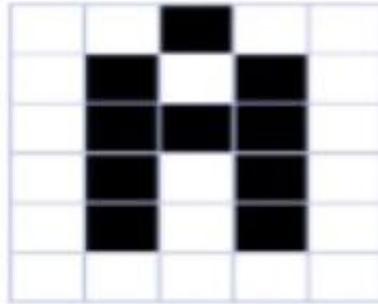


Figure 3 : Représentation de la lettre a sous la forme d'un groupe de pixels [3]

Dans les images monochromes, les pixels donnent les différences exactes entre eux, ainsi que la même chose pour les images couleur, en fonction de la quantité. Par exemple, un pixel est représenté par trois octets pour chaque couleur, ce qui correspond aux images couleur (RGB) [3].

5.2 Dimension & Résolution

La dimension dans ce cas est la taille de l'image, et on le trouve sous la forme d'une matrice portant des valeurs numériques représentant l'intensité de la lumière appelée pixel, et pour obtenir le nombre total de pixels dans l'image, on multiplie le nombre de lignes dans le nombre de colonnes.

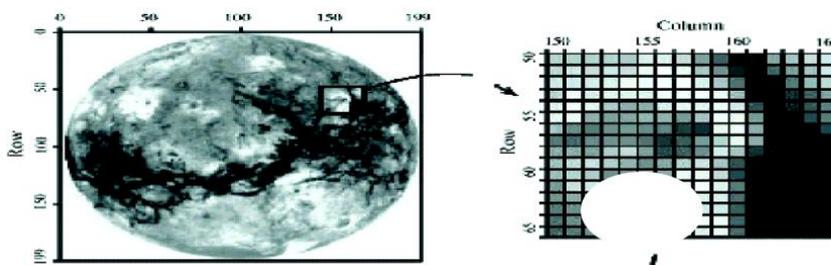


Figure 4 : Représentation de dimension d'une image [3]

Par contre la résolution est la clarté ou la finesse de détails atteinte par un moniteur dans la production d'images. Sur les moniteurs d'ordinateurs, la résolution est exprimée en nombre de pixels par unité de mesure (pouce ou centimètre). On utilise aussi le mot résolution pour

désigner le nombre total de pixels affichables horizontalement ou verticalement sur un moniteur, plus grand est ce nombre, meilleure est la résolution. [3]

5.3 Voisinage L'image

Est un ensemble de pixels qui s'entrelacent et repose sur ce principe, la surface de l'image est formée de deux : forme rectangle ou d'une forme hexagone, ce qui donne la notion de voisinage, où le voisinage de pixels est formé par le groupe de pixels autour de ce même pixel. Où cela les opérateurs de voisinage prennent en compte l'information portée par les pixels voisins du pixel étudié. Nous mentionnons deux types de voisinage : • Voisinage à 4 • Voisinage à 8 [3]

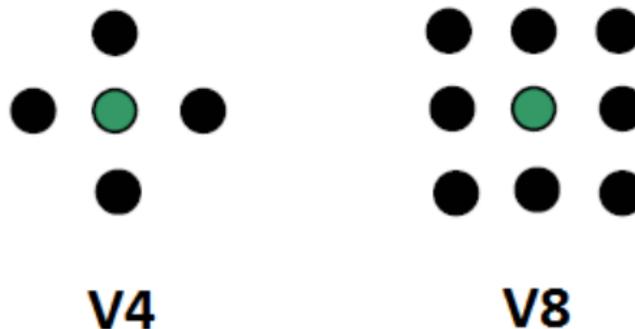


Figure 5 : voisinage à 4 et voisinage à 8. [3]

5.4 Niveau de gris

Le niveau de gris est défini en tout point où ce dernier porte la valeur de l'intensité lumineuse, de sorte que la couleur du pixel prend des valeurs allant du noir au blanc. C'est-à-dire pour représenter des images en niveaux de gris, chaque pixel peut prendre une valeur comprise entre 0 et 255. Plus le nombre de bits reflétant la couleur de chaque pixel de l'image est grand, plus les niveaux de gris sont élevés [3]

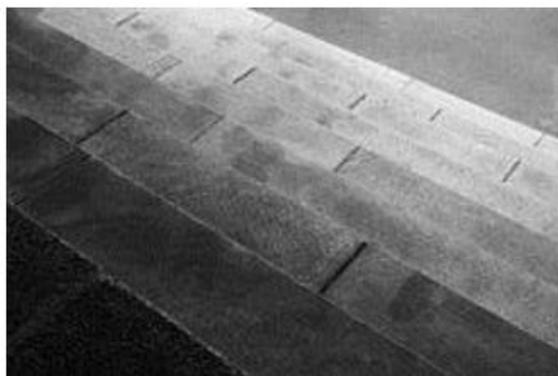


Figure 6 : exemple de niveaux de gris [3]

5.5 Contraste

C'est l'opposition marquée entre deux régions d'une image, plus précisément entre les régions sombres et les régions claires de cette image. Le contraste est défini en fonction des luminances de deux zones d'images. Si L_1 et L_2 sont les degrés de luminosité respectivement de deux zones voisines A_1 et A_2 d'une image, le contraste ' C ' est défini par le rapport : $C = \frac{L_1 - L_2}{L_1 + L_2}$. [3]

5.6 Luminance L'image

Est définie comme un ensemble de points où elle se caractérise par différents degrés de luminosité, ce dernier étant défini par l'intensité lumineuse d'une surface à travers la zone visible de cette surface. Dans le cas de l'observateur distant, la notion de luminance est utilisée comme un télégramme correspondant à la luminosité de l'objet.

Les caractéristiques de la meilleure Luminance sont :

1. Des images lumineuses (brillantes)
2. Le degré de contraste est meilleur : évitez les images où la plage de contraste allant du blanc au noir, ces images entraînent une perte de détail dans les parties sombres ou claires.
3. Aucun de parasites. [3]

5.7 Bruit

Montrée après le processus d'émission et de réception sous la forme d'un signal clair dans l'image où de ce que l'on appelle du parasite, cette variante s'appelle la différence de densité de pixels si on la compare au pixel qui l'entoure, cela produit du bruit sous forme d'erreurs causées par des capteurs électroniques ou de scanners de moins bonne qualité. Ils apparaissent en petits points à la surface de l'image, comme illustrée dans la figure suivante. Image avec bruit Image sans bruit [3]



Image avec bruit



Image sans bruit

Figure 7 : Image avec et sans bruit [3]

5.8 Contour

L'image est composée d'objets et n'interfère pas les uns avec les autres en raison de l'existence de frontières qui les séparent les uns des autres et sont appelées le terme "contour", ou entre les deux pixels qui représentent une grande différence au niveau du gris. Sur la base des points qui séparent deux textures différentes dans l'image ou plutôt de la neutralisation, l'extraction ou la composition du "contour". [3]

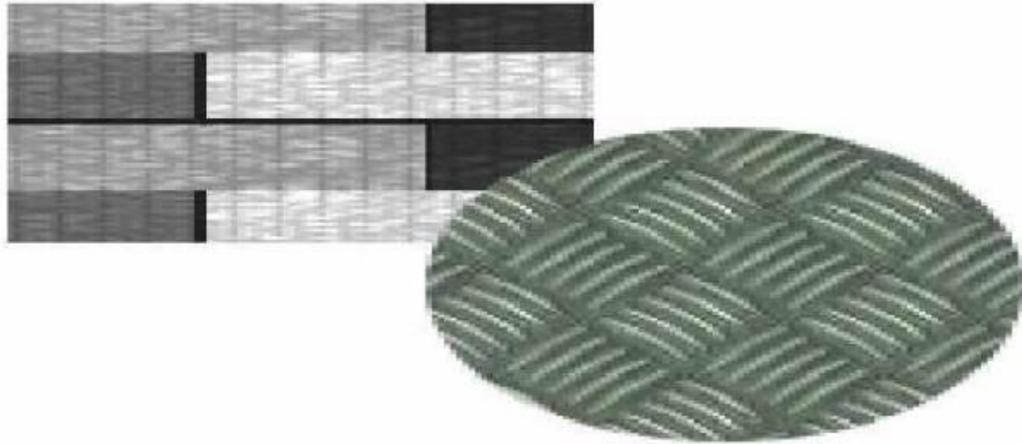


Figure 8 : exemple de contour [3]

6. Traitement d'image

6.1 C'est quoi le traitement d'images

Le traitement d'image est le processus de transformation d'une image en une forme numérique et d'exécution de certaines opérations pour en tirer des informations utiles. Le système de traitement d'image traite généralement toutes les images comme des signaux 2D lors de l'application de certaines méthodes de traitement de signal prédéterminées.

Il existe cinq principaux types de traitement d'image :

1. **Visualisation** - Trouver des objets qui ne sont pas visibles dans l'image
2. **Reconnaissance** - Distinguer ou détecter des objets dans l'image
3. **Netteté et restauration** - Créez une image améliorée à partir de l'image d'origine
4. **Reconnaissance de formes** - Mesurez les différents motifs autour des objets dans l'image
5. **Récupération** - Parcourez et recherchez des images dans une grande base de données d'images numériques similaires à l'image d'origine. [5]

6.2 Étapes fondamentales du traitement d'image

Acquisition d'image

L'acquisition d'images est la première étape du traitement d'images. Cette étape est également connue sous le nom de prétraitement. Cela implique de récupérer l'image à partir d'une source, généralement une source matérielle.

Amélioration d'images

L'amélioration d'image est le processus qui consiste à faire ressortir et à mettre en évidence certaines caractéristiques intéressantes d'une image qui a été obscurcie. Cela peut impliquer de modifier la luminosité, le contraste, etc.

Restauration d'images

La restauration d'image est le processus d'amélioration de l'apparence d'une image. Cependant, contrairement à l'amélioration d'image, la restauration d'image se fait à l'aide de certains modèles mathématiques ou probabilistes.

Traitement des images couleur

Le traitement d'images en couleurs comprend un certain nombre de techniques de modélisation des couleurs dans un domaine numérique. Cette étape a pris de l'importance en raison de l'utilisation importante d'images numériques sur Internet.

Ondelettes et traitement multirésolution

Les ondelettes sont utilisées pour représenter des images à divers degrés de résolution. Les images sont subdivisées en ondelettes ou régions plus petites pour la compression des données et pour la représentation pyramidale.

Compression

La compression est un processus utilisé pour réduire le stockage nécessaire pour enregistrer une image ou la bande passante nécessaire pour la transmettre. Cela se fait notamment lorsque l'image est destinée à être utilisée sur Internet.

Traitement morphologique

Le traitement morphologique est un ensemble d'opérations de traitement permettant de transformer des images en fonction de leurs formes.

Segmentation

La segmentation est l'une des étapes les plus difficiles du traitement d'images. Il s'agit de partitionner une image en ses éléments constitutifs ou objets.

Représentation et description

Après qu'une image a été segmentée en régions dans le processus de segmentation, chaque région est représentée et décrite sous une forme appropriée pour un traitement

informatique ultérieur. La représentation traite des caractéristiques de l'image et des propriétés régionales. La description traite de l'extraction d'informations quantitatives qui aident à différencier une classe d'objets de l'autre.

Reconnaissance

La reconnaissance attribue une étiquette à un objet en fonction de sa description. [5]

Un système de traitement numérique d'images est composé de :

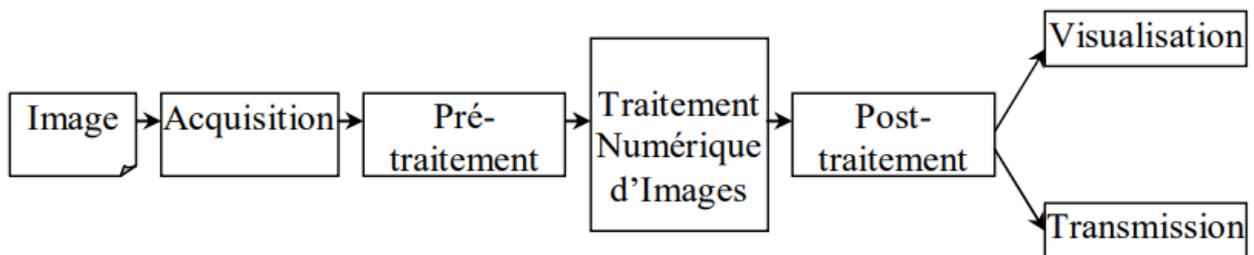


Figure 9 : Schéma d'un système de traitement d'images [5]

6.3 Applications du traitement d'images

- **Récupération d'images médicales**

Le traitement d'images a été largement utilisé dans la recherche médicale et a permis des plans de traitement plus efficaces et plus précis. Par exemple, peut être utilisé pour la détection précoce du cancer du sein à l'aide d'un algorithme sophistiqué de détection de nodules dans les scintigraphies mammaires. Étant donné que l'utilisation médicale nécessite des processeurs d'images hautement qualifiés, ces applications nécessitent une mise en œuvre et une évaluation importante avant de pouvoir être acceptées.

- **Technologies de détection du trafic**

Dans le cas des capteurs de trafic, nous utilisons un système de traitement d'images vidéo ou VIPS. Celui-ci consiste en a) un système de capture d'images b) un système de télécommunication et c) un système de traitement d'images. Lors de la capture vidéo, un VIPS dispose de plusieurs zones de détection qui émettent un signal "on" chaque fois qu'un véhicule entre dans la zone, puis émettent un signal "off" chaque fois que le véhicule sort de la zone de détection. Ces zones de détection peuvent être configurées pour plusieurs voies et peuvent être utilisées pour détecter le trafic dans une station particulière.

- **Reconstruction d'images**

Le traitement d'image peut être utilisé pour récupérer et remplir les parties manquantes ou corrompues d'une image. Cela implique l'utilisation de systèmes de traitement d'images qui ont été formés de manière approfondie avec des ensembles de données de photos existants pour créer de nouvelles versions de photos anciennes et endommagées.

- **Détection faciale**

L'une des applications les plus courantes du traitement d'image que nous utilisons aujourd'hui est la détection de visage. Il s'agit de systèmes d'apprentissage en profondeur où la machine est d'abord entraînée avec les caractéristiques spécifiques des visages humains, telles que la forme du visage, la distance entre les yeux, etc. Après avoir appris à la machine ces caractéristiques du visage humain, elle commencera à accepter tous objets dans une image qui ressemblent à un visage humain. La détection de visage est un outil essentiel utilisé dans la sécurité, la biométrie et même les filtres disponibles sur la plupart des applications de médias sociaux de nos jours.[5]

6.4 Avantages du traitement d'image

La mise en œuvre de techniques de traitement d'images a eu un impact massif sur de nombreuses organisations technologiques. Voici quelques-uns des avantages les plus utiles du traitement d'image, quel que soit le domaine d'opération :

- ✓ L'image numérique peut être rendue disponible dans n'importe quel format souhaité (image améliorée, radiographie, négatif photo, etc.)
- ✓ Il aide à améliorer les images pour l'interprétation humaine
- ✓ Les informations peuvent être traitées et extraites des images pour une interprétation automatique
- ✓ Les pixels de l'image peuvent être manipulés à n'importe quelle densité et contraste souhaités
- ✓ Les images peuvent être stockées et récupérées facilement
- ✓ Il permet une transmission électronique facile des images à des fournisseurs tiers [5]

7. Super Résolution

7.1 Définition

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 un plus haut résolution. Le sujet a été initialement mentionné au milieu des années 80 et le terme "super-résolution" lui-même est apparu vers 1990. Cependant, cette idée a gagné en

popularité au fil des ans et est devenue particulièrement utile aujourd'hui. Bien que de nombreuses approches aient existé, elles partagent grosso modo la même base consistant à régénérer les pixels manquants à partir d'une image basse résolution grâce à des algorithmes particuliers. [6]

7.2 Pourquoi la super résolution

L'objectif central de la super-résolution (SR) est de générer une image de résolution supérieure à partir d'images de résolution inférieure. L'image haute résolution offre une densité de pixels élevée et donc plus de détails sur la scène d'origine. Le besoin de haute résolution est courant dans les applications de vision par ordinateur pour de meilleures performances dans la reconnaissance des formes et l'analyse des images. [7]

La haute résolution est importante en imagerie médicale pour le diagnostic. De nombreuses applications nécessitent un zoom sur une zone d'intérêt spécifique dans l'image où une haute résolution devient essentielle, par ex. applications de surveillance, médico-légales et d'imagerie par satellite.

7.3 Domaines d'application

Il est couramment utilisé dans les applications suivantes :

- **Surveillance** : pour détecter, identifier et effectuer une reconnaissance faciale sur des images basse résolution obtenues à partir de caméras de sécurité.
- **Médical** : 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.
- **Médias** : la super résolution peut être utilisée pour réduire les coûts de serveur, car les médias peuvent être envoyés à une résolution inférieure et mis à l'échelle à la volée.

[8]

7.4 Nature du problème

Le problème est que les images haute résolution ne sont pas toujours disponibles. En effet, la configuration de l'imagerie haute résolution s'est avérée coûteuse et n'est pas toujours réalisable en raison des limitations inhérentes à la technologie de fabrication des capteurs et des optiques.

Comme les anciens caméras était par nature de faible résolution. Aussi, pour certains domaines il n'est pas pratique d'installer des caméras de surveillance de hautes résolutions (faute d'espace de stockage parce qu'une caméra de surveillance fonctionne 24h sur 24h).

Ces problèmes peuvent être surmontés grâce à l'utilisation d'algorithmes de traitement d'images, relativement peu coûteux, qui ont donné naissance au concept de super-résolution. Il offre un avantage car il peut coûter moins cher et les systèmes d'imagerie actuels à faible résolution peuvent être utilisés.

7.5 Méthodes et techniques de super-résolution

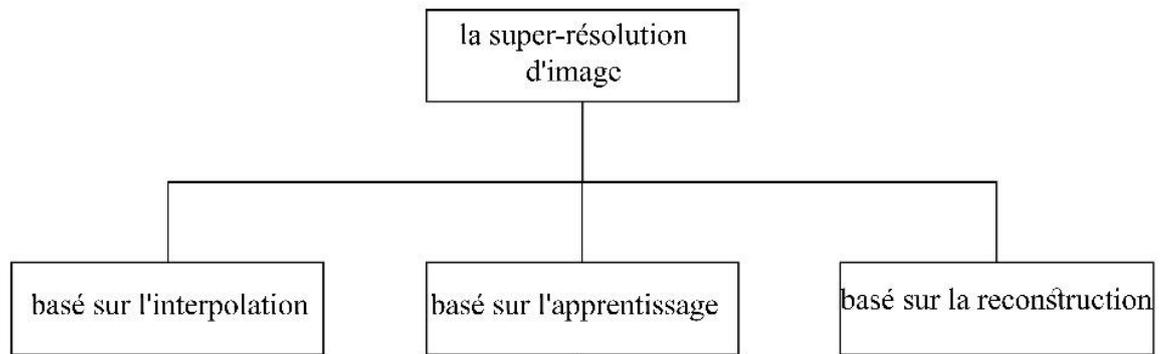


Figure 10 : Classification de l'approche de la super-résolution d'image. . [9]

L'approche de la super-résolution d'image peut être divisée en trois classes. La première comprend des méthodes basées sur l'interpolation telles que les interpolations bilinéaire, bicubique et Lanczos. Bien que les méthodes basées sur l'interpolation soient considérablement rapides, leur effet d'amélioration de la résolution est relativement faible. La seconde comprend la méthode basée sur la reconstruction utilisant l'optimisation inverse. Cette méthode nécessite un processus d'optimisation itératif complexe au lieu du processus de formation. Le troisième comprend des méthodes basées sur l'apprentissage telles que la mise en correspondance des correctifs, l'apprentissage automatique, CNN et le réseau contradictoire génératif. Parmi les trois approches de super-résolution susmentionnées, la méthode basée sur l'apprentissage tend à fournir les meilleures performances visuelles. Cependant, dans cette méthode basée sur l'apprentissage, un processus de préformation est nécessaire et les performances de super-résolution dépendent fortement de la base de données de formation. [9]

8. Conclusion

Nous avons consacré ce chapitre à présenter le concept de l'image et ses différents formats et caractéristiques, et nous avons également parlé du traitement d'image et la super résolution et ses domaines d'utilisation. Dans le deuxième chapitre, nous expliquerons en détail l'apprentissage profond et les réseaux de neurones plus en détail et leur utilisation dans la super résolution.

Chapitre 02

L'intégration du Deep Learning dans la super résolution des images

1. Introduction :

À ses débuts, l'intelligence artificielle (IA) s'est fortement concentrée sur les systèmes basés sur des règles qui feraient des prédictions à l'aide d'ensembles prédéfinis de règles qui devaient être fournies par un expert en la matière. Cependant, ces systèmes étaient fragiles et s'appuyaient sur ces "opinions d'experts", ce qui les a finalement fait tomber en désuétude. Au fur et à mesure que l'échelle et la quantité de données augmentaient, ces méthodes ont été remplacées par une approche plus axée sur les données, l'apprentissage automatique et l'apprentissage profond.

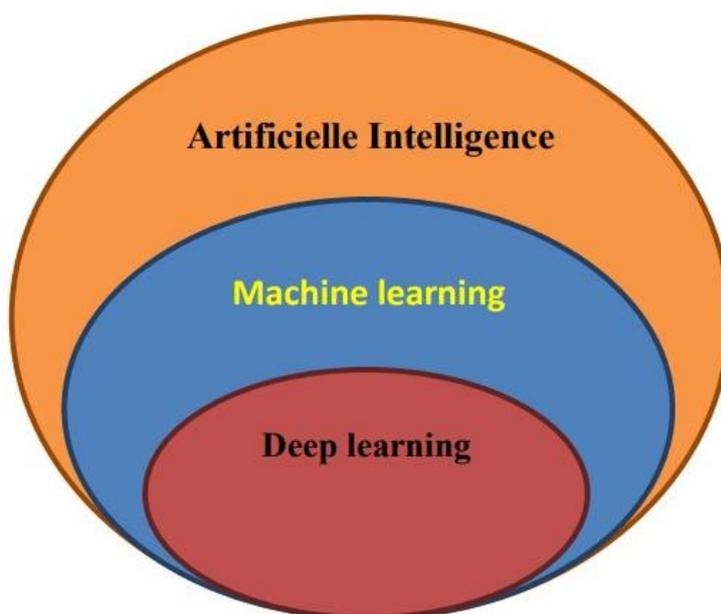


Figure II.1: La relation entre l'intelligence artificielle, le ML et le deep Learning [10]

2. Définition de l'apprentissage profond (deep Learning) :

Deep Learning est une méthode d'apprentissage automatique impliquant l'utilisation d'un réseau de neurones profonds artificiels. Tout comme le cerveau humain est constitué de cellules nerveuses ou de neurones qui traitent les informations en envoyant et en recevant des signaux, l'apprentissage par réseau neuronal profond consiste en des couches de « neurones » qui communiquent entre elles et traitent les informations.

Le « profond » dans Deep Learning indique le nombre de couches au sein du réseau ; plus le nombre de couches est profond, plus le réseau est profond.

Le réseau neuronal profond peut fonctionner avec des données étiquetées ainsi que des données non étiquetées, ce qui permet un apprentissage supervisé et non supervisé.

Cependant, une grande quantité de données est nécessaire pour entraîner le réseau neuronal profond, ce qui peut aider à produire des résultats plus précis.

Ces réseaux sont également capables de reconnaître les erreurs ou les pertes de sortie et de les corriger sans intervention humaine. [11]

3. Historique de l'apprentissage profond

Le premier modèle mathématique d'un réseau de neurones a été créé par Walter Pitts et Warren McCulloch en 1943, qui a démontré le processus de pensée du cerveau humain. De là a commencé le voyage du réseau neuronal profond et de l'apprentissage en profondeur.

- **1957** - Frank Rosenblatt soumet un article intitulé "Le Perceptron : un automate de perception et de reconnaissance", qui consiste en un algorithme ou une méthode de reconnaissance de formes utilisant un réseau de neurones à deux couches.
- **1965** - Alexey Ivakhnenko et V.G. Lapa a développé le premier réseau neuronal fonctionnel et Alexey Ivakhnenko a créé un réseau neuronal profond à 8 couches en 1971 qui a été démontré dans le système d'identification informatique, Alpha. C'était la véritable introduction à l'apprentissage en profondeur.
- **1980** - Kunihiko Fukushima a développé le "Neocognitron", un réseau de neurones profonds artificiels avec des couches multiples et convolutives pour reconnaître les modèles visuels.
- **1985** - Terry Sejnowski crée NETtalk, un programme qui apprend à prononcer des mots anglais.
- **1989** - Yann LeCun, utilisant un réseau de neurones profonds à convolution, développe un système capable de lire les chiffres manuscrits.
- **2009** - Comme les modèles d'apprentissage en profondeur nécessitent une énorme quantité de données étiquetées pour s'entraîner à l'apprentissage supervisé, Fei-Fei Li lance ImageNet, qui est une grande base de données d'images étiquetées.
- **2012** - Les résultats de "The Cat Experiment" menée par Google Brain sont publiés. Cette expérience était basée sur un apprentissage non supervisé dans lequel le réseau neuronal profond travaillait avec des données non étiquetées pour reconnaître des modèles et des caractéristiques dans les images de chats. Cependant, il ne pouvait reconnaître correctement que 15% des images.
- **2014** - Facebook a développé DeepFace, un système d'apprentissage en profondeur pour identifier et taguer les visages des utilisateurs sur les photographies. [11]

4. Domaine d'application de l'apprentissage

L'éducation en profondeur a été développée dans une large mesure, balayant les différents secteurs des TIC, par exemple, la reconnaissance visuelle du panneau de signalisation par un robot ou un véhicule indépendant et aussi la compréhension de l'image avec des réseaux convolutifs profonds.

De nouvelles classes de réseaux de neurones ont été développées et conviennent parfaitement à des applications telles que la traduction de texte et la classification d'images.

L'apprentissage en profondeur est apparu dans de nombreux autres domaines, notamment l'étude de l'ADN dans le domaine du diagnostic de certains cas médicaux, ainsi que dans le domaine de la sécurité informatique et non informatique. Il peut dessiner des tableaux de la même manière que le peintre apprit, et on le retrouve dans la science de l'éducation assistée par ordinateur et dans le domaine de l'intelligence artificielle en général.

Aujourd'hui, les entreprises utilisent de nombreuses applications pratiques du domaine de l'apprentissage en profondeur, notamment :

- **Reconnaissance de la parole** : Prenez, par exemple, Xbox, Skype, Google Now et Siri d'Apple, qui utilise des techniques d'apprentissage approfondies dans leurs systèmes pour reconnaître les modèles de parole et les sons de l'homme.
- **Reconnaissance d'Image** : L'application pratique de la reconnaissance d'images est la traduction automatique d'images et la description de scènes. Cette application est très efficace dans le domaine des enquêtes pour déterminer l'activité criminelle parmi des milliers d'images fournies et les voitures peuvent en bénéficier grâce à l'identification d'images en utilisant la technologie de la caméra à 360 degrés.
- **Traitement du langage naturel** : Les réseaux de neurones constituent depuis de nombreuses années une composante essentielle de l'apprentissage en profondeur dans le traitement et l'analyse de textes écrits. Il peut également être utilisé pour découvrir et analyser les notes du médecin ou les reportages.
- **Systèmes de recommandation** : Amazon et Netflix ont tous deux intégré avec succès le concept du système de recommandation avec la

possibilité de reconnaître votre degré d'intérêt la prochaine fois en fonction du comportement passé. Il est également utilisé dans d'autres domaines tels que la musique et d'autres domaines.

- Dans le même temps, les interfaces homme-machine (IHM) ont beaucoup évolué. Y compris la souris et le clavier où ils ont été remplacés par des gestes ..., suscitant un regain d'intérêt pour l'IA et l'apprentissage en profondeur. [3]

5. Pour quoi le deep Learning ?

Les algorithmes de ML fonctionnent bien pour une grande variété de problèmes. Cependant ils ont échoué à résoudre quelques problèmes majeurs de l'IA telle que la reconnaissance vocale et la reconnaissance d'objets.

Le développement du deep Learning fut motivé en partie par l'échec des algorithmes traditionnels dans de telle tâche de l'IA.

Mais ce n'est qu'après que de plus grandes quantités de données ne soit disponibles grâce notamment au Big Data et aux objets connectés et que les machines de calcul soient devenues plus puissantes qu'on a pu comprendre le potentiel réel du Deep Learning.

Une des grandes différences entre le Deep Learning et les algorithmes de ML traditionnelles c'est qu'il s'adapte bien, plus la quantité de données fournie est grande plus les performances d'un algorithme de Deep Learning sont meilleures. Contrairement à plusieurs algorithmes de ML classiques qui possèdent une borne supérieure à la quantité de données qu'ils peuvent recevoir des fois appelée "plateau de performance", les modèles de Deep Learning n'ont pas de telles limitations (théoriquement) et ils sont même allés jusqu'à dépasser La performance humaine dans des domaines comme le traitement d'image.

Autre différence entre les algorithmes de ML traditionnelles et les algorithmes de Deep Learning c'est l'étape de l'extraction de caractéristiques. Dans les algorithmes de ML traditionnelles l'extraction de caractéristiques est faite manuellement, c'est une étape difficile et coûteuse en temps et requiert un spécialiste en la matière alors qu'en Deep Learning cette étape est exécutée automatiquement par l'algorithme.

Apprentissage automatique vs Apprentissage profond

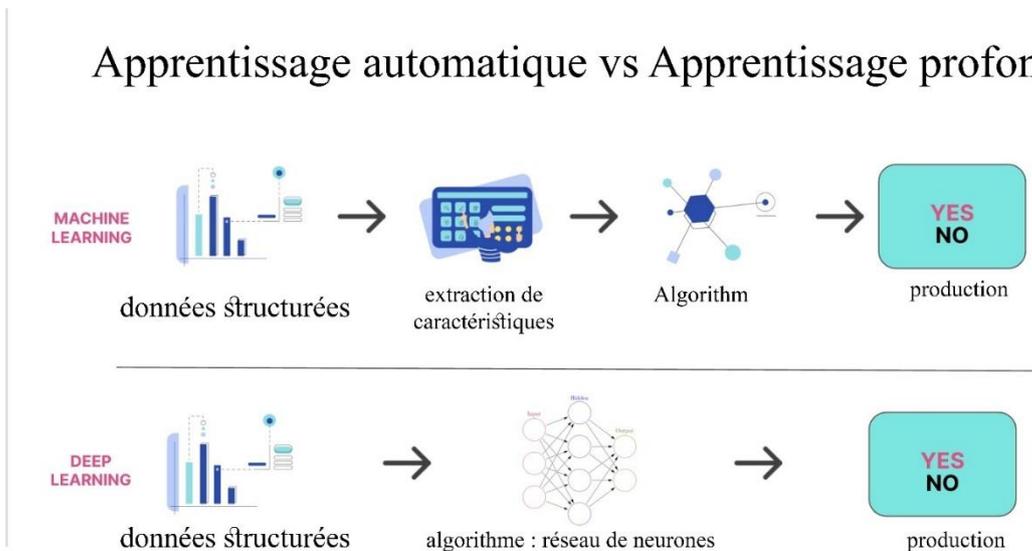


Figure II.2: Difference entre Machine Learning & Deep Learning [12]

6. Fonctionnement du deep Learning

Le *deep Learning* s'appuie sur un réseau de neurones artificiels s'inspirant du cerveau humain. Ce réseau est composé de dizaines voire de centaines de « couches » de neurones, chacune recevant et interprétant les informations de la couche précédente. Le système apprendra par exemple à reconnaître les lettres avant de s'attaquer aux mots dans un texte, ou détermine s'il y a un visage sur une photo avant de découvrir de quelle personne il s'agit.

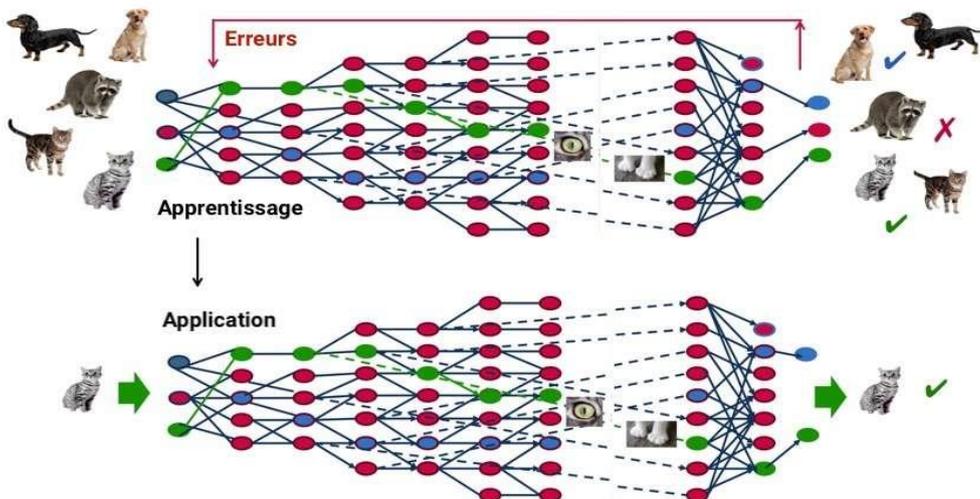


Figure II.3: le processus d'auto-apprentissage pour l'identification d'un chat[13]

À chaque étape, les « mauvaises » réponses sont éliminées et renvoyées vers les niveaux en amont pour ajuster le modèle mathématique. Au fur et à mesure, le programme réorganise les informations en blocs plus complexes. Lorsque ce modèle est par la suite appliqué à d'autres cas, il est normalement capable de reconnaître un chat sans que personne ne lui ait jamais indiqué qu'il n'a jamais appris le concept de chat. Les données de départ sont essentielles : plus le système accumule d'expériences différentes, plus il sera performant. [13]

7. Méthodes d'apprentissage automatique

Pour veiller à ce que les connexions au sein d'un réseau de neurones artificiels soient correctement établies, il faut au préalable procéder à son « entraînement ». On peut distinguer ici deux procédures de base : l'apprentissage supervisé et l'apprentissage non supervisé.

Apprentissage supervisé

Dans le cadre de l'apprentissage supervisé, un **résultat concret est défini** pour chacune des différentes options d'entrée. Par exemple, si on présente au système une photographie de chats afin qu'il la reconnaisse, il faut **contrôler le fonctionnement du système** et donner un retour d'information pour déterminer si l'image est correctement reconnue ou non. Cette méthode permet de modifier les pondérations au sein du réseau de neurones artificiels et d'optimiser l'algorithme.

Apprentissage non supervisé

Dans le cas d'un apprentissage non supervisé, le résultat de la tâche considérée n'est pas prédéterminé. C'est le **système lui-même qui tire les conséquences à partir des seules informations saisies**, en s'appuyant sur la règle d'apprentissage de Hebb ou la théorie de la résonance adaptative. [14]

8. Un neurone artificiel, qu'est-ce que c'est ?

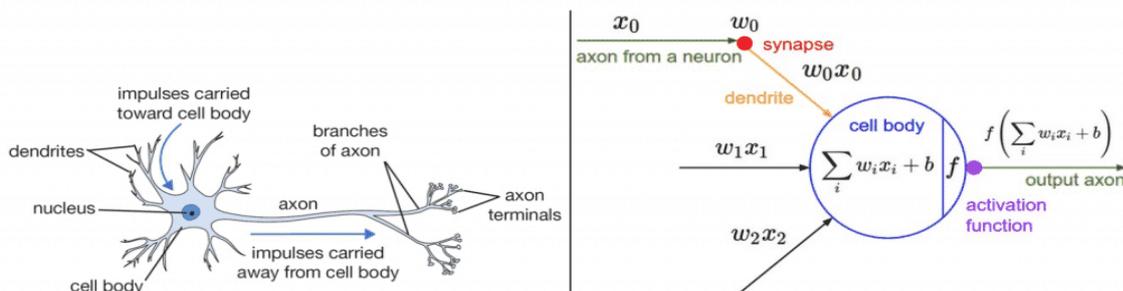


Figure II.4 : Modèle d'un neurone biologique (à gauche) et son modèle mathématique (à droite) [16]

Un réseau de neurones est un ensemble d'algorithmes inspirés par le cerveau humain. Le but de cette technologie est de simuler l'activité du cerveau humain, et plus spécifiquement la reconnaissance de motifs et la transmission d'informations entre les différentes couches de connexions neuronales.

Un Deep Neural Network, ou réseau de neurones profond, se distingue par une particularité : il est **composé d'au moins deux couches**. Ceci lui permet de traiter les données de manière complexe, en employant des modèles mathématiques avancés. [15]

9. Types de réseaux d'apprentissage en profondeur

9.1 Réseau de neurones feedforward

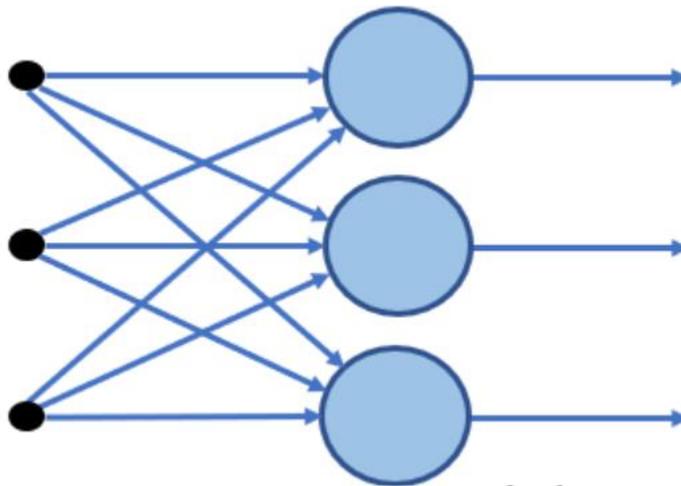


Figure II.5 : Schéma d'un réseau de neurones feedforward[16]

- Ce type de réseau neuronal est le réseau neuronal très basique où le contrôle de flux se produit à partir de la couche d'entrée et va vers la couche de sortie.
- Ces types de réseaux n'ont qu'une seule couche ou une seule couche cachée
- Étant donné que les données ne se déplacent que dans une direction, il n'y a pas de technique de rétropropagation dans ce réseau
- Dans ce réseau, la somme des poids présents dans l'entrée est introduite dans la couche d'entrée
- Ces types de réseaux sont utilisés dans l'algorithme de reconnaissance faciale utilisant la vision par ordinateur. [16]

9.2. Réseaux de neurones à fonction de base radiale

- Ce type de réseau neuronal a généralement plus d'une couche, de préférence deux couches
- Dans ce type de réseau, la distance relative de n'importe quel point au centre est calculée et la même est transmise à la couche suivante

- Les réseaux à base radiale sont généralement utilisés dans les systèmes de restauration de l'alimentation pour rétablir l'alimentation dans les plus brefs délais afin d'éviter les pannes. [16]

9.3. Perceptron multicouche

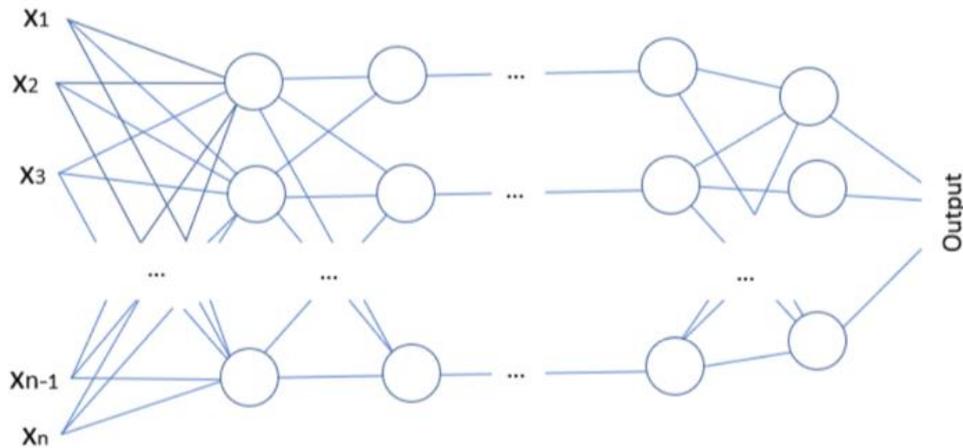


Figure II.6 : Perceptron multicouche [16]

- Ce type de réseau a plus de 3 couches et est utilisé pour classer les données qui ne sont pas linéaires
- Ces types de réseaux sont entièrement connectés à chaque nœud.
- Ces réseaux sont largement utilisés pour la reconnaissance vocale et d'autres technologies d'apprentissage automatique. [16]

9.4. Réseau de neurones à convolution (CNN)

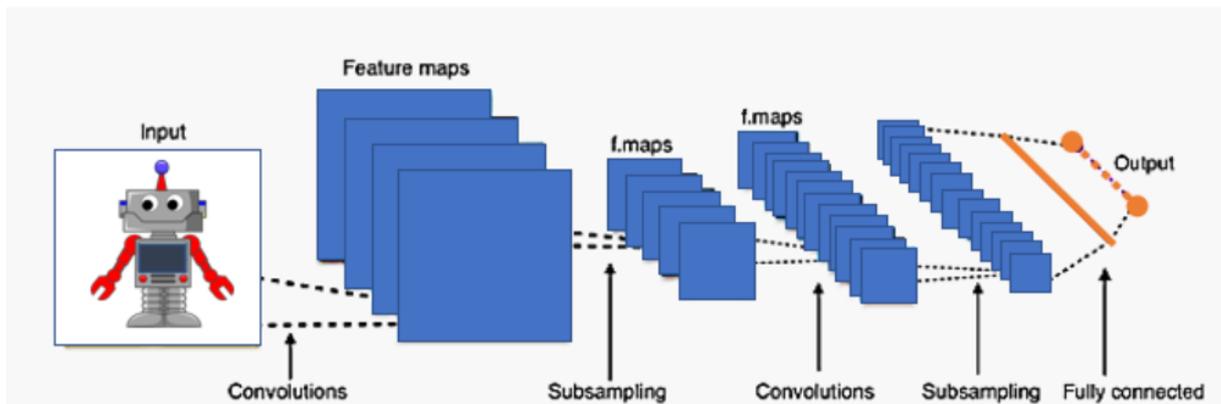


Figure II.7 : Schéma d'un Réseau de neurones à convolution (CNN) [16]

- CNN est l'une des variantes du perceptron multicouche.
- CNN peut contenir plus d'une couche de convolution et puisqu'il contient une couche de convolution, le réseau est très profond avec moins de paramètres.

- CNN est très efficace pour la reconnaissance d'images et l'identification de différents modèles d'images. [16]

9.5. Réseau neuronal récurrent

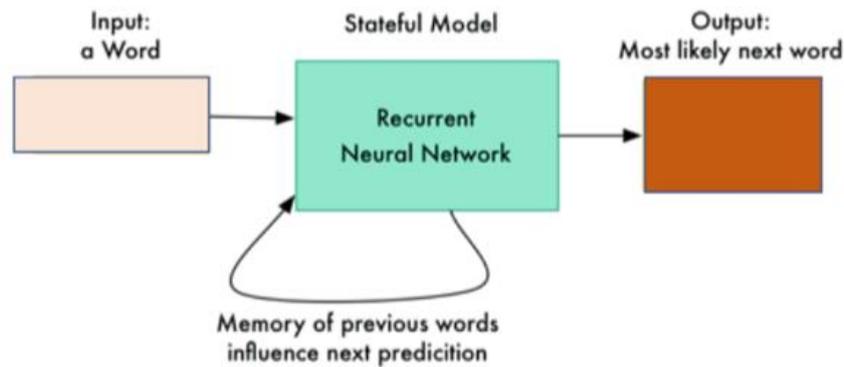


Figure II.8 : Schéma d'un Réseau neuronal récurrent[16]

- RNN est un type de réseau neuronal où la sortie d'un neurone particulier est renvoyée en tant qu'entrée au même nœud.
- Cette méthode aide le réseau à prédire la sortie.
- Ce type de réseau est utile pour maintenir un petit état de mémoire très utile pour développer le chat bot
- Ce type de réseau est utilisé dans le développement de chat bots et les technologies de synthèse vocale. [16]

9.6. Réseau neuronal modulaire

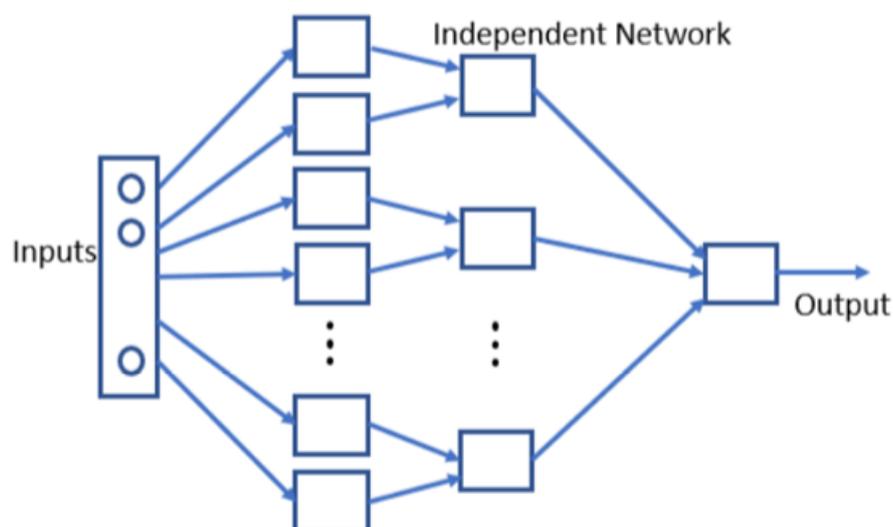


Figure II.9 : Schéma d'un Réseau neuronal modulaire[16]

- Ce type de réseau n'est pas un réseau unique mais une combinaison de plusieurs petits réseaux de neurones.
- Tous les sous-réseaux forment un grand réseau de neurones et tous travaillent indépendamment pour atteindre un objectif commun.
- Ces réseaux sont très utiles pour décomposer le petit-grand problème en petits morceaux et le résoudre ensuite. [16]

9.7. Modèles de séquence à séquence

- Ce type de réseau est généralement une combinaison de deux réseaux RNN.
- Le réseau fonctionne sur l'encodage et le décodage c'est-à-dire qu'il se compose de l'encodeur qui est utilisé pour traiter l'entrée et il y a un décodeur qui traite la sortie
- Généralement, ce type de réseau est utilisé pour le traitement de texte où la longueur du texte entré n'est pas la même que celle du texte sorti. [16]

10. L'intégration du deep Learning dans la super résolution des images

Avec les progrès des techniques d'apprentissage en profondeur ces dernières années, les modèles de SR basés sur l'apprentissage en profondeur ont été activement explorés et atteignent souvent des performances de pointe sur divers critères de SR. Une variété de méthodes d'apprentissage en profondeur a été appliquée pour résoudre des tâches de SR, allant de la première méthode basée sur les réseaux de neurones convolutifs (CNN) aux récentes approches prometteuses de SR basées sur les réseaux antagonistes génératifs.

Il existe de nombreuses méthodes utilisées pour résoudre cette tâche. Nous couvrirons les éléments suivants :

SRCNN

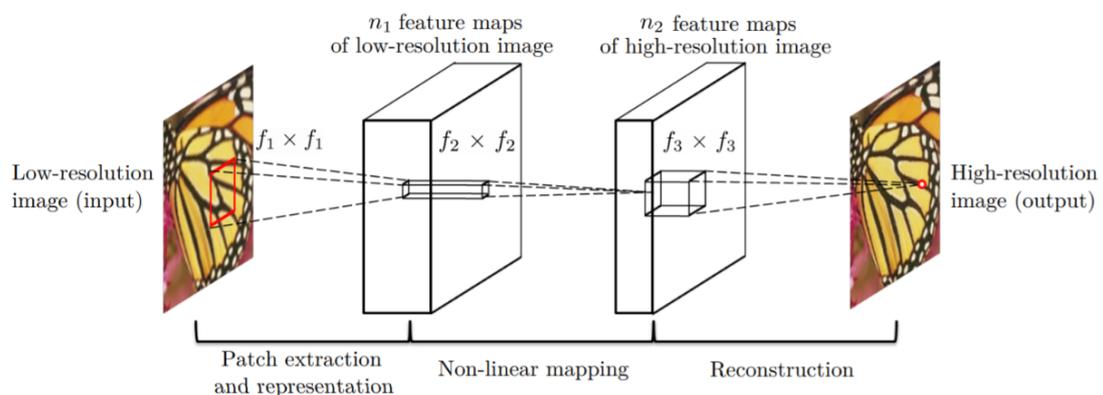


Figure II.10 Architecture du réseau de neurones convolutif à super résolution (SRCNN)[17]

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 patches 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. [17]

VDSR

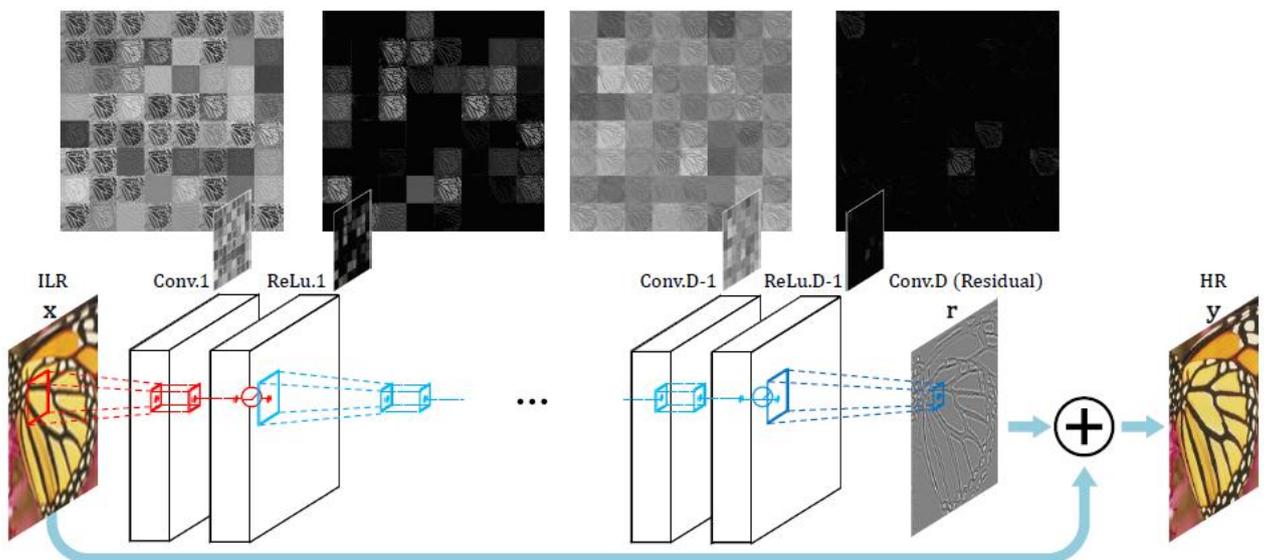


Figure II.11 Architecture du réseau de neurones convolutif profond (VDSR) [18]

Very Deep Super Résolution (VDSR) est une amélioration de SRCNN, avec l'ajout des fonctionnalités suivantes :

Comme son nom l'indique, un réseau profond avec de petits filtres convolutifs 3×3 est utilisé à la place d'un réseau plus petit avec de grands filtres convolutifs. Ceci est basé sur l'architecture VGG.

Le réseau essaie d'apprendre le résidu de l'image de sortie et de l'entrée interpolée, plutôt que d'apprendre le mappage direct (comme SRCNN), comme illustré dans la figure ci-dessus. Cela simplifie la tâche. L'image basse résolution initiale est ajoutée à la sortie réseau pour obtenir la sortie HR finale.

L'écrêtage dégradé est utilisé pour former le réseau profond avec des taux d'apprentissage plus élevés. [18]

FSRCNN

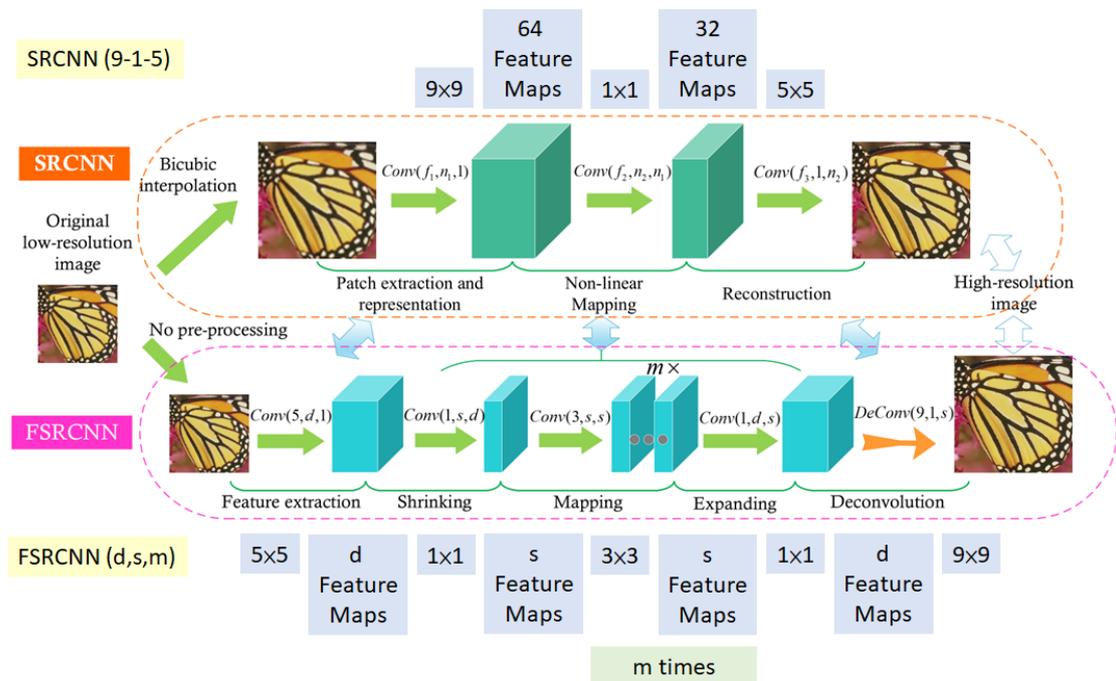


Figure II.12: Comparaison entre SRCNN et FSRCNN [19]

Comme on peut le voir dans la figure ci-dessus, les principaux changements entre SRCNN et FSRCNN sont :

1. Il n'y a pas de prétraitement ni de sur échantillonnage au début. L'extraction des caractéristiques a eu lieu dans l'espace à basse résolution.
 2. Une convolution 1×1 est utilisée après la convolution initiale 5×5 pour réduire le nombre de canaux, et donc moins de calcul et de mémoire, de la même manière que le réseau Inception est développé.
 3. Plusieurs convolutions 3×3 sont utilisées, au lieu d'avoir un grand filtre convolutif, similaire au fonctionnement du réseau VGG en simplifiant l'architecture pour réduire le nombre de paramètres.
 4. Le sur échantillonnage est effectué en utilisant un filtre de convolutif appris, améliorant ainsi le modèle.
- FSRCNN obtient finalement de meilleurs résultats que SRCNN, tout en étant plus rapide. [19]

SRGAN

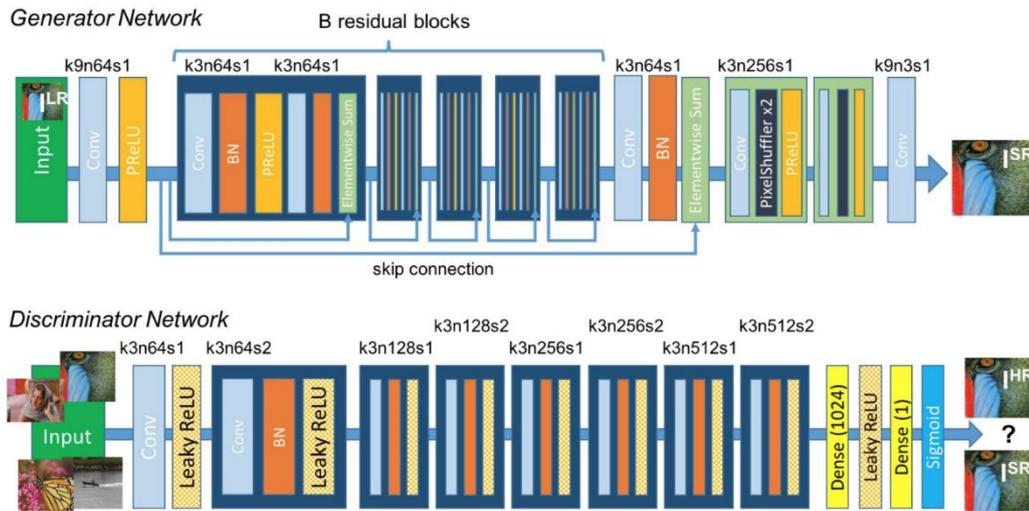


Figure II.13: Architecture du générateur et du réseau discriminateur[20]

Network architecture of SRGAN

SRGAN utilise une architecture basée sur GAN pour générer des images visuellement agréables. Il utilise l'architecture réseau SRResnet comme backend et utilise une perte multitâche pour affiner les résultats. La perte se compose de trois termes :

1. Perte MSE capturant la similarité des pixels
2. Perte de similarité perceptuelle, qui est utilisée pour capturer des informations de haut niveau en utilisant un réseau profond
3. Perte contradictoire du discriminateur

Bien que les résultats obtenus aient des valeurs de PSNR comparativement plus faibles, le modèle a obtenu plus de MOS, c'est-à-dire une meilleure qualité perceptuelle dans les résultats. [20]

EDSR

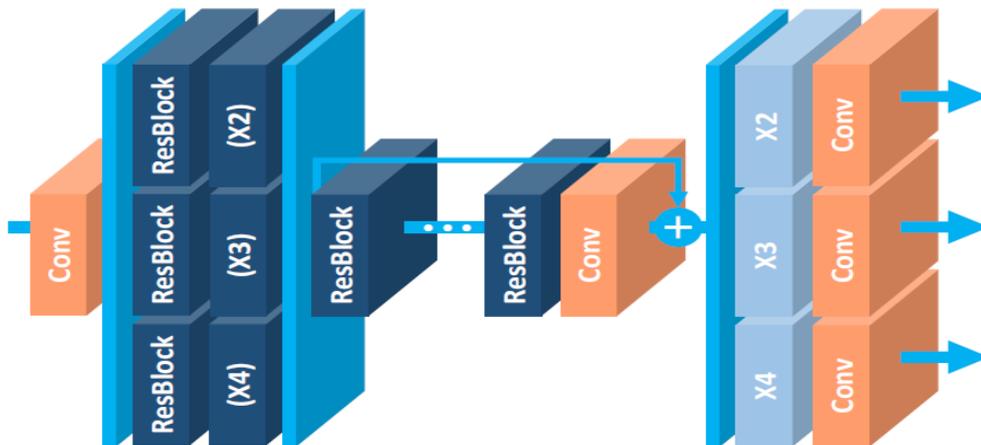


Figure II.14: The architecture of the proposed single-scale SR network (EDSR). [21]

L'architecture EDSR est basée sur l'architecture SRResNet, composée de plusieurs blocs résiduels. Le bloc résiduel dans EDSR est illustré ci-dessus. La principale différence avec SRResNet est que les couches de normalisation par lots sont supprimées. L'auteur déclare que BN normalise l'entrée, limitant ainsi la portée du réseau ; la suppression de BN entraîne une amélioration de la précision. Les couches BN consomment également de la mémoire, et leur suppression entraîne une réduction de mémoire allant jusqu'à 40 %, ce qui rend la formation du réseau plus efficace. [21]

LAPSRN

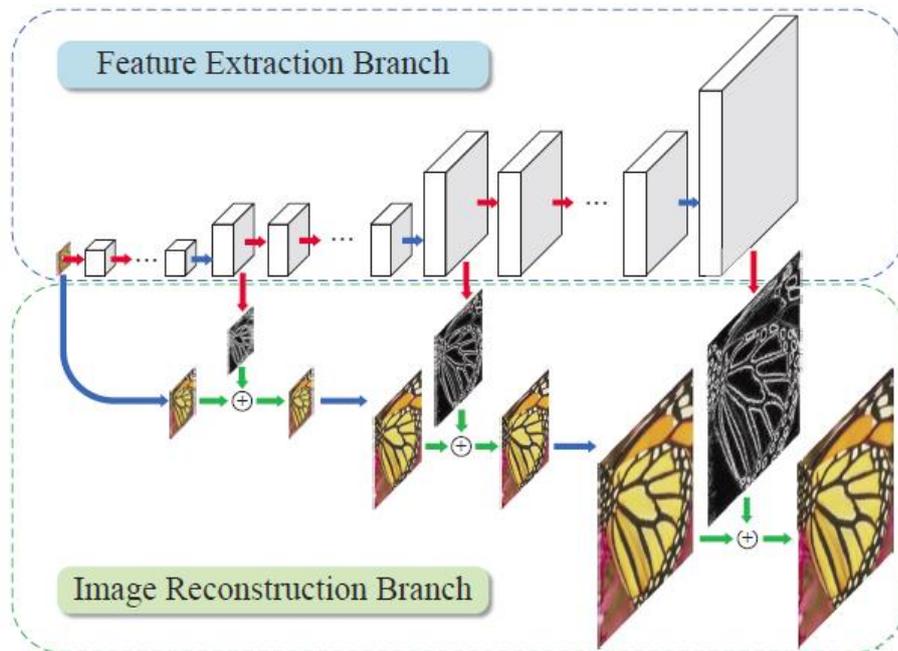


Figure II.15: Architecture LapSRN. [22]

LAPSRN, ou MS-LAPSRN, consiste en une structure pyramidale laplacienne qui peut mettre à l'échelle des images à 2x, 4x et 8x en utilisant une approche étape par étape.

Comme on peut le voir sur la figure ci-dessus, LAPSRN se compose de plusieurs étapes. Le réseau se compose de deux branches : la branche Extraction de caractéristiques et la branche Reconstruction d'images. Chaque étape itérative se compose d'un bloc d'intégration de fonctionnalités et d'un bloc de suréchantillonnage de fonctionnalités, comme illustré dans la figure ci-dessous. L'image d'entrée est passée à travers une couche d'intégration de caractéristiques pour extraire les caractéristiques dans l'espace à basse résolution, qui est ensuite suréchantillonnée à l'aide de la convolution de transposition. La sortie apprise est une image résiduelle qui est ajoutée à l'entrée interpolée pour obtenir l'image haute résolution. La sortie du bloc de suréchantillonnage de fonctionnalités est également transmise à l'étape suivante, qui est utilisée pour affiner la sortie haute résolution

de cette étape et la mettre à l'échelle au niveau suivant. Étant donné que les sorties de résolution inférieure sont utilisées pour affiner les étapes ultérieures, il existe un apprentissage partagé qui aide le réseau à mieux fonctionner. [22]

SelNet

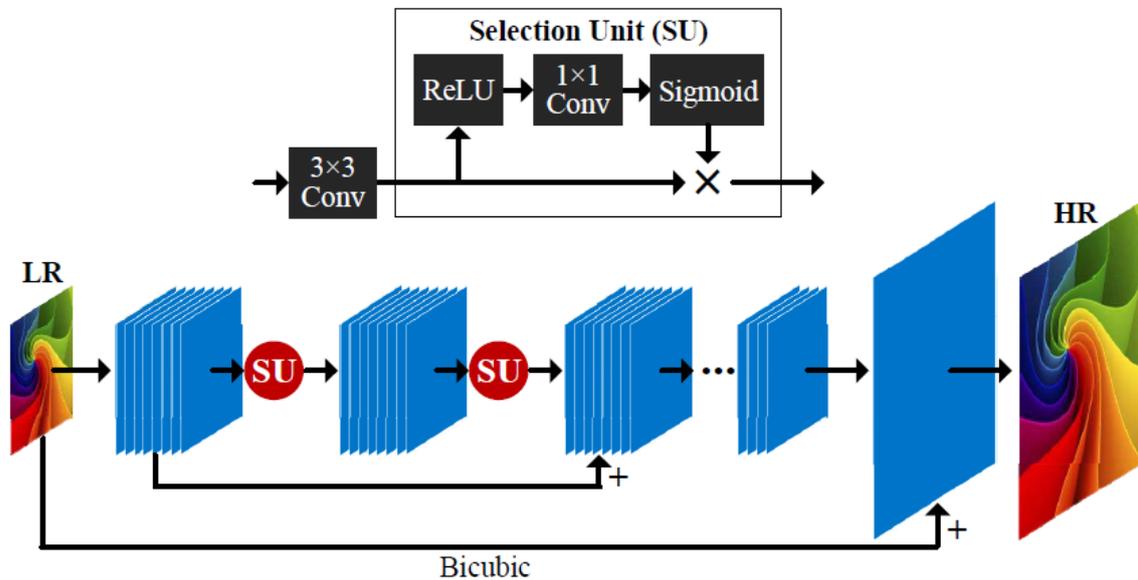


Figure II.16: Unité de sélection (SU) et architecture de réseau SelNet[23]

SelNet propose une nouvelle unité de sélection à la fin des blocs convolutifs qui aide à décider quelles informations transmettre, de manière sélective. Un module de sélection consiste en une activation ReLU suivie d'une convolution 1×1 et d'un déclenchement sigmoïde. Une Unité de Sélection est la multiplication d'un Module de Sélection et d'une connexion d'identité.

Une couche de sous-pixel (similaire à ESPCN) est conservée vers la fin du réseau pour obtenir une mise à l'échelle apprise. Le réseau apprend une image HR résiduelle, qui est ensuite ajoutée à l'entrée interpolée pour obtenir l'image HR finale. [23]

11. Jeux de données

Voici quelques-uns des ensembles de données couramment utilisés pour former des réseaux à super-résolution.

1. **DIV2K** : est un jeu de données super-résolution à image unique populaire qui contient 1 000 images avec différentes scènes et est divisé en 800 pour la formation, 100 pour la validation et 100 pour les tests. Il a été collecté pour les Super-Resolution Challenges afin d'encourager la recherche sur la super-résolution d'image avec une dégradation plus

- réaliste. Ce jeu de données contient des images basse résolution avec différents types de dégradations. [22]
2. Le jeu de données **Urban100** contient 100 images de scènes urbaines. Il est couramment utilisé comme ensemble de test pour évaluer les performances des modèles de super-résolution. [23]
 3. Le jeu de données **Set14** est un jeu de données composé de 14 images couramment utilisées pour tester les performances des modèles de super-résolution d'image. [26]
 4. Le jeu de données **Set5** est un jeu de données composé de 5 images ("bébé", "oiseau", "papillon", "tête", "femme") couramment utilisées pour tester les performances des modèles de super-résolution d'image. [27]
 5. **BSD** est un ensemble de données fréquemment utilisé pour le débruitage d'image et la super-résolution. Parmi les sous-ensembles de données, BSD100 est un ensemble de données d'images classiques comprenant 100 images de test proposées par Martin et al. L'ensemble de données est composé d'une grande variété d'images allant des images naturelles à des objets spécifiques tels que des plantes, des personnes, des aliments. [28]

12. Fonctions de perte

Les diverses fonctions de perte qui peuvent être utilisées pour former les réseaux.

1. Perte de pixels : il s'agit du type de fonction de perte le plus simple et le plus courant utilisé dans la formation de réseaux à super résolution. L2, L1 ou une métrique de différence est utilisée pour évaluer le modèle. L'entraînement avec perte de pixels optimise le PSNR, mais n'optimise pas directement la qualité perceptuelle, et génère donc des images qui pourraient ne pas plaire à l'œil humain.

2. Perte de perception : la perte de perception tente de faire correspondre les caractéristiques de haut niveau d'une image générée avec une image de sortie HR donnée. Ceci est réalisé en prenant un réseau préformé, comme VGG, et en utilisant la différence des sorties de caractéristiques entre les images prédites et de sortie comme perte. Cette fonction de perte est introduite dans SRGAN.

3. Perte Charbonnier : Cette fonction de perte est utilisée dans LapSRN au lieu de la perte L2 générique. Les résultats montrent que la perte de Charbonnier traite mieux les valeurs aberrantes et produit des images plus nettes par rapport à celles générées avec la perte L2, qui sont généralement plus lisses.

4. Perte de texture : Introduite dans EnhanceNet, cette fonction de perte tente d'optimiser la matrice Gram des sorties de fonctionnalités inspirées de la fonction de

perte de transfert de style. Cette fonction de perte entraîne le réseau à capturer les informations de texture dans une image HR.

5. Perte contradictoire : utilisée dans toutes les architectures liées au GAN, la perte contradictoire aide à tromper le discriminateur et produit généralement des images qui ont une meilleure qualité de perception. ESRGAN en ajoute une variante supplémentaire en utilisant le discriminateur relativiste, et en demandant ainsi au réseau non seulement de rendre les fausses images plus réelles, mais aussi de rendre les vraies images plus fausses. [24]

13. Métrique

Les différentes mesures utilisées pour comparer les performances des différents modèles.

1.PSNR : Le rapport signal/bruit de crête est la technique la plus couramment 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).

$$\text{MSE} = \frac{1}{N} \sum_{i=1}^N (I(i) - \hat{I}(i))^2,$$

$$\text{PSNR} = 10 \cdot \log_{10} \left(\frac{L^2}{\text{MSE}} \right).$$

2. SSIM : Cette métrique est utilisée pour comparer la qualité perceptuelle de deux images à l'aide de la formule ci-dessous, avec la moyenne (μ), la variance (σ) et la corrélation (c) des deux images.

$$\text{SSIM}(x, y) = \frac{(2\mu_x \mu_y + c_1)(2\sigma_{xy} + c_2)}{(\mu_x^2 + \mu_y^2 + c_1)(\sigma_x^2 + \sigma_y^2 + c_2)}$$

3. MOS : Mean Opinion Score est un moyen manuel de déterminer les résultats d'un modèle, où les humains sont invités à évaluer une image entre 0 et 5. Les résultats sont agrégés et le résultat moyen est utilisé comme métrique. [17]

14. Conclusion

Dans ce chapitre on a présenté les notions importantes qui sont en relation avec l'apprentissage profond (définition, Architectures...etc.). Aussi qu'une vision générale sur

l'apprentissage profond (supervise, non supervise, faiblement supervisées), on a parlé aussi du l'intégration du deep Learning dans la super résolution des images et les méthodes utilisées pour résoudre cette tâche. Le prochain chapitre, traite les détails de la conception, ainsi que la méthode et les outils utilisés pour la réalisation de notre application.

Chapitre 03

*Mise en œuvre et exploitation
d'un apprentissage profond
pour la super-résolution des
images*

1. Introduction

Pour atteindre l'objectif de la super résolution, nous allons déployer le réseau de neurones à convolution à super-résolution (SRCNN) à l'aide de Keras.

Le réseau SRCNN est un réseau de neurones à convolution profonde qui apprend le mappage de bout en bout d'images basse résolution à haute résolution. En conséquence, nous pouvons l'utiliser pour améliorer la qualité d'image des images à faible résolution. [17]

Pour évaluer les performances de ce réseau, nous avons utilisé des métriques de qualité d'image :

- Rapport signal sur bruit de crête (PSNR),
- Erreur quadratique moyenne (MSE), et
- Indice de similarité structurelle (SSIM).

2. Environnement de mise en œuvre adopté :

Le Deep Learning nécessite un matériel spécial et une très puissante (GPU) dont on ne dispose pas, c'est pour cela nous avons utilisé la plateforme google colab pour pouvoir implémenter et valider notre travail :

- **Google Colab** est un service en nuage (cloud) offert par Google (gratuit), basé sur Jupyter Notebook, permettant d'améliorer les compétences de codage en langage de programmation Python. De développer des applications d'apprentissage automatique notamment approfondi à l'aide de bibliothèques populaires telles que Keras, TensorFlow, et Open CV. Cette plateforme permet d'entraîner des modèles de Machine Learning directement dans le cloud.

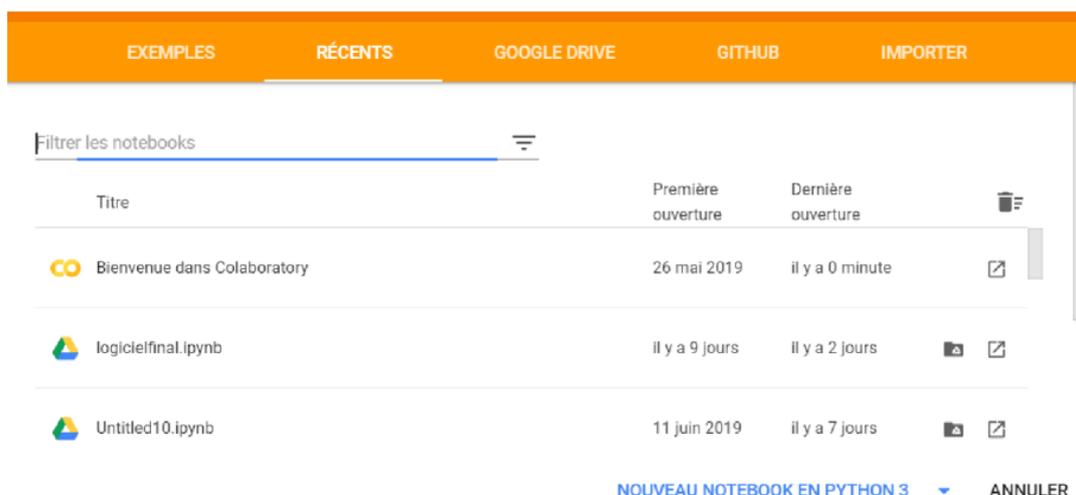


Figure III.1 : Environnement de Google Colab

- **Python**

Python est un langage de programmation de haut niveau utilisé pour la programmation générale. Il fournit des constructions qui permettent une programmation claire à petite et à grande échelle. Python dispose d'un système de type dynamique et d'une gestion automatique de la mémoire. Il prend en charge **tous les paradigmes** de programmation (procédural, orientée objet, fonctionnelle et logique), et dispose d'une bibliothèque standard vaste et complète. Les interpréteurs de Python sont disponibles pour de nombreux systèmes d'exploitation

3. Mise en œuvre

Alors, Pour atteindre notre objectif (SR), nous allons déployer le réseau de neurones à convolution à super-résolution (SRCNN).

SRCNN est une architecture CNN simple composée de trois couches : une pour l'extraction de patch, le mappage non linéaire et la reconstruction.

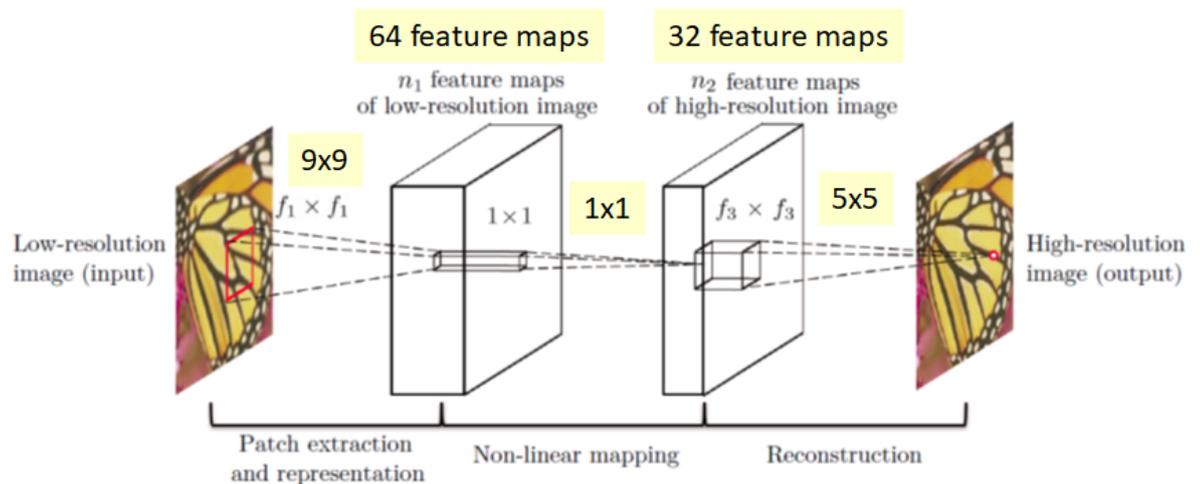


Figure III.2 : Architecture du (SRCNN) [17]

Au cours de ce projet, nous avons appris à :

- Utiliser les métriques de qualité d'image PSNR, MSE et SSIM,
- Traiter les images avec OpenCV,
- Construire des réseaux de neurones profonds à Keras,
- Entraîner le réseau SRCNN
- Évaluer le réseau SRCNN

Préparation de l'environnement de travail

- **Préparation des images**

1. Téléchargement de :

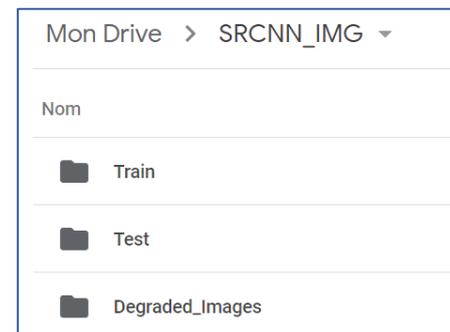
- deux jeux de données (Set14 + Set5.) composés de 19 images couramment utilisées pour tester les performances des modèles de super-résolution d'image.
- 91 images pour l'entraînement du réseau.

A partir de <http://mmlab.ie.cuhk.edu.hk/projects/SRCNN.html> .

2. Création d'un dossier **SRCNN_IMG** Dans **google drive**

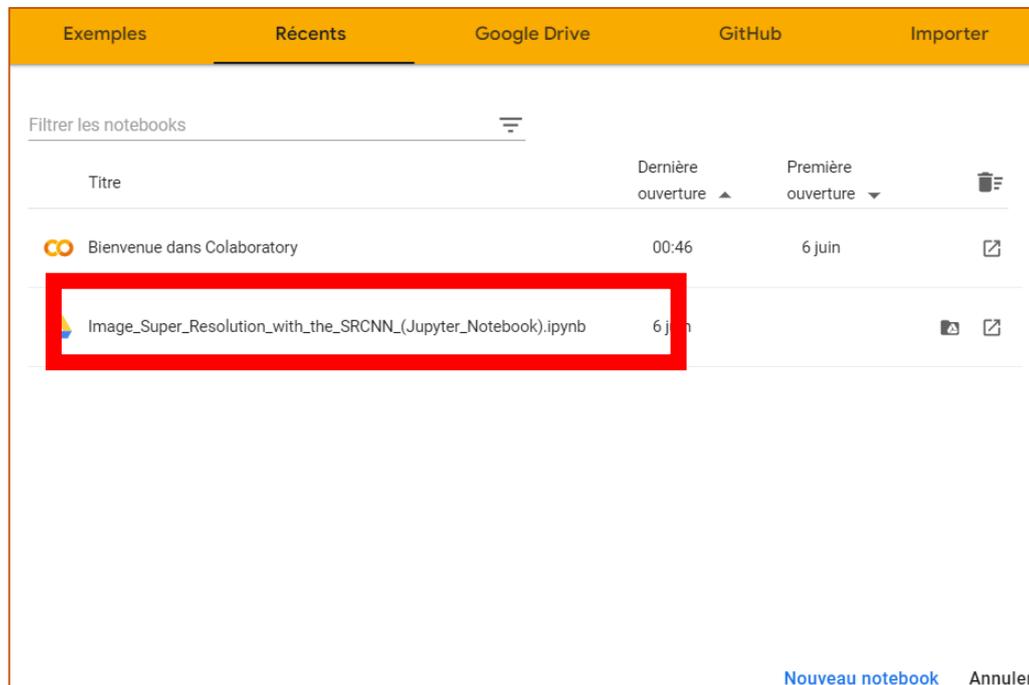
Qui contient 3 autres dossiers

- ✓ **Dossier Train** : qui contient les 91 images
- ✓ **Dossier Test** : qui contient les 19 images
- ✓ **Dossier Degraded_images (vide)**



- **Création d'un nouveau notebook nommé**

image Super Resolution with the SRCNN.ipynb



- **Connecter google colab avec google drive**

```
[ ] from google.colab import drive
drive.mount('/content/drive/')
```

Maintenant, Commençons à travailler sur le code

- **Importation de packages**

Nous importerons les bibliothèques et les packages que nous utiliserons dans ce projet

```
import sys
import keras
import cv2
import numpy
import matplotlib
import skimage
```

1. **Keras** : est une bibliothèque open source écrite en python et permettant d'interagir avec les algorithmes de réseaux de neurones profonds et de machine Learning, notamment Tensorflow.
2. **OpenCV** (pour Open Computer Vision) est une bibliothèque graphique libre, initialement développée par Intel, spécialisée dans le traitement d'images en temps réel.
3. **Numpy** : est une bibliothèque permettant d'effectuer des calculs numériques avec Python. Elle introduit une gestion facilitée des tableaux de nombres, des fonctions sophistiquées (diffusion).
4. **Matplotlib** : est une bibliothèque de traçage pour le langage de programmation Python et son extension mathématique numérique NumPy. Elle fournit une API orientée objet permettant d'incorporer des graphiques
5. **skimage** : est un package Python open source conçu pour le prétraitement d'images.

- **Mesure de qualité d'image**

Pour évaluer les performances de ce réseau, nous utilisons, via une fonction **compare_images**, trois métriques de qualité d'image : le rapport signal/bruit de crête (PSNR), l'erreur quadratique moyenne (MSE) et l'indice de similarité structurelle (SSIM)

```
def psnr(target, ref):
    # assume RGB image
    target_data = target.astype(float)
    ref_data = ref.astype(float)
    print(target_data.shape)
    print(ref_data.shape)
    diff = ref_data - target_data
    diff = diff.flatten('C')
    rmse = math.sqrt(np.mean(diff ** 2.))
    return 20 * math.log10(255. / rmse)

# define function for mean squared error (MSE)
def mse(target, ref):
    err = np.sum((target.astype('float') - ref.astype('float')) ** 2)
    err /= float(target.shape[0] * target.shape[1])
    return err

# define function that combines all three image quality metrics
def compare_images(target, ref):
    scores = []
    scores.append(psnr(target, ref))
    scores.append(mse(target, ref))
    scores.append(ssim(target, ref, multichannel =True))
    return scores
```

Nous préparerons des images dégradées en introduisant des distorsions de qualité via le redimensionnement.

- Dégradation des images (distorsions de qualité) via le redimensionnement avec opencv
- Conception d'une fonction « prepare_images »

```
def prepare_images(path, factor):  
  
    # loop through the files in the directory  
    for file in os.listdir(path):  
  
        # open the file  
        img = cv2.imread(path + '/' + file)  
  
        # find old and new image dimensions  
        h, w, _ = img.shape  
        new_height = int(h / factor)  
        new_width = int(w / factor)  
  
        # resize the image - down  
        img = cv2.resize(img, (new_width, new_height), interpolation =  
int(cv2.INTER_LINEAR))  
  
        # resize the image - up  
        img = cv2.resize(img, (w, h), interpolation = cv2.INTER_LINEAR)  
  
        # save the image  
        print('Saving {}'.format(file))  
        cv2.imwrite('/content/drive/MyDrive/SRCNN_IMG/Degraded_Images/  
{}/'.format(file), img)
```

```
✓ 0s ▶ prepare_images('/content/drive/MyDrive/SRCNN_IMG/Test/Set14Set5', 2)
```

```
Saving lenna.bmp  
Saving baboon.bmp  
Saving zebra.bmp  
Saving barbara.bmp  
Saving flowers.bmp  
Saving baby_GT.bmp  
Saving butterfly_GT.bmp  
Saving woman_GT.bmp  
Saving coastguard.bmp  
Saving monarch.bmp
```

- **Test des images basses résolution**

Pour nous assurer que nos métriques de qualité d'image sont calculées correctement et que les images ont été effectivement dégradées, Nous avons **calculé le PSNR, MSE et SSIM** entre nos images de référence et les images dégradées que nous venons de préparer.

```
# tester les images générées en utilisant les métriques de qualité d'image
for file in os.listdir('/content/drive/MyDrive/SRCNN_IMG/Degraded_Images/'):
    # images cibles et de référence ouvertes
    target = cv2.imread('/content/drive/MyDrive/SRCNN_IMG/Degraded_Images/{
}'.format(file))
    ref = cv2.imread('/content/drive/MyDrive/SRCNN_IMG/Test/Set14Set5/{
}'.format(file))

    # calculer le score
    scores = compare_images(target, ref)

    # affiche les trois partitions
    print('{}\nPSNR: {}\nMSE: {}\nSSIM: {}\n'.format(file, scores[0], scores[1], scores[2]))
```

```
baby_GT.bmp
PSNR: 34.3718064097
MSE: 71.2887458801
SSIM: 0.935698787272
```

```
barbara.bmp
PSNR: 25.9066298376
MSE: 500.655085359
SSIM: 0.809863264641
```

```
bird_GT.bmp
PSNR: 32.8966447287
MSE: 100.123758198
SSIM: 0.953364486603
```

```
butterfly_GT.bmp
PSNR: 24.7820765603
MSE: 648.625411987
SSIM: 0.879134476384
```

- **Construction du modèle SRCNN**

Maintenant que nos images basse résolution sont disponibles et que les trois mesures de qualité d'image fonctionnent correctement, nous construisons le modèle SRCNN. Dans Keras, c'est aussi simple que d'ajouter des calques les uns après les autres.

Comme on a vu dans la **Figure III.3** l'architecture du réseau SRCNN contient 3 couches

- 1 ère couche → 64 filtres avec taille (9 sur 9)
- 2 -ème couche → 32 filtres avec taille (1 sur 1)
- 3 -ème couche → 1 filtre avec taille (5 sur 5)

```
# définir le modèle SRCNN
def model():

    # définir le type de modèle
    SRCNN = Sequential()

    # ajouter des calques de modèle
    SRCNN.add(Conv2D(filters=64, kernel_size = (9, 9), kernel_initializer='glorot_uniform', activation='relu', padding='valid', use_bias=True, input_shape=(None, None, 1)))

    SRCNN.add(Conv2D(filters=32, kernel_size = (1, 1), kernel_initializer='glorot_uniform', activation='relu', padding='same', use_bias=True))

    SRCNN.add(Conv2D(filters=1, kernel_size = (5, 5), kernel_initializer='glorot_uniform', activation='linear', padding='valid', use_bias=True))

    # définir l'optimiseur
    adam = Adam(lr=0.0003)

    # modèle de compilation

    SRCNN.compile(optimizer=adam, loss='mean_squared_error', metrics=['mean_squared_error'])

    return SRCNN
```

- **SRCNN Training**

Et maintenant, l'étape la plus importante est l'étape d'entraînement du notre réseau (SRCNN)

- Lecture les fichiers (Train/ Test) contenant les images dont nous avons besoin
- création d'un fichier 'SRCNN_check.h5' pour contenir les poids calculés (résultat d'entraînement)
- Lancement de l'entraînement avec `srcnn_model.fit (...)` pour commencer l'entraînement

Nombre d'Epochs utilisée est **200** (1 Epoch Lorsqu'un ensemble de données entier est passé une seule fois dans le réseau de neurones)

```

from keras.callbacks import ModelCheckpoint

srcnn_model = model()
print(srcnn_model.summary())
data, label = read_training_data("./crop_train.h5")
val_data, val_label = read_training_data("./test.h5")

checkpoint = ModelCheckpoint("SRCNN_check.h5", monitor='val_loss', verbose=1, save_best_only=True,
                             save_weights_only=False, mode='min')
callbacks_list = [checkpoint]

history = srcnn_model.fit(data, label, batch_size=128, validation_data=(val_data, val_label),
                          callbacks=callbacks_list, shuffle=True, epochs=200, verbose=0)
    
```

```

Epoch 1: val_loss improved from inf to 0.00212, saving model to SRCNN_check.h5
Epoch 2: val_loss improved from 0.00212 to 0.00152, saving model to SRCNN_check.h5
Epoch 3: val_loss improved from 0.00152 to 0.00132, saving model to SRCNN_check.h5
Epoch 4: val_loss improved from 0.00132 to 0.00127, saving model to SRCNN_check.h5
      ↓
Epoch 199: val_loss did not improve from 0.00101
Epoch 200: val_loss did not improve from 0.00101
    
```

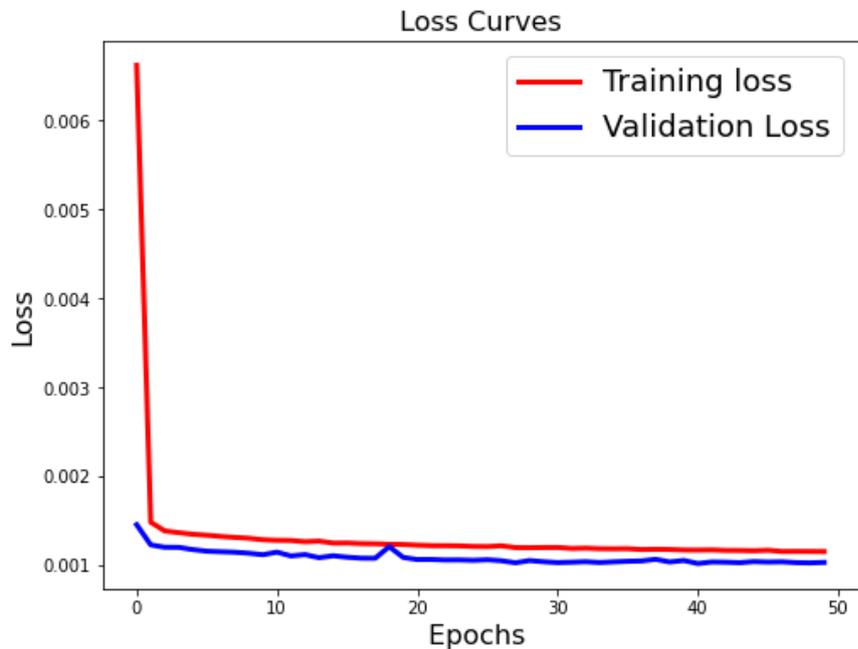


Figure III.3 : Perte d'entraînement/perte de validation

4. Résultats obtenus

Et enfin, après avoir défini notre modèle, nous pouvons l'utiliser pour la super-résolution sur toutes nos images d'entrée. De plus, après traitement, nous pouvons calculer le PSNR, MSE et SSIM sur les images que nous produisons.

```
ref, degraded, output, scores = predict('/content/drive/MyDrive/SRCNN/DEGRADE
D/ butterfly_GT.bmp')

# imprimer tous les scores pour toutes les images
print('Degraded Image: \nPSNR: {}\nMSE: {}\nSSIM: {}\n'.format(scores[0][0],
scores[0][1], scores[0][2]))
print('Reconstructed Image: \nPSNR: {}\nMSE: {}\nSSIM: {}\n'.format(scores[1]
[0], scores[1][1], scores[1][2]))

# afficher les images sous forme de sous-parcelles
fig, axs = plt.subplots(1, 3, figsize=(20, 8))
axs[0].imshow(cv2.cvtColor(ref, cv2.COLOR_BGR2RGB))
axs[0].set_title('Original')
axs[1].imshow(cv2.cvtColor(degraded, cv2.COLOR_BGR2RGB))
axs[1].set_title('Degraded')
axs[2].imshow(cv2.cvtColor(output, cv2.COLOR_BGR2RGB))
axs[2].set_title('SRCNN')

for ax in axs:
    ax.set_xticks([])
    ax.set_yticks([])
```

```
Degraded Image:
PSNR: 24.752462626436284
MSE: 653.0634118967453
SSIM: 0.8788410517538944
```

```
Reconstructed Image:
PSNR: 29.697470976575204
MSE: 209.14838621909328
SSIM: 0.9465641938618182
```



Le résultat obtenu sur les images du test Set5+ Set14 (19 images)

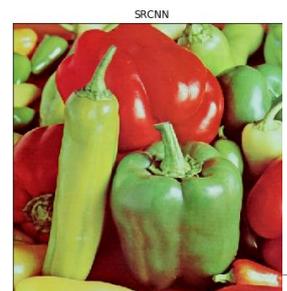
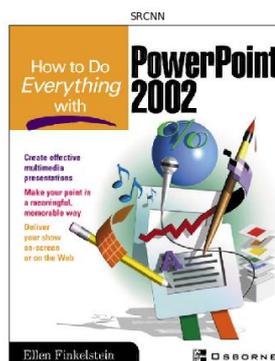
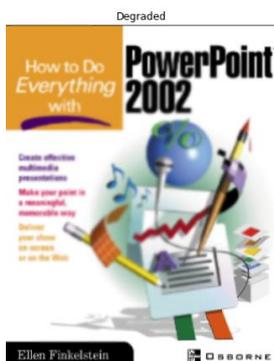
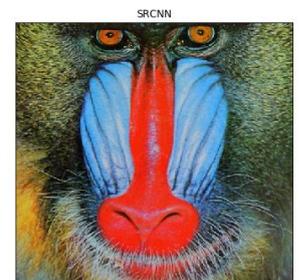
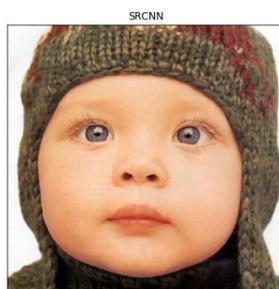
Nom d'image		PSNR	MSE	SSIM
butterfly_GT	Dégradé	24.752	653.063	0.878
	SRCNN	29.697	209.148	0.946
baboon.bmp	Dégradé	22.285	1152.628	0.632
	SRCNN	23.008	975.738	0.707
baby_GT.bmp	Dégradé	34.252	73.270	0.934
	SRCNN	36.110	47.772	0.954
barbara.bmp	Dégradé	25.888	502.708	0.808
	SRCNN	26.526	434.062	0.851
bird_GT.bmp	Dégradé	32.971	98.406	0.953
	SRCNN	36.162	47.197	0.966
coastguard.bmp	Dégradé	27.368	357.534	0.752
	SRCNN	28.745	260.408	0.825
comic.bmp	Dégradé	23.635	844.542	0.829
	SRCNN	25.566	541.464	0.893
face.bmp	Dégradé	30.882	159.2134	0.800
	SRCNN	31.576	135.680	0.8262
flowers.bmp	Dégradé	27.2486	367.564	0.869
	SRCNN	29.3835	224.824	0.896
foreman.bmp	Dégradé	32.0448	121.819	0.938
	SRCNN	35.482	55.206	0.960
lenna.bmp	Dégradé	31.358	142.673	0.845
	SRCNN	32.987	98.043	0.867
monarch.bmp	Dégradé	30.044	193.101	0.943

	SRCNN	34.3108	72.295	0.963
Papper.bmp	Dégradé	31.447	139.791	0.837
	SRCNN	32.633	106.373	0.8487
ppt3.bmp	Dégradé	24.652	668.194	0.9251
	SRCNN	26.999	389.227	0.9438
woman_GT.bmp	Dégradé	29.382	224.880	0.9337
	SRCNN	32.961	98.650	0.9612
zebra.bmp	Dégradé	27.776	325.49	0.8922
	SRCNN	30.714	168.630	0.933
man.bmp	Dégradé	27.142	376.682	0.820
	SRCNN	28.953	248.2304	0.8743
bridge.bmp	Dégradé	25.948	495.859	0.7821
	SRCNN	27.315	361.979	0.851
head_GT.bmp	Dégradé	30.914	158.0296	0.8264
	SRCNN	31.595	135.084	0.8264

Moyenne PSNR de la méthode SRCNN : 30,56438421

Moyenne MSE de la méthode SRCNN : 242,6321263

Moyenne SSIM de la méthode SRCNN : 0,889084211





5. Discussion

Grâce à sa structure légère, SRCNN a atteint des performances supérieures par rapport aux dernières approches. Nous supposons que des performances supplémentaires peuvent être obtenues en explorant davantage de filtres et de stratégies d'entraînement différents.

- 1^{ère} couche → 128 filtres avec taille (9 sur 9)
- 2^{-ème} couche → 64 filtres avec taille (3 sur 3)
- 3^{-ème} couche → 1 filtre avec taille (5 sur 5)

```
def model():
    .
    .
    .
    # ajouter des calques de modèle
    SRCNN.add(Conv2D(filters=128, kernel_size = (9, 9), kernel_initializer='glorot_uniform', activation='relu', padding='valid', use_bias=True, input_shape=(None, None, 1)))

    SRCNN.add(Conv2D(filters=64, kernel_size = (3, 3), kernel_initializer='glorot_uniform', activation='relu', padding='same', use_bias=True))

    SRCNN.add(Conv2D(filters=1, kernel_size = (5, 5), kernel_initializer='glorot_uniform', activation='linear', padding='valid', use_bias=True))
    .
    .
    .
    return SRCNN
```

- Des poids préformés pour le SRCNN, inclus dans le fichier **3051crop_weight_200.h5**, [30] peuvent être chargés pour nous faire gagner du temps de formation de notre réseau neuronal profond

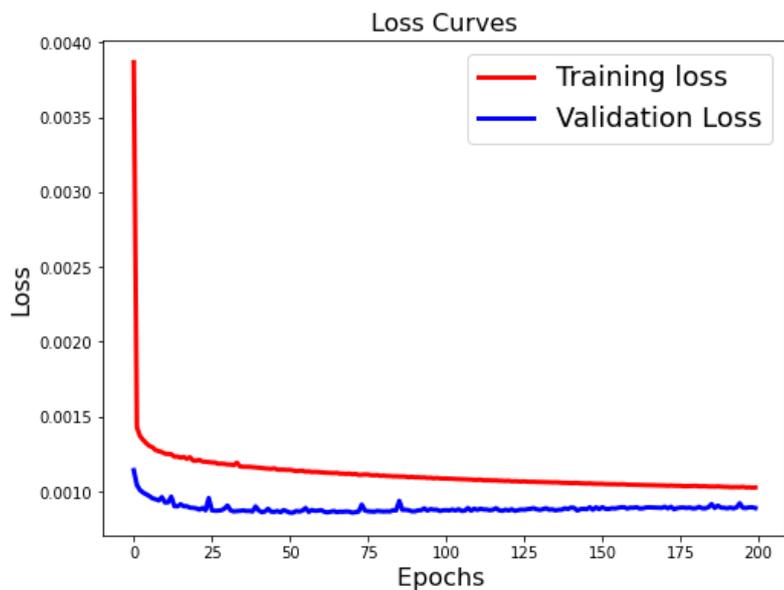
`srcnn.load_weights('/3051crop_weight_200.h5')`

Training 2:

```

Epoch 1: val_loss improved from inf to 0.00114, saving model to SRCNN_check.h5
Epoch 2: val_loss improved from 0.00114 to 0.00104, saving model to SRCNN_check.h5
Epoch 3: val_loss improved from 0.00104 to 0.00101, saving model to SRCNN_check.h5
Epoch 4: val_loss improved from 0.00101 to 0.00099, saving model to SRCNN_check.h5

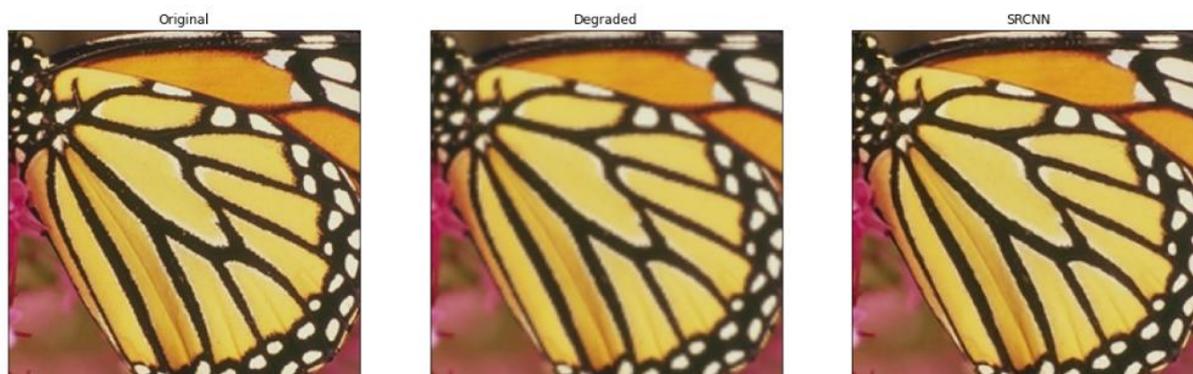
      ↓
Epoch 198: val_loss did not improve from 0.00086
Epoch 199: val_loss did not improve from 0.00086
Epoch 200: val_loss did not improve from 0.00086
    
```



• **Résultat obtenu**

PSNR: 24.752462626436284
 MSE: 653.0634118967453
 SSIM: 0.8788410517538944

Reconstructed Image:
 PSNR: 30.38213886603309
 MSE: 178.64370982552802
 SSIM: 0.9519721343623928



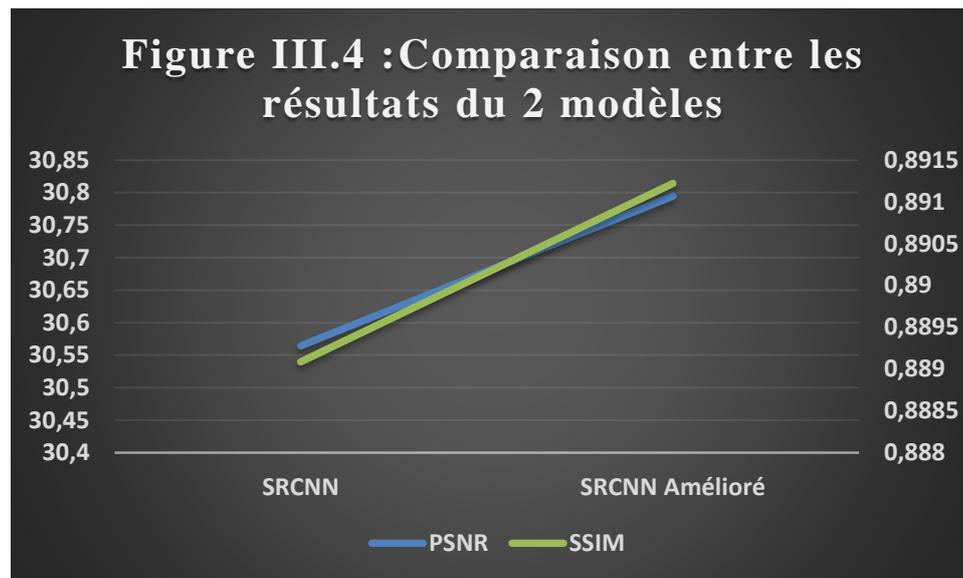
Nom d'image		PSNR	MSE	SSIM
butterfly_GT	Dégradé	24.752	653.063	0.878
	SRCNN	30.382	178.643	0.951
baboon.bmp	Dégradé	22.285	1152.628	0.632
	SRCNN	23.0356	23.0356	0.709
baby_GT.bmp	Dégradé	34.252	73.270	0.934
	SRCNN	36.259	46.162	0.9549
barbara.bmp	Dégradé	25.888	502.708	0.808
	SRCNN	26.575	429.164	0.855
bird_GT.bmp	Dégradé	32.971	98.406	0.953
	SRCNN	36.547	43.199	0.969
coastguard.bmp	Dégradé	27.368	357.534	0.752
	SRCNN	28.674	264.724	0.829
comic.bmp	Dégradé	23.635	844.542	0.829
	SRCNN	25.882	0.899	0.893
face.bmp	Dégradé	30.882	159.2134	0.800

	SRCNN	31.641	133.684	0.828
flowers.bmp	Dégradé	27.2486	367.564	0.869
	SRCNN	29.660	210.951	0.898
foreman.bmp	Dégradé	32.0448	121.819	0.938
	SRCNN	36.146	47.371	0.962
lenna.bmp	Dégradé	31.358	142.673	0.845
	SRCNN	33.108	95.357	0.868
monarch.bmp	Dégradé	30.044	193.101	0.943
	SRCNN	35.132	59.829	0.965
Pepper.bmp	Dégradé	31.447	139.791	0.837
	SRCNN	32.783	102.770	0.849
ppt3.bmp	Dégradé	24.652	668.194	0.9251
	SRCNN	27.143	376.613	0.947
woman_GT.bmp	Dégradé	29.382	224.880	0.9337
	SRCNN	33.626	84.627	0.965
zebra.bmp	Dégradé	27.776	325.49	0.8922
	SRCNN	30.406	177.661	0.932
man.bmp	Dégradé	27.142	376.682	0.820
	SRCNN	28.953	248.2304	0.8743
bridge.bmp	Dégradé	25.948	495.859	0.7821
	SRCNN	27.475	348.895	0.856
head_GT.bmp	Dégradé	30.914	158.0296	0.8264
	SRCNN	31.669	132.807	0.828

Moyenne PSNR avec la méthode SRCNN Amélioré : 30,79455789

Moyenne MSE avec la méthode SRCNN Amélioré : 158,138

Moyenne SSIM avec la méthode SRCNN Amélioré : 0,891221053



Au vu des résultats, notre proposition « d'exploiter davantage les filtres pour des performances supplémentaires » permet effectivement d'améliorer davantage une amélioration de la résolution des images.

Le Code Source :

<https://colab.research.google.com/drive/1WOu04CAOQ5pf6OeqADjHIKcnCH-Kvg0n?usp=sharing>

6. Conclusion

Ce chapitre, représente une description générale de la mise en œuvre de notre travail. Nous avons présenté Google colab et Les propriétés qu'il offre, ainsi que le langage de programmation Python utilisé.

Nous avons ensuite détaillé le modèle qui a servi dans notre étude, puis présenté et discuté les résultats obtenus. Enfin, nous avons apporté une amélioration à notre modèle SRCNN pour de meilleurs résultats.



Conclusion général

Conclusion Générale :

La finalité de ce travail est de reconstruire des images haute résolution (HR) à partir d'images basse résolution (LR). Nous avons donné une explication complète sur l'apprentissage en profondeur et les réseaux de neurones.

Et puis nous avons parlé d'intégrer le deep Learning avec la super-résolution, et Pour atteindre notre objectif, nous avons construit un réseau de neurones à convolution à super-résolution (SRCNN) à l'aide de keras et l'entraîner.

Nous avons obtenu de bons résultats sur les données utilisées, les images de test de basse résolution ont été nettement et visiblement améliorées en utilisant la technique proposée.

Au final, nous avons supposé que des performances supplémentaires pouvaient être obtenues en explorant davantage de filtres et de stratégies d'entraînement différents. Notre hypothèse s'est vérifiée et nous avons obtenu de meilleurs résultats.

Les technologies de la super résolution des images sont toujours d'actualité. L'amélioration de la qualité d'une image est une tâche continue, et le résultat obtenu par un algorithme peut être utilisé par un autre algorithme pour augmenter encore sa qualité. L'amélioration de la qualité des séquences vidéo est un autre domaine prometteur pour la super résolution.



Bibliographie

- [1] : <https://fr.wikipedia.org/wiki/Image>
- [2] : https://fr.wikipedia.org/wiki/Image_num%C3%A9rique
- [3] : **Manseur Ali, (2019)** L'apprentissage profond (Deep Learning) pour la classification des lames anapath numérisées
- [4] : <https://www.informatique-enseignant.com/image-bitmap-ou-vectorielle/>
- [5] : <https://www.simplilearn.com/image-processing-article>
- [6] : <https://www.shimmercat.com/image-api-feature/superresolution.html>
- [7] : https://homepages.inf.ed.ac.uk/rbf/CVonline/LOCAL_COPIES/AV1011/Super_Resolution_CVonline.pdf
- [8] : <https://blog.paperspace.com/image-super-resolution/>
- [9] : <https://www.mdpi.com/2227-7390/8/11/1907/htm>
- [10] : <https://master-iesc-angers.com/artificial-intelligence-machine-learning-and-deep-learning-same-context-different-concepts/>
- [11] : <https://blog.quantinsti.com/introduction-deep-learning-neural-network/>
- [12] : <https://www.jedha.co/formation-ia/vraie-difference-machine-learning-deep-learning>
- [13] : <https://www.futura-sciences.com/tech/definitions/intelligence-artificielle-deep-learning-17262/>
- [14] : <https://www.ionos.fr/digitalguide/web-marketing/search-engine-marketing/quest-ce-quun-reseau-neuronal-artificiel/>
- [15] : <https://datascientest.com/deep-neural-network>
- [16] : <https://www.educba.com/deep-learning-networks/>
- [17] : **Chao Dong, Chen Change Loy, Kaiming He, Xiaoou Tang**, Image Super-Resolution Using Deep Convolutional Networks , 31 Jul 2015
- [18] : **Jiwon Kim, Jung Kwon Lee and Kyoung Mu Lee , Accurate**, Image Super-Resolution Using Very Deep Convolutional Networks , Seoul National University, Korea 11 Nov 2016
- [19] : **Chao Dong, Chen Change Loy, and Xiaoou Tang** , Accelerating the Super-Resolution Convolutional Neural Network , Department of Information Engineering, The Chinese University of Hong Kong , 1 Aug 2016

- [20] : **Christian Ledig, Lucas Theis, Ferenc Huszar, Jose Caballero, Andrew Cunningham, Alejandro Acosta, Andrew Aitken, Alykhan Tejani, Johannes Totz, Zehan Wang, Wenzhe Shi** , Photo-Realistic Single Image Super-Resolution Using a Generative Adversarial Network , 25 May 2017
- [21] : **Bee Lim, Sanghyun Son, Heewon Kim , Seungjun Nah , Kyoung Mu Lee** , Enhanced Deep Residual Networks for Single Image Super-Resolution , Department of ECE, ASRI, Seoul National University, 08826, Seoul, Korea , 10 Jul 2017
- [22] : **Wei-Sheng Lai, Jia-Bin Huang, Narendra Ahuja, and Ming-Hsuan Yang** , Fast and Accurate Image Super-Resolution with Deep Laplacian Pyramid Networks , 9 Aug 2018
- [23] : **Chaofeng , Wang Zheng Li , Jun Shi** , Lightweight Image Super-Resolution with Adaptive Weighted Learning Network , Shanghai University Shanghai, China , 4 Apr 2019
- [24] : <https://blog.paperspace.com/image-super-resolution/>
- [25] : <https://paperswithcode.com/dataset/div2k>
- [26] : <https://paperswithcode.com/dataset/urban100>
- [27] : <https://paperswithcode.com/dataset/set14>
- [28] : <https://paperswithcode.com/dataset/set5>
- [29] : <https://paperswithcode.com/dataset/bsd>
- [30] : https://drive.google.com/file/d/14unwod8IhjMiBtTEGafqRSC6SdY_Tre2/view?usp=sharing