

UNIVERSITÉ ABDELHAMID BEN BADIS-MOSTAGANEM

Faculté des Sciences Exactes et d'Informatique

Département de mathématiques et informatique

Filière: Informatique

RAPPORT DE PROJET DE MASTER EN INFORMATIQUE

Option: **Intelligence Artificielle pour l'internet
des objets (IA4IoT)**

**THÈME: Fusion d'Exposition pour l'Amélioration de la
Qualité d'Image**

Présenté par :

- KECIR Ibrahim El Khalil

Soutenu le: samedi 14/06/2025 ,Devant le jury composé de :

Dr.Bessaoud Karim	Université de Mostaganem	Président
Dr.Lattar hafsa	Université de Mostaganem	Examineur
Dr.Moumene Mohammed El Amine	Université de Mostaganem	Encadreur

Remerciements

Avant toute chose, je tiens à exprimer ma gratitude profonde à **Dieu** Tout-Puissant, source de toute sagesse et de toute force, qui m'a guidé tout au long de mon parcours universitaire et m'a permis d'atteindre cette étape importante de ma vie.

Je remercie chaleureusement tous les enseignants qui m'ont accompagné depuis la première année (L1) jusqu'à la fin de mon Master (M2). Chacun d'eux, à travers ses cours, ses conseils et sa bienveillance, a contribué à forger ma compréhension du domaine de l'informatique et à éveiller ma passion pour l'intelligence artificielle.

Mes remerciements les plus sincères vont à **monsieur Moumene Mohammed El Amine**, mon encadrant, pour son accompagnement, sa disponibilité, sa patience et ses précieux conseils qui ont grandement contribué à la réalisation de ce projet. Son expertise et son soutien ont été d'une aide inestimable.

J'adresse également ma reconnaissance à tout **le département d'informatique** de notre université, pour la qualité de la formation reçue et les moyens mis à disposition pour nous permettre de progresser dans un environnement scientifique stimulant. Enfin, je remercie avec respect **les membres du jury** pour le temps qu'ils consacrent à l'évaluation de mon travail, ainsi que pour leurs remarques constructives qui ne manqueront pas d'enrichir davantage mon projet.

Dédicaces

Aujourd'hui, je souhaite exprimer ma profonde gratitude envers toutes les personnes qui ont contribué, de près ou de loin, à la réalisation de ce projet. C'est avec beaucoup d'émotion que je dédie ce travail à ceux qui ont été mes sources d'inspiration, mes guides et mes soutiens indéfectibles.

Tout d'abord, je dédie ce mémoire à **mes chers parents**. Votre amour inconditionnel, votre patience et votre soutien constant ont été la force motrice derrière chacun de mes pas. Merci de m'avoir toujours cru capable, même dans les moments de doute. Votre présence, vos prières et vos sacrifices m'ont accompagné tout au long de ce parcours.

À **mes enseignants et mentors**, je vous exprime ma plus profonde reconnaissance.

Grâce à votre dévouement, votre exigence bienveillante et votre passion pour l'enseignement, j'ai pu acquérir des connaissances solides et développer une vraie rigueur scientifique. Merci pour vos conseils, vos encouragements et vos critiques constructives qui m'ont toujours poussé à aller plus loin.

Je dédie également ce travail à **mes camarades de classe et amis**, pour tous les échanges, les travaux de groupe, les moments de stress partagés et surtout l'entraide constante. Vous avez rendu ce parcours académique plus riche, plus humain et plus agréable.

Enfin, je n'oublie pas toutes les personnes qui, d'une manière ou d'une autre, ont contribué à l'accomplissement de ce mémoire. Qu'il s'agisse d'un conseil technique, d'un mot de soutien ou d'un simple geste d'encouragement, chaque contribution a compté.

À vous tous, je dédie ce travail avec une sincère reconnaissance et une profonde gratitude.

Liste des figures

<i>Figure</i>	<i>Titre de La Figure</i>	<i>Page</i>
<i>Figure 1.2</i>	<i>Image photographique</i>	<i>9</i>
<i>Figure 1.3</i>	<i>Image Matriciel Vectoriel</i>	<i>9</i>
<i>Figure 1.4.1</i>	<i>image présentant l'affichage des pixels</i>	<i>10</i>
<i>Figure 1.4.3</i>	<i>image présentant Taille d'une image</i>	<i>11</i>
<i>Figure 1.4.5</i>	<i>image présentant Contraste dans une images</i>	<i>12</i>
<i>Figure 1.4.6</i>	<i>image présentant Contraste dans une images</i>	<i>12</i>
<i>Figure 1.4.7</i>	<i>image présentant Contours avec plusieurs algorithme</i>	<i>13</i>
<i>Figure 1.4.8</i>	<i>image présentant Histogramme d'une image</i>	<i>14</i>
<i>Figure 1.5.1</i>	<i>image présentant image binaire</i>	<i>15</i>
<i>Figure 1.5.2</i>	<i>image présentant image en Niveaux de Gris</i>	<i>15</i>

Figure 1.5.3	<i>image présentant en Couleur</i>	16
Figure 1.6.1	<i>les domaines l'Intelligence Artificielle</i>	17
Figure 1.6.1	<i>ia faible vs ia forte</i>	18
Figure 1.7.2	<i>Catégories d'apprentissage automatique</i>	22
Figure 2.4.1	<i>Biological Neural Network vs Artificial Neural Network</i>	25
Figure 2.2	<i>Images ayant diverses expositions.</i>	24
Figure 2.6.4	<i>Structure de U-Net</i>	35
Figure 3.1.4	<i>Biological Neural Network vs Artificial Neural Network</i>	39
Figure 3.1.6	<i>Architecture de réseau neuronal simple</i>	40
Figure 3.1.7	<i>Architecture de réseau neuronal profond</i>	41
Figure 3.2.2.3	<i>les fonctions d'activation</i>	44
Figure 3.2.2.4	<i>Types de pooling</i>	45
Figure 3.2.2.6	<i>fonction Batch normalization</i>	46
Figure 3.3.1	<i>google colab</i>	49
Figure 3.4	<i>scènes aléatoires "mon dataset"</i>	50
Figure 3.6	<i>diagramme courbe de perte du modèle</i>	51
Figure 3.7	<i>résultats de model les images fusionnées</i>	55

Liste des abréviation

<i>Mot abrégé</i>	<i>Signification</i>
RVB / RGB	red-green-blue
HDR	High Dynamic Range
IA	intelligence artificiel
ML	Machine learning
DL	Deep learning
SVM	Support Vector Machine
GOF AI	good old-fashion artificial intelligence
CNN	Convolutional Neural Network
RNN	Recurrent neural networks
DNN	Deep Neural Network
ACP	Analyse en composantes principales
SVD	Décomposition en valeurs singulières
ANN	Les réseaux neuronaux artificiels
FC	fully connected layer
BN	Batch Normalization
MSE	Mean Squared Error / Erreur Quadratique Moyenne
MAE	Mean Absolute Error

LDR	Low Dynamic Range
MEF	Multi-Exposure Image Fusion

Table de matières

Remerciements	2
Dédicaces	2
Liste des figures	3
Liste des abréviations	4
Table de matières	5
Introduction Générale	8
Chapitre 1 : Introduction à la Vision par Ordinateur et à l'IA	9
1.1 Introduction	9
1.2 Définition de l'image	9
1.3 Types d'images en vision par ordinateur	9
1.4 Concepts et définitions de base	10
1.4.1 Pixel	10
1.4.2 Codage de pixel	10
1.4.3 Taille d'une image	11
1.4.4 Luminance	11
1.4.5 Contraste	12
1.4.6 Bruit	12
1.4.7 Contour	13
1.4.8 Histogramme	14
1.5 Codage des couleurs en vision par ordinateur	15
1.5.1 Images Binaires	15
1.5.2 Images en Niveaux de Gris	15
1.5.3 Images en Couleur	16
1.6 Introduction à l'Intelligence Artificielle (IA)	16
1.6.1 Qu'est-ce que l'Intelligence Artificielle (IA) ?	16
1.6.2 Historique et évolution de l'Intelligence Artificielle (IA)	18
1.6.3 Applications de l'Intelligence Artificielle (IA)	19
1.7 Apprentissage Automatique (Machine Learning)	20
1.7.1 Définition	20
1.7.2 Méthodes D'apprentissage Automatique (Machine Learning)	20
1.8 Conclusion	22

Chapitre 2 : État de l'Art	23
2.1 Introduction à la Fusion d'Expositions	23
2.2 Définition de la Fusion d'Expositions	23
2.3 Distinction entre la Fusion d'expositions et le HDR	24
2.4 Utilisation et Applications	25
2.5 Objectifs de la Fusion d'Expositions	25
2.5.1 Améliorer la fiabilité:	26
2.5.2 Améliorer la capacité d'information:	26
2.5.3 Optimiser la plage dynamique:	26
2.5.4 Réduire le bruit numérique:	26
2.5.5 Améliorer l'analyse informatique des images:	26
2.6 Techniques de Fusion d'Expositions	26
2.6.1 Approches Traditionnelles de la Fusion d'Expositions	26
2.6.2 Approche proposée par Mertens (Fusion d'exposition)	30
2.6.3 Approche proposée par Moumene, Nourine, Ziou (Fast Exposure Fusion Based on Histograms Segmentation)	32
2.6.4 Approches Avancées	32
Chapitre 3 :Méthodologie et implémentation et résultats	35
3.1 Apprentissage Profond (Deep Learning)	35
3.1.1 Définition du Deep Learning	35
3.1.2 Quelle est la différence entre le Deep learning et le Machine Learning ?	36
3.1.3 Comparaison entre Machine Learning et Deep Learning	36
3.1.4 Définition et principes fondamentaux des réseaux neuronaux artificiels	37
3.1.5 Comment fonctionnent les réseaux neuronaux ?	38
3.1.6 Architecture de réseau neuronal simple	39
3.1.7 Architecture de réseau neuronal profond	40
3.1.8 Quels sont les différents types de réseaux neuronaux ?	40
3.2 Réseaux de Neurones Convolutionnels (CNN)	41
3.2.1 Introduction	41
3.2.2 Fonctionnement et architecture	41
3.2.2.1 Couche d'entrée	41
3.2.2.2 Couche de Convolution (Convolutional Layer)	41
3.2.2.3 Fonction d'Activation (ReLU)	42
3.2.2.4 Couche de Pooling (Sous-échantillonnage)	43
3.2.2.5 Couche Complètement Connectée (Fully Connected Layer)	44
3.2.2.6 Couche de Normalisation (Batch Normalization)	44
3.2.2.7 Fonction de Perte (Loss Function)	45
3.3 Environnement de Développement	46
3.3.1 Google Colab, Python et ses bibliothèques	46
3.4 Préparation des Données	48
3.5 Architecture du Modèle : U-Net amélioré	49
3.5.1 Principe du U-Net	49

3.6 Compilation du modèle	50
3.7 Différence entre mon approche et celle de Meertens et al.	51
3.7.1 Objectif de la méthode	51
3.7.2 Approche technique	51
3.7.3 Architecture	52
3.7.4 Performances et résultats	52
3.7.5 Avantages et limites	53
3.7 Résultats	53
Conclusion et Perspectives	55
Perspectives d'amélioration	56
Conclusion	56
Références	58

Introduction Générale

❖ L'intelligence artificielle et l'essor du deep learning

L'intelligence artificielle (IA) constitue un champ en forte croissance qui cherche à conférer aux machines des aptitudes pour apprendre et prendre des décisions. Depuis sa création dans les années 1950, l'intelligence artificielle a subi de nombreuses transformations, particulièrement avec l'émergence du machine learning et du deep learning.

Également connu sous le nom d'apprentissage profond, le deep learning repose sur l'utilisation de réseaux neuronaux artificiels pour traiter des représentations complexes à partir d'immenses volumes de données. Cette avancée a révolutionné divers domaines, notamment la reconnaissance d'images, l'analyse du langage naturel et l'automatisation des processus industriels.

❖ L'intelligence artificielle et l'amélioration de la qualité d'image

En matière de traitement d'image, l'intelligence artificielle et l'apprentissage profond ont considérablement optimisé la qualité visuelle des images. Des techniques comme la super-résolution, l'atténuation du bruit et la combinaison d'images ont profité des progrès réalisés dans le domaine des réseaux de neurones convolutifs (CNN) et des modèles génératifs (GANs). Ces méthodes permettent la reconstruction d'images de qualité supérieure grâce à des données issues de diverses perspectives.

❖ Fusion d'Exposition et Objectifs du Projet

Améliorer la qualité d'image représente un enjeu important pour plusieurs secteurs, tels que la photographie numérique, la vision par ordinateur et la surveillance. Une solution avancée pour l'amélioration des images pourrait être la fusion d'expositions. Plusieurs images enregistrées à différentes expositions sont combinées pour un résultat final présentant une meilleure représentation du détail et un fort contraste.

L'intégration d'exposition est un terme libéral impliquant plusieurs approches, allant des approches classiques d'ordre mathématique aux solutions contemporaines dont Intelligence artificielle ou Computer Vision font partie. L'apparition de bibliothèques comme OpenCV en Python a été une aubaine pour développer et intéresser ces techniques dans des systèmes sophistiqués de traitement d'images.

Dans le présent projet, nous allons explorer les différentes approches, comparer leurs avantages et leurs inconvénients, et mettre en œuvre des améliorations d'images fusionnées basées sur des méthodes basiques et des algorithmes de deep learning.

Ce document présente un état de l'art des techniques existantes, les outils technologiques utilisés, ainsi que les contributions proposées dans le cadre de ce projet.

Chapitre 1 : Introduction à la Vision par Ordinateur et à l'IA

1.1 Introduction

L'image constitue l'un des éléments fondamentaux en vision par ordinateur, permettant aux machines d'interpréter visuellement le monde qui les entoure. Grâce aux avancées en intelligence artificielle et en traitement d'images, les systèmes informatiques peuvent analyser, segmenter et extraire des informations précieuses à partir d'images numériques. La vision par ordinateur est omniprésente dans divers domaines, tels que la surveillance, la médecine, la robotique et la reconnaissance faciale.

Dans ce chapitre, nous présentons les concepts clés liés à la représentation et au traitement des images dans le contexte de la vision par ordinateur.

1.2 Définition de l'image

Une image est une représentation visuelle d'un objet, d'une scène ou d'un concept. En termes numériques, elle est souvent constituée d'un ensemble de pixels, chacun étant défini par des paramètres tels que la couleur et la luminosité [1] [2]. Les images peuvent être acquises par des capteurs optiques comme les caméras ou les scanners, ou créées directement par des logiciels informatiques [3].

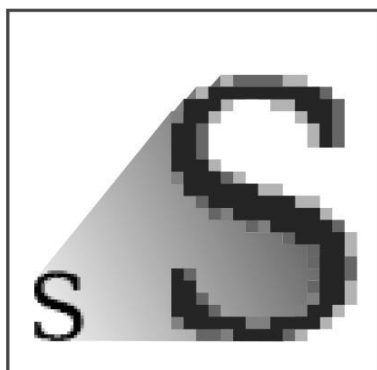


Figure 1.2: Image photographique

1.3 Types d'images en vision par ordinateur

En vision par ordinateur, les images peuvent être classées en plusieurs catégories en fonction de leur structure et de leur mode de représentation :

- **Images vectorielles** : Ces images sont définies par des formules mathématiques qui décrivent des lignes, des courbes et des formes géométriques. Elles sont souvent utilisées pour les logos et les illustrations [1].
- **Images matricielles (bitmap)** : Ces images sont constituées d'une matrice de pixels, où chaque pixel a une valeur numérique représentant sa couleur ou son intensité [1] [3].



Matriciel



Vectoriel

Figure 1.3: Image Matriciel Vectoriel

1.4 Concepts et définitions de base

1.4.1 Pixel

Élément de base d'une image numérique, défini par une valeur d'intensité lumineuse et/ou une couleur.

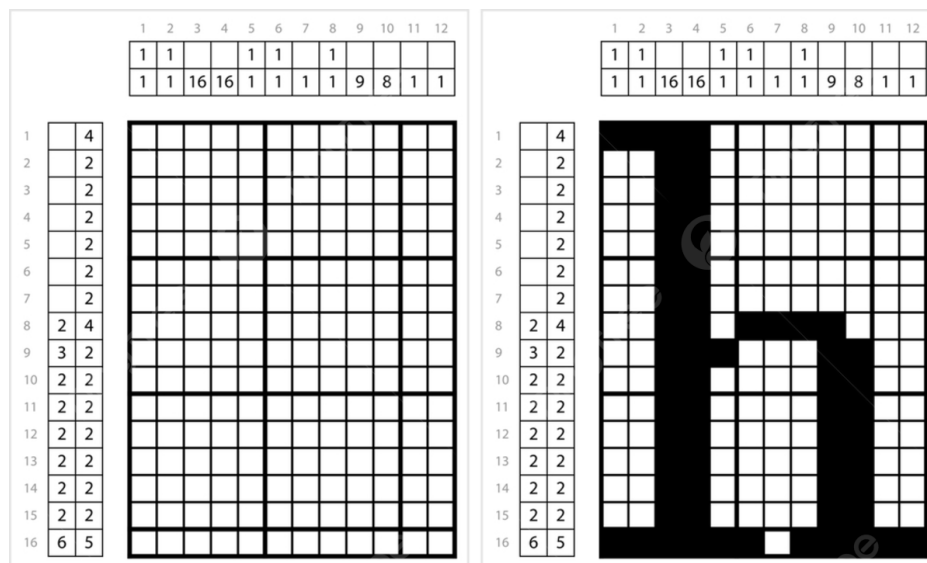


Figure 1.4.1: image présentant l'affichage des pixels

1.4.2 Codage de pixel

Un codage de pixel est plus une définition de la manière dont la couleur et l'intensité des pixels sont représentées dans le numérique d'une image. Elle a des effets directs sur la qualité et sur la richesse des couleurs.

Profondeur d'image

1. **8 bits :**
 - i. 256 options pour un niveau de gris ou une couleur.
 - ii. Pour des images simples, généralement en niveau de gris.
2. **16 bits :**
 - i. 65 536 niveaux de gris ou couleurs.
 - ii. Pour des images de haute qualité où des variations subtiles ont à être perçues.
3. **24 bits :**
 - i. 16,7 millions de couleurs (8 bits par canal pour le système RGB).
 - ii. Pourbonnes images enconcelées colorées avec une bonne profondeur de détails.

À mesure que la profondeur augmente, avec elle, le niveau de qualité visuelle s'accroît et simultanément, le nombre d'I/O dont une image a besoin augmente.

1.4.3 Taille d'une image

Les dimensions d'une image sont exprimées par deux valeurs :

- Largeur en pixels : Nombre de pixels sur l'axe horizontal.
- Hauteur en pixels : Nombre de pixels sur l'axe vertical.

Par exemple, une image de 1920x1080 pixels a une largeur de 1920 pixels et une hauteur de 1080 pixels, ce qui est couramment utilisé pour la résolution HD (haute définition).[3] [4] [5]

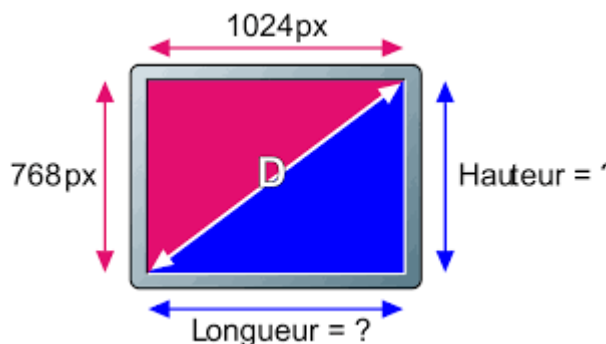


Figure 1.4.3: image présentant Taille d'une image

1.4.4 Luminance

Définition

La luminance est la densité spatiale de flux lumineux visible, qui reflète la sensation visuelle de luminosité d'une surface [6]. En photographie et en traitement d'image, elle est utilisée pour évaluer la brillance d'une couleur ou d'un élément dans une composition [7].

Influence sur la Perception des Contrastes

- Contraste Élevé : Un grand écart entre les zones de luminance différente permet de distinguer facilement les objets et les détails dans une image.
- Contraste faible : Un faible contraste peut rendre les détails plus difficiles à discerner, affectant ainsi la clarté de l'image.

1.4.5 Contraste

Le contraste est une mesure essentielle en photographie et en traitement d'image, représentant la différence entre les parties sombres et claires d'une image. Il joue un rôle crucial dans la perception visuelle, affectant la clarté et la profondeur des images.

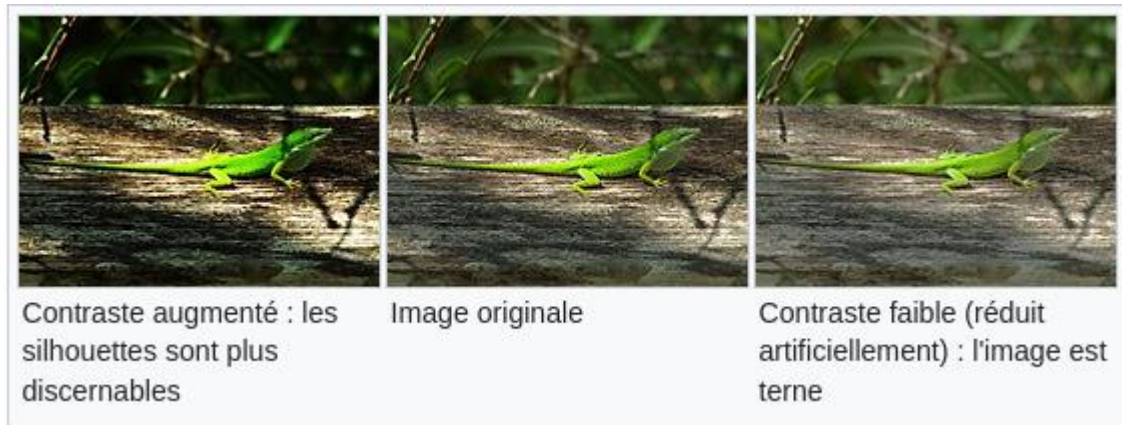


Figure 1.4.5: image présentant Contraste dans une images

1.4.6 Bruit

Le bruit est une variation non désirée des niveaux de gris ou de couleur dans une image.

Il est souvent perçu comme des grains ou des pixels de couleurs aléatoires qui ne correspondent pas aux détails de la scène capturée.

Types de Bruit

Il existe principalement deux types de bruit numérique :

- **Bruit de Luminance** : Visible sous forme de taches plus foncées ou plus claires, donnant un aspect granuleux à l'image. Il affecte principalement la luminosité des pixels [8].
- **Bruit de Chrominance** : Apparaît sous forme de variations de couleur aléatoires, souvent visibles comme des taches colorées réparties sur l'image [8].



Figure 1.4.6: image présentant Contraste dans une images

1.4.7 Contour

Un contour est caractérisé par une rupture ou une discontinuité d'intensité ou de couleur dans une image, souvent suivant une direction spécifique. Il est généralement associé à des changements brusques de luminance ou de couleur entre deux zones adjacentes [9].

Détection des Contours:

La détection des contours implique l'identification des zones où ces changements brusques se produisent. Les algorithmes couramment utilisés incluent :

- **Sobel** : Utilise des masques pour calculer le gradient de l'image et détecter les contours.
- **Canny** : Considéré comme l'un des meilleurs algorithmes pour la détection de contours, il utilise une approche analytique pour optimiser la détection et la localisation des contours [3] [4].
- **Roberts** : Utilise des masques pour détecter les contours en calculant le gradient de l'image.

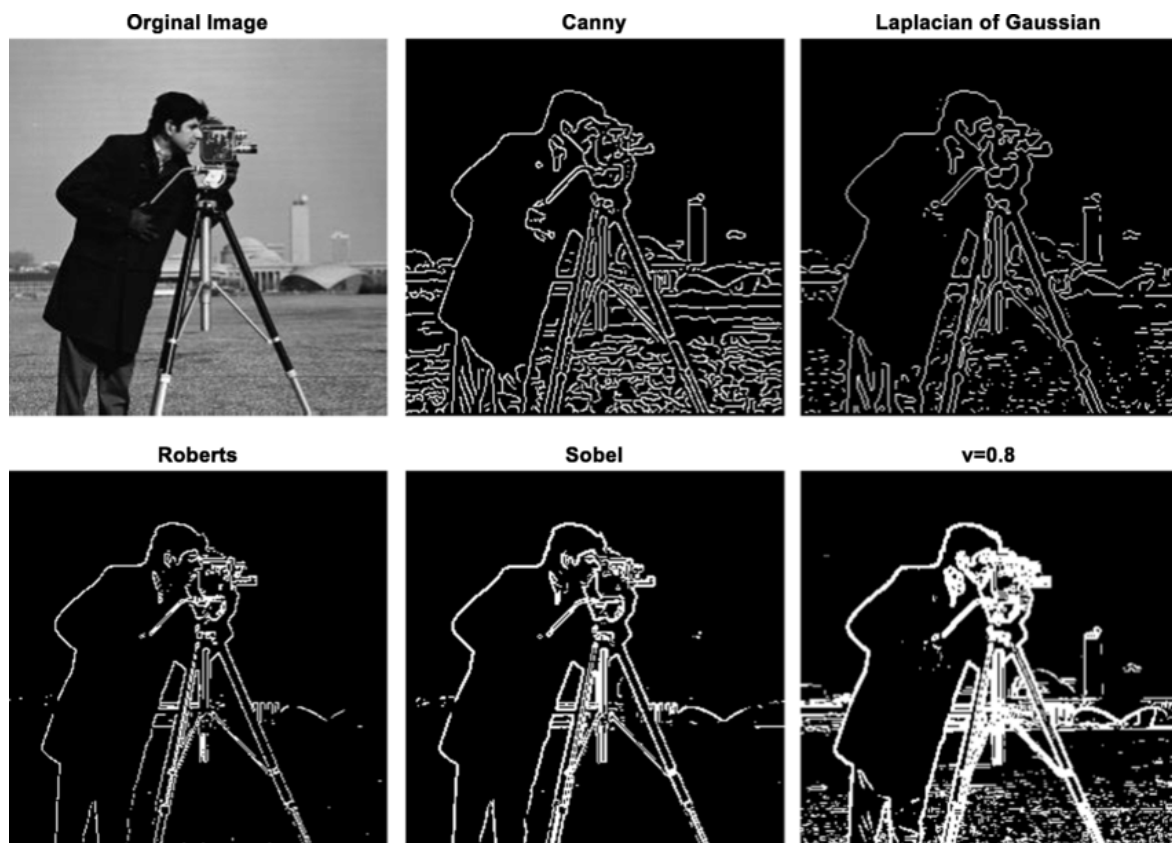


Figure 1.4.7: image présentant Contours avec plusieurs algorithmes

1.4.8 Histogramme

L'histogramme d'une image est défini comme une fonction qui associe à chaque niveau d'intensité le nombre de pixels ayant cette valeur. Pour une image en niveaux de gris, il s'agit généralement d'un graphique avec 256 valeurs possibles (de 0 pour le noir à 255 pour le blanc) sur l'axe des abscisses et le nombre de pixels pour chaque intensité sur l'axe des ordonnées [10] [11].

Types d'Histogrammes

- **Histogramme Monochrome** : Utilisé pour les images en niveaux de gris, il montre la distribution des intensités lumineuses.
- **Histogramme Couleur** : Pour les images en couleur, on peut avoir des histogrammes séparés pour chaque canal de couleur (RVB) ou un histogramme combiné dans l'espace couleur [12].
- **Histogramme Local** : Calculé pour une partie spécifique de l'image, utile pour analyser des zones particulières.

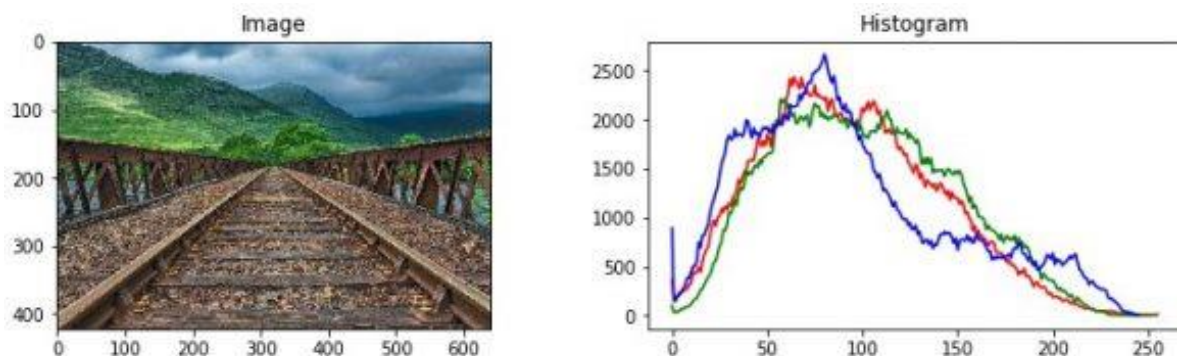


Figure 1.4.8: image présentant Histogramme d'une image

1.5 Codage des couleurs en vision par ordinateur

1.5.1 Images Binaires

- Définition : Composées uniquement de pixels noirs ou blancs, représentés par 0 ou 1.
- Utilisation : Souvent utilisées pour la segmentation d'objets où les différences de couleur ne sont pas nécessaires, seulement les contours et les formes.

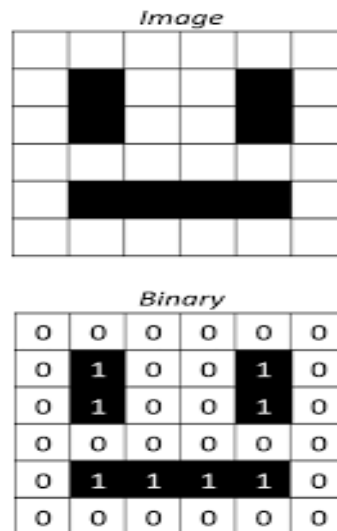


Figure 1.5.1: image présentant image binaire

1.5.2 Images en Niveaux de Gris

- Définition : Chaque pixel est défini par une seule valeur d'intensité lumineuse.
- Codage : Généralement codées sur 8 bits, offrant 256 niveaux de gris (valeurs de 0 à 255), où 0 représente le noir et 255 le blanc.



Figure 1.5.2: image présentant image en Niveaux de Gris

1.5.3 Images en Couleur

- Définition : Utilisent des modèles de couleur pour représenter les différentes couleurs en combinant plusieurs canaux.
- Modèle RVB (Rouge, Vert, Bleu) : Le modèle le plus courant où chaque couleur est une combinaison des trois canaux primaires (RVB). Chaque canal est généralement codé sur 8 bits, permettant ainsi de représenter 16,7 millions de couleurs différentes (256 x 256 x 256). [15]



Figure 1.5.3 : image présentant en Couleur

1.6 Introduction à l'Intelligence Artificielle (IA)

1.6.1 Qu'est-ce que l'Intelligence Artificielle (IA) ?

L'intelligence artificielle (IA) désigne la faculté des systèmes informatiques de reproduire des capacités cognitives humaines, telles que la reconnaissance vocale, le processus de prise de décision ou l'analyse d'images. Le domaine d'étude a été établi dans les années 1950 grâce à des personnalités influentes comme Alan Turing, qui a suggéré le test de Turing pour évaluer la capacité d'une machine à penser comme un être humain, et John McCarthy, l'inventeur du terme « intelligence artificielle », qui l'a présenté lors de la conférence de Dartmouth en 1956.

[13].

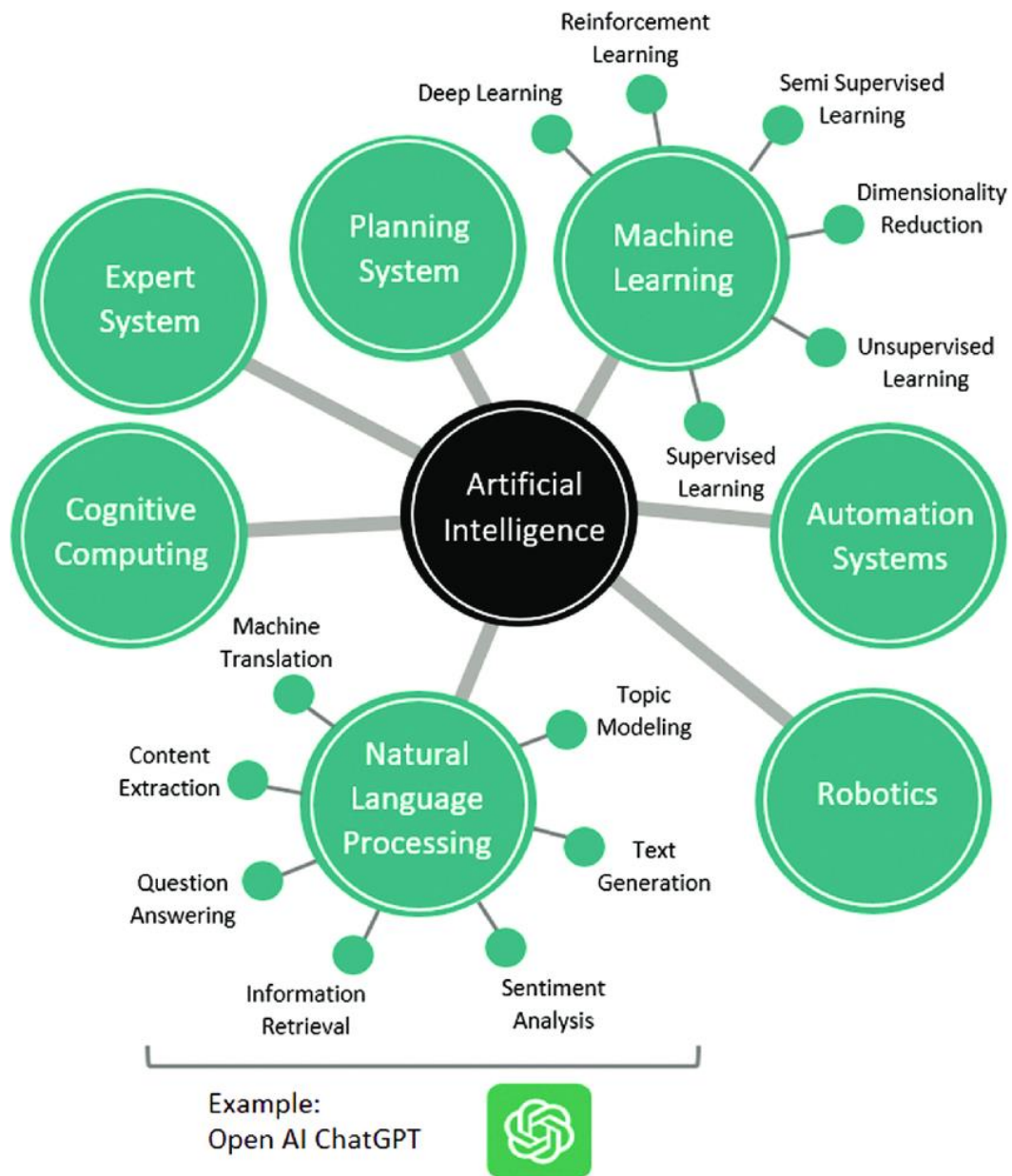


Figure 1.6.1 : les domaines l'Intelligence Artificielle

Les catégories de l'IA:

L'IA peut être divisée en deux grandes catégories :

IA Faible (ou IA étroite)

Destinée à accomplir des missions spécifiques dans un secteur déterminé. Exemples : assistants numériques tels que Siri, systèmes de recommandation sur Netflix, ou chatbots pour l'assistance à la clientèle [26] [27].

Elle manque de conscience et ne peut pas généraliser ses connaissances à d'autres domaines.

IA Forte (ou IA générale):

- Théorique à ce stade, elle vise à reproduire toutes les capacités cognitives humaines.
- Elle pourrait apprendre, comprendre des contextes complexes et résoudre divers problèmes sans limitation de domaine.
- Cette forme d'IA reste un objectif futur dans les recherches scientifiques [26] [27].

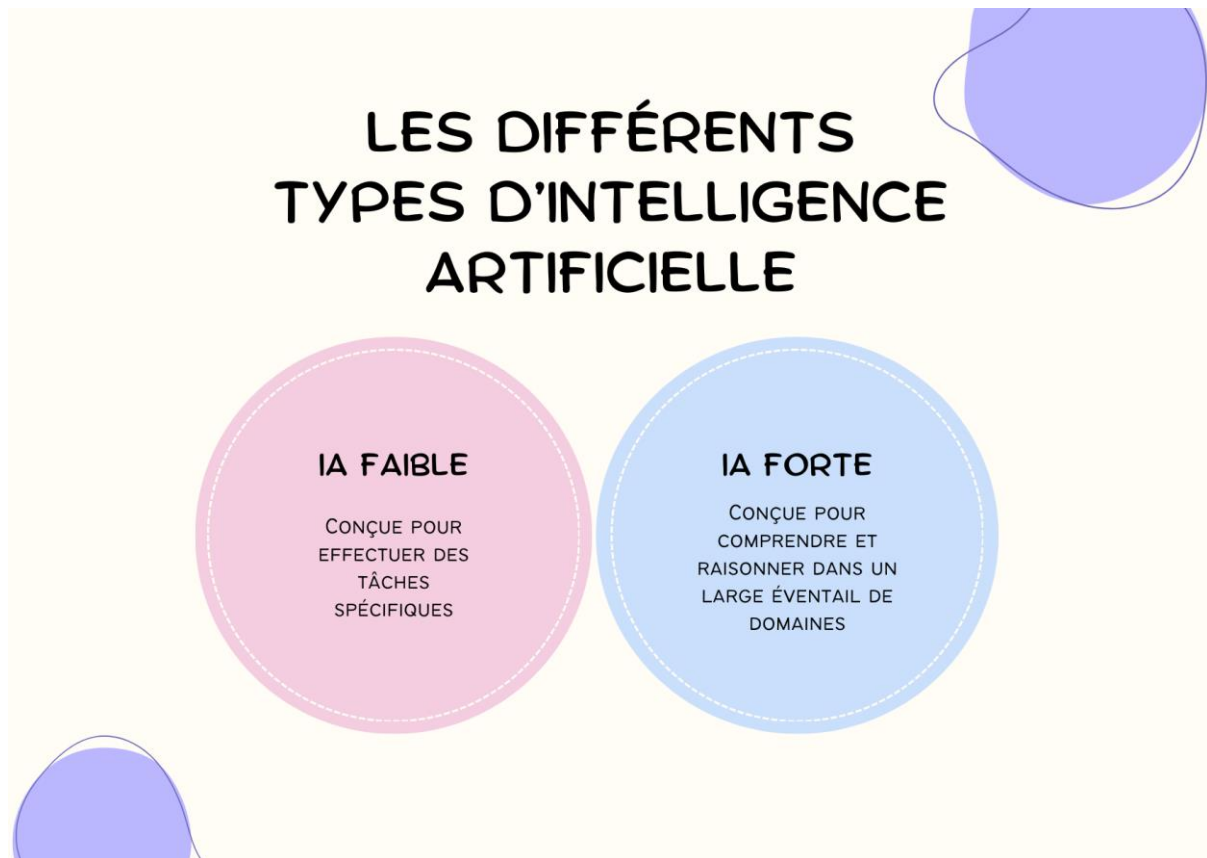


Figure 1.6.1 : ia faible vs ia forte

1.6.2 Historique et évolution de l'Intelligence Artificielle (IA)

L'évolution de l'intelligence artificielle (IA) peut être divisée en plusieurs périodes clés, chacune marquée par des avancées technologiques et des changements de paradigme.

Années 1950–1970 : Débuts de l'IA symbolique:

Fondations : L'IA a été officiellement introduite en 1956 lors de la conférence de Dartmouth, où le terme "intelligence artificielle" a été inventé par John McCarthy. Les premières recherches se concentrent sur les systèmes basés sur des règles et la manipulation de symboles, une approche connue sous le nom d'IA symbolique ou "Good Old-Fashioned AI" (GOFAI) [30] .

Projets emblématiques :

Logic Theorist (1956), développé par Allen Newell et Herbert Simon, capable de prouver des théorèmes mathématiques .

MYCIN (1970), un système expert conçu pour diagnostiquer des maladies du sang et prescrire des traitements [28].

Limites : Ces systèmes étaient rigides et dépendaient fortement de règles explicites, ce qui limitait leur capacité à s'adapter à des situations imprévues [30].

Années 1980–1990 : L'essor des réseaux de neurones artificiels:

Retour des réseaux neuronaux : Après une période d'intérêt limité durant les années 1970 (connue comme l'"hiver de l'IA"), la redécouverte de l'algorithme de rétropropagation a ravivé les recherches sur les réseaux neuronaux artificiels dans les années 1980 [29] [31].

Développements clés :

Introduction des réseaux convolutifs (CNN) pour la reconnaissance d'images.

Popularité croissante des machines à vecteurs de support (SVM) et des réseaux neuronaux récurrents (RNN) [29].

Transition vers une approche basée sur les données : Contrairement aux systèmes experts, cette période marque un passage vers des méthodes apprenant directement à partir des données, ouvrant la voie au machine learning moderne [29].

Années 2000–présent : L'avènement du machine learning et du deep learning

Big Data et GPU : L'augmentation massive des données disponibles via Internet et le développement de processeurs graphiques puissants ont permis l'entraînement de modèles complexes [32] [33].

1.6.3 Applications de l'Intelligence Artificielle (IA)

L'intelligence artificielle est un domaine vaste avec des applications variées dans divers secteurs. Voici quelques-unes des principales applications de l'IA :

1. Vision par ordinateur (ex. reconnaissance faciale)

- Reconnaissance faciale : Utilisée pour la sécurité, le verrouillage des appareils, et la surveillance. Les systèmes de reconnaissance faciale peuvent identifier les individus en comparant leurs traits avec des bases de données existantes [34].
- Applications diverses : La vision par ordinateur est également utilisée dans les voitures autonomes, la détection d'objets, et l'analyse d'images médicales [34].

2. Traitement automatique du langage (chatbots, traduction)

- Chatbots : Ces assistants virtuels utilisent le traitement automatique du langage naturel (NLP) pour interagir avec les utilisateurs, offrant un support client en temps réel et personnalisé [35].
- Traduction automatique : Les technologies NLP permettent la traduction instantanée de textes et de voix, facilitant la communication internationale [36].

3. Santé (diagnostic médical assisté)

- Diagnostic médical : L'IA analyse des données médicales pour détecter des maladies, comme le cancer, et identifier des prédispositions génétiques. Les modèles de deep learning améliorent la précision et la rapidité des diagnostics .
- Imagerie médicale : L'IA aide à analyser des images médicales pour détecter des anomalies et réduire les erreurs humaines [37].

4. Transport (voitures autonomes)

- Voitures autonomes : L'IA est utilisée pour la navigation, la détection d'objets, et la prise de décision en temps réel, permettant aux véhicules de se déplacer sans intervention humaine [34].

5. Sécurité (détection d'intrusions, cybersécurité)

- Détection d'intrusions : Les systèmes IA peuvent identifier les menaces potentielles en analysant les modèles de trafic réseau et détecter les attaques informatiques.
- Cybersécurité : L'IA aide à protéger les systèmes contre les attaques en analysant les comportements suspects et en mettant en place des mesures préventives.

Ces applications montrent comment l'IA transforme divers secteurs en améliorant l'efficacité, la précision, et la sécurité.

1.7 Apprentissage Automatique (Machine Learning)

1.7.1 Définition

Le machine learning (ML) est une branche de l'intelligence artificielle (IA) qui vise à permettre aux ordinateurs et aux machines d'imiter la façon dont les humains apprennent, d'effectuer des tâches de manière autonome et d'améliorer leur performance et leur précision à mesure qu'ils gagnent en expérience et qu'ils sont exposés à de nouvelles données. [38]

1.7.2 Méthodes D'apprentissage Automatique (Machine Learning)

Apprentissage supervisé:

L'apprentissage supervisé, également connu sous le nom de machine learning supervisé, se définit par l'utilisation de jeux de données étiquetés pour entraîner des algorithmes à classer des données ou à prédire des résultats avec précision. À mesure que les données d'entrée alimentent le modèle, ce dernier y applique des pondérations jusqu'à obtenir le bon ajustement. Cette étape intervient dans le cadre du processus de validation croisée pour s'assurer que le modèle évite tout surajustement ou sous-ajustement. L'apprentissage supervisé permet aux entreprises de résoudre divers problèmes concrets à grande échelle, comme le classement des spams dans un dossier autre que la boîte de réception. Parmi les méthodes utilisées dans l'apprentissage supervisé, citons les réseaux neuronaux, les classificateurs Bayes naïfs, la régression linéaire, la régression

logistique, la forêt d'arbres décisionnels et les algorithmes SVM (machine à vecteurs de support).

Apprentissage non supervisé:

L'apprentissage non supervisé, également connu sous le nom de machine learning non supervisé, utilise des algorithmes de machine learning pour analyser et regrouper des jeux de données non étiquetées (sous-ensembles appelés clusters). Ces algorithmes découvrent des modèles ou des groupes de données cachés sans intervention humaine. La capacité de l'apprentissage non supervisé à découvrir des similitudes et des différences dans les informations le rend idéal pour l'analyse exploratoire des données, les stratégies de vente croisée, la segmentation de la clientèle et la reconnaissance d'images et de formes. Il permet également de réduire le nombre de caractéristiques dans un modèle, avec la réduction de dimensionnalité. L'analyse en composantes principales (ACP) et la décomposition en valeurs singulières (SVD) sont deux approches courantes utilisées à cet effet. Parmi les autres algorithmes utilisés dans l'apprentissage non supervisé, citons les réseaux neuronaux, le clustering k-means ou partitionnement en k-moyennes et le regroupement probabiliste.

Apprentissage semi-supervisé

L'apprentissage semi-supervisé offre un juste milieu entre l'apprentissage supervisé et l'apprentissage non supervisé. Pendant l'entraînement, il utilise un jeu de données étiqueté plus petit pour guider la classification et l'extraction des caractéristiques à partir d'un jeu de données non étiqueté plus volumineux. L'apprentissage semi-supervisé peut être utile lorsque les données étiquetées sont insuffisantes pour un algorithme d'apprentissage supervisé. Ce type d'apprentissage constitue également une bonne alternative quand l'étiquetage d'un volume suffisant de données est trop coûteux.

Apprentissage par renforcement

L'apprentissage par renforcement est un modèle de machine learning similaire à l'apprentissage supervisé, à la différence que l'algorithme n'est pas entraîné sur des données d'échantillon. Ce modèle apprend au fur et à mesure en procédant par essais et erreurs. Une séquence de résultats positifs sera renforcée afin d'élaborer la meilleure recommandation ou politique pour un problème donné.

Le système IBM Watson qui a remporté le concours Jeopardy! En 2011 en est un bon exemple. Le système utilisait l'apprentissage par renforcement pour savoir quand tenter de répondre (ou de poser une question), quelle case sélectionner sur le tableau et combien miser, en particulier sur les doubles quotidiens.

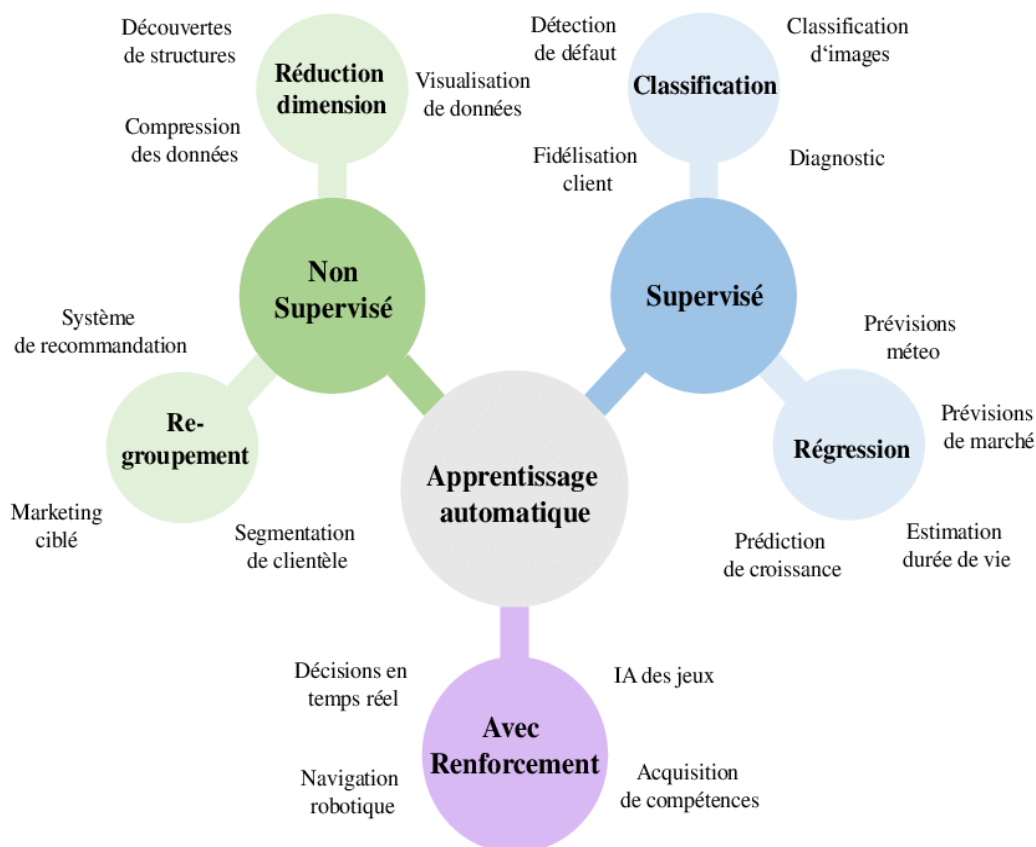


Figure 1.7.2: Catégories d'apprentissage automatique

1.8 Conclusion

Dans ce chapitre, nous avons présenté les concepts généraux des images en vision par ordinateur, notamment les types d'images, leurs propriétés et leurs représentations numériques. Ces connaissances sont essentielles pour bien comprendre la façon dont les systèmes de vision artificielle perçoivent le monde.

Le chapitre suivant abordera le traitement d'images et des méthodes plus avancées de reconnaissance visuelle utilisant l'intelligence artificielle (IA) et l'apprentissage automatique, où une image est analysée, comprise et exploitée par une machine.

Chapitre 2 : État de l'Art

2.1 Introduction à la Fusion d'Expositions

Il s'agit d'un processus hautement technique de traitement d'image, offrant la possibilité de fusionner plusieurs images de la même scène prises sous diverses expositions en une image virtuelle de meilleure qualité. Cela aide à retenir les détails dans les zones d'éclairages faibles et très forts, inhérent à certains appareils photo ou caméras pour conserver toute information.

2.2 Définition de la Fusion d'Expositions

Le processus de fusion d'expositions, de par sa nature, admet de consister dans la composition de deux ou plusieurs images prises à différents temps d'exposition en vue de la détermination d'un troisième qui conserve une quantité appropriée d'information de chacune des prises. L'objectif est d'obtenir une image plus riche en détails, adaptée à la perception humaine et à l'analyse informatique. Ce procédé vise à implanter des informations en surplus provenant des différentes expositions aux fins d'amélioration de la qualité visuelle et de la robustesse du traitement automatique d'images.

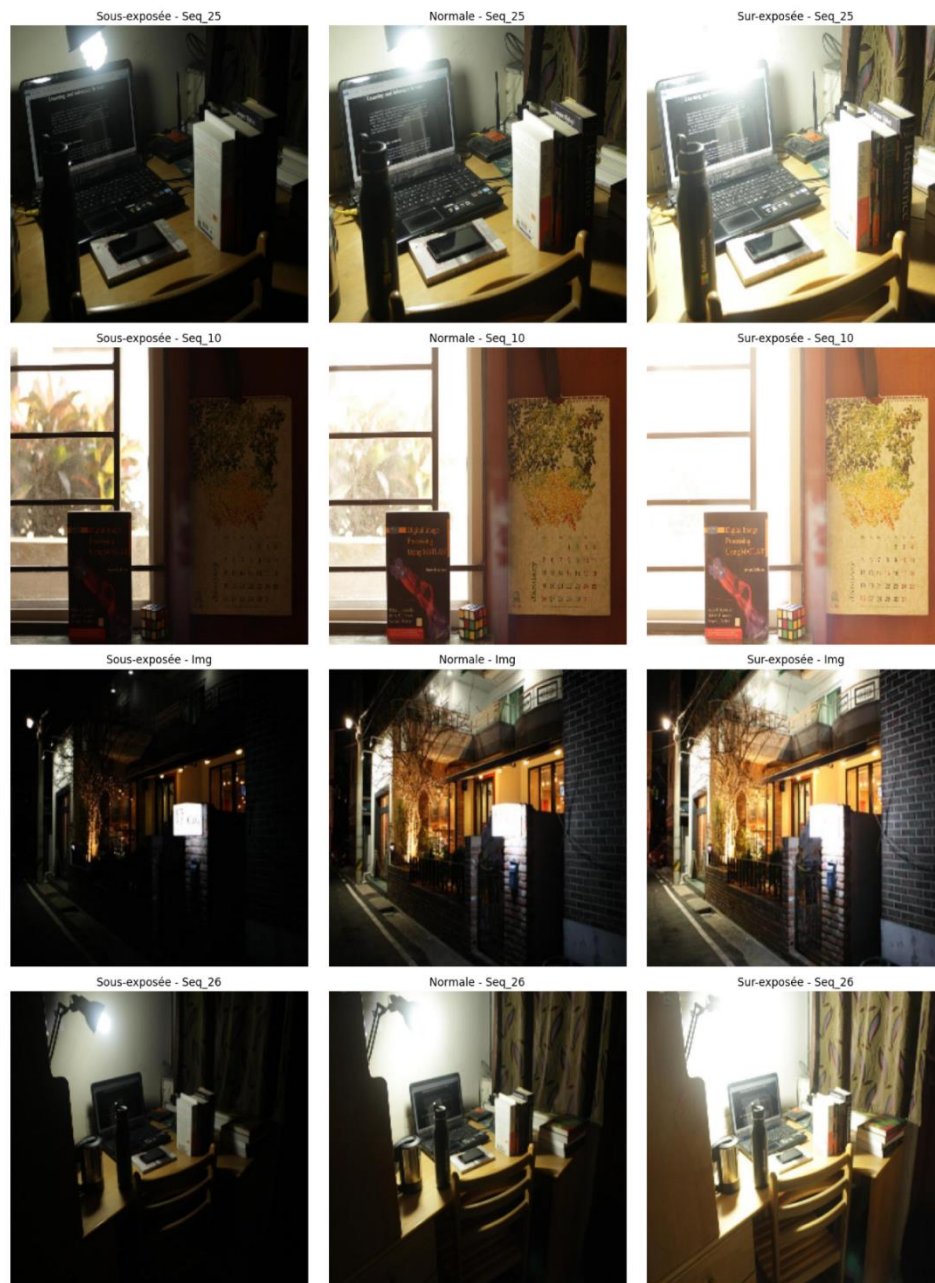


Figure 2.2 : Images ayant diverses expositions.

2.3 Distinction entre la Fusion d'expositions et le HDR

L'intégration d'expositions et la technologie HDR (High Dynamic Range) représentent deux méthodes cruciales en traitement d'images, destinées à optimiser la qualité visuelle en tirant parti de divers niveaux d'exposition d'une scène. Même si ces deux méthodes visent à optimiser visuellement, elles diffèrent en termes de méthodologies et de processus de traitement.

Critère	Fusion d'Expositions	HDR (High Dynamic
---------	----------------------	-------------------

		<i>Range)</i>
Définition	Assemblez plusieurs images afin de répartir les zones sombres et lumineuses sans élargir la gamme dynamique. [16]	Génère une image à gamme dynamique étendue à l'aide d'un fichier HDR intermédiaire, suivi par un processus de tone mapping pour la visualisation .[17]
Format de sortie	Image en plage dynamique standard (SDR), prête pour l'affichage classique [17].	Image HDR nécessitant un Tone Mapping pour être visualisée sur des écrans non HDR [18].
Traitement	Sélectionne les meilleures parties de chaque exposition et les fusionne directement [19].	Génère un fichier HDR intermédiaire (32 bits), puis applique une compression tonale pour adapter la dynamique [17].
Complexité	Méthode plus simple et rapide, sans conversion HDR ni gestion de courbe tonale [20].	Processus complexe incluant la génération HDR et l'ajustement des tons [17].
Utilisation	Idéal pour la vision par ordinateur, l'imagerie médicale ou les corrections locales [20].	Utilisé en photographie artistique, cinéma, ou pour les écrans HDR compatibles [16].
Risques d'artefacts	Moins d'artefacts si l'alignement est réussi (ex : décalages corrigés automatiquement) [18].	Halos ou effets irréalistes possibles si le Tone Mapping est mal paramétré [20].

2.4 Utilisation et Applications

La fusion d'expositions est largement utilisée dans plusieurs domaines :

- **Photographie HDR :** Destinée à saisir des scènes à large gamme dynamique, telles que les paysages comportant des ciels éclatants et des ombres marquées.
- **Vision par ordinateur :** Dans le but d'améliorer la qualité des images employées dans des algorithmes de détection d'objets ou de reconnaissance de formes.
- **Imagerie médicale et suivi :** Pour obtenir des images plus exactes et détaillées, ce qui est essentiel pour le diagnostic ou le suivi.

2.5 Objectifs de la Fusion d'Expositions

La fusion d'expositions est une technique essentielle dans le traitement d'images, visant à améliorer la qualité et la fiabilité des images en combinant plusieurs prises à des niveaux d'exposition différents. Les principaux objectifs de cette technique sont :

2.5.1 Améliorer la fiabilité:

La fusion d'expositions permet de réduire les pertes d'informations dues aux limitations des capteurs d'images. En fusionnant différentes expositions, on parvient à un résultat plus exact et conforme à la réalité, puisque les éléments manquants dans une seule image peuvent être restitués grâce aux autres photographies.[22]

2.5.2 Améliorer la capacité d'information:

Chaque image capturée avec une exposition différente contient des détails spécifiques, comme les ombres ou les hautes lumières. La fusion d'expositions permet d'exploiter ces informations complémentaires pour obtenir une image plus riche et détaillée, offrant ainsi une représentation plus complète de la scène [21].

2.5.3 Optimiser la plage dynamique:

En fusionnant plusieurs expositions, on étend la plage dynamique de l'image finale. Cela permet de mieux représenter les zones sombres et claires sans perte de détails, ce qui est particulièrement utile pour capturer des scènes à haute contraste [23].

2.5.4 Réduire le bruit numérique:

Dans les zones sous-exposées, le bruit numérique peut être important. En fusionnant plusieurs expositions, il est possible d'atténuer ce bruit en conservant les pixels les plus fiables, ce qui améliore la qualité globale de l'image [22].

2.5.5 Améliorer l'analyse informatique des images:

Dans le cadre de la vision par ordinateur, une image mieux équilibrée en termes d'exposition facilite les traitements ultérieurs tels que la segmentation, la reconnaissance d'objets ou la classification. Une image fusionnée offre plus de détails pertinents pour ces analyses, augmentant ainsi leur précision [22].

Ces objectifs sont cruciaux dans divers domaines, notamment la photographie HDR, l'imagerie médicale, la vision par ordinateur et la surveillance.

2.6 Techniques de Fusion d'Expositions

2.6.1 Approches Traditionnelles de la Fusion d'Expositions

Les méthodes traditionnelles de fusion d'expositions privilégient des algorithmes déterministes et des techniques de traitement d'image basées sur des critères locaux. Deux approches principales se distinguent :

A. Moyenne Pondérée

Cette méthode combine les pixels des images d'entrée en utilisant des poids calculés à partir de mesures locales (luminosité, saturation, contraste). La formule est :

Principe : Combine les pixels à l'aide de poids basés sur des mesures locales (luminosité, saturation, contraste).

Formule :

$$F(x, y) = \sum_i = 1Nw_i(x, y) \cdot I_i(x, y) / \sum_i = 1Nw_i(x, y)$$

Avantages :

- *Simplicité d'implémentation et faible coût computationnel [24] [25].*
- *Efficace pour des scènes statiques avec des écarts d'exposition modérés [24].*

Limites :

- Risque de flou et de perte de détails dans les zones à forts contrastes [24].
- Sensibilité aux artefacts de fantômes (défauts d'alignement) [24].
- Inadapté aux scènes dynamiques ou avec mouvements [25].

B. Sélection des Meilleurs Pixels :

Principe :

La méthode de sélection des meilleurs pixels consiste à choisir, pour chaque position de pixel dans l'image finale, celui qui présente la meilleure exposition parmi les différentes images d'entrée. Le critère de sélection peut inclure la luminance optimale (proche d'un seuil médian) ou l'intensité maximale pour préserver les zones les mieux exposées. Les pixels ainsi sélectionnés sont directement copiés dans l'image finale sans mélange ni interpolation.

Avantages :

- Conservation des régions bien exposées : Les détails et les couleurs des zones correctement exposées restent fidèles aux images originales.
- Réduction des artefacts de bruit : En excluant les pixels surexposés ou sous-exposés, le bruit est limité dans l'image résultante.
- Simplicité algorithmique : Cette technique est facile à implémenter et ne nécessite pas de calculs lourds comme dans les approches multi-échelles.

Limites :

- Discontinuités visibles : L'absence de mélange peut créer des bordures nettes et visibles entre les zones sélectionnées, compromettant l'harmonie globale.

- Dépendance au recalage précis : Pour éviter des artefacts tels que des fantômes ou des décalages, un alignement parfait des images d'entrée est indispensable.
- Inadaptation aux scènes dynamiques : Les mouvements ou variations significatives entre les images rendent cette méthode inefficace.

C. Fusion Pyramidale (Pyramides Laplaciennes/Gaussiennes)

Principe :

Cette méthode repose sur la décomposition multi-échelle. Les images sont transformées en pyramides (Laplacienne ou Gaussienne), permettant de séparer les couches de base et les couches de détails. Les poids associés à chaque niveau sont ajustés pour fusionner les informations pertinentes tout en conservant les fréquences spatiales.

Avantages :

1. Préservation des détails fins :
 - La séparation des couches hautes fréquences (contraste local) et basses fréquences permet d'obtenir un résultat détaillé et réaliste.
2. Réduction des artefacts de coutures :
 - Grâce à un mélange progressif entre les niveaux de la pyramide, les transitions sont lissées, ce qui limite les artefacts visibles.

Limites :

1. Halos artificiels :
 - Ces halos apparaissent souvent autour des contours forts lorsque les coefficients Laplaciens présentent des différences importantes en magnitude entre les images.
2. Perte de contraste global :
 - Si les poids appliqués ne tiennent pas compte des variations lumineuses extrêmes, l'image fusionnée peut manquer de contraste.
3. Complexité accrue :
 - Les calculs multi-échelles augmentent le temps de traitement et requièrent des ressources plus importantes, ce qui peut être un défi dans des systèmes nécessitant une efficacité élevée.

Méthodes spatiales (traditionnelles)

Ces techniques agissent directement sur les pixels d'une image, c'est-à-dire sans transformation de domaine. Elles utilisent les règles de pondération spatiale pour combiner les images de différentes expositions. La qualité de la fusion dépend des pondérations spatiales attribuées aux pixels.

1. **Bruce et al. (2014)** ont proposé une méthode basée sur l'entropie locale. Chaque pixel est pondéré selon l'entropie calculée dans un voisinage circulaire après normalisation logarithmique. Bien que la fusion offre une haute entropie, les couleurs peuvent être irréalistes. C'est une technique traditionnelle avec un traitement pixel à pixel. [69]
2. **Lee et al. (2018)** ont introduit une méthode adaptative de pondération basée à la fois sur la luminosité globale et le gradient global de l'image. Cette technique appartient aux approches traditionnelles pixel-based. [76]
3. **Kinoshita et al.** ont présenté une segmentation de la scène selon la distribution de lumière afin de mieux ajuster les expositions. Cela relève des méthodes traditionnelles pour améliorer l'équilibre d'exposition. [77]
4. **Xu et al. (2019)** ont utilisé un modèle de Retinex multi-échelle pour extraire la carte d'illumination, qui est ensuite utilisée pour la pondération. C'est une méthode de fusion traditionnelle fondée sur des caractéristiques physiques. [78]
5. **Ulucan et al. (2020)** ont proposé une méthode combinant l'embedding linéaire et le masquage par watershed. Les poids de fusion sont ajustés en fonction des caractéristiques spatiales. Cette technique est également classée comme méthode traditionnelle pixel-based. [79]
6. **Raman et Chaudhuri (2011)** ont utilisé un **filtre bilatéral** pour lisser les cartes de poids et préserver les détails fins. Il s'agit d'une approche traditionnelle souvent utilisée pour atténuer les artefacts comme les halos. [80]

Méthodes basées sur les patches (traditionnelles)

7. **Ma et al. (2015, 2017)** ont proposé la méthode SPD-MEF, qui décompose les patches en trois composants indépendants : intensité, structure, et moyenne. Une vérification de la cohérence structurelle est utilisée pour éviter les effets fantômes, ce qui fait de cette méthode une référence dans les techniques patch-based traditionnelles.
8. **Li et al. (2020, 2021)** ont amélioré cette approche en rendant le processus multi-échelle, réduisant les artefacts et améliorant la vitesse. Ils ont aussi introduit un facteur de conservation des bords et des courbes de pondération plus précises.
9. **Wang et al. (2019)** ont remplacé les patches fixes par une segmentation en **super-pixels** pour adapter dynamiquement les régions fusionnées. Cette méthode améliore la précision de fusion tout en appartenant à la catégorie traditionnelle.

Méthodes par optimisation (traditionnelles)

10. **Shen et al.** ont introduit un cadre basé sur le **random walk** et un modèle probabiliste global pour estimer les poids. Cela permet une fusion efficace tout en considérant les relations locales et globales.
11. **Song et al. (2016)** ont utilisé des **contraintes de gradient** dans un cadre de maximum de vraisemblance (MAP) pour renforcer les détails dans l'image finale.
12. **Ma et al. (2017)** ont proposé une méthode d'optimisation basée sur l'indice **SSIM**, optimisant directement la similarité structurelle entre les images source et l'image fusionnée.

Méthodes du domaine transformé (traditionnelles)

Dans ces techniques, les images sont d'abord converties dans un domaine fréquentiel ou multi-résolution (pyramide, ondelettes, etc.), où les coefficients sont fusionnés : ensuite, une transformée inverse est mise en œuvre pour reconstruire l'image finale.

13. **Burt et al. (1983)** ont été les pionniers avec leur méthode de pyramide de gradient, utilisée plus tard dans diverses tâches de fusion d'images.
14. **Mertens et al. (2007)** ont proposé un cadre de fusion multi-échelle avec **pyramide de Laplace**. La pondération se base sur le contraste, la saturation et l'exposition. Cette méthode est devenue une référence.
15. **Li et al. (2013)** ont proposé une décomposition en deux couches (base et détail), les poids étant affinés avec des filtres guidés pour limiter les artefacts.
16. **Nejati et al. (2017)** ont conçu une méthode rapide où les images sont décomposées avec un filtre guidé, les couches fusionnées par pondération selon une fonction d'exposition.
17. **Zhang et al. (2016)** ont développé une méthode dans le **domaine des gradients** inspirée de la perception humaine, en reconstruisant l'image via les gradients fusionnés. [80]
18. **Shao et al. (2020)** ont utilisé la **représentation clairsemée (sparse representation)** avec des dictionnaires d'exposition pour estimer les pondérations des patches.

19. **Martorell et al. (2019)** ont appliqué la **DCT sur fenêtre glissante** sur des patches en espace YCbCr, combinant les coefficients DCT selon leur importance.

Synthèse :

Les méthodes traditionnelles, bien que efficaces pour des scènes simples, peinent à gérer les défis des scènes complexes (mouvements, écarts d'exposition extrêmes). Les approches pyramidales, bien que supérieures à la moyenne pondérée, nécessitent des optimisations (profondeur de pyramide, normalisation robuste) pour atténuer les halos [25]. Les travaux récents intègrent des filtres guidés ou des mesures de dissimilarité couleur pour améliorer les poids [24](#), mais ces innovations dépassent le cadre des approches purement traditionnelles.

2.6.2 Approche proposée par Mertens (Fusion d'exposition)

L'approche d'Exposure Fusion, développée par Mertens et ses collaborateurs, cherche à combiner directement plusieurs images LDR (Low Dynamic Range) capturées à différentes expositions, sans nécessité de produire une image HDR intermédiaire. Cette technique s'est imposée comme une norme pour obtenir une image finale correctement exposée, riche en détails et en couleurs, tout en restant dans la plage dynamique des images d'origine. L'objectif est de sélectionner et combiner, pour chaque pixel, les parties les mieux exposées, les plus contrastées et les plus saturées de chaque image de la séquence, afin d'obtenir une image finale optimisée sur ces trois critères

Étapes principales de l'algorithme

1. Calcul des poids par pixel

Pour chaque pixel de chaque image d'entrée, trois critères sont évalués :

- **Contraste local** : Mesuré via un filtre Laplacien, il favorise les zones riches en détails fins.
- **Saturation** : Mesure l'intensité des couleurs, privilégiant les pixels aux couleurs vives.
- **Bien-exposition** : Privilégie les pixels dont l'intensité est proche de la valeur médiane (typiquement 0.5 dans une plage normalisée), c'est-à-dire ni sous-exposés ni surexposés.

Un poids est calculé pour chaque pixel en multipliant les mesures de ces trois critères, ce qui donne une carte de poids pour chaque image [62] .

2. Construction des pyramides de Laplace

Chaque image d'entrée est décomposée en une pyramide de Laplace, c'est-à-dire une représentation multi-échelle qui sépare les détails à différents niveaux de résolution.

Cette étape permet de réaliser une fusion progressive, du plus fin au plus grossier, limitant ainsi les artefacts de transition [62][63].

3. Fusion pondérée des pyramides

À chaque niveau de la pyramide, les images sont fusionnées en utilisant les cartes de poids calculées précédemment. Chaque coefficient de la pyramide fusionnée est obtenu par une moyenne pondérée des coefficients correspondants des images d'entrée [62][63].

4. Reconstruction de l'image finale

La pyramide fusionnée est ensuite reconstruite (par sommation des niveaux) pour obtenir l'image finale, qui combine les meilleures parties de chaque image d'origine, tout en conservant une apparence naturelle et une plage dynamique LDR [62][63].

Avantages et spécificités

- **Pas besoin d'informations sur les temps d'exposition ou la courbe de réponse de la caméra** : L'algorithme fonctionne directement sur les images LDR, sans calibration spécifique.
- **Préservation du contraste local** : Grâce au Laplacien et à la fusion multi-échelle, l'image finale conserve les détails fins et les transitions naturelles.
- **Simplicité et efficacité** : La méthode est rapide et facile à implémenter, ce qui explique son adoption dans de nombreux logiciels de traitement d'images.

2.6.3 Approche proposée par Moumene, Nourine, Ziou (Fast Exposure Fusion Based on Histograms Segmentation)

Cette méthode permet une fusion rapide d'images pour le traitement multi-exposition en temps réel, grâce à la segmentation par histogrammes. Au lieu de calculer la pondération par pixel comme dans les méthodes classiques, l'image est divisée en trois régions : les ombres, les hautes lumières et les zones intermédiaires, selon les histogrammes des images sous-exposées et surexposées.

Les pixels des zones d'ombre et de hautes lumières sont directement intégrés à l'image fusionnée, réduisant ainsi les calculs et évitant l'influence de pixels totalement saturés ou totalement sombres. Seules les zones intermédiaires sont fusionnées par une simple pondération basée sur leur luminance relative.

Outre la fusion, la méthode propose un système automatique de prise d'exposition optimale, basé sur les régions segmentées et testé sur un smartphone (Nokia N900). Les résultats expérimentaux en termes de qualité sont comparables, voire supérieurs, à ceux de Mertens et Photomatix. En termes de temps de traitement, l'image prend près de 0,3 s, soit 29 ips en C++. [75]

2.6.4 Approches Avancées

Combinaison d'expositions par le biais de réseaux de neurones convolutifs (CNN)

Les méthodes traditionnelles de fusion d'expositions s'appuient généralement sur des critères explicites ou une sélection manuelle pour combiner diverses images prises avec différentes expositions. Suite aux progrès récents en intelligence artificielle, notamment dans le domaine du deep learning, des techniques fondées sur les CNN ont émergé, offrant une capacité d'apprentissage automatique pour extraire et fusionner de manière optimale les informations issues des images entrantes.

Un CNN peut apprendre à fusionner plusieurs expositions en suivant un processus structuré et entièrement automatisé grâce à l'apprentissage profond :

- **Entrée** : Les images prises à différentes expositions (par exemple, sous-exposée, normale, surexposée) sont empilées en canaux supplémentaires, formant ainsi une entrée de dimension $H \times W \times 9$ pour trois images RGB. Cela permet au réseau d'avoir accès simultanément à toutes les informations de luminosité et de détail présentes dans chaque prise de vue [57][58].
- **Traitement par couches convolutifs** : Les premières couches du CNN extraient des caractéristiques locales pertinentes de chaque image, telles que les zones bien exposées, les détails fins et les textures. Ces couches apprennent automatiquement à détecter les motifs et informations utiles pour la fusion, sans nécessiter de règles explicites ou de filtrage manuel [57][58].
- **Fusion implicite** : Les features extraites de chaque image sont combinées dans le réseau, soit par addition, soit par concaténation, ou encore via des modules spécialisés (comme l'attention croisée). Le CNN apprend, pendant l'entraînement, à pondérer et à fusionner ces informations pour optimiser la qualité visuelle de l'image finale. Cette fusion est guidée par une fonction de perte qui mesure la qualité de l'image fusionnée par rapport à une référence idéale, ou selon des critères perceptuels [57][58].
- **Sortie** : Le réseau produit une image unique RGB de haute définition, qui fusionne automatiquement les meilleurs éléments de chaque prise (zones bien exposées, détails, couleurs), tout en réduisant les artefacts et en maintenant la cohérence générale [57][58].

Pour faire simple, le CNN a été formé pour identifier et regrouper les informations pertinentes de chaque exposition afin de créer une image fusionnée optimale, sans nécessiter d'intervention manuelle ou d'ajustement de paramètres spécifiques. Il est également capable de s'adapter à divers types de scènes et conditions d'éclairage.

Méthodes basées sur l'apprentissage profond

Ces méthodes utilisent des réseaux neuronaux, le plus souvent des CNN ou GAN, pour apprendre automatiquement à partir d'exemples l'approche optimale de fusion d'images.

20. **Kalantari et al. (2017)** ont été les premiers à proposer une méthode de fusion MEF **supervisée** à l'aide d'un **réseau de neurones convolutifs (CNN)**, avec alignement préalable des images par **optical flow**. C'est une méthode avancée. [70]
21. **DeepFuse (Prabhakar, 2017)** est une méthode **non supervisée** où seul le canal Y est fusionné par CNN, tandis que Cb et Cr sont fusionnés manuellement. Elle utilise une **fonction de perte MEF-SSIM** sans besoin de vérité terrain. [71]
22. **MEFNet (Ma, 2020)** améliore la vitesse et la flexibilité, en travaillant à basse résolution et en utilisant un **réseau d'agrégation de contexte** pour générer les cartes de poids. [72]
23. **UMEF (Qi, 2020)** permet de fusionner plusieurs images d'entrée. Il utilise une **perte combinée** (SSIM + gradient) et peut fonctionner directement en RGB sans changement d'espace couleur.[73]
24. **U2Fusion (2021)** repose sur **DenseNet** et **VGG16**, et s'adapte à plusieurs tâches de fusion (MEF, multimodal, multi-focus). C'est une méthode avancée entièrement non supervisée.
25. **MEF-GAN (Xu, 2021)** est une approche par **GAN** combinant alignement, mécanisme d'attention, et apprentissage adversarial pour produire des fusions visuellement réalistes. [74]

U-Net et Variantes

U-Net est une architecture de réseau de neurones convolutionnels introduite en 2015, initialement conçue pour la segmentation d'images médicales. Sa structure symétrique en "U" comprend deux chemins principaux :

- **Chemin de contraction (encoder)** : Il réduit progressivement la dimension spatiale de l'image tout en augmentant le nombre de canaux, captant ainsi le contexte global de l'image. Cela se fait via des couches de convolutions suivies de sous-échantillonnage (downsampling).
- **Chemin d'expansion (decoder)** : Il reconstruit une sortie haute résolution en combinant les informations contextuelles globales avec des détails fins, grâce à des connexions de saut (skip connections) reliant les couches du chemin de contraction aux couches correspondantes du chemin d'expansion.

Cette architecture permet de produire des cartes de segmentation précises et de conserver les détails fins, même avec peu de données annotées, ce qui est particulièrement utile en imagerie médicale [59] [60] [61].

Pourquoi U-Net est adaptée à la fusion

- **Préservation des détails** : Les skip connections permettent de transférer directement les informations de bas niveau (détails fins) du chemin de contraction

vers le chemin d'expansion, ce qui est crucial pour obtenir une image fusionnée fidèle et riche en détails [61].

- **Flexibilité** : U-Net peut être entraînée pour apprendre à combiner différentes expositions ou modalités d'images tout en maintenant la cohérence spatiale, ce qui la rend adaptée à des tâches de fusion complexes [60].
- **Capacité multi-échelles** : Grâce à son architecture en pyramide, U-Net capture et fusionne des informations à différents niveaux de granularité, permettant une intégration efficace des détails locaux et du contexte global [59].

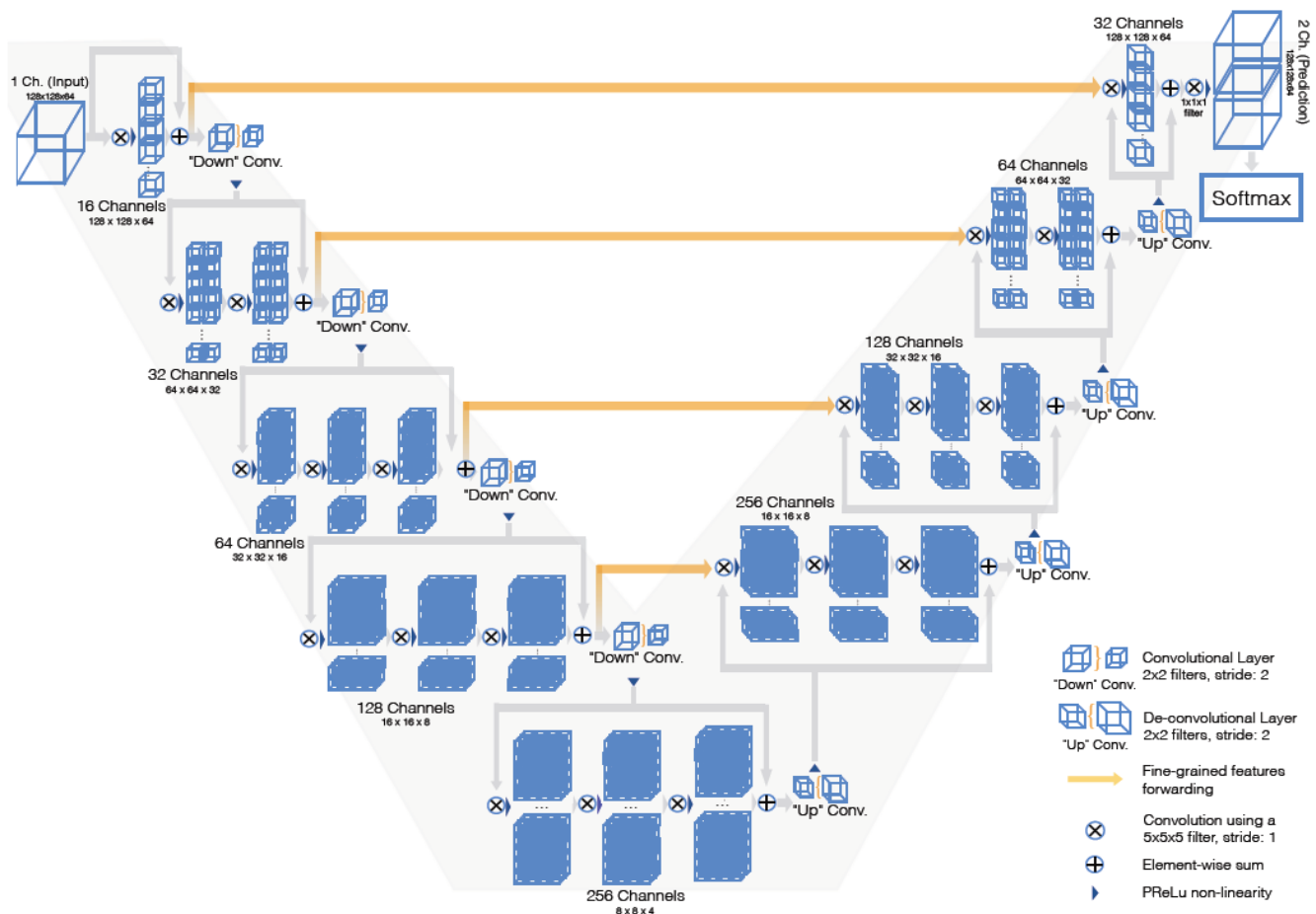


Figure 2.6.4: Structure de U-Net

Chapitre 3 : Méthodologie et implémentation et résultats

3.1 Apprentissage Profond (Deep Learning)

3.1.1 Définition du Deep Learning

Le deep learning, également connu sous le nom d'apprentissage profond, est une sous-branche du machine learning (ML) qui utilise des réseaux de neurones artificiels profonds pour modéliser des fonctions complexes à partir de données brutes [39] [40]. Ces réseaux sont inspirés du fonctionnement du cerveau humain et sont constitués de plusieurs couches de neurones artificiels interconnectés [39]. Plus le nombre de couches est élevé, plus le réseau est capable de traiter des tâches complexes [41].

3.1.2 Quelle est la différence entre le Deep learning et le Machine Learning ?

Contrairement au Machine Learning, qui se repose le plus souvent sur la création manuelle, par le data scientist, de variables explicatives, cette tâche est réalisée automatiquement par l'algorithme en fonction des données d'entraînement dans le cas des réseaux de neurones, car chaque couche peut générer des variables plus avancées, se reposant sur celle de la couche précédente. On peut donc les considérer comme des « générateurs hiérarchiques de variables ».

La conséquence directe est que la quantité de données nécessaires à l'algorithme pour le Deep Learning est bien plus importante que pour le Machine Learning. Autre conséquence : il est pratiquement impossible pour le cerveau humain de comprendre le « raisonnement » d'un algorithme de Deep Learning. Pourtant, les résultats obtenus surperforment ceux des humains en majorité.

3.1.3 Comparaison entre Machine Learning et Deep Learning

Critère	Machine Learning	Deep Learning
Définition	Apprentissage utilisant des algorithmes classiques	Utilise des réseaux de neurones profonds (deep neural networks)
Type de données	Fonctionne bien avec des ensembles de données de petite ou moyenne taille	Nécessite de grandes quantités de données
Extraction des caractéristiques	Nécessite une extraction manuelle par l'expert	Automatique grâce aux couches du réseau
Architecture	Algorithmes simples : régression, SVM, arbres de décision, etc.	Réseaux de neurones profonds avec plusieurs couches cachées
Interprétabilité	Généralement plus facile à comprendre et à expliquer	Souvent considéré comme une « boîte noire », plus difficile à interpréter

Ressources nécessaires	Moins gourmand en calcul, peut fonctionner sur des machines classiques	Nécessite une forte puissance de calcul (GPU, TPU, etc.)
Performance (gros datasets)	Moins performant pour les très grands ensembles de données	Excellente performance sur des données massives
Applications typiques	Diagnostic médical, prédiction de ventes, détection de fraude	Vision par ordinateur, traitement du langage naturel (NLP), reconnaissance vocale

3.1.4 Définition et principes fondamentaux des réseaux neuronaux artificiels

Les réseaux neuronaux artificiels (ANN) sont des modèles mathématiques inspirés du fonctionnement des réseaux biologiques du cerveau humain. Ils sont conçus pour traiter des données en entrée (images, texte, audio, etc.) à travers plusieurs couches interconnectées afin de produire une sortie. Ces systèmes adaptatifs permettent aux ordinateurs de résoudre des problèmes complexes en apprenant à partir de données et en ajustant leurs paramètres automatiquement grâce à l'apprentissage profond [42] [43] [46].

Histoire et évolution des réseaux neuronaux

Origines (années 1940–1980)

- En 1943, Warren McCulloch et Walter Pitts ont proposé le premier modèle de réseau neuronal basé sur la logique seuil, jetant les bases des réseaux neuronaux artificiels [42].
- En 1958, Frank Rosenblatt a introduit le perceptron, un modèle simple de réseau neuronal capable d'apprendre à partir de données étiquetées [43].

Renaissance (années 1980–2000)

- Les années 1980 ont vu l'essor des réseaux multicouches grâce à l'algorithme de rétropropagation, permettant d'entraîner des réseaux plus complexes.
- Les premiers succès dans la reconnaissance d'images et la classification ont démontré le potentiel des réseaux neuronaux pour résoudre des problèmes réels [45].

Révolution du Deep Learning (2000–présent)

- L'utilisation de GPU pour accélérer les calculs a permis d'entraîner des réseaux neuronaux profonds (Deep Learning), ouvrant la voie à des avancées majeures dans la vision par ordinateur et le traitement du langage naturel [45].
- En 2017, l'architecture Transformer a révolutionné le traitement du langage naturel grâce au concept d'attention. Les modèles basés sur cette architecture, comme les LLM (Large Language Models), sont capables de comprendre et générer du texte avec une précision exceptionnelle.

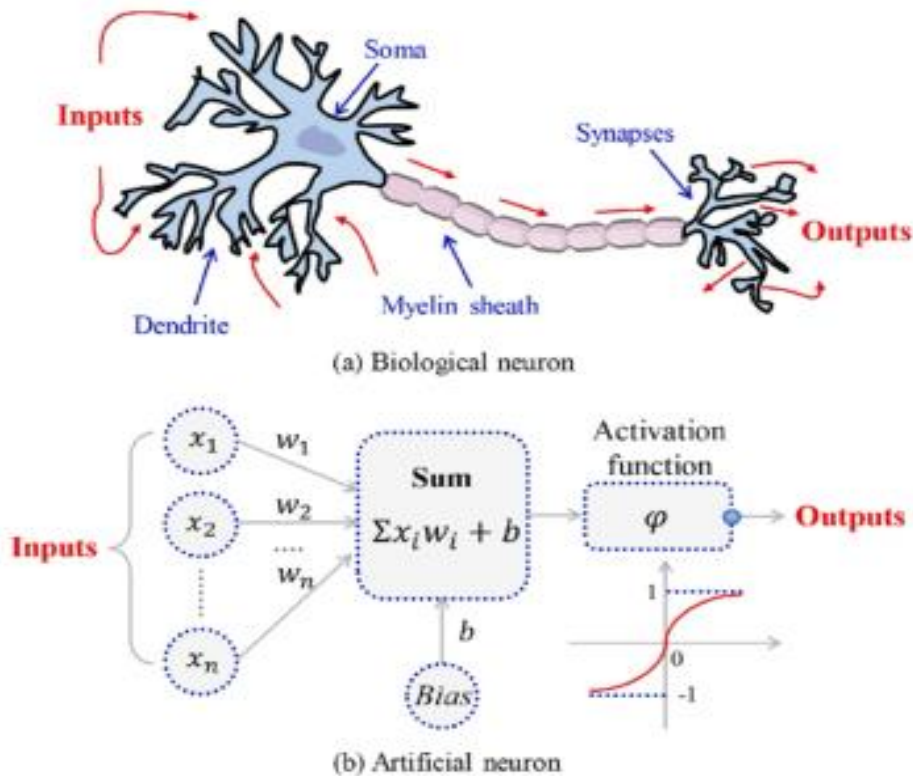


Figure 3.1.4: Biological Neural Network vs Artificial Neural Network

3.1.5 Comment fonctionnent les réseaux neuronaux ?

Le cerveau humain est la source d'inspiration de l'architecture des réseaux neuronaux. Les cellules du cerveau humain, appelées neurones, forment un réseau complexe et hautement interconnecté et s'envoient des signaux électriques les uns aux autres pour aider les humains à traiter les informations. De même, un réseau neuronal artificiel est constitué de neurones artificiels qui travaillent ensemble pour résoudre un problème. Les neurones artificiels sont des modules logiciels, appelés nœuds, et les réseaux neuronaux artificiels sont des programmes logiciels ou des algorithmes qui, à la base, utilisent des systèmes informatiques pour résoudre des calculs mathématiques.

Tout d'abord, chaque neurone de la couche d'entrée reçoit un élément de données brutes, comme les pixels d'une image ou les mots d'une phrase, et attribue un poids à cette entrée, indiquant sa pertinence pour la tâche. Un poids faible (inférieur à 0,5) signifie que l'information a moins de chances d'être pertinente. Ces entrées pondérées sont transmises à des couches cachées, où les neurones ajustent encore l'information. Ce processus se poursuit sur plusieurs couches jusqu'à ce que la couche de sortie fournisse une prédiction finale.

3.1.6 Architecture de réseau neuronal simple

Un réseau neuronal de base comporte des neurones artificiels interconnectés en trois couches :

Couche d'entrée:

Les informations du monde extérieur entrent dans le réseau neuronal artificiel par la couche d'entrée. Les nœuds d'entrée traitent les données, les analysent ou les catégorisent, et les transmettent à la couche suivante.

Couche cachée:

Les couches cachées prennent leurs entrées de la couche d'entrée ou d'autres couches cachées. Les réseaux neuronaux artificiels peuvent avoir un grand nombre de couches cachées. Chaque couche cachée analyse la sortie de la couche précédente, la retravaille et la transmet à la couche suivante.

Couche de sortie:

La couche de sortie donne le résultat final de tous les traitements de données effectués par le réseau neuronal artificiel. Il peut avoir des nœuds uniques ou multiples. Par exemple, si nous avons un problème de classification binaire (oui/non), la couche de sortie aura un nœud de sortie qui fournira le résultat sous forme de 1 ou 0. Cependant, si nous avons un problème de classification multi-classes, la couche de sortie peut être constituée de plus d'un nœud de sortie.[49]

A simple neural network

input layer hidden layer output layer

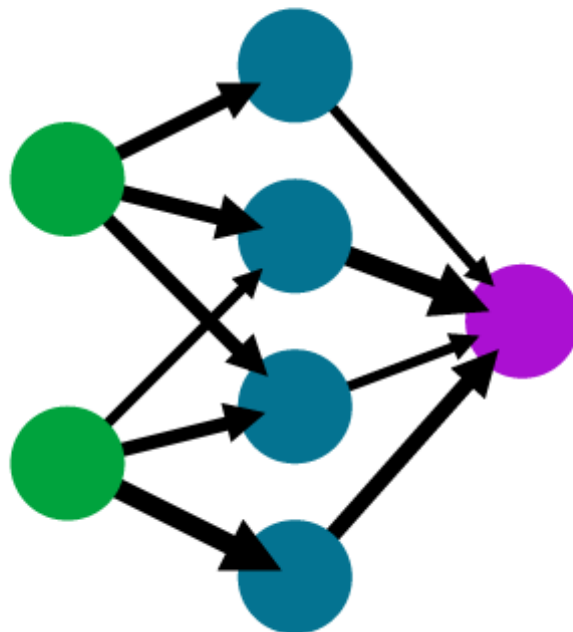


Figure 3.1.6 : Architecture de réseau neuronal simple

3.1.7 Architecture de réseau neuronal profond

Les réseaux neuronaux profonds, également connus sous le nom de réseaux de deep learning, sont des modèles d'apprentissage automatique complexes qui utilisent plusieurs couches de neurones artificiels interconnectés pour traiter et analyser des données.

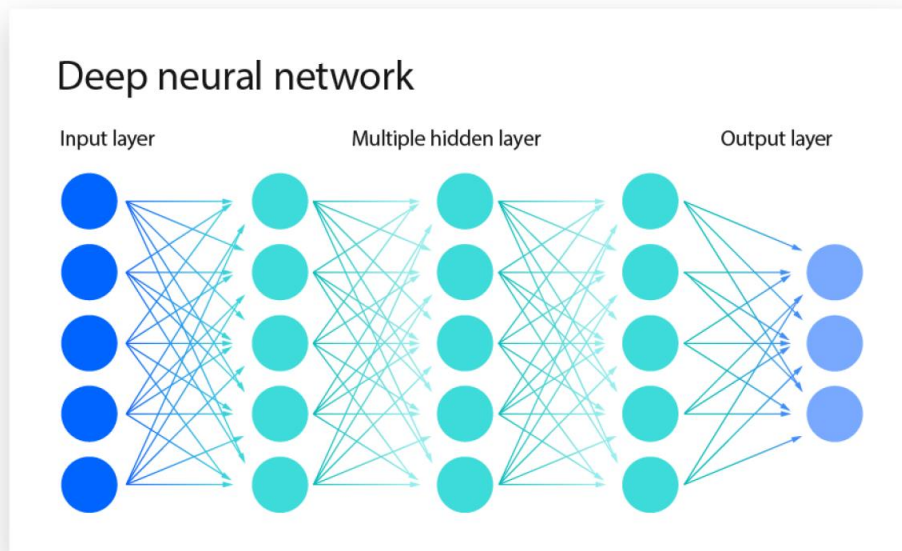


Figure 3.1.7 : Architecture de réseau neuronal profond

3.1.8 Quels sont les différents types de réseaux neuronaux ?

Type de réseau neuronal	Caractéristiques principales	Cas d'utilisation courants
Réseau de type feedforward	Flux de données à sens unique	Prédictions simples, tâches élémentaires
Réseau convolutif (CNN)	Détecte les modèles visuels	Reconnaissance d'images, imagerie médicale
Réseau récurrent (RNN)	Traite les données séquentielles	Séries temporelles, reconnaissance vocale
Mémoire à long terme (LSTM)	Mémoriser les dépendances à long terme	Génération de texte, chatbots
Unité récurrente clôturée (GRU)	LSTM simplifiée	NLP, prédiction de séries temporelles
Fonction de base radiale (RBF)	Utilise la fonction de base radiale	Classification, reconnaissance des formes

Réseau adversarial génératif (GAN)	Génère de nouvelles données	Génération d'images, deepfakes
Transformateur	Mécanisme d'auto-attention	Modèles linguistiques, traduction
Réseau modulaire	Plusieurs sous-réseaux indépendants	Processus multi-tâches complexes
Réseau de neurones à pointes (SNN)	Traitement temporel	Robotique, informatique neuromorphique

3.2 Réseaux de Neurones Convolutionnels (CNN)

3.2.1 Introduction

Les réseaux de neurones convolutionnels (CNN, pour *Convolutional Neural Networks*) sont une catégorie spécifique de réseaux de neurones profonds, conçus pour traiter des données structurées en grille, comme les images. Inventés dans les années 1980 par Kunihiko Fukushima, ils s'inspirent du fonctionnement du cortex visuel des animaux, où les neurones sont organisés pour analyser des régions locales du champ visuel [48].

Les CNN ont révolutionné la vision par ordinateur en surpassant les méthodes classiques dans des tâches telles que la classification d'images, la détection d'objets, et la segmentation. Leur architecture particulière leur permet d'extraire automatiquement des caractéristiques pertinentes à différentes échelles, ce qui les rend très efficaces pour le traitement d'images, mais aussi pour la reconnaissance vocale, le traitement du langage naturel, et d'autres domaines [49] [50].

3.2.2 Fonctionnement et architecture

Un CNN est composé de plusieurs types de couches qui traitent successivement l'information :

3.2.2.1 Couche d'entrée

reçoit les données brutes, par exemple une image représentée par une matrice tridimensionnelle (hauteur, largeur, profondeur — souvent les canaux de couleur RVB) [49].

3.2.2.2 Couche de Convolution (Convolutional Layer)

Opération de base

La couche de convolution applique un filtre (ou noyau) de taille $k \times k$ qui glisse sur l'image d'entrée pour extraire des caractéristiques locales.

- **Filtre (noyau) :** Petite matrice de poids w de dimension $k \times k$

- **Image d'entrée** : Matrice x représentant les pixels ou les valeurs des caractéristiques.
- **Biais** : Un scalaire b ajouté à la somme pondérée.

Sortie

La sortie de la convolution est une feature map (carte de caractéristiques) qui représente la réponse locale du filtre à différentes positions de l'image.

Formule mathématique

Pour chaque position (i,j) de la feature map, la valeur est calculée par :

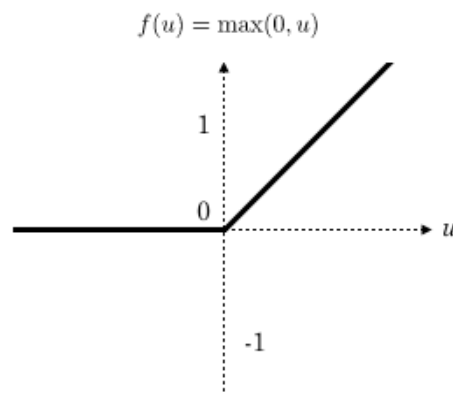
But de la couche de convolution

- Détection de motifs simples : Dans les premières couches, les filtres détectent des motifs élémentaires comme les bords, les contours, ou les textures.
- Détection de motifs complexes : Dans les couches plus profondes, les filtres combinent ces motifs simples pour reconnaître des formes plus complexes, comme des objets ou des parties d'objets.

Cette opération locale et répétée sur toute l'image permet au réseau d'apprendre automatiquement des représentations hiérarchiques des données visuelles.

3.2.2.3 Fonction d'Activation (ReLU)

ReLU (Rectified Linear Unit) est la plus utilisée : $f(x)=\max(0,x)$



Rôle : introduire la non-linéarité, essentiel pour modéliser des fonctions complexes.

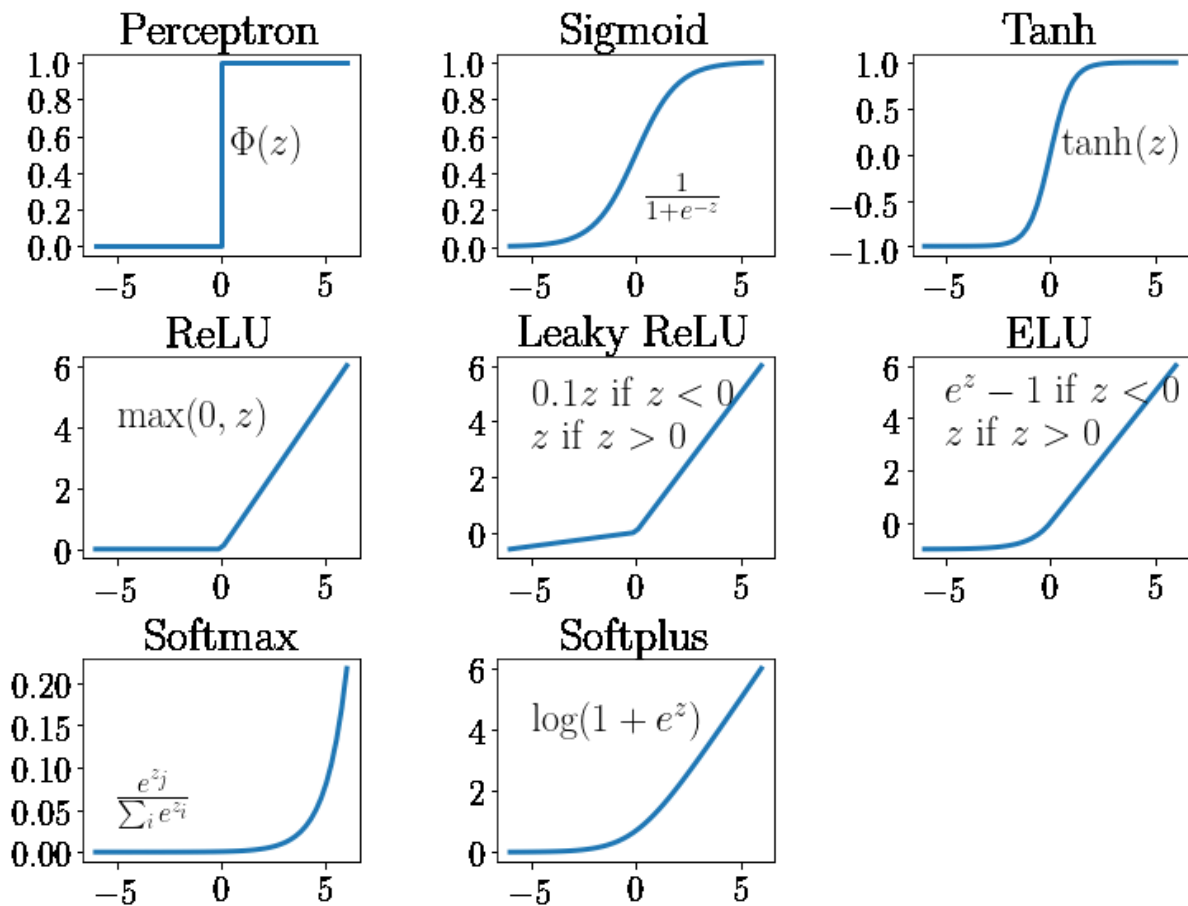


Figure 3.2.2.3 : les fonctions d'activation

3.2.2.4 Couche de Pooling (Sous-échantillonnage)

Types de pooling

- **Max pooling :** sélectionne la valeur maximale dans chaque région locale (patch) de la feature map.
- **Average pooling :** calcule la moyenne des valeurs dans chaque région locale.

Buts principaux de la couche de pooling

- **Réduire les dimensions des données :** en diminuant la taille spatiale (largeur et hauteur) des cartes de caractéristiques, on réduit le nombre de paramètres et la charge computationnelle du réseau.
- **Conserver les caractéristiques dominantes:** le pooling met en avant les informations les plus importantes (par exemple, les activations les plus fortes avec le max pooling).
- **Améliorer la robustesse à la translation :** en réduisant la sensibilité aux petites variations de position des motifs dans l'image, le réseau devient plus stable face aux déplacements ou déformations mineures.

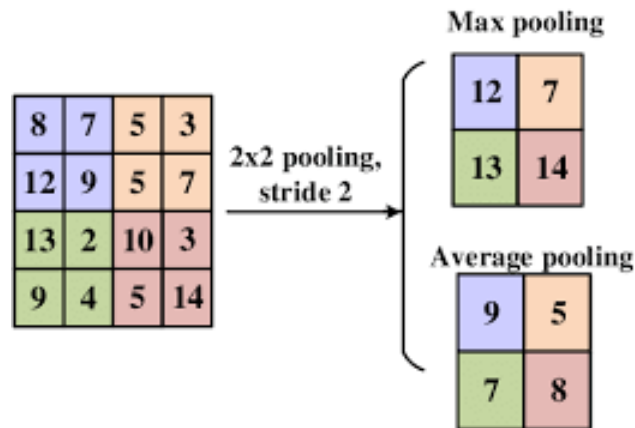


Figure 3.2.2.4 :Types de pooling

3.2.2.5 Couche Complètement Connectée (Fully Connected Layer)

La couche complètement connectée, aussi appelée *fully connected layer* (FC) ou *dense layer*, est une couche de réseau de neurones où chaque neurone de la couche précédente est connecté à tous les neurones de la couche suivante [51] [52]. Cette connexion dense permet au réseau de combiner toutes les caractéristiques extraites par les couches précédentes pour produire la prédiction finale.

Fonctionnement

- La couche FC reçoit en entrée un vecteur (souvent obtenu après aplatissage des cartes de caractéristiques issues des couches convolutionnelles et de pooling).
- Elle applique une transformation linéaire : multiplication par une matrice de poids suivie de l'ajout d'un biais.
- Ensuite, une fonction d'activation non linéaire (comme ReLU ou softmax) est appliquée pour modéliser des relations complexes [51].
- Le résultat est un vecteur de sortie dont la taille correspond au nombre de classes ou de valeurs à prédire.

3.2.2.6 Couche de Normalisation (Batch Normalization)

La Batch Normalisation (BN) est une technique utilisée dans les réseaux de neurones profonds pour standardiser les activations d'une couche. Elle intervient généralement après une couche linéaire (dense ou convolutionnelle) et avant la fonction d'activation.

Principe de fonctionnement:

Pour chaque mini-batch de données, la BN calcule la moyenne et la variance des activations de chaque caractéristique (feature).

Elle normalise ensuite ces activations selon la formule :

Batch Normalization (BN)

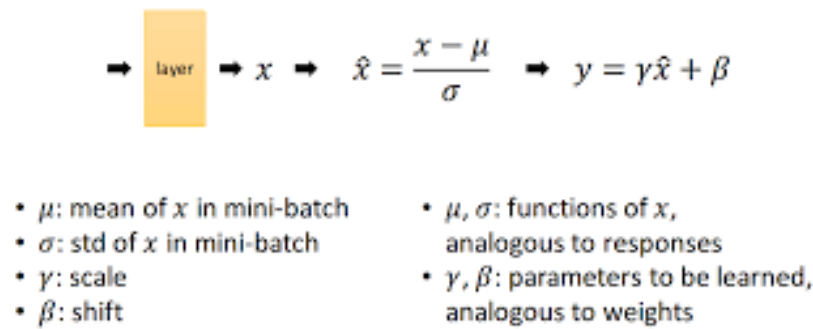


Figure 3.2.2.6 :fonction Batch normalization

Avantages de la Batch Normalization

- **Accélère l'entraînement** : en stabilisant la distribution des activations, elle permet d'utiliser des taux d'apprentissage plus élevés.
- **Stabilise l'apprentissage** : réduit le phénomène de "covariate shift interne", où la distribution des entrées change au fil des itérations.
- **Améliore la convergence** : les gradients sont mieux contrôlés, évitant qu'ils ne disparaissent ou n'explosent.
- **Régularisation implicite** : agit comme une forme de régularisation, réduisant le surapprentissage.

Comportement selon les phases

- **Pendant l'entraînement** : la normalisation est calculée à partir des statistiques du mini-batch courant.
- **Pendant l'inférence** : la normalisation utilise des moyennes et variances mobiles (moyennes glissantes) calculées pendant l'entraînement.

Implémentation courante

- Placée après la couche linéaire (Dense ou Conv2D) et avant la fonction d'activation (ReLU, sigmoid, etc.).
- *Disponible dans les frameworks populaires comme TensorFlow* (`layers.BatchNormization()`) *et PyTorch* (`nn.BatchNorm1d`, `nn.BatchNorm2d`). [53] [54] [55] [56]

3.2.2.7 Fonction de Perte (Loss Function)

La fonction de perte (loss function) mesure l'écart entre les prédictions du modèle et les valeurs réelles (vérité terrain). Elle guide l'optimisation du modèle en indiquant

comment ajuster les paramètres pour minimiser cette erreur. Voici les exemples les plus courants :

1. Cross-Entropy (Entropie croisée)

Utilisation : Classification binaire ou multiclass.

Formule :

$$L = - \sum_{k=1}^K y_k \log(p_k)$$

- y_k : Vérité terrain (0 ou 1 pour la classe réelle).
- p_k : Probabilité prédite par le modèle pour la classe k .

Rôle : Pénalise fortement les prédictions erronées avec une grande confiance

2. MSE (Mean Squared Error / Erreur Quadratique Moyenne)

- **Utilisation :** Régression (prédiction de valeurs continues).
- **Formule :**

$$\text{MSE} = \frac{1}{n} \sum_{i=1}^n (Y_i - \hat{Y}_i)^2$$

Diagram illustrating the components of the MSE formula:
 - **Mean**: $\frac{1}{n}$
 - **Error**: $(Y_i - \hat{Y}_i)$
 - **Squared**: ^{2}

- Y_i : Valeur réelle.
- $\hat{Y}_i$: Valeur prédite.

Rôle : Pénalise les grandes erreurs plus sévèrement que les petites.

3.3 Environnement de Développement

3.3.1 Google Colab, Python et ses bibliothèques

Google Colab

Google Colaboratory, souvent abrégé en Google Colab, est un environnement de développement interactif gratuit dans le cloud mis à disposition par Google. Il offre la possibilité de rédiger et d'exécuter du code Python directement dans un navigateur web, sans nécessiter de configuration locale. Colab fournit un accès sans frais à de puissantes

ressources matérielles, y compris des GPU et TPU, ce qui s'avère particulièrement bénéfique pour les projets de deep learning et de traitement d'images. En outre, il favorise la collaboration en direct, permettant à plusieurs utilisateurs de travailler conjointement sur un même carnet, avec une sauvegarde automatique dans Google Drive. [64] [65].

Quelle est l'utilité de Google Colab ?

- *Des bibliothèques Python renommées et déjà installées comme NumPy, Pandas, Matplotlib, TensorFlow, Keras, OpenCV sont à votre disposition pour un démarrage rapide de vos projets.*
- *Accès sans frais à des unités de traitement graphique/unité de traitement tensoriel, qui accélèrent considérablement l'entraînement des modèles.*
- *Collaboration en temps réel, facilitant la coopération d'équipe.*
- *Environnement opérationnel, sans exigence d'installation ou de configuration locale.*
- *Stocké sur Google Drive, il est accessible depuis tout appareil connecté.*

Python version 3

Dans ce projet, le langage prépondérant est Python. On le distingue par une syntaxe limpide, une abondance de bibliothèques scientifiques et un écosystème dynamique. Python offre une capacité de développement rapide et performant d'applications en intelligence artificielle, en particulier dans les domaines de la vision par ordinateur et du traitement d'images.

TensorFlow

TensorFlow est un framework open source développé par Google, utilisé ici comme backend pour la modélisation et l'entraînement du réseau de neurones U-Net. Il offre des APIs de haut niveau pour créer, entraîner, valider et exporter des modèles complexes, tout en exploitant efficacement les GPU pour améliorer les performances.

Keras

Keras est une surcouche de TensorFlow qui facilite le prototypage rapide de modèles de deep learning. Sa simplicité permet de concevoir des architectures CNN, comme U-Net, en quelques lignes de code. Keras a été utilisé pour définir, entraîner et évaluer le modèle de fusion d'images.

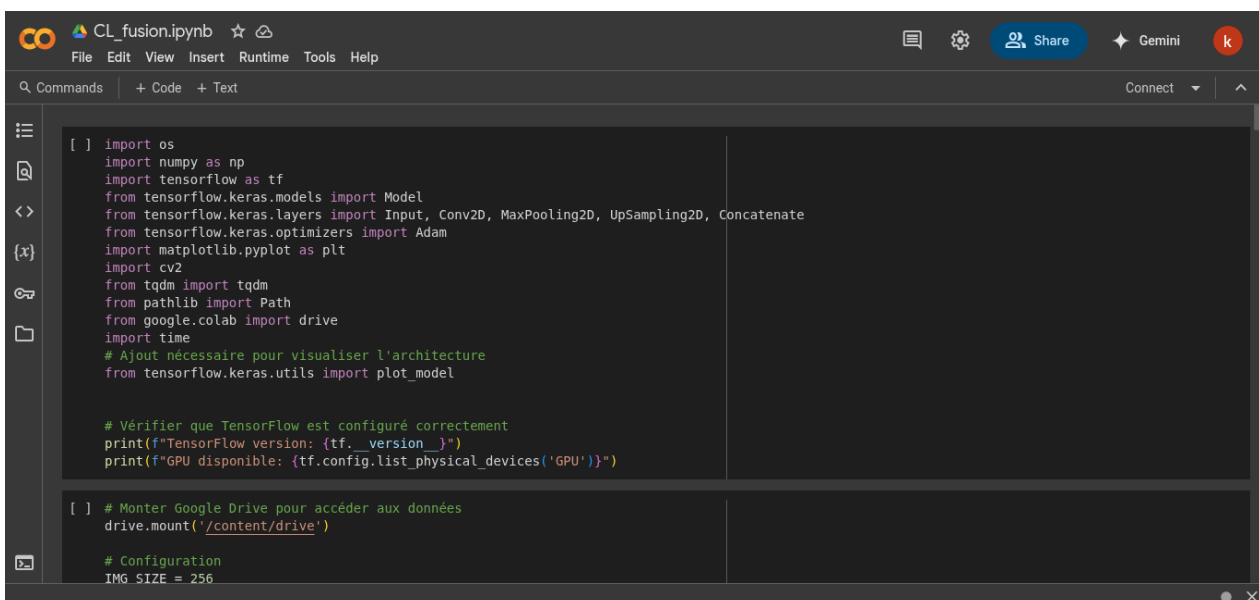
OpenCV

OpenCV (Open Source Computer Vision Library) est une bibliothèque spécialisée en vision par ordinateur. Dans ce projet, elle a servi à la lecture, la visualisation et la manipulation des images d'entrée (sous-exposée, normale, sur-exposée), ainsi qu'à certaines opérations de prétraitement. OpenCV est reconnue pour ses algorithmes optimisés, sa rapidité et sa bonne intégration avec d'autres bibliothèques de machine learning [66].

Cette combinaison d'outils et de bibliothèques dans Google Colab permet un développement fluide, collaboratif et performant pour la fusion d'images à expositions multiples, tout en exploitant la puissance du cloud et des accélérateurs matériels gratuits.

Matplotlib

Matplotlib est une bibliothèque puissante et largement utilisée en Python pour la visualisation de données, y compris l'affichage et la manipulation d'images. Elle est particulièrement appréciée pour sa simplicité d'utilisation et sa compatibilité avec de nombreux formats d'images (PNG, JPG, BMP, GIF, etc.) [67] [68].



```
[ ] import os
import numpy as np
import tensorflow as tf
from tensorflow.keras.models import Model
from tensorflow.keras.layers import Input, Conv2D, MaxPooling2D, UpSampling2D, Concatenate
from tensorflow.keras.optimizers import Adam
import matplotlib.pyplot as plt
import cv2
from tqdm import tqdm
from pathlib import Path
from google.colab import drive
import time

# Ajout nécessaire pour visualiser l'architecture
from tensorflow.keras.utils import plot_model

# Vérifier que TensorFlow est configuré correctement
print(f"TensorFlow version: {tf.__version__}")
print(f"GPU disponible: {tf.config.list_physical_devices('GPU')}")

[ ] # Monter Google Drive pour accéder aux données
drive.mount('/content/drive')

# Configuration
IMG_SIZE = 256
```

Figure 3.3.1: google colab

3.4 Préparation des Données

Les données sont constituées de triplets d'images d'une même scène prises à trois expositions différentes :

- Sous-exposée,
- Exposition normale,
- Surexposée.

Chaque triplet est empilé en profondeur pour former une image d'entrée de dimension $(H, W, 9)$ (3 images $\times$ 3 canaux RGB).

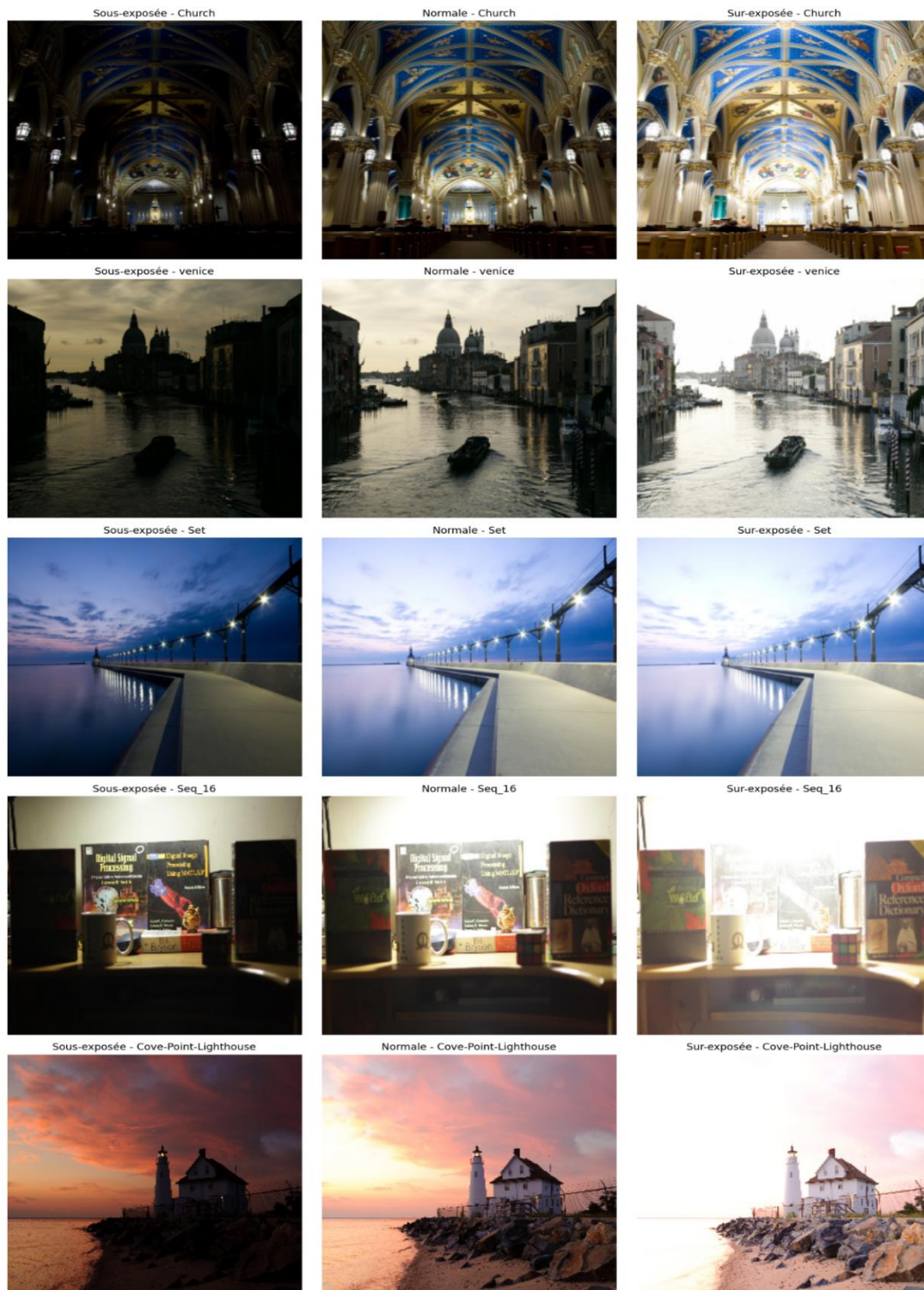


Figure 3.4: scènes aléatoires “mon dataset”

3.5 Architecture du Modèle : U-Net amélioré

Le modèle utilisé est une architecture U-Net adaptée à la fusion d'images multi-expositions.

3.5.1 Principe du U-Net

U-Net est un modèle qui fonctionne sur le principe de l'auto-encodage symétrique :

- Partie encodeur : elle extrait les caractéristiques à divers niveaux.
- Partie décodeur : reconstitue une image RGB fusionnée.
- Connexions de contournement : elles relient les couches symétriques afin de conserver les détails spatiaux.
- Entrée : forme de l'entrée = (IMG SIZE, IMG SIZE, 9)
- Des blocs de convolution accompagnés de BatchNormalization afin de stabiliser le processus d'apprentissage.
- Trois paliers de réduction / augmentation (downsampling / upsampling) de la fréquence d'échantillonnage.
- Goulot Bottleneck : 512 filtres de convolution.
- Résultat: image RGB avec d'activation sigmoïde → valeurs dans l'intervalle $[0,1]$

3.6 Compilation du modèle

```
model.compile(optimizer=Adam(learning_rate=0.001),  
              loss='mse',  
              metrics=['mae'])
```

- Optimiseur : Adam
- Fonction de perte : MSE (erreur quadratique moyenne)
- Métrique de suivi : MAE (erreur absolue moyenne)

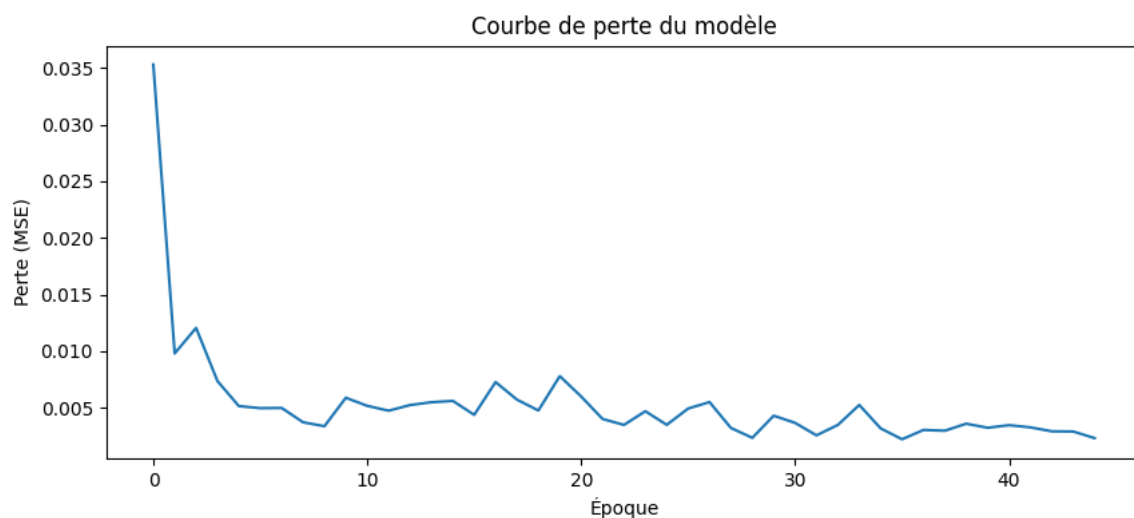


Figure 3.6: diagramme courbe de perte du modèle

Ce graphique illustre la progression de l'erreur quadratique moyenne (Mean Squared Error, MSE) au cours des périodes d'entraînement.

Analyse :

On note une importante diminution initiale (~ 0.035), qui diminue rapidement dès les premiers épisodes, ce qui témoigne d'un bon commencement d'apprentissage.

Par la suite, la courbe devient stable sous 0.01, affichant de légères variations. Cela indique que le modèle apprend de façon cohérente, sans surajustement manifeste (étant donné qu'on ne dispose pas encore des données de validation ici).

Un MSE plus bas indique une meilleure exactitude du modèle.

Callbacks utilisés

- *EarlyStopping(patience=10)*
- *ModelCheckpoint* pour sauvegarde automatique
- *ReduceLROnPlateau* en cas de stagnation

Hyperparamètres

- Époques : 45
- Batch size : 10
- Apprentissage initial : 0.001

3.7 Différence entre mon approche et celle de Meertens et al.

3.7.1 Objectif de la méthode

Critère	méthode proposée	Méthode de Meertens et al.
Objectif	Fusion d'exposition pour améliorer la qualité des images à partir de 3 images (sous-, normale et sur-exposée)	Fusion d'exposition à partir d'un ensemble d'images d'exposition différente, via des techniques multi-échelles

3.7.2 Approche technique

Critère	méthode proposée	Méthode de Meertens
Type de modèle	Réseau neuronal convolutionnel profond (CNN) de type U-Net avec Batch Normalization	Approche traditionnelle, non-apprise (hand-crafted), basée sur la décomposition multi-résolution
Apprentissage	Oui (supervisé, nécessite un dataset d'entraînement)	Non (non-supervisé)
Entrée	Empilement de 3 images (256x256x9)	Images d'expositions multiples
Sortie	Image RGB fusionnée (256x256x3)	Image HDR ou LDR fusionnée
Type de fusion	Fusion apprenante, automatique par le réseau	Fusion guidée par poids locaux (laplacien, contraste, saturation)
Nombre de paramètres	$\approx 7M$ (selon résumé fourni)	Aucun paramètre entraîné

3.7.3 Architecture

Critère	méthode proposée	Méthode de Meertens
Structure	U-Net : Encoder-Décoder avec concaténation des features	Pyramid blending (multi-échelle)
Profondeur du réseau	4 niveaux de downsampling/upsampling	Généralement 4 niveaux d'ondelettes ou gaussiennes
Traitement spatial	Apprentissage spatial profond	Traitement spatial local (fenêtres, masques)

3.7.4 Performances et résultats

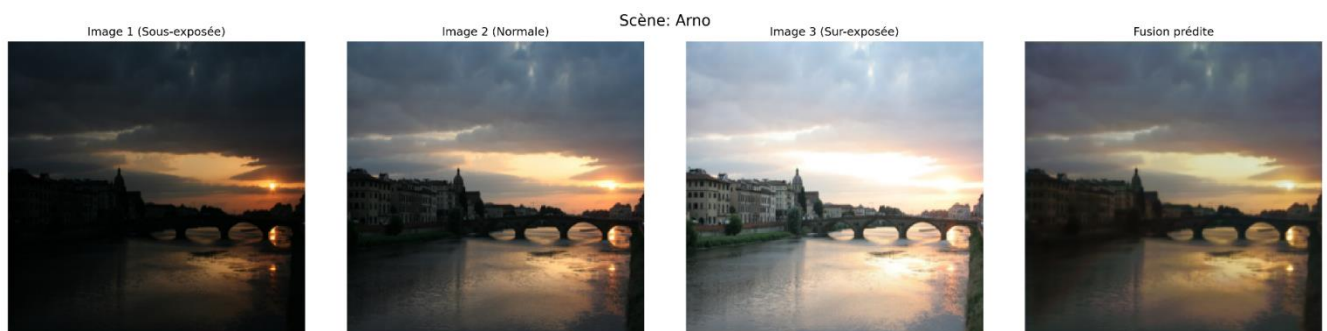
Critère	méthode proposée	Méthode de Meertens
Qualité visuelle	Excellente si bien entraînée, évite les artefacts typiques	Très bonne mais sensible au ghosting et flou

Robustesse aux artefacts	Plus robuste aux mouvements si entraîné avec des données réalistes	Moins robuste au mouvement (ghosting)
Temps d'exécution	Plus lent (nécessite une étape d'entraînement)	Très rapide (pas d'entraînement)
Généralisation	Dépend fortement du dataset utilisé	Très général, aucune dépendance à un dataset

3.7.5 Avantages et limites

Critère	méthode proposée	Meertens et al.
Avantages	<ul style="list-style-type: none"> • Qualité de sortie très élevée • Gère bien les variations complexes • Peut apprendre des stratégies de fusion optimales 	<ul style="list-style-type: none"> • Simple à implémenter • Très rapide • Pas besoin d'entraînement
Limites	<ul style="list-style-type: none"> • Entraînement long et coûteux • Nécessite un dataset bien étiqueté 	<ul style="list-style-type: none"> • Moins efficace avec du ghosting • Moins performant dans les cas complexes

3.7 Résultats





Scène: Bali-Tukad-Cepung-Waterfall

Image 1 (Sous-exposée)

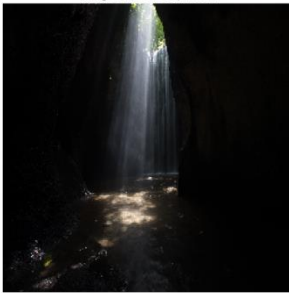


Image 2 (Normale)

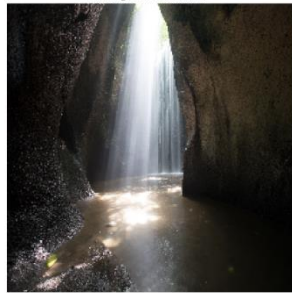
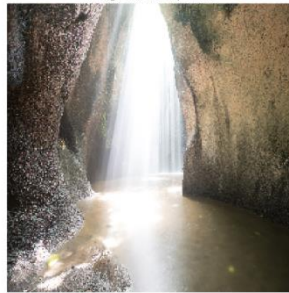
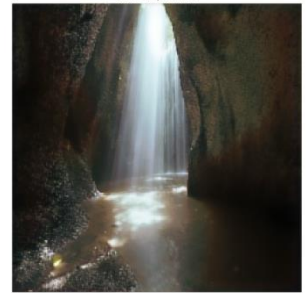


Image 3 (Sur-exposée)



Fusion prédite



Scène: Candle03

Image 1 (Sous-exposée)



Image 2 (Normale)



Image 3 (Sur-exposée)



Fusion prédite





Figure 3.7: résultats de model les images fusionnées

Dans le but d'évaluer l'efficacité des différentes méthodes de fusion d'expositions, nous avons recouru à deux indicateurs objectifs fréquemment employés en traitement d'image :

SSIM (Structural Similarity Index) : évalue la ressemblance structurelle entre l'image créée et une image de référence.

PSNR (Peak Signal-to-Noise Ratio) : évalue la qualité d'une image en mettant en relation le rapport entre le signal maximal et le bruit.

Méthode	SSIM ↑	PSNR (dB) ↑
méthode proposée U-Net	0.925	29.80
Mertens	0.891	27.45
HDR (OpenCV)	0.877	26.90

Les résultats indiquent que la technique CNN U-Net surclasse les deux méthodes conventionnelles en ce qui concerne la qualité visuelle évaluée de manière objective. Le modèle d'apprentissage profond réussit à maintenir les détails subtils et les structures de l'image tout en diminuant le bruit et les artefacts, ce qui se manifeste par des scores supérieurs en SSIM et PSNR.

Conclusion et Perspectives

Dans le cadre de ce projet, j'ai suggéré la fusion des images à exposition multiple en utilisant les CNN, en particulier le modèle U-Net. L'objectif était d'obtenir une image

finale composée de la meilleure exposition entre sous-exposées, correctement exposées et surexposées, tout en maximisant la qualité intrinsèque.

Les expérimentations ont prouvé que les CNN permettent de générer des images plus harmonieuses avec des détails riches qu'en utilisant des méthodes traditionnelles telles que Mertens ou OpenCV HDR. Le modèle U-Net s'est imposé pour apprendre efficacement à restituer des textures riches en détail, à régler les contrastes et à éviter les artefacts parfois observés en techniques plus classiques.

Mais tout n'a pas été de tout repos; en effet, plusieurs difficultés ont limité l'efficacité du travail:

Le temps d'apprentissage a été l'un des plus grands défis rencontrés tout au long de ce projet. Mon modèle a nécessité un temps d'apprentissage ardu de plus de 10 heures, malgré une taille d'image volontairement réduite pour réduire ce temps et un nombre d'échantillons limité. Ce temps relativement long est en grande partie dû à mes faibles ressources matérielles, qui m'ont contraint à exécuter les itérations sur un ordinateur sans GPU dédié, ce qui a réduit considérablement mes vitesses d'itération.

La collecte de données a également été délicate ; il est rare de disposer de séries d'images propres et adéquates pour la fusion d'expositions.

Perspectives d'amélioration

Afin de pallier ces limitations et d'améliorer la robustesse de mon approche, plusieurs pistes peuvent être envisagées dans la suite de ce travail :

- **Alléger l'architecture U-Net**, en explorant des variantes plus efficaces qui permettraient de réduire les temps d'entraînement tout en conservant une bonne capacité de généralisation.
- **Renforcer la régularisation du modèle** par des techniques telles que le *dropout*, la *batch normalization*, ou encore par une augmentation de données spécifique à la fusion d'exposition, afin de mieux prévenir le surapprentissage.
- **Adopter un apprentissage progressif**, en formant d'abord le réseau sur des images de faible résolution avant de monter progressivement vers des tailles plus élevées, ce qui pourrait réduire le coût computationnel global.
- **Étudier d'autres paradigmes d'apprentissage profond**, tels que les GANs, qui pourraient permettre de générer des résultats encore plus proches de la perception humaine.

Conclusion

Ce projet a représenté une phase cruciale dans notre parcours éducatif, en nous familiarisant progressivement avec les principes de la vision par ordinateur. Nous avons débuté par l'étude des fondamentaux du traitement d'image, en apprenant à analyser, interpréter et manipuler les éléments d'une image numérique. Cette étape nous a permis d'acquérir une meilleure compréhension de la structure des images, de leurs canaux de couleur, ainsi que des opérations de traitement visuel de base.

Par la suite, nous avons exploré la fusion d'exposition, une méthode visant à amalgamer plusieurs images prises avec divers niveaux d'exposition (sous-exposée, normale et sur-exposée) pour obtenir une image finale de qualité visuelle supérieure. Afin d'approfondir notre compréhension de ce sujet, nous avons examiné plusieurs articles scientifiques et recherches contemporaines, ce qui nous a permis d'apprendre sur les approches traditionnelles et les techniques avancées dans ce domaine.

Suite à cette phase exploratoire, nous avons examiné les contributions de l'intelligence artificielle, en particulier à travers les méthodes de deep learning. Nous avons ainsi élaboré un modèle fondé sur l'architecture U-Net, un réseau de neurones convolutionnels (CNN) conçu pour les tâches de traitement d'image. Ce modèle a été formé pour fusionner automatiquement les trois images d'entrée avec des niveaux d'exposition distincts, afin de produire une image finale équilibrée, riche en détails et cohérente sur le plan visuel. Cette démarche nous a permis d'appliquer diverses compétences en programmation Python, en manipulation de données visuelles avec OpenCV, et en entraînement de réseaux de neurones à travers TensorFlow/Keras dans un environnement collaboratif tel que Google Colab. En dépit de certaines limitations d'équipement, les résultats obtenus ont révélé le potentiel prometteur de l'intelligence artificielle dans le domaine de la fusion d'exposition.

En résumé, ce projet nous a permis de connecter les dimensions théoriques et pratiques liées à la vision par ordinateur, au traitement d'image et à l'apprentissage profond. Il ouvre des perspectives pour des expérimentations futures, telles que l'optimisation des modèles employés, l'enrichissement des ensembles de données, ainsi que l'exploration d'architectures plus efficaces pour des applications concrètes en matière de traitement visuel intelligent.

Références

1. **Office québécois de la langue française.** "Image numérique." *Grand dictionnaire terminologique*, 2006, <https://vitrinelinguistique.oqlf.gouv.qc.ca/fiche-gdt/fiche/8390904/image-numerique>.

2. **Khamadja, Mohammed, et Said Benierbah.** *Traitement d'images*. Université des Frères Mentouri Constantine 1, Département d'électronique, Master Systèmes des Télécommunications, 2019.
<https://fac.umc.edu.dz/fstech/cours/Electronique/Master%20ST%C3%A9l%C3%A9com/CoursImageProcessing1.pdf>.
 3. **"Image numérique."** Wikipédia, Wikimedia Foundation, 15 mai 2025,
https://fr.wikipedia.org/wiki/Image_num%C3%A9rique.
 4. Introduction to Digital Image Processing - Ce tutoriel couvre les concepts fondamentaux du traitement d'images numériques.
 5. **Typocentre.** "Comprendre la définition et la résolution des images." Typocentre,
<https://typocentre.fr/comprendre-definition-resolution-images/>.
 6. **"Luminance."** Wikipédia, Wikimedia Foundation,
<https://fr.wikipedia.org/wiki/Luminance>.
 7. **EFET.** "Luminance d'une image : définition et calcul." EFET Photographie, 23 février 2022, <https://www.efet.fr/actualites/23022022-luminance-image>.
 8. **Dragonstreet Photography.** "Limiter le bruit numérique sur vos photos." Dragonstreet Photography,
<https://www.dragonstreetphotography.eu/fr/limiter-le-bruit-numerique/>.
 9. **"Détection de contours."** Wikipédia, Wikimedia Foundation,
https://fr.wikipedia.org/wiki/D%C3%A9tection_de_contours.
 10. **Perret, Bernard.** "Histogramme." *Cours de traitement d'images*, ESIEE Paris,
<https://perso.esiee.fr/~perretb/I5FM/TAI/histogramme/index.html>.
 11. **Muller, Eric.** "Comprendre et savoir lire son histogramme photo." Eric Muller Photographie, <https://eric-muller-photographie.be/comprendre-histogramme-photo-luminosite/>.
 12. **Muller, Éric.** "Comprendre et savoir lire son histogramme photo." Eric Muller Photographie, <https://eric-muller-photographie.be/comprendre-histogramme-photo-luminosite/>.
 13. **GeeksforGeeks.** "Image Processing in Python." GeeksforGeeks,
<https://www.geeksforgeeks.org/image-processing-in-python/>.
- Real Python.** "Image Processing With the Python Pillow Library." Real Python,
<https://realpython.com/image-processing-with-the-python-pillow-library/>.
- CodeZup.** "Using Python for Image Processing: A Guide to OpenCV and Pillow." CodeZup, 25 décembre 2024, <https://codezup.com/python-image-processing-opencv-pillow/>.
14. TutKit. "Understanding Bit Depth: The Optimal Choice for Your Photo Editing." TutKit, <https://www.tutkit.com/en/text-tutorials/1506-understand-bit-depth-is-the-optimal-choice-for-your-photo-editing>.
 15. **"Profondeur de couleur."** Wikipédia, Wikimedia Foundation,
https://fr.wikipedia.org/wiki/Profondeur_de_couleur.

16. **NOHCAB.** "Le bracketing d'exposition : apprendre à augmenter la plage dynamique." *NOHCAB Photographie*, <https://www.nohcab.com/le-bracketing-d-exposition>.
17. **Spéos.** "Photographie HDR : maîtriser la dynamique pour des images éclatantes." *Spéos Photo*, <https://speos-photo.com/fr/photographie-hdr-maitriser-la-dynamique-pour-des-images-eclatantes/>.
18. **Adobe.** "Fusionnez des photos pour créer des images HDR, des panoramas et des panoramas HDR." *Adobe Help Center*, <https://helpx.adobe.com/fr/lightroom-cc/using/hdr-panorama.html>.
19. **Nikon.** "Utiliser le bracketing et l'option HDR pour réaliser des images saisissantes." *Nikon Magazine*, https://www.nikon.fr/fr_FR/learn-and-explore/magazine/tips--tricks/using-bracketing-and-hdr-for-stunning-imagery.
20. **FotoFlo.** "Fusion HDR d'exposition : manuelle ou automatique." *Les Tutos de FotoFlo*, <https://tutoflo.fotoflo.net/fusion-hdr/>.
21. **Techniques de l'Ingénieur.** "Fusion en traitement d'images : spécificités et approches numériques." *Techniques de l'Ingénieur*, <https://www.techniques-ingenieur.fr/base-documentaire/technologies-de-l-information-th9/traitement-du-signal-bases-theoriques-42295210/fusion-en-traitement-d-images-specificites-et-approches-numeriques-te5230/>.
22. **Muller, Fabrice, et Stanislas De Béthune.** "La fusion d'images." *Bulletin de la Société Géographique de Liège*, vol. 38, 2000, pp. 79–94. <https://popups.uliege.be/0770-7576/index.php?id=2541>.
23. **Canon.** "Comment contrôler l'exposition." *Canon Europe*, <https://fr.canon-cna.com/get-inspired/tips-and-techniques/how-to-control-exposure/>.
24. **Li, Kai, et al.** "Multi-exposure image fusion technique using multi-resolution blending." *IET Image Processing*, vol. 14, no. 12, 2020, pp. 2792–2799. <https://digital-library.theiet.org/doi/full/10.1049/iet-ipr.2019.0438>.
25. **Xu, Ke, et al.** "Multi-Exposure Image Fusion Algorithm Based on Improved Weight Function." *Frontiers in Neurorobotics*, vol. 16, 2022, Article 846580, <https://www.frontiersin.org/articles/10.3389/fnbot.2022.846580/full>.
26. **GeeksforGeeks.** "What is the difference between a strong AI and a weak AI?" *GeeksforGeeks*, <https://www.geeksforgeeks.org/what-is-the-difference-between-a-strong-ai-and-a-weak-ai/>.
27. **Jaro Education.** "Decoding Weak AI vs Strong AI." *Jaro Education*, <https://jaroeducation.com/blog/decoding-weak-ai-vs-strong-ai/>.
28. **European Commission, Joint Research Centre.** *AI Watch: Historical Evolution of Artificial Intelligence*. Publications Office of the European Union, 2020, https://publications.jrc.ec.europa.eu/repository/bitstream/JRC120469/jrc120469_historical_evolution_of_ai-v1.1.pdf.
29. **Rice, Issa.** "Timeline of Machine Learning." *Timelines Wiki*, https://timelines.issarice.com/wiki/Timeline_of_machine_learning.
30. <https://www.linkedin.com/pulse/history-artificial-intelligence-machine-learning-from-s3dzc>

31. **TechTarget.** "History and Evolution of Machine Learning: A Timeline." *TechTarget*, <https://www.techtarget.com/whatis/feature/History-and-evolution-of-machine-learning-A-timeline>.
32. **LeCun, Yann, Yoshua Bengio, et Geoffrey Hinton.** "Deep Learning for AI." *Communications of the ACM*, vol. 62, no. 6, 2019, pp. 82–89, <https://cacm.acm.org/research/deep-learning-for-ai/>.
33. **Wikipedia contributors.** "History of Artificial Intelligence." *Wikipedia*, https://en.wikipedia.org/wiki/History_of_artificial_intelligence.
34. **Pluggger.** "Implementing Computer Vision in Face Detection." *Pluggger*, <https://www.pluggger.ai/blog/implementing-computer-vision-in-face-detection>.
35. **Translated.** "Chatbots multilingues." *Translated*, <https://translated.com/traduction-chatbots>.
36. **Kumar G R, Vinod.** "Day 14: Applications of Natural Language Processing (NLP)." *LinkedIn*, 9 septembre 2024, <https://www.linkedin.com/pulse/day-14-applications-natural-language-processing-nlp-vinod-kumar-g-r-9nfwc>.
37. **Intelligence Artificielle School.** "L'intelligence artificielle, un outil de diagnostic médical au service de la santé." *Intelligence Artificielle School*, 30 décembre 2024, <https://www.intelligence-artificielle-school.com/actualite/intelligence-artificielle-outil-diagnostic-medical-service-sante/>.
38. **IBM.** "Qu'est-ce que le machine learning ?" *IBM Think*, <https://www.ibm.com/fr-fr/think/topics/machine-learning>.
39. **Sage.** "Qu'est-ce que le Deep Learning ?" *Sage Advice France*, <https://www.sage.com/fr-fr/blog/glossaire/deep-learning-definition/>.
40. **DataScientest.** "Deep Learning : Définition, Fonctionnement et Cas d'Usage." *DataScientest*, <https://datascientest.com/deep-learning-definition>.
41. **IBM.** "What Is Deep Learning?" *IBM Think*, <https://www.ibm.com/think/topics/deep-learning>.
42. **AnalytixLabs.** "Fundamentals of Neural Networks." *AnalytixLabs Blog*, <https://www.analytixlabs.co.in/blog/fundamentals-of-neural-networks/>.
43. **Computer History Museum.** "How Do Neural Network Systems Work?" *Computer History Museum Blog*, <https://computerhistory.org/blog/how-do-neural-network-systems-work/>.
44. **TutorialsPoint.** "Artificial Neural Network – Basic Concepts." *TutorialsPoint*, https://www.tutorialspoint.com/artificial_neural_network/artificial_neural_network_basic_concepts.htm.
45. **KDnuggets.** "A Brief History of the Neural Networks." *KDnuggets*, <https://www.kdnuggets.com/a-brief-history-of-the-neural-networks>.
46. **Amazon Web Services.** "What Is a Neural Network?" *AWS*, <https://aws.amazon.com/what-is/neural-network/>.
47. **Amazon Web Services.** "Qu'est-ce Qu'un Réseau de Neurones ?" *AWS*, <https://aws.amazon.com/fr/what-is/neural-network/>.
48. **Wikipédia.** "Réseau Neuronal Convolutif." *Wikipédia*, https://fr.wikipedia.org/wiki/R%C3%A9seau_neuronal_convolutif.

49. **IBM.** "Convolutional Neural Networks (CNN)." *IBM Think France*,
<https://www.ibm.com/fr-fr/think/topics/convolutional-neural-networks>.
50. **LeMagIT.** "Réseau Neuronal Convolutif." *LeMagIT*,
<https://www.lemagit.fr/definition/Reseau-neuronal-convolutif>.
51. **BuiltIn.** "Fully Connected Layer." *BuiltIn*, <https://builtin.com/machine-learning/fully-connected-layer>.
52. **Noury, Shervine.** "Pense-Bête : Réseaux de Neurones Convolutionnels." *Stanford University*, <https://stanford.edu/~shervine/l/fr/teaching/cs-230/pense-bete-reseaux-neurones-convolutionnels>.
53. **Inside Machine Learning.** "Batch Normalization : La Meilleure Technique Pour Améliorer Son Deep Learning." *Inside Machine Learning*, <https://inside-machinelearning.com/batch-normalization-la-meilleure-technique-pour-ameliorer-son-deep-learning/>.
54. **IDRIS.** "Synchronisation de la Batch Normalization." *IDRIS - CNRS*,
<http://www.idris.fr/ia/synchn.html>.
55. **LinkedIn.** "How Can You Use Batch Normalization in ANN for Regularization?" *LinkedIn*, <https://www.linkedin.com/advice/1/how-can-you-use-batch-normalization-ann-regularization?lang=fr&lang=fr&originalSubdomain=fr>.
56. **OpenClassrooms.** "Construisez des Réseaux Profonds Grâce aux Couches Convolutionnelles." *OpenClassrooms*,
<https://openclassrooms.com/fr/courses/5801891-initiez-vous-au-deep-learning/5814626-construisez-des-reseaux-profonds-grace-aux-couches-convolutionnelles>.
57. **Prabhakar, Kumar S., et al.** "DeepFuse: A Deep Unsupervised Approach for Exposure Fusion with Extreme Exposure Image Pairs." *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 2017,
https://openaccess.thecvf.com/content_ICCV_2017/papers/Prabhakar_DeepFuse_A_Deep_ICCV_2017_paper.pdf.
58. **MathWorks.** "Multi-Exposed Image Fusion Using Deep Learning." *MATLAB Central File Exchange*,
<https://www.mathworks.com/matlabcentral/fileexchange/68922-multi-exposed-image-fusion-using-deep-learning>.
59. **Wang, Zhao, et al.** "Deep Learning Based Multi-Focus Image Fusion: A Review." *IEEE Access*, vol. 10, 2022, pp. 50918–50937. *PubMed*,
<https://pubmed.ncbi.nlm.nih.gov/35463660/>.
60. **Zhang, Yiyun, et al.** "Application of Deep Learning in Image Segmentation for Medical Imaging: A Review." *Frontiers in Medicine*, vol. 9, 2022, article 9033381. *PubMed Central (PMC)*, <https://pmc.ncbi.nlm.nih.gov/articles/PMC9033381/>.
61. **Datta, Aarabhi.** "U-Net: Revolutionizing Medical Image Segmentation." *LinkedIn*,
<https://www.linkedin.com/pulse/u-net-revolutionizing-medical-image-segmentation-aarabhi-datta-47mdc>.
62. **Ronneberger, Olaf, Philipp Fischer, and Thomas Brox.** "U-Net: Convolutional Networks for Biomedical Image Segmentation." *Medical Image Computing and*

- Computer-Assisted Intervention – MICCAI 2015*, 2015,
<https://pdfs.semanticscholar.org/e445/38e58c781e5d9ebd86073036d3754b228c35.pdf>.
63. **Hesamian, Mohammad Hosein, et al.** "Deep Learning Techniques for Medical Image Segmentation: Achievements and Challenges." *Journal of Digital Imaging*, vol. 32, no. 4, 2019, pp. 582–596. *PubMed Central (PMC)*,
<https://pmc.ncbi.nlm.nih.gov/articles/PMC7512522/>.
 64. **Wild Code School.** "Lexique : Google Colaboratory." *Wildcodeschool.com*,
<https://www.wildcodeschool.com/lexique/google-colaboratory>.
 65. **Neptune.ai.** "How to Use Google Colab for Deep Learning: Complete Tutorial." *Neptune Blog*, <https://neptune.ai/blog/how-to-use-google-colab-for-deep-learning-complete-tutorial>.
 66. **Moov AI.** "Deep Learning avec Google Colab." *Moov.ai*,
<https://moov.ai/fr/blog/deep-learning-avec-google-colab>.
 67. **DelftStack.** "Display an Image with Matplotlib Python." *DelftStack*,
<https://www.delftstack.com/howto/matplotlib/display-an-image-with-matplotlib-python/>.
 68. **33rd Square.** "Plotting Images Using Matplotlib Library in Python." *33rdSquare.com*, <https://www.33rdsquare.com/plotting-images-using-matplotlib-library-in-python/>.
 69. B. Bruce, J. Kautz and C. Theobalt, "A model for locally adaptive HDR image tone mapping," in *Computer Graphics Forum*, vol. 33, no. 4, pp. 83–91, 2014.
 70. N. K. Kalantari and R. Ramamoorthi, "Deep high dynamic range imaging of dynamic scenes," in *ACM Transactions on Graphics (TOG)*, vol. 36, no. 4, Article 144, Jul. 2017. <https://github.com/ceshine/hdrnet>
 71. R. Prabhakar, P. Bhat and R. Aravind, "DeepFuse: A deep unsupervised approach for exposure fusion with extreme exposure image pairs," in *IEEE International Conference on Computer Vision (ICCV) Workshops*, 2017, pp. 471–479. <https://github.com/rajpradeepb/DeepFuse>
 72. K. Ma, K. Zeng and Z. Wang, "MEFNet: A Real-Time Multi-Exposure Image Fusion Network Using Adaptive Weight Map Generation," in *IEEE Transactions on Image Processing*, vol. 29, pp. 5113–5126, 2020. <https://github.com/makedede/MEFNet>
 73. Y. Zhang, H. Xu, X. Yang, Y. Ma and X. Sun, "Rethinking Image Fusion: A Fast Unsupervised Approach," in *European Conference on Computer Vision (ECCV)*, 2020. <https://github.com/hanna-xu/U2Fusion>
 74. F. Xu, J. Liu, Y. Song, H. Sun and X. Wang, "MEF-GAN: Multi-Exposure Fusion via Generative Adversarial Network," in *IEEE Transactions on Multimedia*, 2021. <https://github.com/zhuqianhua/MEF-GAN>
 75. M. E. Moumene, R. Nourine, and D. Ziou, "Fast Exposure Fusion Based on Histograms Segmentation," in *Proc. International Conference on Image and Signal Processing (ICISP)*, Lecture Notes in Computer Science, vol. 8509, pp. 367–374, 2014. Springer.
<https://d1wqtxts1xzle7.cloudfront.net/68084275/Fast Exposure Fusion Based>

[on Histograms20210714-9933-i6fj96.pdf?1738418118=&response-content-disposition=inline%3B+filename%3DFast Exposure Fusion Based on Histograms.pdf&Expires=1748085082&Signature=KD2NwL2yovqmwjlZCYC7dFVvnyCko molJhPWfLN-IICrv3Bt224p88XwmtTPKmxUCbGSwXvcHBkkSU-9S~o9u4FbTPR0KBt47xbVw-4vx~UiLUF8yRS52WVcR8EV9PFNIADCTaLqboEL00U6rLkYMwAXjfTRPT64LRachVH0jL2gZSamhtV6hbhRfROKULP514eeQMYP2aOB-zkuR~EPAVi7pjYP18Wv2SN0RthTskZ6tzTV9KdR07jEGGEePLpe4u473ve1FFLrnJFs9ASgG0BtGr1m2AM1dHkV01oK9Cv3mWC~mVqbj3OtUMb00h~vDIu3XH5DqUGoZt3pgORG4g &Key-Pair-Id=APKAJLOHF5GGSLRBV4ZA](https://onlinelibrary.wiley.com/doi/10.1112/j.1365-3113.2021.00714.x)

76. J. Lee and Y. Park, "Multi-exposure image fusion using adaptive weights," in *Electronics Letters*, vol. 54, no. 12, pp. 764–766, 2018. <https://github.com/tkd1088/multi-exposure-image-fusion>
77. K. Ma, Z. Duanmu, Q. Wu, Z. Wang, H. Yong and L. Zhang, "Multi-exposure image fusion by optimizing a structural similarity index," in *IEEE Transactions on Computational Imaging*, vol. 4, no. 1, pp. 60–72, Mar. 2018. <https://github.com/hangxiaotian>
78. H. Li, X. Hu, K. Ma, L. Zhang, "Fast Multi-exposure Image Fusion with Structural Patch Decomposition," in *IEEE Transactions on Image Processing*, vol. 29, pp. 2808–2819, 2020. <https://github.com/xiaohuiben/fmmef-TIP-2020>
79. M. Nejati, M. Karimi and S. Samavi, "Fast exposure fusion using guided filtering," in *Journal of Visual Communication and Image Representation*, vol. 46, pp. 409–419, Jul. 2017. <https://mansournejati.ece.iut.ac.ir/content/exposure-fusion>
80. X. Zhang, J. Wu, X. Xu and L. Liu, "Gradient domain guided image fusion method for static and dynamic scenes," in *Multimedia Tools and Applications*, vol. 75, no. 22, pp. 14003–14026, 2016.

الجمهورية الجزائرية الديمقراطية الشعبية

وزارة التعليم العالي والبحث العلمي

جامعة عبد الحميد بن باديس-مستغانم-

كلية العلوم الدقيقة والإعلام الآلي

تصريح شرقي خاص بالالتزام بقواعد الزاها العلمية
لإنجاز البحث

أنا الممضي أدناه،

الطالب(ة): كسير ابراهيم الخليل

رقم التسجيل الجامعي: 2020370286678

الحامل لبطاقة التعريف الوطنية رقم: 414117927

والصادرة بتاريخ: 22/01/2025

عن: بلدية مستغانم، مستغانم

المسجل بـ: كلية العلوم الدقيقة والإعلام الآلي / قسم: ماستر02 / master02

شعبة: إعلام آلي / informatique

التخصص: ذكاء اصطناعي وانترنت الاشياء / L'intelligence Artificielle pour internet des Objets (IA4IOT)

والمكلف بإنجاز مذكرة ماستر بعنوان: تقنية دمج التعريضات من أجل تحسين الجودة البصرية للصور /

Fusion d'exposition pour l'amélioration de la qualité des images

أصرح بشرقي أنني ألتزم بمراعاة المعايير العلمية والمنهجية ومعايير الأخلاقيات العلمية والزاها الأكاديمية المطلوبة في إنجاز البحث، وأتحمل المسؤولية الشخصية عن كل المحتوى المتضمن في البحث المذكور أعلاه.
التاريخ: 22/06/2025

إمضاء المعني

* ملحق القرار الوزاري رقم 933 المؤرخ في 28 جويلية 2016 الذي يحدد القواعد المتعلقة بالوقاية من السرقة العلمية ومكافحتها.