

République Algérienne Démocratique et Populaire
Ministère de l'Enseignement Supérieur et de la Recherche Scientifique

Université Mohamed Sadik Benyahia de Jijel
Faculté des Sciences Exactes et informatique
Département d'Informatique



Mémoire Fin d'étude
Option : Informatique Légale et Multimedia

Thème

Utilisation des réseaux de neurones
convolutifs pour la suppression
automatique des filigranes

Présenté par :

Krine Zohra.

Abdi Ahlem .

Encadré par :

Boudjadja Rima

Promotion : 2020.

Remerciements

Nous adressons en premier lieu notre reconnaissance à ALLAH, notre DIEU unique et miséricordieux, de nous avoir donné santé et courage, pour pouvoir achever ce modeste travail. Nous l'implorons de nous guider sur le droit chemin et de nous envelopper de sa bonté et de sa bénédiction.

AMEN!

Ce travail a été effectué au sein du département d'informatique, de l'université M.S. Benyahia, de Jijel, sous la direction de Mme Rima Boudjadja, enseignante au sein du même département. Nous lui adressons nos plus vifs et sincères remerciements et gratitude pour avoir accepté la responsabilité d'encadrer notre mémoire et la confiance qu'elle nous a accordée; pour les orientations judicieuses qu'elle nous a prodiguées dans certains moments d'égarement et son soutien moral. Sa disponibilité permanente durant la réalisation de ce travail, sa courtoisie et sa gentillesse ne se sont jamais démenties. C'est tout à son honneur.

Nous adressons nos plus vifs remerciements aux membres du jury qui nous ont fait l'honneur d'accepter d'examiner notre mémoire. Qu'ils trouvent ici, l'expression de nos sentiments les plus empressés.

Un grand merci à l'ensemble des enseignants de la Faculté des Sciences Exactes et Informatique pour leur patience et leur dévouement envers les étudiants.

Nous tenons à remercier aussi Dr Abdi Ilham, Mr Benzaid mohammed said et Mlle Chalouche Meriem pour leur soutien désintéressé, et leurs précieux conseils tout au long de la préparation de ce Mémoire. Toute notre sympathie pour eux.

Nous ne saurons oublier nos camarades de promotion. Qu'ils trouvent ici, le témoignage de notre sympathie.

Des remerciements chaleureux et particuliers vont à l'encontre de nos parents, frères et soeurs pour leur soutien et leur enthousiasme.

Nous remercions toute personne ayant consacré un moment, pour nous aider, nous conseiller, ou nous encourager.

A tous, nous disons : soyez-en sincèrement remerciés.

Dédicace

Dédicas 1

A mes Parents

Aucun mot ne pourra exprimer la profondeur de mes sentiments pour vous. C'est avec le plus grand amour et dévouement que je vous offre ce modeste travail, en hommage de ma gratitude qui ne sera jamais assez pour tous vos sacrifices et votre abnégation. Je ne vous remercierai jamais assez. J'espère simplement que vous êtes fiers de moi. Qu'ALLAH, notre DIEU, le tout puissant, unique et miséricordieux, vous bénisse et vous prête santé et longue vie, afin que je puisse toujours vous servir et vous combler.

A mes frères et à ma soeur

Vous étiez et resterez toujours les meilleurs amis pour moi. Veuillez trouver en ce travail, le témoignage de toute mon affection et de ma tendresse.

A celui que j'aime beaucoup et qui ma soutenue tout au long de ce travail mon fiancé Mehdi, et à sa famille.

A toute ma famille.

A ma chère binôme Abdi Ahlem .

A toutes mes amis et mes collègues .

A tous ceux qui me sont chers.

A moi-même.

Zohra

Dédicace

Dédicas 2 :

A mes parents .

A mon frères et mes soeur .

A toute ma famille.

Ma chère binôme « Krine zohra » et à toute sa famille

A toutes mes amis et mes collègues .

Ahlem

Table des matières

Liste des abréviations	i
Dédicace	ii
Dédicace	iii
Table des matières	iv
Liste des tableaux	ii
Table des figures	iv
Liste des abréviations	1
Introduction	2
1 Les réseaux de neurones et l'apprentissage profond	5
Les réseaux de neurones et l'apprentissage profond	5
1.1 Introduction	5
1.2 Description des réseaux de neurones	5
1.3 Les neurones biologiques et artificiels	7
1.4 Le neurone formel	7
1.4.1 Le perceptron simple	8
1.4.2 Le perceptron multicouche (PMC)	9
1.4.3 Fonctions d'activation	10
1.4.4 La rétropropagation	11
1.5 L'Apprentissage	11
1.5.1 Apprentissage supervisé	12
1.5.2 Apprentissage non supervisé	12
1.6 L'apprentissage profond	13
1.7 Les réseaux de neurones convolutifs	13
1.7.1 Architecture des réseaux de neurones convolutifs	14

1.8	Auto-encodeur (AE)	17
1.8.1	Architecture des auto-encodeurs	18
1.8.2	Apprentissage des auto-encodeurs	19
1.8.3	Auto-encodeurs débruiteur	19
1.9	Conclusion	21
2	Filigrane numérique	23
	Filigrane numérique	23
2.1	Historique	24
2.2	Aux origines du tatouage	24
2.2.1	La cryptographie	24
2.2.2	La stéganographie	25
2.3	Types d'insertion	25
2.3.1	Méthodes additives	25
2.3.2	Méthodes substitutives	26
2.4	Type de filigrane	26
2.4.1	Filigrane visible	26
2.4.2	Filigrane invisible :	27
2.5	Critères de dissimulation d'information	27
2.5.1	Robustesse	27
2.5.2	Capacité	27
2.5.3	Imperceptibilité	28
2.6	Applications de filigrane numérique des images	28
2.7	La suppression de filigrane par des applications	29
2.8	Outils d'évaluation	30
2.8.1	Evaluation subjective	30
2.8.2	Evaluation objective	33
2.8.3	Catégories des métriques objectives	33
2.9	Conclusion	39
3	Conception et implémentation	41
	Conception et implémentation	41
3.1	Introduction :	41
3.2	Configuration matérielle	41
3.2.1	Configuration matérielle distante (Google Colab)	41
3.3	Environnement de développement logiciel	41
3.3.1	Python	41
3.3.2	Tensorflow	42
3.3.3	Keras	42

3.4	Dataset	42
3.4.1	Découpage de la base d'images	42
3.4.2	Fonction de redimensionnement	42
3.5	Présentation de l'application	42
3.5.1	Auto-encodeur (AE)	43
3.5.2	Combinaison auto-encodeur débruiteur	43
3.5.3	Configuration du CNN	43
3.5.4	les méthodes d'évaluation :	44
3.6	Résultats d'application	46
3.6.1	Les résultats obtenus par l'auto-encodeur convolutif	46
3.6.2	les résultats obtenus par l'autoencodeur débruiteur convolutif	47
3.6.3	Évaluation par CW-SSIM :	51
3.7	Les résultat obtenus par un débruiteur convolutif	52
3.7.1	Pearson coefficient de corrélation(PCC)	54
3.8	Conclusion	54
	Conclusion	55
	Bibliographie	56
	Résumé	58

Liste des tableaux

2.1	Echelle d' évaluation de qualité d'image	32
3.1	Configuration de notre CNN	45
3.2	Tableau des PCC calculés	54

Table des figures

1.1	Mise en correspondance neurone biologique /neurone artificiel	7
1.2	Structure d'un perceptron simple	8
1.3	Structure d'un perceptron multicouche. [9]	10
1.4	Apprentissage des réseaux de neurone par l'algorithme de rétropropagation	12
1.5	Architecture standard d'un réseau de neurones convolutifs	14
1.6	Exemple de convolution avec un filtre 3 x 3	15
1.7	Exemples de max et moyenne pooling sur une fenêtre 2x2 et pas 2	16
1.8	Exemple de la fonction ReLu	17
1.9	Les différentes couches d'un CNN	17
1.10	Principe d'un auto-encodeur sur une image	18
1.11	Image avec le bruit poivre et sel	20
1.12	Image avec le bruit Gaussienne	20
1.13	Le bruit de speckle(bruit multiplicatif)	21
1.14	Autoencodeur débruiteur	22
2.1	Filigrane visible	26
2.2	Filigrane invisible	27
2.3	Logo Gimp	29
2.4	Logo Photoshop	30
2.5	Méthode à double stimulus	31
2.6	Méthode à simple stimulus	32
2.7	Schéma de métrique sans référence	37
3.1	L'architecture de notre Auto-encodeur (AE)	43
3.2	Combinaison auto-encodeur débruiteur	43
3.3	Composants de notre réseau de neurones	44
3.4	Plan d'évaluation	45
3.5	Image résultat d'auto-encodeur convolutif.	46
3.6	Graphes de Précision et d'Erreur de l'auto-encodeur convolutif.	46
3.7	Graphe de SSIM entre les images originales et les images obtenues par l'auto-encodeur convolutif.	47
3.8	Images résultats d'auto-encodeur convolutif débruiteur	47

3.9	Graphe de la Précision pour l’auto-encodeur convolutif débruiteur	48
3.10	Graphe de l’erreur pour l’auto-encodeur convolutif débruiteur	48
3.11	SSIM	48
3.12	Images résultats d’auto-encodeur débruiteur (text court)	49
3.13	Graphe de la Précision pour l’auto-encodeur débruiteur	49
3.14	Graphe de l’erreur pour l’auto-encodeur débruiteur	49
3.15	Graphe de SSIM	50
3.16	Images résultats d’auto-encodeur débruiteur (texte aléatoire).	50
3.17	Graphe de la Précision et de l’erreur pour l’auto-encodeur débruiteur	51
3.18	SSIM des images résultats (texte aléatoire).	51
3.19	Images résultats	51
3.20	Graphe de la Précision et de l’erreur pour le débruiteur(cwssim)	52
3.21	Graphe de SSIM à gauche et de CW-SSIM à droit	52
3.22	Images résultats débruiteur (texte aléatoire).	53
3.23	Graphe de la Précision et de l’erreur pour débruiteur	53
3.24	SSIM des images résultats d’un débruiteur(texte aléatoire).	54

Liste des abréviations

RLC	Run Length Coding
ITU	Information Telecommunication Union
DSCQS	Double Stimuli Continuous Quality Scale
DSIS	Double Stimulus Impairment scale
SSCQE	Single Stimulus Continuous Quality Evaluation
MOS	Mean Opinion Score
PSNR	Peak Signal Noise Ratio
EQM	Erreur Quadratique Moyenne
SNR	Signal Noise Ratio
SSIM	Structural SIMilarity
ANNs	Artificial Neural Networks
BIQI	Blind Image Quality Index
SVM	Support Vector Machine
FC	Couche entièrement connectée
DIVIINE	Distortion Identification-based Image Verity and INtegrity Evaluation
NR	Métriques sans référence
NSS	natural scène statistics
RNA	réseaux de neurones artificiels
CNN	Convolutional Neural Network
DNN	Deep Neural Network
RMSE	Root Mean Square Error
PMC	Perceptron Multi-Couche
ReLU	Rectified Linear Unit
AE	AutoEncoder
PCC	Pearson coefficient de corrélation

Introduction

La croissance démographique fulgurante et les besoins sans cesse croissants en infrastructures et biens d'équipements et de consommation, conjugués aux mutations sociopolitiques dans de nombreux pays, ont provoqué une expansion des nouvelles technologies de l'information et des communications (NTIC) et des supports audio-visuels de divertissement. Ces TIC jouent un rôle majeur dans le développement de la globalisation, notamment du fait de la réduction des coûts de communication. L'essor des NTIC s'est fait d'une manière plus rapide que prévu. En effet, plus de la moitié de la population mondiale a aujourd'hui, accès à ces nouvelles technologies notamment grâce à la révolution de la téléphonie mobile (Smartphones, tablettes,...). Avec le développement des récentes technologies des réseaux multimédias, le niveau d'exigence en qualité de service des systèmes multimédia s'accroît rapidement.

Le monde entier vit l'ère du tout numérique, où l'imagerie prend une place primordiale dans la société, parce que devenue incontournable dans les activités socio-économiques et culturelles quotidiennes et notamment dans la communication. Dès lors, les nouveaux défis des chercheurs et ingénieurs du domaine de l'imagerie et de la vidéo numériques, consistent à chercher une amélioration continue de la qualité de service des équipements d'imagerie et de vidéos numériques.

A la fin des années 80 Yann le Cun[1] a développé un type de réseau particulier, dénommé 'réseau de neurone convolutionnel'. Il consiste en une forme particulière de réseau neuronal multicouche dont l'architecture des connexions est inspirée de celle du cortex visuel des mammifères, plus précisément du cerveau humain. Par exemple, chaque élément n'est connecté qu'à un petit nombre d'éléments voisins dans la couche précédente.

Aujourd'hui, les réseaux de neurones convolutionnels (ou convolutifs) ont de nombreuses applications dans des domaines technologiques et industriels très variés. Nous pouvons citer à titre non exhaustif, le traitement de l'image (reconnaissance des formes, des motifs et des caractères...), du signal (traitement de la parole, filtrage des sons...) et des langues (traduction automatique...), ou encore le contrôle (diagnostic de pannes, contrôle qualité, diagnostic médical, robotique,...) et l'optimisation (planification, gestion des ressources, régulation des transports,...).

En 1995, Yann le cun[1] et deux autres ingénieurs ont développé un système automatique

de lecture de chèques qui a été déployé largement dans le monde. À la fin des années 90, ce système lisait entre 10 % et 20 % de tous les chèques émis aux États-Unis. Mais ces méthodes étaient plutôt difficiles à mettre en oeuvre avec les ordinateurs de l'époque, et malgré ce succès, les réseaux convolutionnels et les réseaux neuronaux plus généralement ont été délaissés par la communauté de la recherche entre 1997 et 2012.

En 2011 et 2012 trois événements ont soudainement changé la situation. Tout d'abord, les GPU (Graphical Processing Unit) capables d'effectuer plus de mille milliards d'opérations par seconde sont devenus disponibles pour un prix moins cher. Ces puissants processeurs spécialisés, initialement conçus pour le rendu graphique des jeux vidéo, se sont avérés être très performants pour les calculs des réseaux neuronaux.

Deuxièmement, des expériences menées simultanément à Microsoft, Google et IBM avec l'aide du laboratoire de Geoffrey Hinton ont montré que les réseaux profonds pouvaient diminuer de moitié les taux d'erreurs des systèmes de reconnaissance vocale.

Troisièmement plusieurs records en reconnaissance d'image ont été battus par des réseaux de neurones convolutionnels. L'événement le plus marquant a été la victoire éclatante de l'équipe de Toronto (Canada) dans la compétition de reconnaissance d'objets Image Net .[1] La diminution des taux d'erreurs était qualifiée d'une véritable révolution. Du jour au lendemain, la majorité des équipes de recherche en parole et en vision ont abandonné leurs méthodes préférées et sont passées aux réseaux de neurones convolutionnels et autres réseaux neuronaux. L'industrie d'Internet a immédiatement saisi l'opportunité et a commencé à investir massivement dans des équipes de recherche et développement en apprentissage profond.

D'autre part, nous avons constaté depuis quelques temps sur les photos des appareils photographiques, les photos contenues sur la toile et les applications de capture, que la majorité de ces photos comporte un titre ou une expression sous forme, soit d'un nom de marque ou un logo, soit une signature ou une adresse d'un site web. Ces motifs sont appelés filigranes. Au départ, ils ont été conçus dans le but de protéger les photos et les images contre les falsifications et les vols. Mais ces filigranes peuvent-être un handicap pour qui, veut utiliser ou exploiter ces images.

Ce qui nous intéresse, dans le cas de notre étude, c'est que le filigrane peut cacher une partie du contenu de l'image ou de la photo, ce qui altère sa qualité, surtout si elle comporte une multitude de détails. Dans ce cas, la compréhension ou la lecture de toute l'image (ou d'une partie) sera faussée et sa qualité rendue médiocre. De ce qui précède, notre but est de supprimer le filigrane de manière automatique en utilisant Deep Learning, c'est-à-dire, de façon plus explicite, nous concevons des réseaux de neurones convolutionnels capables de supprimer cette marque et d'effectuer un 'montage' qui soit le plus fidèle possible à la photo originale, sans passer par cette même photo de départ, tout en conservant sa qualité première.

Le deep learning ou apprentissage profond est un type d'intelligence artificielle dérivé du machine learning (apprentissage automatique) où l'ordinateur est capable d'apprendre par

lui-même, contrairement à la programmation où l'ordinateur se contente d'exécuter des règles déterminées au préalable.

Cet apprentissage profond est basé sur un réseau de neurones artificiels. Il est composé de dizaines, parfois de centaines de couches de neurones, chacune recevant les informations de la couche précédente. Le système apprendra par exemple à déterminer s'il y a un visage sur une photo avant de découvrir de quelle personne il s'agit, ou de reconnaître les lettres avant de s'attaquer aux mots dans un texte.

Le présent mémoire s'articule autour de trois chapitres :

- Le premier présente un aperçu général sur le vaste domaine des réseaux de neurones artificiels. Nous y présentons les concepts de bases du perceptron, du perceptron multicouches, la rétro propagation et les différents types de réseaux de neurones et nous abordons l'évolution majeure de ces dernières années : les réseaux de neurones profonds (deep learning). Nous détaillons l'architecture typique d'un réseau de neurones convolutionnel, qui est une partie fondamentale des réseaux profonds, par la présentation des couches qui le constituent.
- Le deuxième chapitre est consacré aux filigrane numérique. Nous sommes intéressées dans ce chapitre aux différents types d'insertion et nous avons présenté deux méthode pour la suppression de filigrane . Nous avons aussi présenté les notions liées aux méthodes de mesure de la qualité des images.
- Dans le troisième chapitre nous étudions et implémentons un débruiteur qui Supprime le filigrane. nous présentons quelques résultats expérimentaux obtenus par notre débruiteur.

Nous clôturons notre étude par une conclusion générale.

Les réseaux de neurones et l'apprentissage profond

1.1 Introduction

Un réseau de neurones artificiels[1] est un système dont la conception est à l'origine, inspirée du fonctionnement des neurones biologiques. Si nous comparons l'ordinateur et le cerveau, pratiquement, l'ordinateur est bien plus puissant. Il est largement supérieur au cerveau humain pour effectuer des calculs, mémoriser et classer des informations. Mais, pour des opérations cérébrales comme raisonner, analyser son environnement et communiquer, l'être humain dépasse très largement la machine.

Les expériences montrent que l'être humain peut reconnaître l'image d'une personne ou d'un objet en une fraction de seconde. Ce qui correspond au temps d'une communication entre les neurones du cerveau. Un ordinateur suivant l'architecture de Von Neuman ne peut rien effectuer dans un temps aussi court, sur cette tâche spécifique. C'est alors que rentrent dans la partie, les réseaux de neurones artificiels. Basés sur l'architecture d'un cerveau, ce paradigme informatique permet de concurrencer l'homme sur les tâches citées précédemment.

Les réseaux de neurones sont généralement optimisés par des méthodes d'apprentissage. Ils sont placés dans la famille des applications statistiques, qu'ils enrichissent avec un ensemble de paradigmes qui permettent de créer des classifications rapides (comme par exemple : les réseaux de Kohonen). En modélisation des circuits biologiques, ils permettent de tester quelques hypothèses fonctionnelles issues de la neurophysiologie, ou encore les conséquences de ces hypothèses pour les comparer au réel.

1.2 Description des réseaux de neurones

Il existe plusieurs types de réseaux suivant la façon dont sont connectés les neurones entre eux. En informatique, les neurones sont calculés via un programme informatique, mais ils peuvent être parfois réalisés sur un circuit électronique. Un réseau de neurones possède deux états, un état d'apprentissage et un état de fonctionnement optimal. Afin de former le réseau, il doit y avoir une période d'enseignement et de répétition, tout comme un enfant

qui développe son cerveau grâce à l'enseignement de ses parents.

Les réseaux de neurones apprennent par un processus de rétro propagation. Cela implique la comparaison de la sortie du réseau de neurones avec la sortie pour laquelle il a été conçu. Cette différence sert à modifier les poids des connexions entre les neurones du réseau et à le rendre plus précis. Une fois le réseau suffisamment calibré, il atteint un niveau d'autonomie exceptionnel où il n'est plus nécessaire de le superviser.

Prenons un exemple, où nous supposons que nous avons enseigné à un réseau de neurones à distinguer une voiture automobile d'un engin de travaux publics. Pour cela nous lui avons montré 15 voitures et 15 engins différents, en lui expliquant bien à quoi correspondait chaque image.

Il atteint alors un état d'autonomie et est capable de distinguer par lui-même si les nouvelles entrées correspondent plutôt à une voiture ou à un engin. Si on lui montre une variante, par exemple, une voiture plus grande il va être capable de catégoriser cet objet, en voiture et non en engin de travaux publics. Simplement sur la seule base de son expérience et sans aucune aide, tout comme un être humain. Au final, nous avons appris à un ordinateur à apprendre. . . Les réseaux de neurones sont utilisés dans la vie de tous les jours pour plusieurs problématiques complexes comme :

- L'estimation des trafics maritime, ferroviaire, routier ;
- Les prévisions météorologiques ;
- La classification d'espèces animales et végétales, par analyse de l'ADN ;
- Les banques, afin de vérifier les montants des transactions financières ;
- La poste pour trier le courrier en fonction du code postal ;
- Les opérations boursières et bien d'autres utilisations ;

La recherche sur les réseaux de neurones a gagné en popularité et a atteint le sommet au début des années 1990. Jusqu'en 2006, les scientifiques n'ont pas su former les réseaux de neurones à dépasser les approches plus traditionnelles, sauf pour quelques problèmes spécialisés. Ce qui a changé à partir de cette date, c'est la découverte de techniques d'apprentissage dans les réseaux de neurones profonds. Ces techniques sont maintenant connues comme apprentissage profond .

Elles ont été développées d'avantage et aujourd'hui les réseaux de neurones et l'apprentissage profond ont obtenus des performances exceptionnelles sur de nombreux problèmes importants. Ils sont déployés à grande échelle par des entreprises telles que Microsoft, Google , IBM et Facebook.

Dans l'approche conventionnelle de la programmation, nous disons à l'ordinateur ce qu'il faut faire, diviser un grand problème en de nombreuses petites tâches définies avec précision, que l'ordinateur peut facilement effectuer.

En revanche, dans un réseau de neurones, nous ne disons pas à l'ordinateur comment résoudre

notre problème. Au lieu, il apprend des données d'observation, trouvant sa propre solution au problème en question.

1.3 Les neurones biologiques et artificiels

Le cerveau est l'organe le plus complexe du corps humain. L'unité fondamentale du cerveau humain est le neurone. Un petit morceau de cerveau de la taille du grain de riz, contient plus de 10. 000 neurone.[2] A sa base, le neurone est optimisé pour recevoir des informations d'autres neurones, traiter ces informations de manière unique et envoyer son résultat à d'autres cellules.

Le neurone reçoit ses entrées le long d'une structure en forme d'antenne appelée dendrite. Après avoir été pondéré par la force de leurs connexions, les entrées sont additionnées dans le corps de la cellule. Cette somme est ensuite transformée en un nouveau signal qui se propage le long de l'axone de la cellule et envoyée à d'autres cellules.

Nous pouvons traduire cette compréhension fonctionnelle des neurones de notre cerveau en un modèle artificiel que nous pouvons représenter sur notre ordinateur. Dans le domaine des technologies de l'information, un réseau de neurones est un système logiciel et / ou matériel qui imite le fonctionnement des neurones biologiques. Les réseaux neuronaux,[3] aussi appelés réseaux de neurones artificiels (RNA ou ANN, en anglais).

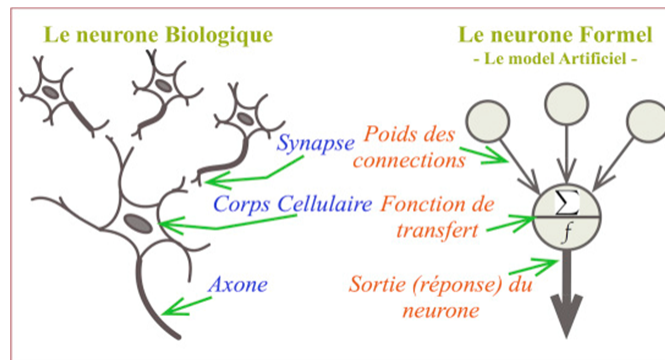


FIGURE 1.1 – Mise en correspondance neurone biologique /neurone artificiel

1.4 Le neurone formel

Le neurone formel[4][5] est un dispositif à plusieurs entrées et une sortie, qui prend pour modèle schématiquement le neurone biologique : il correspond respectivement aux dendrites et au point de départ de l'axone (cône d'émergence du neurone biologique).

Le neurone formel est une modélisation mathématique sous forme d'une fonction algébrique non linéaire, qui reprend les principes du fonctionnement du neurone biologique et dont la

valeur dépend des paramètres appelés coefficients ou poids. Il constitue l'unité minimale d'un réseau de neurones artificiels.

1.4.1 Le perceptron simple

Un réseau de neurones est une structure de réseau constituée d'un nombre de noeuds interconnectés par des liaisons directionnelles. Chaque noeud représente une unité de traitement et les liaisons représentent les relations causales entre les noeuds.

Le réseau le plus simple est celui de neurones monocouche appelé perceptron terme forgé par Franck Rosenblatt en 1958 [6] et l'assimilant à un modèle de rétine artificiel. Ce réseau est représenté par une couche de neurones d'entrée et de plusieurs neurones de sortie. Il constitue l'un des tous premiers algorithmes d'apprentissage supervisé. Il s'agit d'un neurone artificiel inspiré par la théorie cognitive de Friedrich Hayek et celle de Donald Hebb [7]. Dans sa version la plus simple, le perceptron n'a qu'une seule sortie.

La structure d'un perceptron est représentée dans la figure 1.2. Les valeurs d'entrée (x_1, x_2, \dots, x_n) et les poids associés aux entrées ($w_{1j}, w_{2j}, \dots, w_{nj}$) sont les variables de la fonction de combinaison qui détermine la somme pondérée du neurone (net_j). Cette somme est ensuite passée à la fonction d'activation qui détermine la valeur de sortie du neurone (o_j). Enfin, le neurone corrige ses poids afin de s'approcher de la valeur attendue (S).

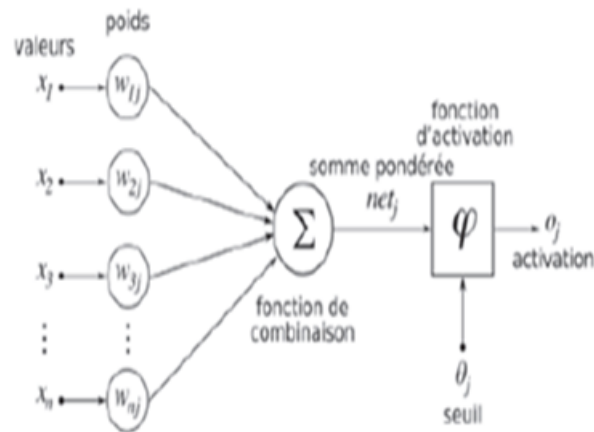


FIGURE 1.2 – Structure d'un perceptron simple

La somme pondérée des entrées par les poids (weights) w_i , associés aux entrées est appelée potentiel, noté S .

$$S = \sum_{i=0}^n w_{ij}x_j + b \quad (1.1)$$

S : est la somme pondérée du neurone net_j .

w_{ij} : est le poids de l'entrée i .

x_i : est la valeur de l'entrée i .

b : entrée spéciale (biais).

- **Biais(bias)** :c'est une valeur d'entrée qui est toujours égale à 1, ce qui permet une normalisation grâce à la valeur du poids correspondant ; [8]
- **La fonction d'activation** (ϕ) : transforme la somme pondérée pour obtenir la valeur de sortie. Le choix de la fonction de sortie dépend de l'application et du comportement souhaité ; [8]

1.4.2 Le perceptron multicouche (PMC)

Le perceptron multicouche [5] est un classifieur linéaire de type réseau neuronal formel organisv en plusieurs couches au sein desquelles une information circule de la couche d'entrée vers la couche de sortie uniquement. Les connexions sont toujours dirigées des entrées à travers le réseau vers les sorties, sans interconnexions entre les neurones de la même couche.

Les perceptrons multicouches sont capables de traiter des données qui ne sont pas linéairement séparables. Avec l'arrivée des algorithmes de rétro propagation, ils deviennent le type de réseaux de neurones le plus utilisé. Les MLP (Multi Layer Perceptron) sont généralement organisés en trois couches, la couche d'entrée, la couche intermédiaire (dite couche cachée) et la couche de sortie.

Les trois couches d'un MLP et le rôle de chacune :

1. **Couche d'entrée** :La première couche elle est complètement connectée vers l'avant et reçoit toutes les valeurs d'entrées du réseau à des fins d'apprentissage. Donc le nombre de neurones dans la couche d'entrées est égal au nombre de données en entrées (ou égale au nombre de dimensions du problème) ;
2. **Couches cachées** :couches qui succèdent à la couche d'entrée, constituées d'une ou de plusieurs couches intermédiaires. Elles relient la couche d'entrées à la couche de sorties. Le choix du nombre de couches et la taille (nombre de neurones) de chaque couche est empirique ;
3. **Couche de sortie** :La troisième couche ou la couche de résultat. Elle donne le résultat obtenu par le réseau. Donc, le nombre de neurones dans la couche de sorties égale au nombre de classes ;

La figure 1.3 illustre la structure d'un MLP présentant trois neurones en entrée, deux couches cachées et deux neurones en sortie.

Lorsque tous les neurones d'une couche sont connectés aux neurones de la couche suivante, on parle alors de couches complètement connectées.[3]

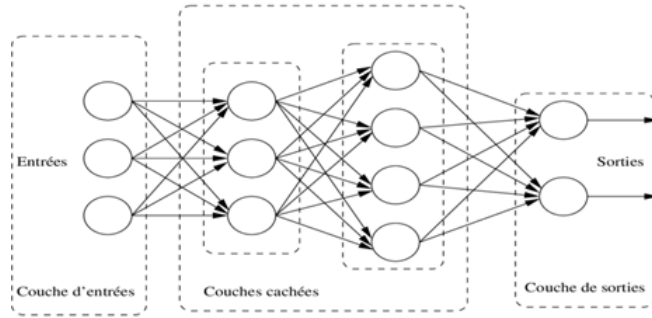


FIGURE 1.3 – Structure d’un perceptron multicouche. [9]

1.4.3 Fonctions d’activation

Dans le domaine des réseaux de neurones artificiels, la fonction d’activation est une fonction mathématique non linéaire appliquée à un signal en sortie d’un neurone artificiel. Le terme de fonction d’activation est synonyme de potentiel d’activation en biologie. C’est un seuil de stimulation qui une fois atteint entraîne une réponse du neurone. Les fonctions d’activation sont utilisées selon leurs caractéristiques d’étendue de non linéarité, de différentiabilité de convergence vers l’identité en 0 de continuité ou de monotonie.

Plusieurs fonctions d’activation ont été mises au point. Nous retenons quelques unes.

1. **Fonction Sigmoidé (Marche Douce ou Logistique) :**

Elle est souvent utilisée comme fonction d’activation pour les réseaux de neurones ,dans lesquels les valeurs de sortie désirées sont soit binaires soit dans un intervalle compris entre 0 et 1. Le but de cette fonction est de réduire la valeur d’entrée à un intervalle entre 0 et 1.

Si la valeur en entrée est un très grand nombre positif, la fonction convertira cette valeur en une probabilité de 1. A l’inverse, si la valeur en entrée est un très grand nombre négatif, la fonction convertira cette valeur en une probabilité de 0. L’équation ci-dessous représente la fonction Sigmoidé

$$\sigma(x) = \frac{1}{1 + e^x} \tag{1.2}$$

2. **Fonction Tanh (Tangente Hyperbolique) :**

Cette fonction ressemble à la fonction Sigmoidé[5]. La différence avec la fonction Sigmoidé est que la fonction Tanh produit un résultat compris entre -1 et 1. La fonction Tanh est en terme général préférable à la fonction Sigmoidé car elle est centrée sur zéro. Les grandes entrées négatives tendent vers -1 et les grandes entrées positives tendent vers 1.

3. **Fonction ReLU(Unité de Rectification Linéaire) :**

Elle est inspirée par des motivations biologiques.[10] Elle a été introduite pour la première fois en 2011 par Glorot et al.[10], Les avantages de ReLU incluent le calcul efficace,

l'activation clairsemée, etc.

Elle est favorisée pour les réseaux neuronaux profonds et plus vite[10][11],L'équation ci-dessous représente la fonction ReLU :

$$f(x) = \max(0, x) \quad (1.3)$$

Si l'entrée est négative la sortie est 0 .Et si elle est positive alors la sortie est x. Cette fonction est la plus utilisée.

4. **Fonction Softmax :**

Est également appelée fonction exponentielle normalisée. Elle est couramment utilisée pour représenter une distribution de probabilité pour les K sorties possibles. Ainsi, la fonction softmax a été fortement utilisée dans les tâches de classification. Elle est définie comme :

$$F(x)_i = \frac{e^x}{\sum_{k=1}^k e^x k} \quad (1.4)$$

1.4.4 La rétropropagation

La technique de rétropropagation (Backpropagation en anglais) est une méthode qui permet de calculer le gradient de l'erreur pour chacun des neurones du réseau de la dernière couche vers la première. On appelle souvent technique de rétropropagation du gradient, l'algorithme classique de correction des erreurs basé sur le calcul du gradient grâce à la rétropropagation, mais cela n'est pas toujours le cas. La correction des erreurs peut se faire selon d'autres méthodes. Le plus souvent, dans le cas des réseaux de neurones, la méthode de correction d'erreurs agit en corrigeant de manière significative les coefficients synaptiques qui contribuent à engendrer une erreur importante tout en pondérant également les neurones générant une erreur moins conséquente.

La procédure de rétropropagation peut se résumer dans les étapes suivantes :

- Initialiser tous les poids à de petites valeurs aléatoires dans l'intervalle [-0.5, 0.5].
- Normaliser les données d'entraînement.
- Permuter aléatoirement les données d'entraînement.
- Pour chaque donnée d'entraînement n :
 1. Calculer les sorties observées en propageant les entrées vers l'avant.
 2. Ajuster les poids en rétropropageant l'erreur observée. Répéter les étapes 1 et 2 jusqu'à un nombre maximum d'itérations ou jusqu'à ce que l'erreur soit en dessous d'un certain seuil.

1.5 L'Apprentissage

L'apprentissage[1] est souvent considéré comme la caractéristique principale de l'intelligence. De nombreuses études en intelligence artificielle (IA) ont donc été consacrées à l'apprentissage et aux méthodes qui permettent à une machine, d'étendre ses connaissances. Le

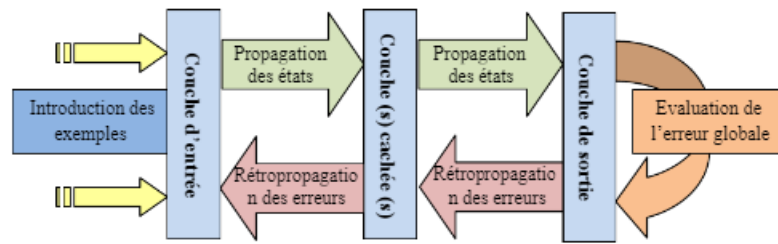


FIGURE 1.4 – Apprentissage des réseaux de neurone par l’algorithme de rétropropagation

principe de l’apprentissage devient alors le suivant : les informations captées ne servent pas uniquement à prendre une décision mais également à améliorer le réseau. Pour cette tâche, c’est l’apprentissage artificiel par induction qui est retenu par les scientifiques. Mais quelque soit le mode d’apprentissage nous distinguons habituellement deux situations : l’apprentissage supervisé et l’apprentissage non supervisé.

1.5.1 Apprentissage supervisé

Il est généralement destiné à reproduire un processus quelconque dont nous connaissons uniquement quelques variables et les résultats correspondants à obtenir.[12]. Dans ce cas, les données servant à l’apprentissage peuvent être représentées par le couple (entrée/sortie désirée, correspondante). Le processus est le suivant : nous modifions la force des connexions tant que le résultat obtenu correspondant à chaque entrée n’est pas assez proche de la sortie souhaitée.

Les réseaux à apprentissage supervisé [13] les plus utilisés sont les réseaux multicouches avec rétropropagation du gradient. La méthode la plus connue d’apprentissage supervisé est la classification.

1.5.2 Apprentissage non supervisé

Les réseaux à apprentissage non supervisé [13] sont utilisés lorsque les classes de données à obtenir ne sont pas connues à l’avance. Dans ce cas, les données servant à l’apprentissage sont uniquement des entrées. Comme c’est le cas pour les réseaux à apprentissage non supervisé, nous modifions la force des connexions entre les neurones du réseau. Or ici, il n’est pas possible de le faire en fonction d’une erreur rencontrée sur la réponse souhaitée. Dans ce type de réseaux, aucune réponse n’est connue a priori, les données étant uniquement des entrées. L’information utile se trouve donc uniquement dans les données servant de base pour l’apprentissage et en particulier dans les doublons de celles-ci

1.6 L'apprentissage profond

L'apprentissage profond [1] (deep Learning) est une classe de méthodes dont les principes sont connus depuis la fin des années 80, mais dont l'utilisation ne s'est vraiment généralisée que depuis environ 2012. L'apprentissage profond peut être vu comme un réseau multicouche avec de nombreuses couches intermédiaires.

Le Deep Learning est un ensemble de méthodes d'apprentissage qui tentent de modéliser des données avec architectures complexes, combinant différentes transformations non linéaires. Les bases élémentaires de l'apprentissage en profondeur sont les réseaux de neurones, qui sont combinés pour former les réseaux de neurones profonds.

Ces techniques ont permis des progrès significatifs dans les domaines du son et le traitement d'images, y compris la reconnaissance faciale, la reconnaissance vocale, la vision par ordinateur, le traitement automatique du langage, la classification des textes (par exemple reconnaissance du spam). Les applications potentielles sont très nombreuses. Un spectaculaire exemple est le programme Alpha Go[14], qui a appris à jouer le jeu de go par la méthode d'apprentissage en profondeur et a battu le champion du monde en 2016

Il existe plusieurs types d'architectures pour les réseaux de neurones :

- Les perceptrons multicouches, les plus anciens et les plus simples ;
- Les Réseaux de Neurones Convolutionnels (CNN), particulièrement adaptés au traitement d'images ;

L'architecture profonde est capable d'extraire automatiquement des caractéristiques de l'image.

Par exemple les premières couches extraient des contours que les couches suivantes forment en des concepts de plus en plus complexes et abstraits : des formes, des objets, des parties d'objets en objets, etc.

1.7 Les réseaux de neurones convolutifs

les réseaux neuronaux convolutifs désignés par l'acronyme CNN (Convolutional Neural Network)[15] renvoient à un terme mathématique : le produit de convolution. En termes simples, l'idée est qu'on applique un filtre à l'image d'entrée, ils comportent deux parties bien distinctes. En entrée, une image est fournie sous la forme d'une matrice de pixels. Elle a 2 dimensions pour une image en niveaux de gris. La couleur est représentée par une troisième dimension de profondeur 3 pour représenter les couleurs fondamentales (Rouge, Vert et Bleu). La première partie d'un CNN est la partie convolutive à proprement parler. Elle fonctionne comme un extracteur de caractéristiques des images. Une image est passée à travers une succession de filtres, ou noyaux de convolution, créant de nouvelles images appelées cartes de convolutions. Certains filtres intermédiaires réduisent la résolution de l'image, par une opération de maximum ou de moyenne(pooling). Au final, les cartes de convolutions sont mises à plat et concaténées en un vecteur de caractéristiques, appelé code CNN.

en sortie de la partie convolutive est ensuite branchées en entrée d'une deuxième partie, constituée de couches entièrement connectées (perceptron multicouche). Le rôle de cette partie est de combiner les caractéristiques du code CNN.

Les réseaux neuronaux convolutifs ont de larges applications dans la reconnaissance d'image et vidéo.

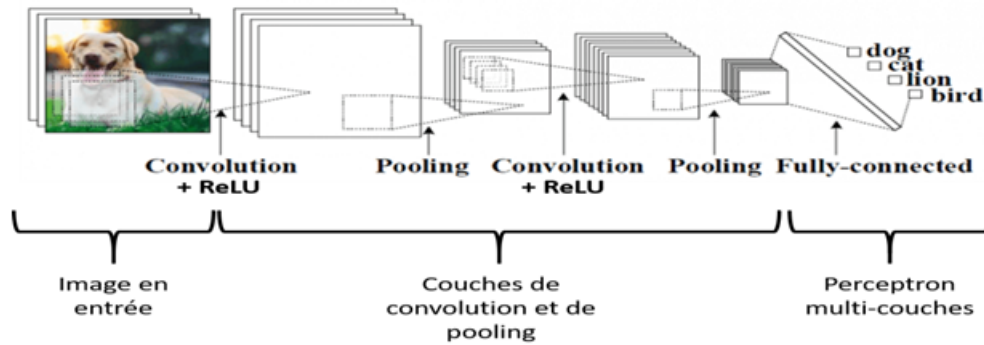


FIGURE 1.5 – Architecture standard d'un réseau de neurones convolutifs

1.7.1 Architecture des réseaux de neurones convolutifs

La couche de convolution(CONV)

La convolution est très classique en traitement d'images Elle consiste à déplacer un filtre (ou kernel en anglais) en le faisant glisser sur une image et à réaliser une convolution (un produit matriciel) de ce filtre avec l'image.

La couche de convolution est le bloc de construction de base d'un CNN. Trois paramètres permettent de dimensionner le volume de la couche de convolution : la profondeur, le pas et la marge.[15]

1. **La profondeur de la couche** :nombre de noyaux de convolution (ou nombre de neurones associés à un même champ récepteur).
2. **Le pas** :lorsque nous effectuons une convolution, nous choisissons ce qu'on appelle le pas. stride, il contrle le chevauchement des champs récepteurs. Plus le pas est petit, plus les champs récepteurs se chevauchent et plus le volume de sortie sera grand.
3. **La marge(à 0) ou zero padding** :parfois, il est commode de mettre des zéros à la frontière du volume d'entrée. La taille de ce zero-padding est le troisième hyper paramètre. Cette marge permet de contrôler la dimension spatiale du volume de sortie. En particulier, il est parfois souhaitable de conserver la même surface que celle du volume d'entrée.

Avec un padding de 0, aucun contour de 0 n'est rajouté. Avec un padding de 1, un contour de 0 est rajouté. Avec un padding de 2, deux contours de 0 sont rajoutés, et ainsi de suite.

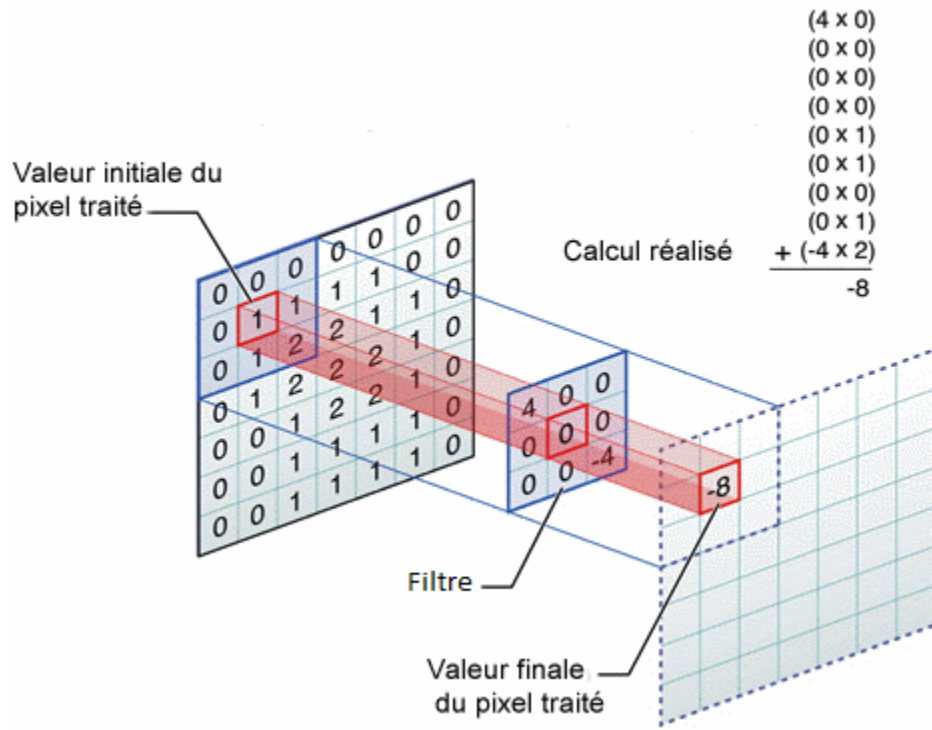


FIGURE 1.6 – Exemple de convolution avec un filtre 3 x 3

Avec un stride de 1, le filtre se déplace d'un pixel à la fois. Avec un stride de 2, de deux pixels à la fois.

Nous pouvons savoir la taille de l'image en sortie d'une convolution à partir de la taille de l'image d'entrée (W), la taille du filtre (F), le padding (P) et le stride (S) :

$$\text{Taille sortie convolution} = \left\{ \frac{W - F + 2P}{S} + 1 \right\} \quad (1.5)$$

Souvent, nous considérons un pas $S=1$. Nous calculons donc la marge de la manière suivante :

$$P = \frac{F - 1}{2} \quad (1.6)$$

si nous souhaitons un volume de sortie de même taille que le volume d'entrée, dans ce cas particulier la couche est dite : connectée localement.

Couche de pooling

La couche de pooling est une forme de sous-échantillonnage de l'image entrée. En extrayant les valeurs importantes des pixels, elle permet de réduire une image (réduit la taille spatiale) tout en conservant les caractéristiques pertinentes. La méthode la plus utilisée est le Max Pooling . Elle consiste à extraire les valeurs les plus grandes des pixels. Pour ce faire, nous avons un filtre qui se déplace sur la surface de notre image. à chaque position de filtre, nous extrayons la valeur la plus haute et nous ne retenons que celle là. Cela produit une

nouvelle image avec uniquement les valeurs remarquables de cette image.

Le Pooling est très important. En réduisant l'image, le nombre de données traitées diminue et donc le temps de calcul sera lui aussi réduit. Cela n'est pas négligeable. Il existe d'autres méthodes que le Max Pooling par exemple :

- **Average Pooling** : Moyenne de toutes les valeurs recouvertes par la tuile ;
- **Pooling stochastique** : Ne retient qu'une seule valeur, comme le Max Pooling, mais en se basant sur une méthode probabiliste ;

Cependant, en pratique le Max Pooling est la méthode qui donne le plus souvent les meilleurs résultats. Elle est donc généralement privilégiée.

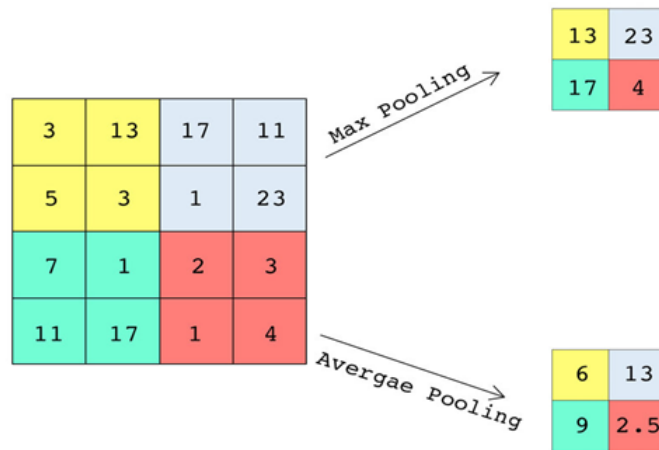


FIGURE 1.7 – Exemples de max et moyenne pooling sur une fenêtre 2x2 et pas 2

Couches de correction (RELU)

La fonction ReLU (abréviation de Unités Rectifié linéaires) désigne la fonction réelle non-linéaire définie par :

$$F(x) = \max(0, x) \quad (1.7)$$

La couche de correction ReLU remplace donc toutes les valeurs négatives reçues en entrées par des zéros. Elle joue le rôle de fonction d'activation. La fonctionnalité ReLU est illustrée à la figure ci-dessous.

Subsampling

C'est la fonction inverse de Max-Pooling, ou l'image sera reconstruite en répétant les lignes et les colonnes des données respectivement. Il a pour but d'obtenir la même résolution que l'image d'entrée.

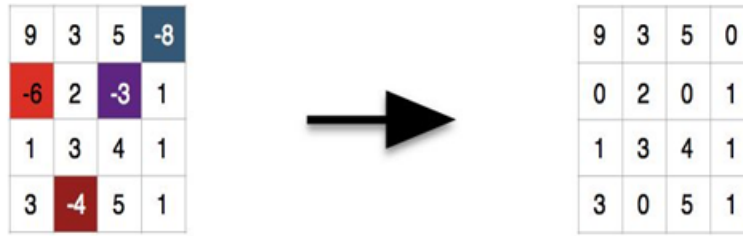


FIGURE 1.8 – Exemple de la fonction ReLu

Couche entièrement connectée (FC)

Après plusieurs couches de convolution et de max-pooling, le raisonnement de haut niveau dans le réseau neuronal se fait via des couches entièrement connectées. Les neurones dans une couche entièrement connectée ont des connexions vers toutes les sorties de la couche précédente. Leurs fonctions d'activations peuvent donc être calculées avec une multiplication matricielle suivie d'un décalage de polarisation.

Les réseaux de neurones convolutifs utilisent relativement peu de pré-traitement. Cela signifie que le réseau est capable de faire évoluer tout seul ses propres filtres (apprentissage sans supervision), ce qui n'est pas le cas d'autres algorithmes plus traditionnels.

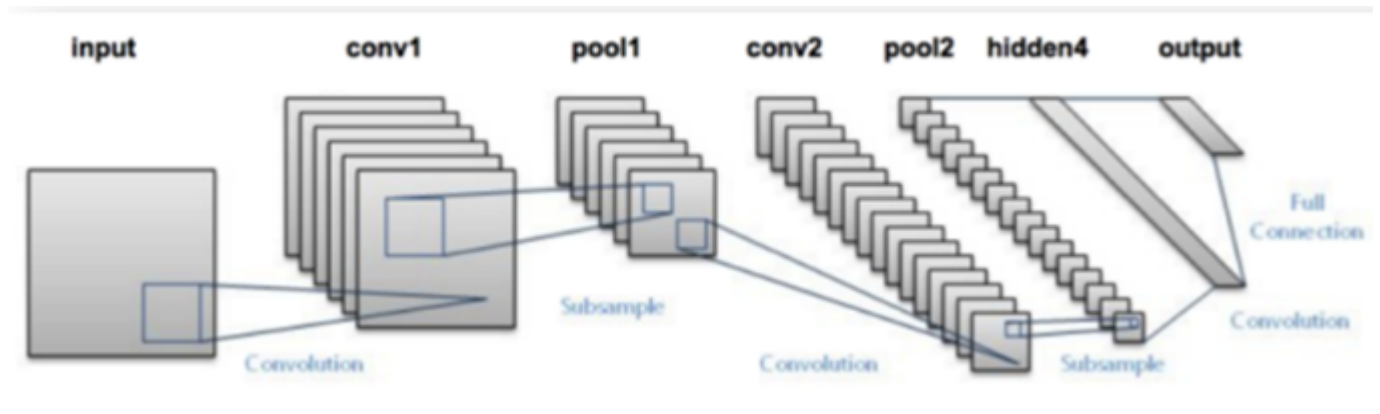


FIGURE 1.9 – Les différentes couches d'un CNN

1.8 Auto-encodeur (AE)

Les réseaux de neurones ne se limitent pas forcément à des problématiques d'apprentissage supervisé, même si cela reste leur utilisation principale. En effet, il existe aussi des réseaux de neurones qui permettent de faire de l'apprentissage non supervisé.

Les plus utilisés dans ce domaine sont probablement les Auto-Encodeurs. Ces réseaux n'ont pas pour objectif de prédire ce qui se trouve dans une image mais seulement de chercher à

trouver une représentation pertinente et compressée de l'image en tirant justement parti du pouvoir de représentation des réseaux de neurones.

Auto Encodeurs : sont des algorithmes d'apprentissage non supervisé à base des réseaux de neurones artificiels, qui permettent de construire une nouvelle représentation d'une image. Généralement, celle-ci est plus compacte, et présente moins de descripteurs, ce qui permet de réduire la dimensionnalité de l'image. L'architecture d'un auto-encodeur est constituée de deux parties : l'encodeur et le décodeur.

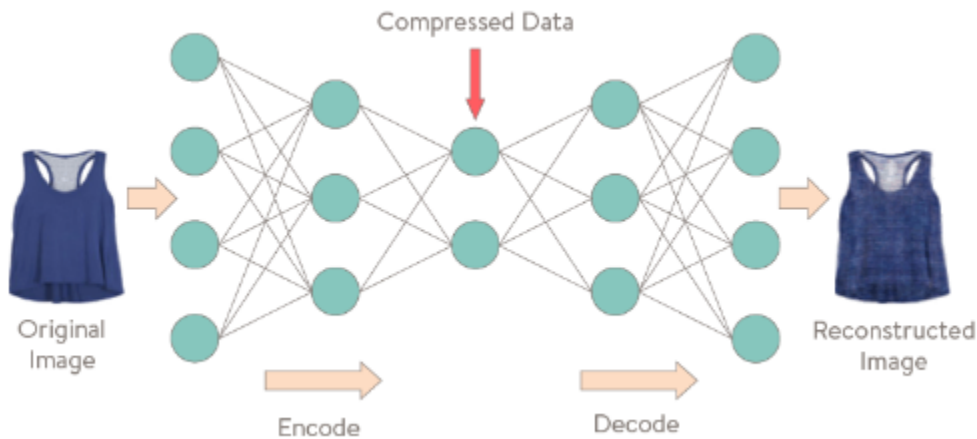


FIGURE 1.10 – Principe d'un auto-encodeur sur une image

1.8.1 Architecture des auto-encodeurs

Cette architecture peut être décomposée en deux parties :

L'encodeur : est constitué par un ensemble de couches de neurones, qui traitent les données afin de construire de nouvelles représentations dites encodées. À leur tour, les couches de neurones du décodeur, reçoivent ces représentations et les traitent afin d'essayer de reconstruire les données de départ. Les différences entre les données reconstruites et les données initiales permettent de mesurer l'erreur commise par l'auto-encodeur. L'entraînement consiste à modifier les paramètres de l'auto-encodeur afin de réduire l'erreur de reconstruction mesurée sur les différents exemples du jeu de données.

Le décodeur : C'est la dernière couche qui contient uniquement la reconstruction des données initiales, mais plutôt à la nouvelle représentation créée par l'encodeur.

L'architecture la plus simple d'un auto-encodeur est semblable à un perceptron multicouche. Cependant, en fonction des données traitées, nous pouvons utiliser différentes topologies de réseaux de neurones. Par exemple, des couches convolutives afin d'analyser des images ou des couches de neurones récurrentes pour traiter des séries temporelles ou des séquences.

1.8.2 Apprentissage des auto-encodeurs

L'idée est simple mais ingénieuse : on entraînera un réseau de neurones à une couche cachée h à prédire en sa sortie z en son entrée x , Une tâche que nous nommons reconstruction. Le critère sera donc la minimisation d'une erreur de reconstruction $L(x, z)$. En principe, la couche cachée devra donc contenir de l'information pertinente à la reconstruction. Par exemple, si la couche cachée contient moins d'unités qu'il y a d'entrées, alors logiquement pour bien reconstruire elle doit apprendre à résumer l'entrée, puisque l'information pour la reconstruction est entièrement contenue dans la couche cachée. Ainsi des caractéristiques pertinentes à la reconstruction devront être extraites.

Il faut noter que les paramètres pour passer de h à z , W_0 , ont une forme matricielle qui est la transposée de la forme des paramètres W . Si les poids sont liés (tied) alors il s'agit effectivement de la transposée et durant l'entraînement le contenu de la matrice sera contraint de rester le même pour les deux directions. Autrement, s'ils ne sont pas liés, ils sont libres de prendre des valeurs différentes.

Il est aussi possible d'effectuer la reconstruction en faisant passer l'information dans plusieurs couches cachées h_1, \dots, h_n . Nous obtenons ainsi un auto-encodeur profond. Comme ce type de réseau contient plusieurs couches, il sera en général préférable de pré-entraîner les couches comme expliqué dans la section sur les architectures profondes, et de les faire correspondre par paires : la première et la dernière, la seconde et l'avant dernière, par les correspondances entre les noms des paramètres de chacune des couches.

Par contre, dans l'auto-encodeur simple, si le nombre d'unités dans la couche cachée est plus grand ou égal au nombre d'entrées, le modèle peut alors apprendre de façon triviale la fonction identité. C'est facile en copiant une à une les valeurs de l'entrée dans la couche cachée, par exemple. Nous n'obtenons donc pas de caractéristiques extraites intéressantes. Pour contourner ce problème, différentes contraintes peuvent être imposées durant l'apprentissage. C'est ce qui est fait avec l'auto-encodeur débruiteur (voir ci dessous).

Pour des modèles effectuant de la reconstruction, Nous pouvons évaluer l'erreur de reconstruction, la différence entre l'entrée originale et l'entrée reconstruite. Cela permet de comparer les modèles entre eux. Pour le faire numériquement, nous pouvons utiliser Le SSIM, EQM. (Voir chapitre 3) pour guider l'entraînement (évaluation objectif).

L'apprentissage de l'auto-encodeur se fait par rétropropagation du gradient.

1.8.3 Auto-encodeurs débruiteur

débruiteur

Bruit :

Le bruit [22][23] est un signal qui dégrade la qualité d'une image (vidéo), il est provoqué par une

perturbation externe. Dans le cas de la transmission d'une image (vidéo) par voie électronique on peut s'attendre à des erreurs se produisant dans le signal de l'image. Ces erreurs peuvent dégrader la qualité de l'image (vidéo) reçues de différentes manières selon le type de canal de transmission (satellite, sans fil, câble réseau...).

Dans ce qui suit, nous allons présenter quatre types de bruit et leurs effets sur la qualité d'une image.

1. **Le bruit additif** : peut être défini de la façon suivante : Etant donnée une image non bruitée g et f la même image avec un bruit additif b , alors chaque pixel j est caractérisé par la relation :

$$f_j = g_j + b_j \quad (1.8)$$

b est une variable aléatoire de moyenne égale à 0.

type de bruit additif

- (a) Le bruit poivre et sel : est obtenu en ajoutant des pixels blancs et des pixels noirs (ou les deux) aléatoirement dans une image.



FIGURE 1.11 – Image avec le bruit poivre et sel

- (b) Le bruit Gaussienne : Est une forme idéalisée de bruit blanc, obtenu en ajoutant à chaque pixel une valeur aléatoire suivant une loi de probabilité Gaussienne.



FIGURE 1.12 – Image avec le bruit Gaussienne

2. **Le bruit multiplicatif** : peut être défini de la façon suivante : Etant donnée une image non bruitée g et f la même image avec un bruit multiplicatif b , alors chaque pixel j est caractérisé par la relation :

$$f_j = g_j * b_j \quad (1.9)$$

b est une variable aléatoire de moyenne égale à 0.

type de bruit multiplicatif

- (a) bruit de speckle : s'appelle également le bruit multiplicatif. il est un problème important dans quelques applications de radar et d'échographie.



FIGURE 1.13 – Le bruit de speckle (bruit multiplicatif)

3. **Le bruit convolutif** : peut être défini de la façon suivante : Etant donnée une image non bruitée g et f la même image avec un bruit additif b , alors chaque pixel j est caractérisé par la relation :

$$f_j = g_j * b_j \quad (1.10)$$

b est une variable aléatoire de moyenne égale à 1.

parmi ce type de bruit, nous trouvons le flow.

Un débruiteur : C'est un algorithme qui enlève le bruit dans l'image, vidéo, son, single.

Auto-encodeur débruiteur

Auto-encodeur débruiteur : est une combinaison d'un auto-encodeur en premier lieu et un débruiteur comme deuxième traitement. L'objectif de cette combinaison est l'amélioration de la convergence du réseau.

Il reçoit une donnée partiellement corrompue comme entrée et apprend à récupérer l'entrée originale débruitée. Cette technique a été introduite avec une approche spécifique d'une bonne représentation. Une bonne représentation est celle qui peut être obtenue de manière robuste à partir d'une entrée corrompue et qui sera utile pour récupérer l'entrée débruitée correspondante.

1.9 Conclusion

Dans ce chapitre nous avons abordé les notions de base d'un réseau neuronal, telles que, le perceptron et le perceptron multicouches. Ce dernier est efficace pour les problèmes logiques mais il est limité pour l'extraction des caractéristiques de l'image et de larges structures (le

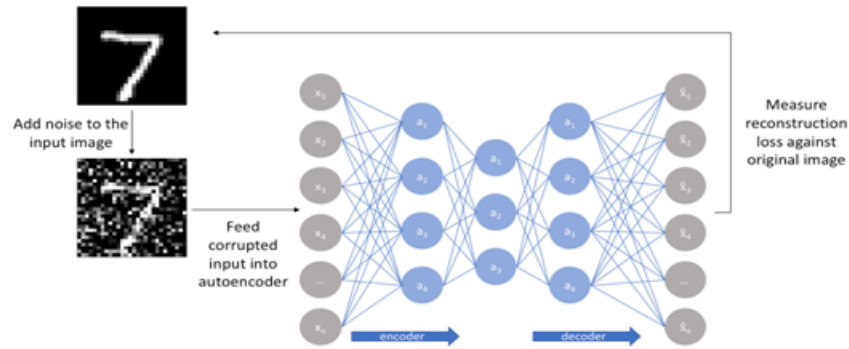


FIGURE 1.14 – Autoencodeur débruiteur

nombre des couches est grand). Nous avons également abordé les fonctions d'activation les plus utilisées puis nous avons expliqué la technique de rétropropagation. Nous avons présenté les notions importantes qui sont en relation avec l'apprentissage profond (définition, Architectures, etc.), ainsi qu'une vision générale sur l'apprentissage profond tel que, les réseaux de neurones convolutifs (CNN) qui est capable d'extraire les caractéristiques (les plus pertinentes) d'une image, par des séquences des couches convolutifs et de réduire les données non pertinentes (pooling). Nous avons aussi présenté un autoencodeur et un autoencodeur combiné avec un débruiteur. Cette combinaison peut être utilisée pour Accélérer l'apprentissage des neurones artificiels et minimiser le temps de travail.

Filigrane numérique

Introduction :

Un filigrane physique est une image ou un motif d'identification sur le papier qui apparaît sous la forme de diverses nuances de clarté / d'obscurité lorsqu'il est vu par lumière transmise (ou lorsqu'il est visualisé par la lumière réfléchie, sur un fond sombre), causée par des variations d'épaisseur ou de densité du papier. Des filigranes ont été utilisés sur des timbres-poste, des devises et d'autres documents gouvernementaux pour décourager la contrefaçon.

Les filigranes varient considérablement dans leur visibilité ; tandis que certains sont évidents lors d'une inspection occasionnelle, d'autres nécessitent une étude pour être repérés. Différents auxiliaires ont été développés, comme un fluide filigrane qui mouille le papier sans l'abîmer. Un filigrane est très utile dans l'examen du papier car il peut être utilisé pour dater, identifier les formats, les marques de fabrique et les emplacements, et déterminer la qualité d'une feuille de papier [16].

Un filigrane numérique est une sorte de marqueur incorporé secrètement dans un signal tolérant au bruit tel que des données audio, vidéo ou image. Il est généralement utilisé pour identifier la propriété du droit d'auteur d'un tel signal. Le filigrane est le processus consistant à cacher des informations numériques dans un signal porteur ; les informations cachées devraient, mais ne doivent pas nécessairement, contenir une relation avec le signal de porteuse. Les filigranes numériques peuvent être utilisés pour vérifier l'authenticité ou l'intégrité du signal porteur ou pour montrer l'identité de ses propriétaires. Il est largement utilisé pour localiser les violations de droits d'auteur et pour l'authentification des billets de banque. Comme les filigranes physiques traditionnels, les filigranes numériques ne sont souvent perceptibles que dans certaines conditions, par exemple après avoir utilisé un algorithme. Si un filigrane numérique déforme le signal porteur de manière à ce qu'il devienne facilement perceptible, il peut être considéré comme moins efficace en fonction de son objectif. Les filigranes traditionnels peuvent être appliqués aux supports visibles (comme les images ou la vidéo), tandis que dans le filigrane numérique, le signal peut être audio, images, vidéo, textes ou modèles 3D. Un signal peut porter plusieurs filigranes différents en même temps.

Contrairement aux métadonnées qui sont ajoutées au signal de porteuse, un filigrane numérique ne change pas la taille du signal de porteuse .

Les propriétés nécessaires d'un filigrane numérique dépendent du cas d'utilisation dans lequel il est appliqué. Pour marquer des fichiers multimédias avec des informations de copyright, un filigrane numérique doit être assez robuste contre les modifications qui peuvent être appliquées au signal porteur. Au lieu de cela, si l'intégrité doit être garantie, un filigrane fragile serait appliqué . La stéganographie et le filigrane numérique utilisent des techniques stéganographiques pour intégrer des données secrètement dans des signaux bruyants. Alors que la stéganographie vise l'imperceptibilité des sens humains, le filigrane numérique tente de contrôler la robustesse comme priorité absolue. Étant donné qu'une copie numérique des données est identique à l'original, le filigrane numérique est un outil de protection passive. Il marque simplement les données, mais ne les dégrade pas et ne contrôle pas l'accès aux données.

2.1 Historique

Les filigranes ont été introduits pour la première fois à Fabriano , en Italie , en 1282[17] sur papier ,mais Le terme filigrane numérique a été inventé par Andrew Tirkel et Charles Osborne en décembre 1992 [18]. La première intégration et extraction réussies d'un filigrane à spectre étalé stéganographique a été démontrée en 1993 par Andrew Tirkel, Charles Osborne et Gerard Rankin[18] .

Les filigranes continuent d'être utilisés aujourd'hui comme marques de fabricant et pour empêcher la falsification.

2.2 Aux origines du tatouage

La cryptographie, la stéganographie et le tatouage sont des techniques très proches les unes des autres puisqu'elles consistent à transmettre une information à caractère confidentielle. Elles répondent toutes les trois à des problèmes de sécurité(Robustesse ,Capacité ,Imperceptibilité). Cette section vise à établir les différences et les similitudes entre ces trois disciplines.

Le tatouage numérique consiste à insérer une marque qui a un lien avec le support numérique.Si la marque insérer dans un support numérique est différentes pour toutes les copies du support de base, on parle alors du **filigrane**

2.2.1 La cryptographie

Puisque le tatouage consiste à transmettre une information non accessible, la discipline est souvent rattachée aux questions de sécurité des données numériques, et donc naturelle-

ment à la discipline de la cryptographie.

La cryptographie[19] est une discipline très vieille, des techniques ont été mises en place dès le Vème siècle. Elle consiste à transformer un message pour qu'il devienne illisible. Seule la connaissance d'une clef et du moyen de cryptage peut permettre de décoder le message afin de le rendre lisible. Alors que pour le tatouage, la donnée tatouée est disponible, diffusée et exploitable, la donnée cryptée est elle inexploitable sans la connaissance des clés de déverrouillage de l'algorithme de cryptage. En fait les deux disciplines sont considérées comme complémentaires puisque d'un coté la cryptographie tend à renforcer le contrôle d'accès aux données, leur authenticité et leur intégrité, d'un autre coté le tatouage tend à lier le contenu des données avec des informations auxiliaires .

2.2.2 La stéganographie

Le terme stéganographie [19] vient du mot Grec steganos signifiant caché et de graphia signifiant écriture, littéralement on traduit par écriture cachée. Elle consiste à dissimuler un message dans un autre.

Ainsi, seule la personne connaissant le procédé de dissimulation peut lire le message caché. Contrairement à la cryptographie, la stéganographie est invisible. La différence entre la stéganographie et le tatouage, est que dans la stéganographie, l'existence d'un message caché doit rester secrète alors que pour le tatouage seul le message doit rester caché mais son existence (tant qu'on ne peut le détecter) peut être connue.

2.3 Types d'insertion

2.3.1 Méthodes additives

Lors de l'insertion, le signal représentant la marque est ajouté à certaines composantes du médium. Pour y parvenir, il s'agit d'adapter la marque au médium, afin que le signal qu'elle représente ne soit ni trop faible (risques de non détectabilité et problèmes de robustesse), ni trop fort (effacement du signal initial, et donc trop grande dégradation de celui-ci).

Le principe des méthodes dites additives consiste à ajouter la marque aux composantes du document en utilisant l'une des équations suivantes [20] :

$$I_w = I_0 + WI_w = I_0 * (1 + W)I = I_0 * e^w \quad (2.1)$$

En pratique, la marque W est ajoutée aux certains composantes caractéristiques de l'image I_0 . Cette opération peut se faire à partir de l'image elle-même ou à partir d'une transformation fréquentielle(les transformations fréquentielles(DCT, DFT),La transformée en ondelettes discrète(DWT))[21]

2.3.2 Méthodes substitutives

Dans les méthodes substitutives, la marque à insérer est substituée à des composantes de l'image originale, ce qui correspond principalement à deux comportements : le premier est un tatouage quantitatif appelé aussi tatouage substitutif avec dictionnaire, le second est un tatouage substitutif avec contraintes. Il consiste à imposer un ensemble de contraintes aux données marquées [22]. Plusieurs techniques substitutives ont été proposées dans la littérature.

La première méthode consiste à remplacer les bits les moins significatifs ou les bits de poids faibles des pixels d'une image par les bits de la marque. Cette méthode a été améliorée dans [23] pour les images couleurs. Une technique très utilisée dans le tatouage des images est appelée la quantification par la modulation d'index (QIM) proposée dans [24][25]. Le principe de cette technique consiste à quantifier l'image en utilisant un ensemble de quantificateurs indexés par la marque à insérer. Ainsi, chaque élément de la marque est attaché à un quantificateur différent, et le tatouage s'effectue par quantification de l'image avec le quantificateur correspondant à la marque.

2.4 Type de filigrane

2.4.1 Filigrane visible

Le filigrane visible est très simple .il altère le signal ou le fichier par exemple ajout d'une image pour en marquer une autre . Il est fréquent que les agences de photo ajoutent un filigrane visible en forme de copyright aux versions de pré-visualisation basse résolution de leurs photos.

Ceci afin d'éviter que ces versions ne se substituent aux versions hautes résolutions payantes. Le filigrane visible est un sujet à controverse. Il y a une branche de chercheurs qui disent que si le filigrane est visible, alors elle peut être facilement attaquée. Néanmoins, nous trouvons des applications qui demandent que le filigrane soit visible, c'est le cas du logo des sociétés dans les programmes télévisuels.



FIGURE 2.1 – Filigrane visible

Dans les techniques de filigrane visible :

- La marque insérée est facilement enlevée par un simple cropping.
- La visibilité de la marque insérée dégrade la qualité visuelle de l'image

2.4.2 Filigrane invisible :

le filigrane invisible est un concept beaucoup plus complexe. Le filigrane invisible modifie le signal d'une manière imperceptible par l'utilisateur final. Pour reprendre l'exemple de l'agence de photo, les photos haute résolution vendues par l'agence possèdent elles au contraire un filigrane invisible, qui ne dégrade donc pas le contenu visuel.



FIGURE 2.2 – Filigrane invisible

Dans un filigrane invisible :

- il n'est pas facile de faire la distinction entre l'image originale et l'image filigranée.
- il est difficile d'enlever ou détruire la marque insérée sans avoir une dégradation de la qualité visuelle de l'image filigranée de manière significative.

2.5 Critères de dissimulation d'information

Les applications de dissimulation d'information sont triées en fonction de trois critères :

2.5.1 Robustesse

le pouvoir de récupérer la marque insérée même si l'image filigranée a été manipulée par des attaques. Il est nécessaire de distinguer plusieurs types d'attaques selon qu'elles sont considérées comme étant bienveillantes ou malveillantes. Les attaques bienveillantes sont les manipulations effectuées de bonne foi par un utilisateur. On retrouve dans cette catégorie : la compression JPEG, certaines transformations géométriques, le filtrage spatial et fréquentiel, l'ajout de bruit, l'impression et la numérisation, la correction gamma et l'égalisation d'histogramme.

2.5.2 Capacité

représente la quantité d'information que l'on veut insérer dans une image. Cette quantité varie selon l'application. En général, quelques bits suffisent pour la protection du droit d'auteurs à l'aide d'un identifiant, mais pas pour insérer un logo de société. En revanche, il est nécessaire de cacher plusieurs bits d'information pour permettre l'authentification des images.

2.5.3 Imperceptibilité

:
le filigrane numérique va certainement introduire des distorsions. Cette contrainte exige que lesdites distorsions soient les plus faibles possibles afin que visuellement l'image filigranée reste fidèle à l'image originale. Pour ce faire, les caractéristiques du système visuel humain (SVH) peuvent être exploitées pour rendre la marque moins perceptible. La qualité de l'image tatouée par rapport à l'image originale peut être évaluée à l'aide d'outils mathématiques tels que le rapport, signal sur bruit de crête (PSNR), la similitude structurale (SSIM), etc. Le critère d'imperceptibilité est une propriété liée uniquement au filigrane numérique invisible.

2.6 Applications de filigrane numérique des images

Le filigrane numérique peut être utilisé pour une large gamme d'applications, telles que :

- Protection du droit d'auteur :
la protection des droits d'auteur a été une des premières applications du filigrane numérique. En cas de litige juridique, le propriétaire d'une image est en mesure d'apporter la preuve qu'il est le propriétaire même si celle-ci a subi des dégradations (attaques)[26]. Une telle application doit assurer une grande robustesse contre les attaques, éviter toute ambiguïté de la preuve et minimiser les distorsions liées à l'insertion de la marque.
- Authentification du contenu d'une image :
l'idée de base de cette application consiste à insérer une marque fragile dans une image qui serve à alerter l'utilisateur face à une éventuelle modification de l'image par une personne non autorisée et à localiser précisément les régions manipulées. Cette application est généralement utilisée dans le domaine juridique et médical.
- Contrôle du nombre de copies :
les données numériques peuvent être dupliquées sans subir de détérioration de la qualité. Dans ce contexte, si une personne détient en main un document numérique, si elle est malintentionnée, elle peut produire illégalement un nombre illimité de copies de ce document avec une qualité égale au document d'origine. Le filigrane numérique peut faire face à cette situation. Des informations relatives au nombre de copies autorisées sont encodées dans la marque. Ce principe a été utilisé dans les vidéos où la marque indique si la vidéo peut être copiée ou non.
- Autres applications :
il existe d'autres applications telles que l'indexation et le contrôle d'accès, etc.

Les filigranes sont habituellement utilisés pour empêcher des images et des photos d'être réutilisées sans la permission de leurs propriétaires. Ils s'avèrent parfois difficiles à supprimer.

2.7 La suppression de filigrane par des applications

Les filigrane sont été conçus dans le but de protéger les photos et les images contre les falsifications et les vols.

sont habituellement utilisés pour empêcher des images et des photos d'être réutilisées sans la permission de leurs propriétaires. Ils s'avèrent parfois difficiles à supprimer.

Il existe plusieurs applications pour supprimer un filigrane sur une image. Nous allons citer quel que application :

- **Gimp** : Le programme GIMP ou GNU(General Public License) Manipulation de l'image , un logiciel open-source gratuit qui peut être téléchargé à partir de gimp.org à beaucoup des mêmes caractéristiques que un programme professionnel, propriétaire de retouche d'image ,iffusé sous la licence GPLv3 comme un logiciel gratuit et libre. Il en existe des versions pour la plupart des systèmes d'exploitation dont GNU/Linux, macOS et Microsoft Windows.

si un filigrane est créé sur un couche dans une image, vous pouvez supprimer la couche de filigrane en utilisant GIMP. Toutefois, si le filigrane est une partie de la couche d'image elle-même, la seule façon de supprimer le filigrane est de modifier l'image.

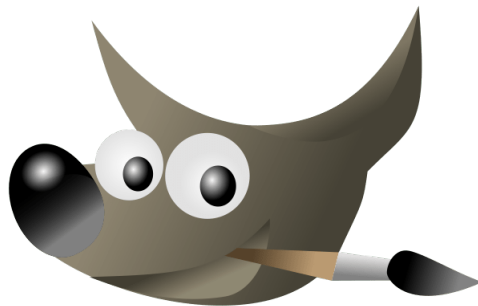


FIGURE 2.3 – Logo Gimp

- **Photoshop** : Les prémices de Photoshop remontent toutefois à 1987, lorsque Thomas Knoll, doctorant à l'Université du Michigan, inventa le logiciel Display pour ses propres besoins de recherches sur l'affichage sur ordinateurs. C'est dans un deuxième temps qu'il s'est associé à son frère John Knoll pour créer la première version de Photoshop, Photoshop 0.87, livrée en exclusivité avec des scanners. La société fut alors rachetée par Adobe en avril 1989, et l'entreprise lança dans le commerce Photoshop 1.0 en 1990 sur MacOS puis en 1992 sur Windows.



FIGURE 2.4 – Logo Photoshop

2.8 Outils d'évaluation

Pour évaluer la qualité d'une image, différentes approches ont été adoptées dans la littérature dont la majorité avait comme objectif l'évaluation d'une version dégradée de l'image originale plutôt que l'évaluation d'une version améliorée.

Nous pouvons classer ces approches en deux grandes classes :

l'approche subjective où des observateurs humains évaluent la qualité de l'image et l'approche objective où un algorithme appliqué à une image donnée lui attribue une note, dite note de qualité. Mais les performances de ces critères (ou algorithmes) ne sont pas toujours à la hauteur des objectifs visés.

2.8.1 Evaluation subjective

Dans la plupart des applications visuelles, l'œil humain est le terminal final qui va recevoir et traiter l'information. Pour cela l'évaluation subjective est considérée comme étant la meilleure manière de mesure de la qualité d'image puisqu'elle est basée sur les appréciations des êtres humains.

Pour augmenter la précision des notes de qualités, les recommandations UIT [27] doivent être respectées pendant la réalisation des tests. Cependant, dans les mêmes conditions, les observateurs peuvent juger très différemment la même image car leurs décisions dépendent de la qualité attendue par chaque observateur.

Protocoles

Il y a essentiellement trois grandes familles communes d'évaluation subjective définies par l'ITU [26] : Echelle continue de la qualité sur stimulus double (DSCQS), Echelle de dégradation sur stimulus double (DSIS) et Evaluation continue de la qualité sur stimulus unique (SSCQE).

Echelle continue de la qualité sur stimulus double (DSCQS)

La méthode à double stimulus est appelée Double Stimuli Continuous Quality Scale (DSCQS). Dans cette méthode, chaque essai se compose d'une paire de stimulus : un stimulus de la référence, et l'autre du test. Le stimulus de test est habituellement la référence après avoir subi un certain type de traitement. Les deux stimuli sont chacun présentés deux fois dans un test, d'une façon alternative, avec l'ordre du choix aléatoirement pour chaque épreuve. Des signaux audio sont employés pour indiquer quand une preuve commence, quand un nouveau stimulus commence, quand voter, et ce qui est le nombre de tests courants dans la séquence de test qui compose une session de tests.(voir figure 2.5).

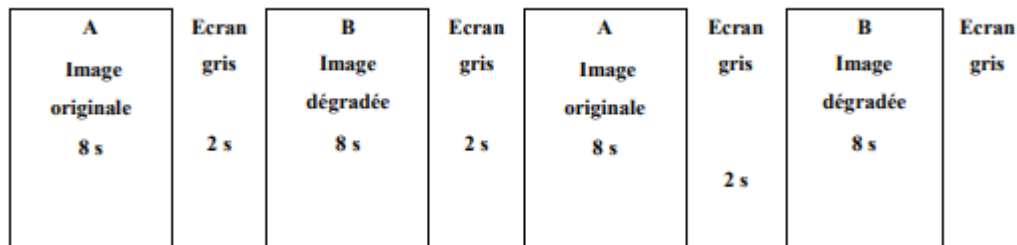


FIGURE 2.5 – Méthode à double stimulus

Echelle de dégradation sur stimulus double (DSIS)

Comme dans la méthode de DSCQS, chaque essai se compose d'une paire de stimulus : la référence et l'essai. Cependant, dans la méthode de Double Stimulus Impairment scale (DSIS), les deux stimulus sont toujours présentés dans le même ordre : la référence est toujours la première, suivi du test. Dans la méthode de DSIS, les observateurs comparent les deux stimuli dans un essai et évaluent la dégradation du stimulus d'essai en ce qui concerne la référence, en utilisant une échelle de dégradation de cinq niveaux. Ainsi, seulement une voix est faite pour chaque essai de DSIS. .

Evaluation continue de la qualité sur stimulus unique (SSCQE)

Au lieu de voir des paires de courtes séquences séparées, les observateurs observent un programme de la durée 20 – 30 minutes en général qui a été traité par le système du test ; la référence n'est pas montrée (Single Stimulus Continuous Quality Evaluation, SSCQE). En utilisant un glisseur, les observateurs évaluent continuellement la qualité instantanément perçue sur l'échelle de DSCQS du mauvais à l'excellent .(voir figure2.6).

Une échelle de notation utilisée pour évaluer la qualité d'une image est donnée dans le tableau 2 Ces échelles peuvent varier de 1 à 5 jusqu'à 100.

Analyse des résultats

Pour chaque paramètre de test, la moyenne et l'intervalle de confiance à 95% de la distribution statistique des notes doivent être calculés.

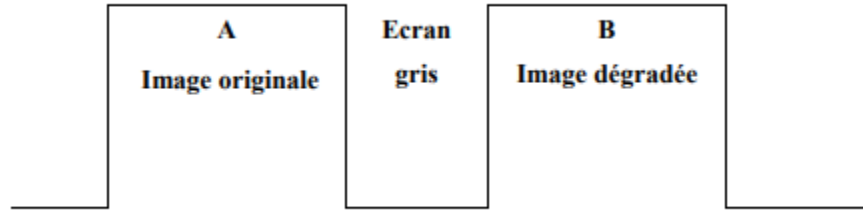


FIGURE 2.6 – Méthode à simple stimulus

Excellente
Bonne
Assez Bonne
Médiocre
Mauvaise

TABLE 2.1 – Echelle d' évaluation de qualité d'image

MOS : Score d'Opinion Moyen Lorsque les tests se terminent, on cherche à trouver la moyenne de la qualité. Une note est attribuée à chaque image en combinant les notes de tous les observateurs. Cette note appelée Mean Opinion Score est donnée par la formule suivante :

$$MOS(i) = \frac{1}{N} \sum_{j=1}^N Note_i(j) \quad (2.2)$$

Où N est le nombre d'observateurs $Note_i(j)$ est la note de l'observateur j pour la dégradation i de l'image.

Intervalle de confiance

Afin de réduire l'impact d'éventuelles erreurs, un intervalle de confiance est souvent associé à chaque note MOS. Il est généralement fixé à 95% et est donné par la relation suivante :

$$[Mos(i) - e_j, Mos(i) + e_j] \quad (2.3)$$

$O\tilde{A}^1$:

$$e_j = 1.95 \frac{\sigma_j}{\sqrt{N}}$$

Et l'écart-type pour chaque présentation σ_j , est donné par :

$$\sigma_j = \sqrt{\frac{\sum_{j=1}^N (Note_i(j) - MOS(i))^2}{N - 1}} \quad (2.4)$$

Une analyse plus approfondie permet de détecter et de rejeter d'éventuelles erreurs.

2.8.2 Evaluation objective

Les méthodes d'évaluation objective sont des méthodes calculatoires basées sur la mise en place d'un algorithme qui calcule la distance entre deux images. Cet algorithme produit des valeurs numériques, exprimant la qualité visuelle, appelées métriques. Les valeurs de ces métriques doivent refléter les notes subjectives données par les observateurs humains lors de test d'évaluation. L'intérêt majeur de ces métriques de qualité réside dans la possibilité de surveiller de manière automatique la qualité visuelle en temps réel.[29][27][30]

Les métriques d'évaluation objective ont plusieurs domaines d'application

- utilisés pour surveiller la qualité de l'image dans les systèmes de contrôle de la qualité.
- utilisés pour évaluer les algorithmes de traitement d'image.
- utilisés pour optimiser les systèmes de transmission et de traitement d'image.

2.8.3 Catégories des métriques objectives

les métriques objectives d'évaluation de la qualité des images peuvent être classées en utilisant un certain nombre de critères tels que le type de domaine (pixel ou transformé), le type de distorsion prédit (bruit, bloc), et le type d'information nécessaire pour évaluer la qualité (image originale, image dégradée). Les mesures de qualité objective proposées dans la littérature peuvent être classées en 3 catégories :

- Métriques à référence complète :
- Métriques sans référence .
- Métriques à référence réduite.

Métriques à référence complète

Ces métriques réalisent une comparaison directe entre l'image jugée et l'image de référence. Cette approche nécessite la disponibilité d'une image ou séquence de référence, ce qui représente une restriction importante à l'usage d'une telle mesure.

Approches basées EQM L'erreur quadratique moyenne est une métrique très populaire et très utilisée dans le domaine d'évaluation image. Grâce à sa simplicité, cette métrique peut facilement être intégrée dans un processus d'optimisation. Plusieurs métriques ont été proposées en fonction de cette erreur quadratique : le rapport signal à bruit (SNR : Signal Noise Ratio) et le rapport signal à bruit crête (PSNR : Peak Signal Noise Ratio).

$$EQM = \frac{1}{M.N} \sum_{i=1}^M \sum_{j=1}^N (I(i, j) - I_d(i, j))^2 \quad (2.5)$$

Avec $I(i, j)$ et $I_d(i, j)$ le pixel de coordonnées (i, j) de l'image originale et de sa version dégradée, N et M représentent respectivement le nombre de lignes et de colonnes de l'image.

Rapport signal sur bruit(SNR) :

$$SNR = 10 \log \left(\frac{\sum_{i=1}^M \sum_{j=1}^N (I(i, j))^2}{EQM} \right) \quad (2.6)$$

PSNR :

$$PSNR = 10 \log \left(\frac{\text{Valeur Crête du signal original}}{EQM} \right) \quad (2.7)$$

Où la valeur crête du signal originale est fixée à 255 pour les images codées sur 8 bits par pixels.

Dans tous les cas, il n'est pas très aisé de pouvoir interpréter les résultats obtenus.

Approches Structurelles(SSIM) Pour pallier la faiblesse des mesures type EQM, basées sur le pixel en général, certains auteurs ont orienté leurs recherches vers les métriques de qualité basées sur une analyse des structures locales du signal image. Parmi les métriques les plus populaires et les plus utilisées, on trouve celles proposées par Wang et al[31], appelées SSIM.

Indice de similarité structurelle (SSIM)

SSIM :est une mesure de similarité entre deux images numériques. Elle a été développée pour mesurer la qualité visuelle d'une image déformée, par rapport à la l'image originale. L'idée de *SSIM* est de mesurer la similarité de structure entre les deux image, plutôt qu'une différence pixel à pixel comme le fait par exemple le *PSNR*. L'hypothèse sous-jacente est que l'oeil humain est plus sensible aux changements dans la structure de l'image.

La métrique SSIM est calculée sur plusieurs fenêtres d'une image. On dénote x et y l'image originale et l'image déformée respectivement.

La similarité compare la luminance, le contraste et structure entre chaque couple de fenêtres.

La luminance est estimée par la mesure de l'intensité moyenne de chaque fenêtre :

$$u_x = \frac{1}{N} \sum_1^N x_i \quad (2.8)$$

N : le nombre de pixels de chaque fenêtre.

x_i : l'intensité d'un pixel.

Le contraste pour chaque fenêtre est mesuré par :

$$\sigma_x = \left(\frac{1}{N-1} \sum_1^N (x_i - u_x)^2 \right)^{\frac{1}{2}} \quad (2.9)$$

La similarité est déterminée par la perte de corrélation entre les deux fenêtres :

$$\sigma_{xy} = \frac{1}{N-1} \sum_1^N (x_i - u_x)(y_i - u_y) \quad (2.10)$$

La fonction de comparaison de luminance notée par $l(x, y)$ est une fonction de x et y . Le contraste entre deux fenêtres noté par $c(x, y)$, comparent les variances de x et y .

La troisième fonction compare les structures des deux fenêtres données par la fonction $s(x, y)$ en fonction de deux fenêtres normalisés.

Finalement la fonction mesurant la similarité, est une fonction de $l(x, y)$, $c(x, y)$ et $s(x, y)$. Elle est de la forme :

$$s(x, y) = f(l(x, y), c(x, y), s(x, y)) \quad (2.11)$$

Wang et Al ont proposé dans [30] fonction de comparaison de la luminance de la forme :

$$I(x, y) = \frac{2u_x u_y + (K_1 L)^2}{u_x^2 u_y^2 + (K_1 L)^2} \quad (2.12)$$

K_1 est une constante de très faible valeur. La constante $(k_1 L)^2$ évite l'instabilité de la fonction de comparaison lorsque $u_x^2 u_y^2$ sont très proches de zéro. Dans le cas des images à niveau de gris, la constante est $L = 255$ (dynamique du capteur).

On remarque que l'équation (2.12) est conforme à la loi de Weber [31], largement utilisée pour la modélisation de l'adaptation à la lumière dans le SVH (système visuel humain). Selon la loi de Weber [32], la variation de luminance est proportionnelle à la luminance de fond. En d'autres termes, le SVH est sensible aux variations relatives de luminance entre les deux signaux ou encore entre les deux images.

La fonction de comparaison de contraste prend une forme similaire :

$$c(x, y) = \frac{2\sigma_x \sigma_y + (K_2 L)^2}{\sigma_x^2 \sigma_y^2 + (K_2 L)^2} \quad (2.13)$$

où K_2 est une constante de très faible valeur.

La comparaison de structure est réalisée par la corrélation entre les deux vecteurs après soustraction de la luminance et normalisation par la variance. Notons que la corrélation entre les deux vecteurs est une mesure simple et effective de la similarité structurelle. Alors la fonction de comparaison structurelle est donnée par :

$$s(x, y) = \frac{2\sigma_{xy} + (K_2 L)^2}{2\sigma_x \sigma_y + (K_2 L)^2} \quad (2.14)$$

Finalement, la mesure de similarité résultante des trois comparaisons est donnée par une expression simplifiée de l'index de similarité structurelle entre x et y :

$$SSIM(x, y) = l(x, y) \cdot c(x, y) \cdot s(x, y) \quad (2.15)$$

où encore sous une forme simplifiée :

$$SSIM(x,y) = \frac{2u_x * u_y + (K_1L)^2}{u_x^2u_y^2 + (K_1L)^2} * \frac{2\sigma_u\sigma_y + (K_2L)^2}{\sigma_x^2\sigma_y^2 + (K_2L)^2} \quad (2.16)$$

L'indice de similarité est exploité pour un choix approprié des constantes K_1 et K_2 pour généraliser l'évaluation à toute l'image.

Complexe Wavelet SSIM (CW-SSIM) est une variante de la métrique SSIM, proposé dans [32] pour mesurer la qualité des images qui ont une distorsion géométrique telle que la rotation, le redimensionnement. Comme le SSIM, la valeur maximale de CW-SSIM est 1 qui indique que les deux images sont parfaitement identiques. L'équation de CW-SSIM est décrite comme suit :

$$CW - SSIM(C_x, X_y) = \frac{2 \sum_{i=1}^N |C_{x,i}| |C_{y,i}| + K}{\sum_{i=1}^N |C_{x,i}|^2 |C_{y,i}|^2 + K} * \frac{2 |\sum_{i=1}^N C_{x,i} * C_{y,i}^*| + K}{2 \sum_{i=1}^N |C_{x,i} * C_{y,i}^*| + K} \quad (2.17)$$

Où C_x est la transformation en ondelettes complexe du signal x et C_y est la transformation en ondelettes complexe du signal y . De plus, K est un petit nombre positif utilisé aux fins de la stabilité de la fonction.

Le CW-SSIM Généralement utiliser quand une image à la même qualité que l'image original.

Similitude structurale à base de gradient Un algorithme amélioré de SSIM a été développé (la similitude structurale à base de Gradient(GSSIM))[32] , qui compare l'information de contour entre les blocs de l'image déformée et l'originale, et remplace la comparaison de contraste $c(x,y)$ et la comparaison de structure $s(x,y)$ par la comparaison de contraste à base gradient $c_g(x,y)$ et la comparaison de la structure $s_g(x,y)$ Respectivement. Il y a un certain nombre de manières d'obtenir l'information de contour, telle que l'algorithme simple de détection de contour.

Deux cartes de gradient x' et y' représentant respectivement l'image originale et déformée sont générées en utilisant l'opérateur de Sobel. En plus les blocs des images X' et Y' sont donnés par x' et y' .La comparaison de contraste à base de gradient $c_g(x,y)$ et la comparaison de structure $s_g(x,y)$ sont décrites par :

$$c_g(x,y) = \frac{2\sigma_{x'}\sigma_{y'} + C_2}{\sigma_{x'}^2 + \sigma_{y'}^2 + C_2} \quad (2.18)$$

$$s_g = \frac{\sigma_{x'y'} + C_3}{\sigma_{x'}\sigma_{y'} + C_3} \quad (2.19)$$

Où σ'_x et σ'_y sont l'écart type des blocs x' et y' respectivement, $\sigma_{x'y'}$ est la covariance des blocs x' et y' et $C2, C3$ sont des petites constantes évitant la dénomination de la valeur zéro. La similitude structurale à base de Gradient (GSSIM) est définie comme suit :

$$GSSIM = [I(x, y)]^\alpha \cdot [c_g(x, y)]^\beta \cdot [s_g(x, y)]^\gamma \quad (2.20)$$

La similitude globale de structure d'image est calculée comme la moyenne de toutes les sous-images (blocs) de GSSIM, les coefficients de pondérations (α , β et γ) sont fixés à la valeur 1.

Métriques sans référence

Cette classe de méthodes de qualité objective ne nécessite pas d'accéder à la vidéo d'origine mais recherche des artefacts par rapport au domaine de pixel d'une vidéo, utilise les informations incorporées dans le train de bits du format vidéo associé ou effectue une évaluation de la qualité sous forme hybride. D'approches basées sur les pixels et sur le flux binaire.

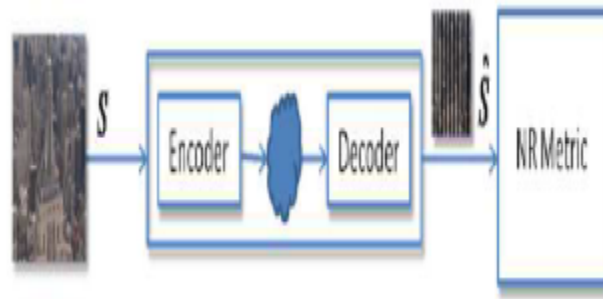


FIGURE 2.7 – Schéma de métrique sans référence

basées sur les statistiques de scènes naturelles (NSS : natural scene statistics)

L'idée principale de ces approches est que les images de scènes naturelles présentent des régularités statistiques affectés par la présence de distorsions, et ces changements statistiques dans la distorsion les images peuvent être mesurés pour évaluer la qualité de l'image. Parmi ces méthodes :

BIQI (Blind Image Quality Index) :

Cette approche a été présentée par Moorthy et Bovik[34], il estime la qualité à partir d'une transformée en ondelettes utilisant la base d'ondelettes de Daubechies[35]. La transformation est effectuée sur trois échelles et trois orientations. Le coefficient de sous-bande de la transformation est paramétré à l'aide d'une distribution généralisée gaussienne GGD. Il y a trois paramètres de la distribution : moyenne, variance et paramètre de forme. Mais deux paramètres seulement sont utilisés qui sont la variance et le paramètre de forme parce

que la moyenne dans la base des ondelettes est égale à zéro. Par conséquent, trois échelles, trois orientations et deux paramètres génèrent 18 caractéristiques pour l'image (3 x 3 x 2). Après avoir extrait les caractéristiques, l'image testée est classée par le classificateur SVM multiclass selon l'une des cinq classes représentant cinq distorsion différentes (jpeg, jpeg2000, bruit blanc, erreur de lecture rapide et flou). Ce classificateur ne donne pas seulement une classification absolue, mais aussi en donnant le montant de chaque distorsion présente dans l'image.

DIVIINE (Distortion Identification-based Image Verity and INtegrity Evaluation) :

Dans[35] , Moorthy et Bovik ont présenté l'algorithme DIIVINE. Les caractéristiques ici sont extraites en décomposant l'image déformée d'abord en utilisant une décomposition en ondelettes utilisant une pyramide orientable sur deux échelles et six orientations (0, 30, 60, 90, 90, 120, 150). Ils ont constaté qu'un degré d'orientation accru améliore la performance. Les coefficients de sous-bande à travers ces orientations et échelles sont ensuite utilisés pour extraire un ensemble de caractéristiques statistiques. Au total , 88 fonctionnalités sont utilisées dans cette méthode. Ensuite, la qualité est estimée en utilisant le même cadre en deux étapes dans [34].

Paramètres d'analyse comparative : Les différents algorithmes d'évaluation de qualité sans référence fournissent différents scores de qualité. Donc, pour comparer la performance de ces algorithmes il faut avoir un système de comparaison, les paramètre utilisé pour faire la comparaison sont cités ci-dessous :

Pearson coefficient de corrélation (PCC) :

Il est utilisé pour mesurer la dépendance entre les variables. Sa valeur est entre $(-1, +1)$ où la valeur proche de $+1$ indique que les deux variables ont une corrélation positive et la valeur proche de -1 indique que les deux variables ont une corrélation négative. Une très faible où un zéro indique que les deux variables ne sont pas corrélées. L'équation de PCC entre X, Y est décrite comme suit :

$$p = \frac{\text{covariance}(x, y)}{\sigma_x \sigma_y} \tag{2.21}$$

Les deux variables X, Y sont le résultat de d'algorithmes et la qualité réelle obtenue par la base de données.

Spearman Corrélation Coefficient (SCC) :

Fournit la relation entre deux variables classées. Sa valeur est entre $(-1, +1)$ avec la même interprétation que celle de PCC. Le SCC est calculé comme suit :

$$p = 1 - \frac{6 * \sum d^2}{n(n^2 - 1)} \tag{2.22}$$

Où (d) est la différence dans les rangs de deux variables X et Y, $d_i = x_i - y_i$. Et n est le nombre de paires de valeurs.

Métriques à référence réduite (RR)

Parmi les méthodes d'évaluation de qualité d'image avec référence réduite citons : Dans l'article[35], le score de qualité est estimé en mesurant les changements d'entropies pondérées des images de référence et déformées dans le domaine des ondelettes. Les coefficients d'ondelettes de l'image naturelle et déformés sont modélisés sous forme de distributions GSM. La métrique de qualité est obtenue en tant que distance entre l'approximation de référence et l'image naturelle d'une image déformée. L'algorithme proposé fonctionne beaucoup mieux que l'algorithme FR MSE.[37]

Dans l'article[38], un nouvel algorithme RR IQA basé sur des statistiques d'image est proposé. Les statistiques sur les images sont modélisées dans le domaine des pixels, basé sur la distribution en dégradé. Les caractéristiques de qualité RR sont extraites sur la base d'un modèle statistique. La qualité est mesurée en déterminant la similarité entre l'image de référence et l'image déformée. La méthode proposée est aussi performante que le PSNR FR largement connu[38]

Dans cet article[38], une tentative d'approximation de la métrique RR SSIM à partir de FR SSIM a été effectuée par transformation de normalisation divisible DNT de l'image naturelle. Outre l'évaluation de la qualité, le concept de réparation d'image a été introduit en comparant les propriétés statistiques de sous-bande de l'image déformée à celles de l'image de référence.

Les caractéristiques sont extraites de l'image de référence en appliquant une transformation linéaire à plusieurs échelles (transformée en ondelettes), puis la représentation de la DNT est calculée en divisant chaque coefficient d'ondelette par une mesure d'énergie locale basée sur les coefficients voisins. Pour un résumé efficace des propriétés statistiques de l'image de référence, une moyenne nulle est ajustée sur les coefficients DNT. Pour l'estimation de RR SSIM, l'effet des distorsions sur les propriétés statistiques doit être cohérent avec le FR SSIM correspondant. La méthode proposée est comparée à FR (PSNR et SSIM) et à RR (ondelettes marginales et DNT marginales) et la mesure proposée est très compétitive dans la plupart des cas.

2.9 Conclusion

Ce chapitre a présenté un état de l'art sur le filigrane des images numériques. Nous sommes intéressés aux types d'insertion du filigrane numérique. Nous avons abordé les Types de filigrane et présenté es des application de suppression d'un filigrane. nous avons aussi présente un état de l'art sur les méthodes d'évaluation de la qualité d'image qui sont divisées en deux familles : méthodes subjectives et méthodes objectives.

Les méthodes subjectives consiste à réaliser des séances de test en respectant des contraintes

précises, ces méthodes sont coûteuses et leurs résultats dépendent fortement des conditions physiques et de l'état émotionnel des observateurs.

Les méthodes objectives sont des méthodes calculatoires basées sur la mise en place d'un algorithme qui calcule la distance entre deux images. Le défaut majeur de ce type d'évaluation c'est que la majorité de ses critères sont calculées par rapport à l'image originale entre deux images. Le défaut majeur de ce type d'évaluation c'est que la majorité de ses critères sont calculées par rapport à l'image originale.

Conception et implémentation

3.1 Introduction :

Dans ce chapitre, nous allons procéder à la conception de notre application. Nous allons présenter aussi la mise en oeuvre de notre application en utilisant le langage Python. En commençant tout d'abord par une présentation du langage de programmation choisi. Ensuite nous décrivons les captures d'écran de l'exécution de notre application.

3.2 Configuration matérielle

3.2.1 Configuration matérielle distante (Google Colab)

Google Collaboratory ou Colab [38] : un outil Google simple et gratuit pour nous initier aux réseaux profonds ou collaborer avec nos collègues sur des projets en science des données. Colab permet d'améliorer nos compétences de codage en langage de programmation Python, de développer des applications en réseaux profonds, en utilisant des bibliothèques populaires telles que Keras, TensorFlow sans installation, ainsi que l'utilisation d'un environnement de développement (Jupyter Notebook) qui ne nécessite aucune configuration. Cependant, chaque 12 heures, la machine virtuelle mise à disposition par Google est réinitialisée, nécessitant un mécanisme de sauvegarde des données en cours. De plus, les documents Colab (Jupyter Notebook) sont enregistrés directement à votre compte Google Drive.

3.3 Environnement de développement logiciel

3.3.1 Python

Python [39] est un langage de programmation de haut niveau interprété et orienté objet. Il est très sollicité par une large communauté de développeurs et de programmeurs. Python est un langage simple et facile à apprendre. Les bibliothèques de python sont disponibles pour la majorité des plateformes et peuvent être redistribuées gratuitement.

3.3.2 Tensorflow

TensorFlow [40] est une bibliothèque de logiciels open source publiée en 2015, par Google pour faciliter la conception, la construction et la formation de modèles d'apprentissage en profondeur pour les développeurs. TensorFlow était prévu à l'origine comme une bibliothèque interne que les développeurs de Google utiliseraient pour construire des modèles en interne. A un niveau élevé, TensorFlow est une bibliothèque Python qui permet aux utilisateurs d'exprimer un calcul arbitraire sous forme de graphique de flux de données. Les noeuds de ce graphique représentent des opérations mathématiques, tandis que les arêtes représentent les données communiquées d'un noeud à un autre. Les données de TensorFlow sont représentées sous la forme de tenseurs, qui sont des tableaux multidimensionnels. Bien que ce cadre pour la réflexion sur le calcul soit précieux dans de nombreux domaines différents, TensorFlow est principalement utilisé pour l'apprentissage profond dans la pratique et la recherche.

3.3.3 Keras

Keras [42] est un Framework de plus haut niveau, il est écrit et entretenu par Francis Chollet et un autre membre de l'équipe Google Brain pour l'utiliser avec python, il aide à créer un modèle avec paramètre prédéfini (fonction de coût, optimiseur, fonctions d'activation ...).

3.4 Dataset

Nous avons utilisé la base d'images Pascal VOC (Visual Object Classe) version 2012, elle contient 11530 images couleur de différentes dimensions pour l'apprentissage distribués sur 20 classes. Dans notre projet nous avons utilisé 2000 images qui ont été redimensionnement avant de les soumettre au réseau. cette base Contient 20 class : avion, vélo, oiseau, bateau, bouteille, autobus, voiture, chat, chaise, vache, table à manger, chien, cheval, moto, la personne, plante en pot, mouton, canapé, train, TV monitor.

3.4.1 Découpage de la base d'images

La base des images est découpée en trois parties :

- Les images d'apprentissage/entraînement (training set), qui vont servir à apprendre le réseau. Cela représente 1800 images ;
- Les images de test (test set), qui servent à évaluer les progrès de notre modèle. Elles présentent 100 images.
- Les images de validation (validation set), qui présentent 100 images.

3.4.2 Fonction de redimensionnement

Les images d'apprentissage ont des tailles différentes, par conséquent, elles doivent être redimensionnées avant d'être utilisées comme entrée du modèle. Tandis que notre système requiert une dimensionnalité d'entrée constante, nous avons sous-échantillonné toutes les images à une résolution fixe de 320 X 320.

3.5 Présentation de l'application

Nous allons maintenant présenter les résultats obtenus grâce à des expériences réalisées par l'application de notre approche de supprimer le filigrane on utilise le deep learning .

3.5.1 Auto-encodeur (AE)

définition : Un auto-encodeur est un réseau de neurones conçu pour l'apprentissage d'une représentation d'un ensemble de données, dans le but de réduire la dimension de cet ensemble. Il permet d'extraire des caractéristiques pertinentes à l'entrée dont le but est de les reconstruire de façon non supervisée. dans notre application on utilise L'auto-encodeur convolutif comme un de pré-entraînement. Les paramètres d'auto-encodeur sont initialisés de manière aléatoire.

objectif : L'intérêt de l'utilisation des auto-encodeurs est d'apprendre une représentation d'un ensemble des données, mais dans notre projet notre but est d'améliorer les résultats de l'auto encodeur débruiteur. Nous avons entraîné un auto-encodeur et l'utilisé pour initialiser les valeurs des poids d'un auto-encodeur débruiteur avec la même architecture.la figure3.1 présente l'architecture de notre réseau.

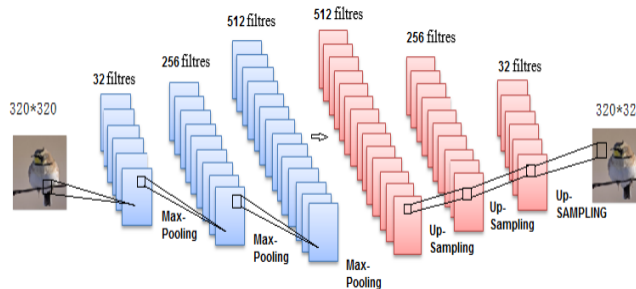


FIGURE 3.1 – L'architecture de notre Auto-encodeur (AE)

3.5.2 Combinaison auto-encodeur débruiteur

L'auto-encodeur débruiteur est une autre variante d'auto-encodeur, Le DAE (Denoising Autoencoder) est un enregistreur qui reçoit des données bruitées en entrée et qui est entraîné à prédire les données originales non bruitées en sortie.

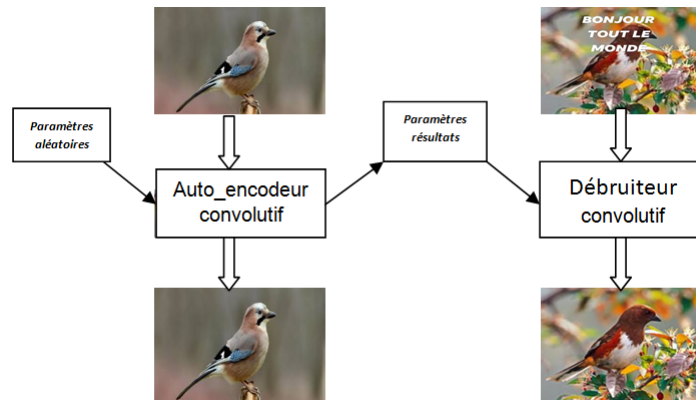


FIGURE 3.2 – Combinaison auto-encodeur débruiteur

3.5.3 Configuration du CNN

Nous allons entraîner un réseau de neurones convolutif. Le réseau possède 13523203 paramètres entraînables et il consiste en 7 couches convolutives. Une couche de Max Pooling est placée après chaque couche

convolutive d'encodage et une couche de Up- Sampling entre chaque deux couches convolutives de décodage. Pour un apprentissage plus rapide, ReLu est la fonction d'activation. La figure 3.3 montre la structure de notre CNN.

```

input_img = Input(shape=(320, 320, 3))
x = Conv2D(32, (5,5), activation='relu', padding='same')(input_img)
x = MaxPooling2D((2, 2), padding='same')(x)
x = Conv2D(256, (5, 5), activation='relu', padding='same')(x)
x = MaxPooling2D((2, 2), padding='same')(x)
x = Conv2D(512, (5, 5), activation='relu', padding='same')(x)
encoded = MaxPooling2D((2, 2), padding='same')(x)

x = Conv2D(512, (5, 5), activation='relu', padding='same')(encoded)
x = UpSampling2D((2, 2))(x)
x = Conv2D(256, (5, 5), activation='relu', padding='same')(x)
x = UpSampling2D((2, 2))(x)

x = Conv2D(32, (5, 5), activation='relu', padding='same')(x)
x = UpSampling2D((2, 2))(x)
decoded = Conv2D(3, (5, 5), activation='sigmoid', padding='same')(x)

```

FIGURE 3.3 – Composants de notre réseau de neurones

D'après la figure 3.3, notre modèle CNN comprend les composants suivants : **resume=*,*,*

- **La couche d'entrée** : sa dimension $320*320*3$, le nombre 3 représente les couleurs Rouge, Vert et Bleu.
 - **La couche de convolution** : dans cette couche la taille du filtre est fixée à $5*5$.
 - Le nombre de filtre où la profondeur a été fixée à 32, 256, 512 successivement pour les couches d'encodage, est l'inverse pour les couches de décodage.
 - La fonction d'activation ReLU : force le réseau à retourner des valeurs positives. Tout nombre inférieur à 0 est converti en 0, cette technique est appliquée pour toutes les couches, sauf la dernière couche qui utilise la fonction sigmoid.
 - **La couche de Pooling** : nous avons utilisé un max pooling de taille $2x2$ pour réduire la taille des paramètres de l'image.
 - **La couche de Up-Sampling** : est la couche inverse de pooling qui permet d'inverser le pooling ou de reconstruire l'image. La taille utilisée est $2x2$.
 - Le pas : à été fixé à 1 dans les couches de Pooling et à 2 dans la couche de Up- Sampling.
 - Dans notre modèle nous n'avons pas utilisé une couche entièrement connectée.
- Les hyperparamètres sont fixés pour toutes les expériences comme suit :
- La taille de mini-batch : 16.
 - La fonction de coût utilisé : MSE.
 - Toutes les couches convolutives utilisent la fonction ReLu sauf la couche de sortie utilise la fonction Sigmoid.
 - L'optimiseur utilisé est adam.
 - La taille des filtres ($5x5$), facteur de sous-échantillonnage ($2x2$).

3.5.4 les méthodes d'évaluation :

Afin d'évaluer la qualité des image construite par notre auto-encodeur débruiteur et pour évaluer l'efficacité de notre débruiteur(Efficacité de la suppression du filigrane) on utilise le SSIM (mesurer la similarité de structure entre deux image)et le CW-SSIM.

Couche	Paramètre
Entrée	image RVB 32X32 pixl
Conv2D(ReLu)	Profondeur : 32 Fenêtre : 5X5 Pas : 1
Couche de Pooling	Fenêtre : 2X2 : Pas : 1
Conv2D(ReLu)	Profondeur : 256 Fenêtre : 5X5 Pas : 1
Couche de Pooling	Fenêtre : 2X2 : Pas : 1
Conv2D(ReLu)	Profondeur : 512 Fenêtre : 5X5 Pas : 1
Couche de Pooling	Fenêtre : 2X2 : Pas : 1
Conv2D(ReLu)	Profondeur : 512 Fenêtre : 5X5 Pas : 1
Couche de UpSampling	Fenêtre : 2X2 : Pas : 2
Conv2D(ReLu)	Profondeur : 256 Fenêtre : 5X5 Pas : 1
Couche de UpSampling	Fenêtre : 2X2 : Pas : 2
Conv2D(ReLu)	Profondeur : 32 Fenêtre : 5X5 Pas : 1
Couche de UpSampling	Fenêtre : 2X2 : Pas : 2
Conv2D(sigmoid)	Profondeur : 32 Fenêtre : 5X5 Pas : 1

TABLE 3.1 – Configuration de notre CNN

- SSIM1 :le SSIM entre les images bruitées et les image reconstruites par le réseau.
- SSIM2 :le SSIM entre les images originales et les image reconstruites par le réseau.

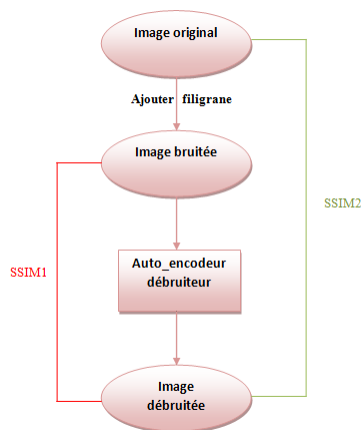


FIGURE 3.4 – Plan d'évaluation

3.6 Résultats d'application

3.6.1 Les résultats obtenus par l'auto-encodeur convolutif

Les résultats obtenus par l'auto-encodeur convolutif sont très proches aux images originales, les résultats sont visualisables dans la figure 3.5.

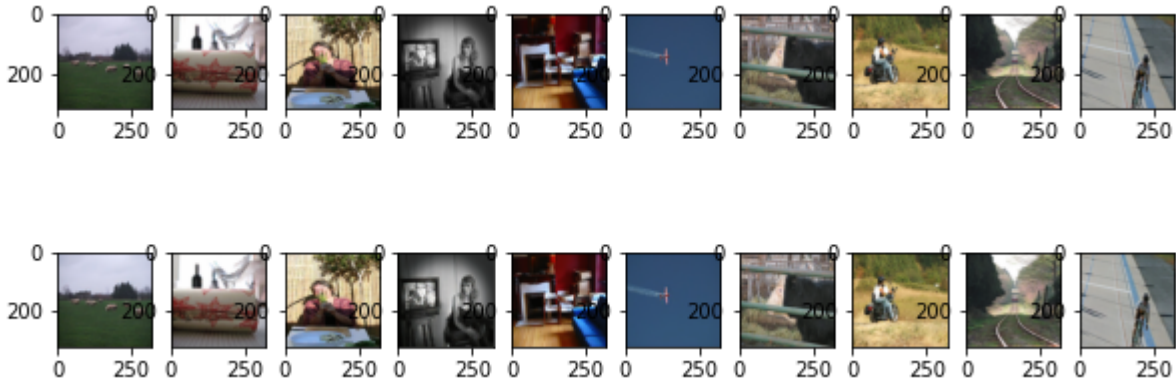


FIGURE 3.5 – Image résultat d'auto-encodeur convolutif.

les graphes de la Figure 3.6 montrent que la précision d'apprentissage et de validation augmente avec le nombre des époques. En même temps, l'erreur d'apprentissage et de validation diminue avec le nombre des époques. Donc à chaque époque notre modèle apprend à reconnaître plus d'informations et de manière plus rapide.

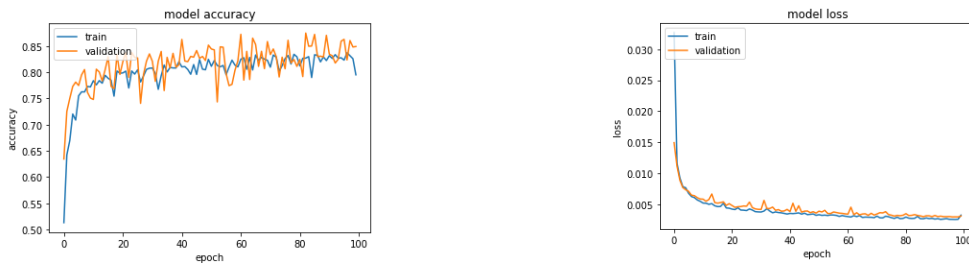


FIGURE 3.6 – Graphes de Précision et d'Erreur de l'auto-encodeur convolutif.

La figure 3.7 montre les valeurs du SSIM entre les images originales et les images reconstruites par notre réseau de neurones. Notons que certaines images ayant un SSIM proche de 1. Elles sont très semblables aux images originales. Ce sont des images simples avec un fond uni et qui contiennent moins de détails.

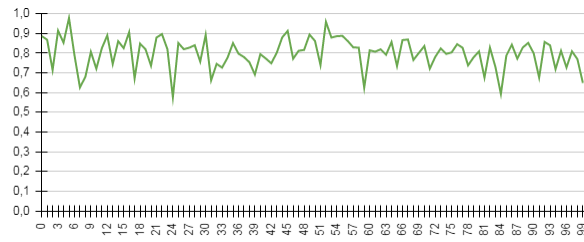


FIGURE 3.7 – Graphe de SSIM entre les images originales et les images obtenues par l’auto-encodeur convolutif.

3.6.2 les résultats obtenus par l’autoencodeur débruiteur convolutif

Le processus d’apprentissage dans l’auto-encodeur débruiteur passe par deux étapes .La première consiste à entraîner le CNN pour qu’il puisse effectuer le rôle d’un auto-encodeur convolutif. Le but de cette partie est d’entraîner le réseau à reproduire l’image à partir de sa version originale. Les paramètres du réseau résultat seront utilisés comme paramètres initiaux de la deuxième partie d’apprentissage. Cette partie permet d’entraîner le réseau à reproduire les images originales à partir de leurs versions filigranées. Les images de la base d’apprentissage de cette étape sont créés en ajoutant des filigranes aux images originales.

Le filigrane "test figrane"

Notre CNN construit une image sans filigrane. Les résultats sont présentés dans la figure 3.8.



FIGURE 3.8 – Images résultats d’auto-encodeur convolutif débruiteur

D’après l’image 3.8 ,nous notons que les images résultantes sont des images sans filigrane et proches des images originales.Pour prouver cette observation, les images doit êtres évaluées par SSIM(évaluation objectif).

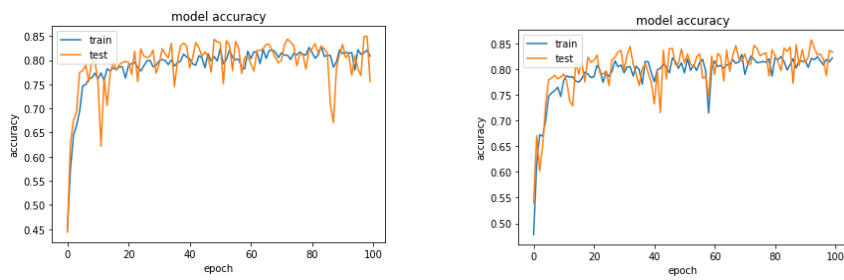


FIGURE 3.9 – Graphe de la Précision pour l’auto-encodeur convolutif débruiteur

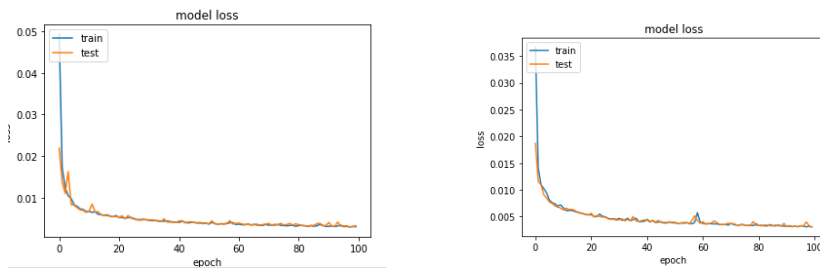


FIGURE 3.10 – Graphe de l’erreur pour l’auto-encodeur convolutif débruiteur

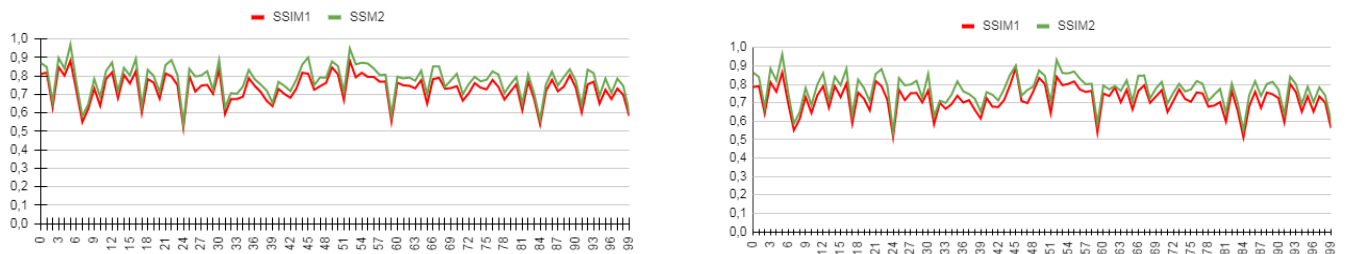


FIGURE 3.11 – SSIM

filigrane : text court

Les résultats sont présentés dans la figure 3.12.



FIGURE 3.12 – Images résultats d’auto-encodeur débruiteur (text court)

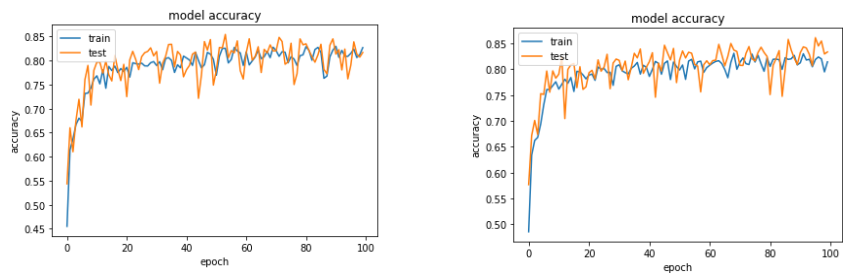


FIGURE 3.13 – Graphe de la Précision pour l’auto-encodeur débruiteur

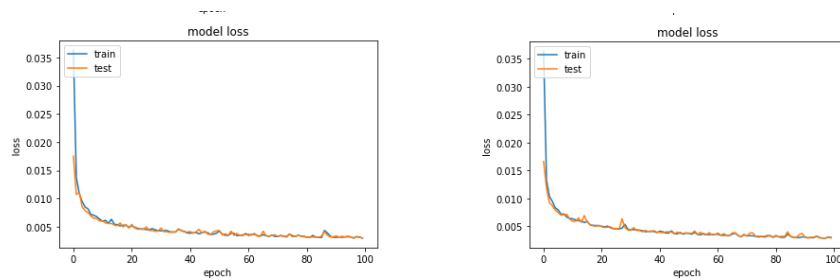


FIGURE 3.14 – Graphe de l’erreur pour l’auto-encodeur débruiteur

D’après les graphes de SSIM résultats présenter dans la figure 3.15 ,On remarque qu’ils sont pa-
reils, donc les images résultats de ces deux traitements sont de la même qualité.
On remarque Aussi ,que notre application capable de reconstruire une image de la même qualité que limage
originale (SSIM=1 dans le graphe)Sans apprendre sur l’image d’origine (image de test).

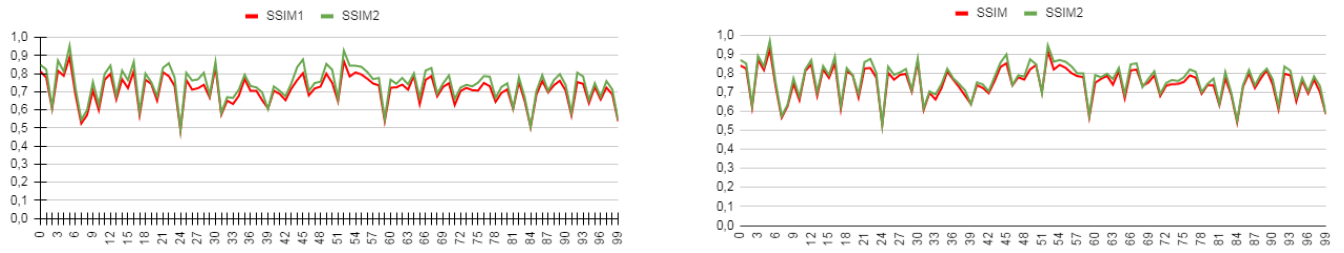


FIGURE 3.15 – Graphe de SSIM

Le filigrane : Texte aléatoire

Notre CNN reconstruit une image sans filigrane. Les résultats sont présentés dans la figure3.16.



FIGURE 3.16 – Images résultats d’auto-encodeur débruiteur (texte aléatoire).

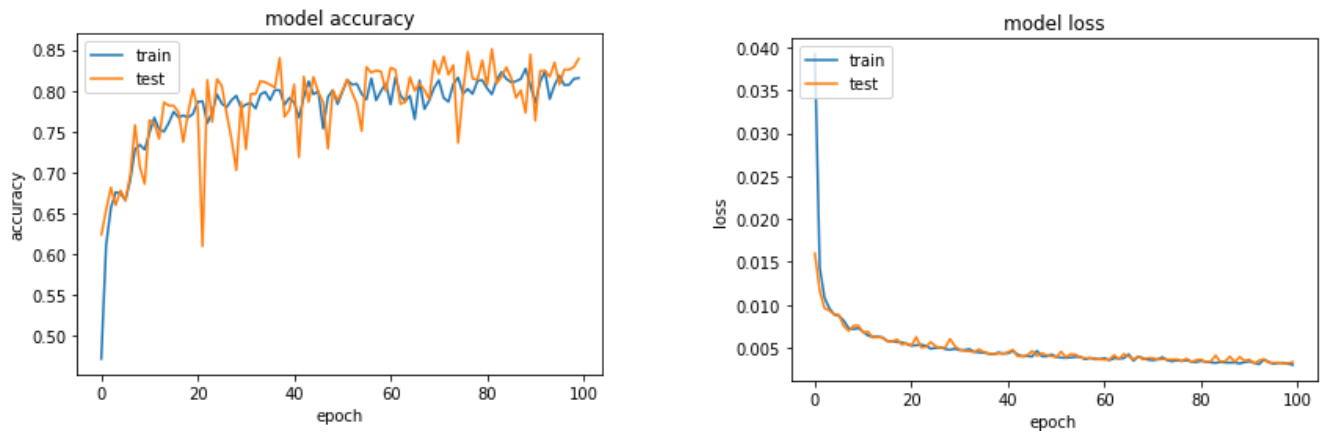


FIGURE 3.17 – Graphe de la Précision et de l’erreur pour l’auto-encodeur débruiteur

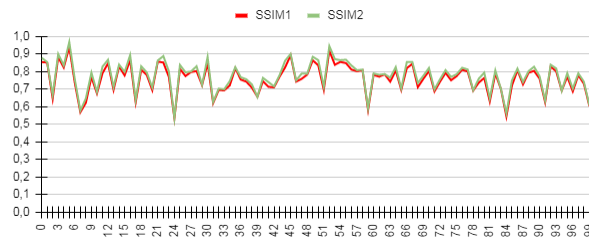


FIGURE 3.18 – SSIM des images résultats (texte aléatoire).

3.6.3 Évaluation par CW-SSIM :

Le filigrane :texte aléatoire :



FIGURE 3.19 – Images résultats .

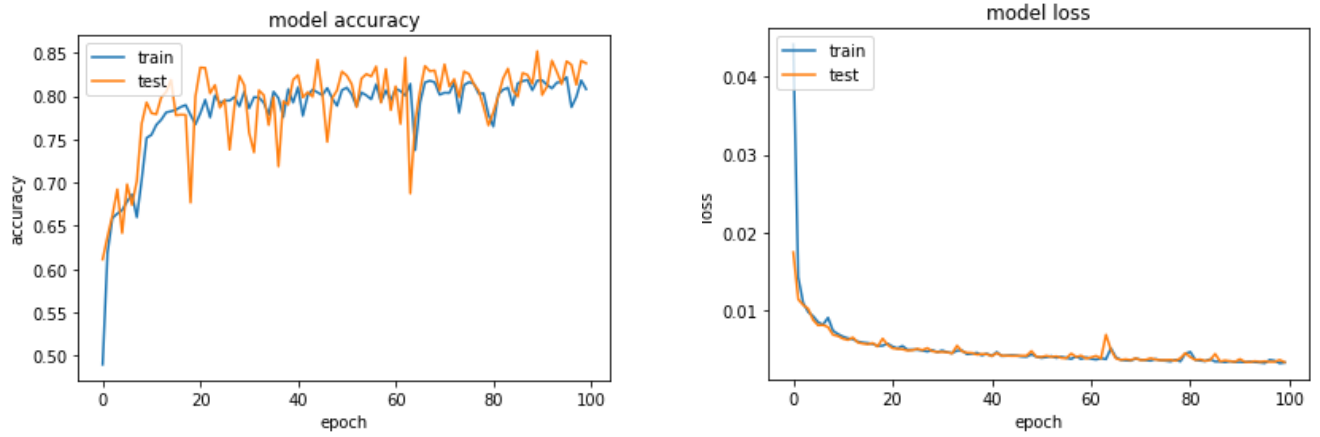


FIGURE 3.20 – Graphe de la Précision et de l’erreur pour le débruiteur(cwssim)

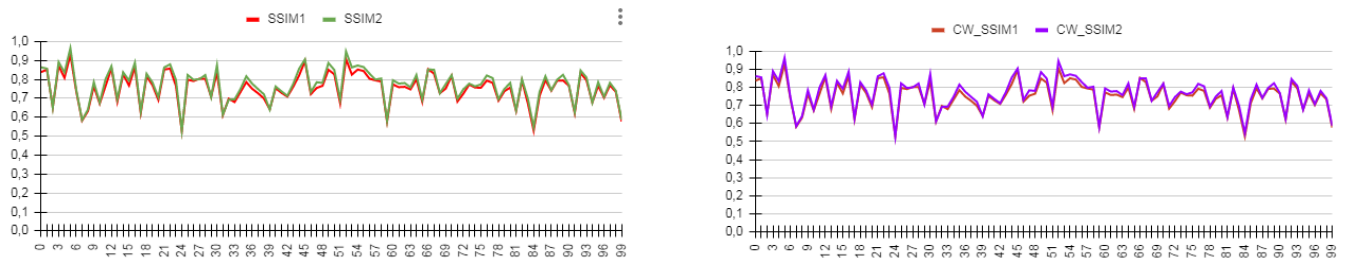


FIGURE 3.21 – Graphe de SSIM à gauche et de CW-SSIM à droit

D’après la figure 3.21 les résultats de SSIM et de CW-SSIM sont les même.

3.7 Les résultat obtenus par un débruiteur convolutif

Le filigrane :texte aléatoire :



FIGURE 3.22 – Images résultats débruiteur (texte aléatoire).

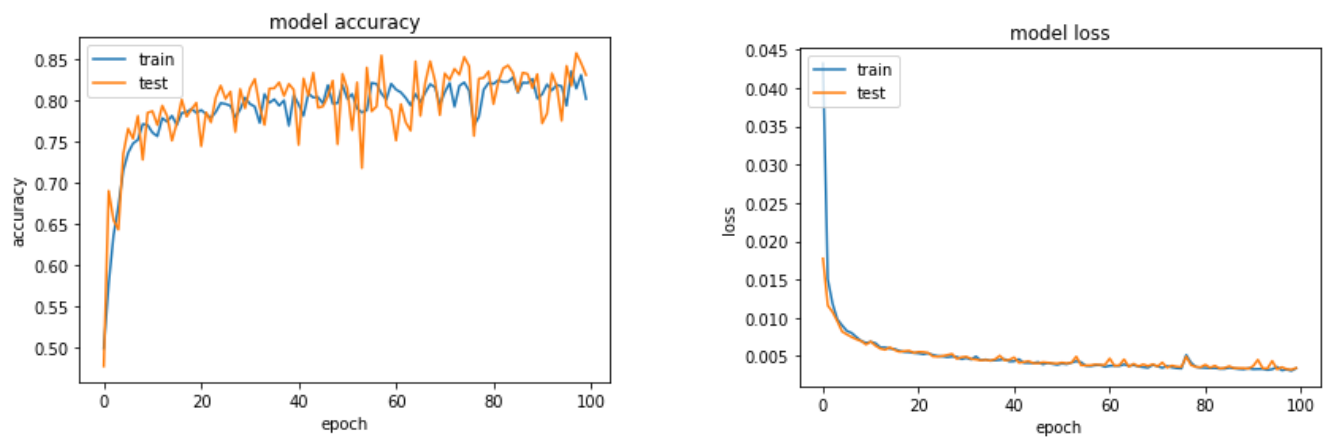


FIGURE 3.23 – Graphe de la Précision et de l'erreur pour débruiteur

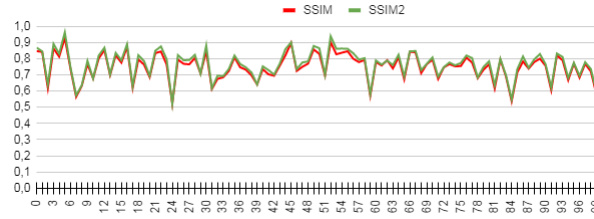


FIGURE 3.24 – SSIM des images résultats d’un débruiteur(texte aléatoire).

3.7.1 Pearson coefficient de corrélation(PCC)

Le PCC est la covariance entre le score SSIM1 et SSIM2, rapporté au produit de leurs écart-types :

$$\frac{\text{Covariance}(SSIM1, SSIM2)}{\sigma_{SSIM1} * \sigma_{SSIM2}} \quad (3.1)$$

Il détermine la corrélation des SSIMs, une grande valeur de PCC (proche à 1) indique une bonne précision de prédiction. Le tableau ci-dessous présente les valeurs de PCC calculées :

filigrane	thème et taille	Auto-encodeur débruiteur
"ILM"	Arial /20	0.9930375173
"ILM"	Arial /50	0.9895909558
"testfiligrane"	Arial /70(noir)	0,9779749979
"testfiligrane"	Arial /70	0,9866073836
text aléatoire	Arial	0,9942491089
text aléatoire	Times New Roman	0,8198216837

TABLE 3.2 – Tableau des PCC calculés

On remarque qu’on a des bonnes résultats, les valeurs de PCC sont élevées (proches à 1 et elles se diminuent avec l’augmentation de bruit), ce qui indique qu’il existe une corrélation positive entre les SSIM1 et les SSIM2.

3.8 Conclusion

Dans ce chapitre nous avons présenté une méthode pour supprimer le filigrane en utilisant les réseaux de neurones profonds (Deep Learning). Nous avons utilisé un autoencodeur convolutif combiné avec un débruiteur. Nous avons présenté dans ce chapitre les différents paramètres de codage des deux réseaux de neurones précédemment cités.

Dans la deuxième partie nous avons présenté les différents résultats obtenus en fonction du SSIM et du PCC , ces résultats ont prouvé l’efficacité de cette combinaison.

Conclusion

Le filigrane n'est pas une technologie nouvelle, mais il est très ancien, tel qu'il est apparu pour la première fois sur papier en 1282, mais avec le développement de la numérisation, il est devenu largement présent sur la plupart des images d'Internet et des appareils photos. Ce filigrane peut-être un handicap pour qui, veut utiliser ou exploiter ces images ,en particulier avec la difficulté d'obtenir l'image originale. Cela a créé un champ de recherche sur les techniques de suppression de Filigrane parce que c'est un obstacle pour les utilisateurs de ces images, et d'autre part il affecte négativement la qualité de l'image car il peut couvrir certains détails importants dans l'image. Et comme nous nous soucions de l'image et sa qualité, nous avons traité le sujet dont nous avons discuté afin de supprimer le filigrane d'une nouvelle manière, en nous appuyant sur le dernier domaine des médias automatisés, qui est l'apprentissage en profondeur. Nous avons suivi les résultats obtenus en évaluant leur qualité par évaluation visuelle et avec l'évaluation objective (SSIM).

Dans la première partie de notre travail, nous avons commencer par un autoencodeur convolutif dans le but de récupérer une image à partir de sa version originale. Il permet d'extraire des caractéristiques pertinentes à l'entrée dont le but est de les reconstruire de façon non supervisée. Les paramètres d'autoencodeur sont initialisés de manière aléatoire.

Les résultats de notre autoencodeur convolutif implémenté a donné des résultats très satisfaisants vis à vis le SSIM et la qualité visuelle. L'objectif de ce réseau dans notre travail n'était pas de reproduire une image à partir de sa version originale mais de préparer le réseau à reproduire une image à partir de sa version filigranée. On peut considéré cette étape comme étant un prétraitement.

Dans la deuxième partie de notre travail, nous avons combiné l'auto-encodeur par un autre réseau de neurones convolutif (débruiteur) et il est utilisé pour initialiser les valeurs des poids d'un auto-encodeur débruiteur avec la même architecture. Nous avons utilisé les paramètres résultats de l'autoencodeur comme paramètres initiaux au débruiteur. Cette combinaison a donné des bonnes qualités subjective(visuelle) et objectives (SSIM, CW-SSIM).

Perspective Selon les résultats obtenus dans les expériences effectuées dans ce travail, nous proposons comme perspectives de :

Tester l'autoencodeur débruiteur sur les images filigranées par image(filigrane image).

Bibliographie

- [1] Yann LeCun. Recherche sur l'intelligence artificielle, 2016.
- [2] Richard M. and David Grubin. *The Secret Life of the Brain*. Joseph Henry Press, 2001.
- [3] Nikhil Buduma and Nicholas Lacascio. *Fundamentals of Deep Learning*. June 2017.
- [4] Martial Mermillod. *Réseaux de neurones biologiques et artificiels ; Vers L'émergence de systèmes artificiels conscients ?*, volume 129. 2016.
- [5] Djeriri Youcef. Les réseaux de neurones artificiels. September 2017.
- [6] Gilbert Saporta. Une brève histoire de l'apprentissage. 2018.
- [7] Barry SMITH. L'esprit connexionniste : une étude de la psychologie de hayek. 1/1999.
- [8] Kary Framling. *Modélisation et apprentissage des préférences par réseaux de neurones pour l'aide à la décision multicritère*. PhD thesis, INSA de Lyon, 1996.
- [9] Marc Sauget. *Parallélisation de problèmes d'apprentissage par réseaux neuronaux artificiels. Application en radiothérapie externe*. PhD thesis, Université de FrancheComté, 2007.
- [10] Antoine Bordes Xavier Glorot and Yoshua Bengio. Deep sparse rectifier neural networks. in proceedings of the fourteenth international conference on artificial intelligence and statistics. page 315 à 323. 2011.
- [11] Ilya Sutskever Alex Krizhevsky and Geoffrey E. Hinton. Imagenet classification with deep convolutional neural networks. page 84 à 90. ACM, May 2017.
- [12] K. Gurney. An introduction to neural networks. 1997.
- [13] Aurélie Fischer. Deux méthodes d'apprentissage non supervisé : synthèse sur la méthode des centres mobiles et présentation des courbes principales. *la Société Française de Statistique*, 17 mars 2014.
- [14] Alphago. <https://fr.wikipedia.org/wiki/AlphaGo>.
- [15] Les réseaux de neurones convolutifs. <https://datasciencetoday.net//deep-learning/lesr{é}seaux-deneurones-convolutifs>.
- [16] N. Golea. Tatouage numérique des images couleurs rgb, 2010.
- [17] *Les Filigranes. Dictionnaire historique des marques du papier dès leur apparition vers 1282 jusqu'en 1600*, volume 4. G CORG-OLMS, 2 edition, 1991.
- [18] *Electronic Watermark , Digital Image Computing, Technology and Applications*. 1993.
- [19] Différence entre la stéganographie et la cryptographie. <https://fr.fondoperlaterra.org/comdifference-between-steganography-and-cryptography>, 2020.
- [20] J. Kilian F. T. Leighton Cox, I. J. and T. Shamoon. Secure spread spectrum watermarking for multimedia, 1997.
- [21] M. Banagar Zarmehi, N. and M. A. Akhaee. Optimum decoder for an additive video watermarking with laplacian noise in h.264.
- [22] M El-Hajji. La sécurité d'images par le tatouage numérique dans le domaine d'ondelettes, 2012.

- [23] N. V. Dharwadkar and B. Amberker. Secure watermarking scheme for color image using intensity of pixel and lsb substitution. *Journal of Computing*, 2009.
- [24] B. Chen and G. W. Wornell. Quantization index modulation : a class of provably good methods for digital watermarking and information embedding, may 2001.
- [25] T. Furon and P. Bas. A new measure of watermarking security applied on qim, 2013.
- [26] M. Khan and T. Shah. A copyright protection using watermarking scheme based on nonlinear permutation and its quality metrics.
- [27] ITU-T Recommendation P.910. Subjective video quality assessment methods for multimedia applications. recommendations of the itu, 2008.
- [28] C.CHARRIER et al. Evaluation de la qualité des images, 2012.
- [29] O. Le Meur P. Le Callet A. Ninassi and D. Barba. Considering temporal variations of spatial visual distortions in video quality assessment. *IEEE Journal of Selected Topics in Signal Processing(JSTSP)*, 2009.
- [30] ITU-R Recommendation BT.1683. Itu-r recommendation bt.1683. objective perceptual video quality measurement techniques for standard definition digital broadcast television in the presence of a full reference., 2011.
- [31] E. R. Kandel. *Principles of Neural Science*. McGraw-Hill., 2013.
- [32] Hamid R. Sheikh. Zhou Wang, Alan C. Bovik and Eero P. Simoncelli. Image quality assessment : From error visibility to structural similarity., 2004.
- [33] Yang C.L Xie S.L Chen, G.H. Gradientbased structural similarity for image quality assessment.
- [34] A. K. Moorthy and A. C. Bovik. A two-step framework for constructing blind image quality indices,, 2010.
- [35] *Ten lectures on wavelets,society for industrial and applies 100 mathematics*. 1992.
- [36] K. Ghazinour O. Alaql and C. C. Lu. Classification of image distortions based on features evaluation, 2015.
- [37] Rajiv Soundararajan and Alan C. Bovik. Rred indices. Reduced reference entropic differencing for image quality assessmen, 2012.
- [38] Zainab Awan Noor Al Madeed and Somaya Al Madeed. Image quality assessment a survey of recent approaches. 2011.
- [39] Shibao Zheng Xiaolin Chen and Rui Zhang. Reduced reference image quality assessment based on image statistics in pixel domain. *Advances on Digital Television and Wirless Multimedia Communications*, 2012.
- [40] Henri Michel. Google colab : Le guide ultime, 4/11/2019.
- [41] tensorflow.
- [42] keras.

Résumé

Dans cette étude nous avons utilisé la méthode d'apprentissage profond et les réseaux de neurones convolutifs. Nous avons expérimenté ces réseaux pour supprimer le filigrane de façon automatique, sans utiliser l'image originale (apprentissage non supervisé). Pour ce faire, nous avons appliqué la combinaison autoencodeur-débruiteur sur un ensemble d'images filigranées. La base des images utilisée est PASCAL VOC dont nous avons testées 2000 images (1800 images pour l'entraînement, 100 images pour la validation et 100 pour le test), nous avons déformé ces images en ajoutant un texte (filigrane). Nous avons procédé à une évaluation des images produites en les comparant aux images originales et les images portant un filigrane, en utilisant la métrique d'évaluation objective, basée sur l'approche structurelle (SSIM). Les résultats étaient, en grande partie, assez satisfaisants. Les réseaux de neurones convolutifs ont pu rétablir et restituer l'authenticité et la qualité des images à 82% de celle des images originales.

Mots-clés : Apprentissage profond , Réseaux de neurones convolutifs , Filigrane , Auto-encodeur débruiteur , Evaluation objective , Qualité de l'image .

Abstract

In this study we used the deep learning method and convoluted neural networks. We experimented these networks to remove the watermark automatically, without using the original image(unsupervised learning).To do this, we applied the autoencoder-denoiser combination on a set of watermarked images. The basis of the images used is PASCAL VOC of which we tested 2000 images (1800 images for training, 100 images for validation and 100 for testing). We evaluated the images produced by comparing them to the original images and the watermark images, using the objective assessment metric, based on the structural approach (SSIM). The results were, to a large extent, quite satisfactory. Convoluted neural networks were able to restore the authenticity and quality of the images to 82% of that of the original images.

Keywords : Deep learning ,convoluted neural networks ,watermark,autoencoder-denoiser, Objective evaluation, Image quality.