



Faculté des Sciences Exactes et Informatique
Département d'Informatique

Mémoire de fin d'études pour l'obtention du diplôme de
Master en Informatique

Option : Informatique Légale et Multimédia

Thème

**Techniques d'apprentissage automatique pour
l'évaluation de la qualité des images fixes.**

Présenté par :

- ZAOUMA Rima
- LAHMAR Amina

Encadré par :

- MAHROUK Zahir

Dédicace

Je me dois d'avouer pleinement ma reconnaissance à toutes les personnes qui m'ont soutenue durant mon parcours, qui ont su me hisser vers le haut pour finaliser ce travail. C'est avec amour, respect et gratitude que,

Je dédie ce travail à :

mon père, qui veillait sur notre confort et notre éducation, j'espère qu'Allah prolonge sa vie afin de qu'il se réjouisse avec plus de succès.

celle qui m'a transmis la vie, l'amour, le courage, à toi **chère mère** toutes mes joies, mon amour et ma reconnaissance.

mes chère frères et mes belles soeurs : Roumaïssa, Mohammed Amine, Sifeddine, Manar, Rokaïa.

toute ma famille.

ma belle Rima et mes collègues de promotion.

des personnes spéciales dans ma vie, qu'ils étaient avec moi tout le temps.

tous ceux ou celles qui me sont chers.

tous mes enseignants tout au long de mes études.

tous ceux qui ont participé à la réalisation de ce travail.

Amina

Dédicace

À

mon cher père, que Dieu lui fasse miséricorde.

ma chère maman, que Dieu la protège.

mes chère frères et mes belles soeurs.

mon fiancé, que Dieu le protège.

toute la famille.

mes amis, et mes collègues de promotion.

je dédie ce modeste travail.

Remerciements

*Avant toute chose, nous rendons grâce à **Allah** le tout puissant qui nous a fait ouvrir les portes du savoir, qui nous a donné le courage, la volonté, la force nécessaire durant tout notre cursus pédagogiques.*

*Notre profonde gratitude à nos **parents** pour leur soutien moral indéfectible.*

*Nous tenons à remercier notre encadreur **Dr MAHROUK ZAHIR**, pour son aide, ses conseils, et ses orientations pour l'accomplissement de ce mémoire.*

*Nous remercions également **les membres du jury** qui nous ont honorés en acceptant l'invitation d'évaluer ce modeste travail.*

*Enfin, nous tenons à exprimer notre reconnaissance à tous nos **amis** et **collègues** pour le soutien moral et matériel.*

Résumé

L'évaluation de la qualité d'image présente un intérêt substantiel pour les services ainsi que pour les systèmes de traitement d'images où le dernier maillon de la chaîne est l'observateur humain.

L'objectif de ce travail est d'exploiter les techniques d'apprentissage supervisé, en particulier les réseaux de neurones convolutifs combinés avec la régression en utilisant les SVMs, pour l'estimation automatique de la qualité de l'image. Le système apprend à partir des étiquettes de la qualité subjective et construit un modèle capable de continuer à fournir une mesure objective toujours correspondre à l'avis de l'homme à toute image qui lui est présentée.

Le but principal était d'optimiser la performance prédictive des mesures développées en fonction de la corrélation, la monotonie et la précision. Le système de prédiction présenté, commence par l'utilisation des CNNs pour l'obtention des cartes de caractéristiques qui permettent d'extraire les caractéristiques les plus pondérantes dans les images étudiées. Ensuite la distribution gaussienne généralisée asymétrique est utilisée pour calculer les caractéristiques statistiques de ces cartes, à la fin un SVR est entraîné avec ces caractéristiques statistiques combinées avec les étiquettes de scores issues de l'évaluation subjective pour prédire les scores de qualité subjectifs des images étudiées.

***Mots-clés** : Évaluation de la qualité des images, réseau de neurones convolutif, statistique des scènes naturelles.*

Abstract

Image quality assessment is of substantial interest for services as well as for image processing systems where the last link in the chain is the human observer.

The objective of this work is to exploit supervised learning techniques, in particular convolutional neural networks combined with regression using SVMs, for automatic image quality estimation. The system learns from the subjective quality labels and builds a model capable of continuing to provide an objective measure that always matches the human opinion of any image presented to it.

The main goal was to optimize the predictive performance of the developed measures in terms of correlation, monotonicity and accuracy. The prediction system presented starts with the use of CNNs to obtain feature maps that allow the extraction of the most prevalent features in the images studied. Then the asymmetric generalized Gaussian distribution is used to compute the statistical features of these maps, at the end a SVR is trained with these statistical features combined with the score labels from the subjective evaluation to predict the subjective quality scores of the studied images.

Keywords : *Image Quality Assessment, Convolutional Neural Network, Nature Scene Statistic.*

TABLE DES MATIÈRES

Table des Matières	i
Table des figures	iv
Liste des tableaux	vi
Liste des acronymes	vii
Introduction générale	1
1 Evaluation du qualité d'image fixé	3
1.1 Introduction	3
1.2 Qualité des images	3
1.2.1 Effets de bloc	4
1.2.2 Effet d'oscillations parasites (ringing effect)	4
1.2.3 Flou	5
1.2.4 Bruit	5
1.3 Evaluation subjective	6
1.3.1 Facteurs influant	7
1.3.2 Protocoles	8
1.3.2.1 Méthode à simple stimules	8
1.3.2.2 Méthode à double stimules	8
1.3.2.3 Méthode comparative	9
1.3.3 Analyse des résultats	9
1.4 Evaluation objective	10
1.4.1 Métriques pour la mesure de la qualité objective	10
1.4.1.1 Métriques à référence complète	11
1.4.1.2 Métriques à référence réduite	14
1.4.1.3 Métrique sans référence	15
1.4.2 Mésures inspirées du Système Visuel Humain	18
1.4.2.1 Système visuel humain	18
1.4.2.2 Caractéristiques psychophysiques de SVH	20
1.4.2.3 Méthodes basées sur SVH	21

1.5	Évaluation des performances des métriques de qualité	21
1.6	Conclusion	23
2	Réseau de neurones artificiels	24
2.1	Introduction	24
2.2	Réseau de neurone biologique	24
2.2.1	Le neurone biologique	24
2.2.2	Fonctionnement du neurone biologique	25
2.3	Réseau de neurone artificielle	26
2.3.1	Du neurone biologique au neurone formel	26
2.3.2	Historique	27
2.3.3	Caractéristiques de RNA	28
2.3.4	Architecture de RNA	30
2.3.5	L'apprentissage	33
2.3.5.1	Apprentissage supervisée	33
2.3.5.2	Apprentissage non supervisée	34
2.3.5.3	Apprentissage Semi-Supervisé	35
2.4	Réseau de neurone convolutionnel	36
2.4.1	Architecture de CNN	37
2.4.1.1	Couche de convolution(CONV)	37
2.4.1.2	Couche de pooling (POOL)	38
2.4.1.3	Couches de correction (RELU)	39
2.4.1.4	Couche entièrement connectée (FC)	40
2.4.1.5	Couche de perte (LOSS)	40
2.5	Classification SVM	41
2.6	Domaines d'application des RNA	43
2.7	Avantages et inconvénients des RNAs	43
2.8	Conclusion	45
3	Implémentation et Résultats	46
3.1	Introduction	46
3.2	Ressources matérielles	46
3.3	Ressources logicielles	46
3.3.1	Python	46
3.3.2	PyCharm	47
3.3.3	PyTorch	47
3.3.4	Anaconda	48
3.3.5	Scikit-Learn	48
3.3.6	Google Colab	48
3.4	Base des images	49
3.5	Interfaces Graphiques Utilisateurs	50
3.6	L'architecture d'application	51
3.6.1	Extraction des caractéristiques (CNN)	52
3.6.2	Evaluation de la qualité d'image sans référence	53
3.6.3	Apprentissage (SVR) et Test (prédit)	56
3.6.4	Evaluation des performance	58

3.7 Conclusion	58
Conclusion générale	59
Bibliographie	vii

TABLE DES FIGURES

1.1	Evaluation de la qualité d'image.	3
1.2	Exmple d'effet de bloc.	4
1.3	Exmple d'effet d'oscillations parasites.	5
1.4	Image dégradée avec le flou.	5
1.5	Exmple du flou volontaire.	5
1.6	Exmple du Bruit poivre et sel.	6
1.7	Image dégradée avec la distorsion Bruit Gaussien.	6
1.8	Présentation des séquences d'évaluation.	7
1.9	Méthode à simple stimules.	8
1.10	Méthode à double stimules.	8
1.11	Catégories d'image objective.	11
1.12	Le schéma de l'algorithme SSIM.	14
1.13	Evaluation de la qualité d'image avec référence réduite.	15
1.14	Résumé du métrique BRISQUE.	17
1.15	Schéma simplifié de Système Visuel Humain.	18
1.16	Schéma d'oeil humain.	18
1.17	Schéma des voies visuelles.	19
1.18	Illustration de l'effet de masquage.	21
2.1	Modèle d'un neurone biologique.	25
2.2	Circulation de l'information dans le neurone biologiques.	26
2.3	Transmission de l'information entre les neurones biologiques.	26
2.4	Modèle d'un neurone formel.	27
2.5	Fonctions d'activations : (a) du neurone «seuil» ; (b) du neurone «linéaire», et (c) du neurone «sigmoïde».	29
2.6	L'architecture de réseau neurone.	30
2.7	Perceptron monocouche (simple).	31
2.8	Réseau neurone multicouche.	31
2.9	Perceptron calculant la fonction XOR.	31
2.10	Réseaux de neurones récurrents.	32
2.11	Architecture d'un réseau de Kohonen.	32
2.12	Architecture d'un Réseau de Hopfield.	33
2.13	Apprentissage supervisé.	33

2.14	Exemple du classification.	34
2.15	Illustration de la différence entre classification linéaire et régression linéaire.	34
2.16	apprentissage non supervisé.	35
2.17	Exemple Clustering.	35
2.18	L'apprentissage Semi-supervisé.	35
2.19	Architecture standard d'un réseau de neurone convolutionnel.	36
2.20	Les Réseaux de Neurones Convolutifs (CNN)	37
2.21	Exemple d'image RGB en couche de convolution.	38
2.22	Pooling avec un filtre 2x2 et un pas de 2.	39
2.23	Représentation graphique de la fonctionnalité Relu.	40
2.24	Schéma de principe d'un SVM multiclassés.	41
2.25	Exemple d'un problème de discrimination à deux classes, avec une séparatrice linéaire.	41
2.26	Exemple d'hyperplans séparateurs.	42
2.27	Exemple d'hyperplans séparateurs.	42
2.28	noyaux linéaires et non linéaires de SVR.	43
2.29	Réseau neurone classique pour reconnaissance d'image.	44
3.1	Logo Python.	47
3.2	Logo PyTorch.	47
3.3	Interface Anaconda.	48
3.4	Logo Scikit-Learn.	48
3.5	Interface Anaconda.	49
3.6	Echantillon des images de la base LIVE.	50
3.7	La fenêtre principale d'application.	51
3.8	Architecture d'application.	52
3.9	Filter de model VGG16.	52
3.10	Cartes des caractéristiques (feature maps).	53
3.11	Code source de méthode d'extraction des caractéristiques.	53
3.12	Code source de MSNC.	54
3.13	Code source AGGD.	54
3.14	Résultat AGGD d'apprentissage (Vécture de caractéristique).	55
3.15	Résultat AGGD d'apprentissage (complète la figure 3.14).	55
3.16	Résultat AGGD de test.	55
3.17	Code source Brisque.	56
3.18	Code source d'apprentissage et de test.	57
3.19	Résultat de test.	57
3.20	Résultat de test (complète la figure 3.19).	57
3.21	Résultat de test (complète la figure 3.20).	58
3.22	Code source de SCC et PCC.	58

LISTE DES TABLEAUX

1.1	Échelle représentative d'un protocole à simple stimulus.	8
1.2	Échelle représentative d'un protocole double stimulus.	9
2.1	Equivalence entre le neurone biologique et le neurone formel.	27
3.1	Type et nombre des images utilisé	50

LISTES DES ACRONYMES

AGGD	<i>Asymmetric Generalized Gaussian Distribution.</i>
CGL	<i>Corps Genouillés Latéraux.</i>
CNN	<i>Réseau Neurone Convolutionnel.</i>
CONV	<i>Convolution.</i>
DMOS	<i>Différence Score Optimal Moyen</i>
DCT	<i>Transformée Cosinus Discrète.</i>
FC	<i>Fully connected.</i>
IA	<i>Intelligence Artificielle.</i>
JPEG	<i>Joint Photographic Experts Group.</i>
LOSS	<i>perte en français.</i>
MSE	<i>Mean Square Error.</i>
MOS	<i>Score Optimal Moyen.</i>
NSS	<i>Natural Scene Statistics.</i>
POOL	<i>Pooling.</i>
PSNR	<i>Rapport crête signal surbruit.</i>
PCC	<i>Pearson Correlation Coefficient.</i>
ReLU	<i>Rectifier Linear Unit</i>
RMSE	<i>Root Mean Square Error.</i>
RGB	<i>Rouge, Green, Bleu.</i>
RNA	<i>Réseau Neurone Artificiel.</i>
SVH	<i>Système Visuel Humain.</i>
SVM	<i>Support Vector Machines.</i>
SSIM	<i>Indice de similarité structurelle.</i>
SCC	<i>Spearman Correlation Coefficient.</i>
UIT	<i>Union Internationale Télécommunications.</i>

INTRODUCTION GÉNÉRALE

Les images ont un rôle important dans les médias, mais leur utilisation dépend de sa qualité.

La qualité d'image est un terme général employé par l'observateur pour évaluer la qualité inhérente de l'image. Elle est principalement subjective, signifiant que chaque observateur évalue la qualité d'image différemment. Certains peuvent penser qu'une image est d'assez haute qualité, mais on peut avoir un individu avec des niveaux plus élevés qui peut indiquer que l'image manque de qualité [2].

Il existe deux façons de l'évaluation de la qualité d'image. La première consiste à mener des tests subjectifs. Ces tests exigent une équipement approprié et des protocoles normalizes pour permettre des coopérations, échanges et comparaisons des résultats fournis par différents laboratoires. Elle est effectuée par les observateurs humains.

Toute fois ces tests sont lourds à mettre en oeuvre, chers et surtout très longs et ne constituent donc pas une solution pratique pour les différents opérateurs. Pour éviter un tel inconvénient, les méthodes objectives représentant la deuxième alternative, ont pour objectif de définir des mesures de qualité qui soient fortement corrélées aux notes de qualité qu'aurait donnée un ensemble d'observateurs.

La validation des résultats obtenus par une méthode objective et sa conformément à l'évaluation subjective nécessite l'implication des procédures ainsi que des standards d'évaluation subjective de la qualité des images. Ces standards fixent plusieurs contraintes pour mener à bien l'évaluation. A cette fin plusieurs bases d'images sont mise en oeuvre qui sont développées par des laboratoires.

Parmi les technologies d'évaluation des images, on trouve les réseaux de neurones artificiels (RNA). Elles permettent aux ordinateurs de résoudre eux-mêmes les problèmes et d'améliorer leurs capacités globales, selon la méthode d'intelligence artificielle (IA) utilisée. Ces réseaux de neurones sont développés à l'aide de l'apprentissage, par lequel le comportement du réseau de neurones sera modifié jusqu'à ce que le comportement souhaité soit atteint.

Notre mémoire consiste à étudier une technique d'apprentissage automatique pour l'évaluation de la qualité des images fixes à l'aide des réseaux de neurones convolutionnels (CNN).

Cette mémoire est divisée en trois chapitres :

1. Dans le premier chapitre on va présenter l'évaluation de la qualité des images fixes : évaluation subjective, évaluation objective et l'évaluation de performance des métriques de qualité.
2. Le deuxième chapitre est consacré à la description des réseaux de neurones artificiels en général et les réseaux convolutifs en particulier.
3. Dans le troisième chapitre, on va montrer la partie expérimentale de notre travail et on discute les différents résultats obtenus.

Notre mémoire se termine par une conclusion générale.

CHAPITRE 1

EVALUATION DU QUALITÉ D'IMAGE FIXÉ

1.1 Introduction

La notion de qualité d'image est avant tout une notion subjective qui dépend de la perception visuelle de l'être humain. Actuellement, l'évaluation de la qualité d'image est très demandée dans diverses applications de communication multimédias.

Les technique d'évaluation de la qualité d'image sont nécessaires pour : l'optimisation, où on maximise la qualité à un coût donné. La surveillance de la qualité dans des applications en temps réel.

Dans ce chapitre nous étudions deux manières de mesure de la qualité des images : subjective et objective.

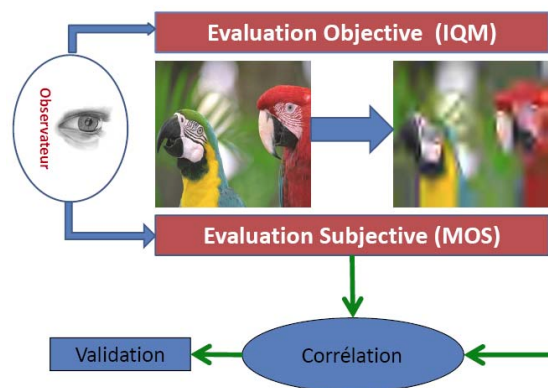


FIGURE 1.1 – Evaluation de la qualité d'image.

1.2 Qualité des images

La qualité d'une image est une caractéristique liée à la mesure des dégradations perçues à travers celle-ci et à la perception visuelle humaine [5].

La qualité d'une image est très liée à la perception, et parmi les nombreux critères entrant dans le jugement de la qualité d'une image, nous pouvons citer [5] :

- **Clarté** : sensation visuelle selon laquelle une partie de l'image paraît émettre plus ou moins de lumière.
- **Teinte** : précision des couleurs.
- **Saturation** : niveau de coloration (couleur vive, pâle etc).
- **Contraste** : différence dans l'intensité ou la teinte des objets qui permet de les rendre distinguables.
- **Distorsions** : ensemble d'aberrations affectant l'image, et dégradant son rendu visuel.

L'image numérique peut subir plusieurs dégradations ou distorsions, nous allons présenter les différents types de dégradation et donner des exemples [25] :

1.2.1 Effets de bloc

La principale source d'erreur lors de la compression JPEG est ce qu'on appelle l'effet de blocs. Visuellement, cette distorsion se manifeste généralement au niveau des frontières entre blocs et apparaît comme des contours verticaux et horizontaux dont la visibilité dépend fortement de la distribution spatiale du signal image. En effet, tous les blocs sont encodés indépendamment les uns des autres. Il peut donc arriver qu'à la frontière entre deux blocs, il y ait une discontinuité facilement perceptible par l'œil humain.



FIGURE 1.2 – Exemple d'effet de bloc.

1.2.2 Effet d'oscillations parasites (ringing effect)

Cette dégradation est due en général à l'étape de quantification ou de décimation des coefficients hautes fréquences. Elle se manifeste sous forme d'oscillations au voisinage des régions à fort contraste et est souvent définie comme un bruit au tour de ces régions. Ce sont les ondelettes dont le support croise le bord d'un objet qui créent ce type d'artefact.



FIGURE 1.3 – Exemple d'effet d'oscillations parasites.

1.2.3 Flou

Effet esthétique qui donne à voir un contour imprécis. Ce type d'erreur qui apparaît principalement dans la compression JPEG et JPEG2000 d'où la perte de netteté de l'image. Cela se caractérise par une image plus floue, dont les bords des objets sont plus diffus.



FIGURE 1.4 – Image dégradée avec le flou.

En plus, le flou apparaît dans l'image de différentes manières (mouvement, perte d'informations et le flou volontaire) :



FIGURE 1.5 – Exemple du flou volontaire.

1.2.4 Bruit

On peut définir le bruit comme étant une dégradation dans l'image, provoquée par une perturbation externe. Généralement, on peut savoir les types d'erreurs à

attendre, et donc le type de bruit sur l'image. On va présenter deux types différents de bruit :

- **Bruit poivre et sel** : on l'appelle aussi le bruit impulsionnel, le bruit de grenaille, ou le bruit binaire. Cette dégradation peut être causée par de fortes perturbations soudaines dans le signal d'image. Son apparence est éparpillée au hasard en pixels blancs ou noirs (ou les deux) sur l'image, Par contre, ce bruit est obtenu en ajoutant n pixels blancs et n pixels noirs aléatoirement dans une image.



FIGURE 1.6 – Exemple du Bruit poivre et sel.

- **Bruit Gaussien** : le bruit gaussien est une forme idéalisée de bruit blanc, qui est provoqué par des fluctuations aléatoires dans le signal. Nous pouvons observer le bruit blanc en regardant une télévision qui est mal accordée à un canal particulier. Si l'image est représentée comme I , et le bruit gaussien par N , alors nous pouvons modéliser une image bruyante en additionnant simplement les deux :

$$F = I + N \quad (1.1)$$



FIGURE 1.7 – Image dégradée avec la distorsion Bruit Gaussien.

1.3 Évaluation subjective

L'évaluation subjective est considérée comme la méthode la plus fiable pour mesurer la qualité réelle et attire directement les observateurs humains. Ce dernier

est guidé pour juger de la qualité de l'image qui lui est présentée sur la base d'une grille d'évaluation de multiples niveaux d'appréciation. A l'issue de ces tests, un score subjectif appelé MOS a été obtenu. Lors de la réalisation de ces tests, il est important de porter une attention particulière à certains facteurs qui peuvent influencer le jugement des participants [1].

1.3.1 Facteurs influant

L'évaluation subjective nécessite de réunir certains dispositifs importants pour obtenir une évaluation la plus fiable possible. Les facteurs influant sur l'évaluation des observateurs humains sont [2] :

- **Distance d'observation** : Doit être réglée sur 4 à 6 fois la hauteur de l'image, et cette distance doit être maintenue tout au long du test.

- **Écran** : Le support d'affichage utilisé comme écran doit être calibré pour qu'il fonctionne dans les meilleures conditions. Si cet étalonnage n'est pas effectué, la couleur affichée peut être différente du stimulus d'origine.

- **Conditions de visualisation** : L'environnement et l'éclairage de la salle affecte effectivement la perception des stimuli (séquence de test). Un éclairage élevé risque de troubler l'observateur et ainsi perturber sa perception. En revanche, la couleur de fond de la salle affecte l'apparence de certaines couleurs.

- **Le choix des images** : Un panel d'images de contenus visuels assez variés et le plus représentatif possible des scènes, liées à l'application visée, doit être utilisé.

- **Facteurs psychologiques** : Les tests doivent commencer par une session d'introduction qui permet au surveillant de mieux comprendre le test. Il est également préférable de passer de nombreux tests courts et variés (environ 10 minutes) plutôt que de forcer l'observateur à des sessions longues et fatigantes (environ 30 minutes) qui peuvent conduire à un jugement rapide et perverti.

- **Observateurs** : Il est préférable d'avoir autant d'observateurs que possible (16 observateurs). Ils peuvent être experts ou débutants, et avant chaque séance, les observateurs seront sélectionnés en fonction de leur acuité visuelle.

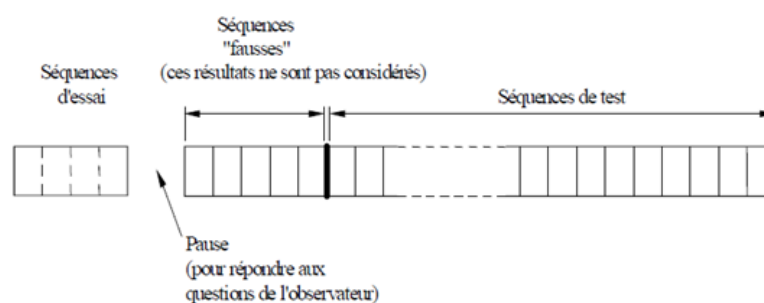


FIGURE 1.8 – Présentation des séquences d'évaluation.

1.3.2 Protocoles

L'union internationale des télécommunications (UIT) propose trois différentes méthodes d'évaluation subjective [3] :

1.3.2.1 Méthode à simple stimulus

Dans cette méthode l'observateur doit juger la qualité d'image sans aucune information relative à l'image d'origine [6]. Les différentes images à évaluer sont affichées une à une avec un temps mort entre chaque image permettant d'évaluer la qualité de l'image qui vient d'être montrée. Dans ce type de protocole, un seul stimulus peut être jugé à la fois, son nom est donc «protocole de stimulus unique» [3].



FIGURE 1.9 – Méthode à simple stimulus.

Excellente
Bonne
Assez bonne
Médiocre
Mauvaise

TABLE 1.1 – Échelle représentative d'un protocole à simple stimulus.

On peut aussi utiliser une échelle de qualité d'image comportant 5, 6, 7 ou 100 niveaux de qualité d'image [3].

1.3.2.2 Méthode à double stimulus

Le test à double stimuli oblige l'observateur à juger de la similitude entre deux images. Il lui est demandé d'évaluer la fidélité de l'image par rapport à sa version originale. La similarité entre les deux images représente la fidélité [3].

Habituellement, l'image d'origine est affichée en premier, puis l'écran gris est affiché, puis l'image à évaluer est affichée, puis le deuxième écran gris est affiché afin que le spectateur puisse l'évaluer. Cependant, il est important d'appliquer le même temps de présentation aux deux stimuli [3].

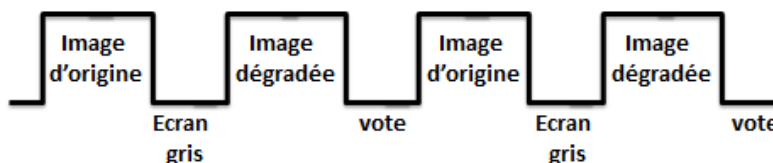


FIGURE 1.10 – Méthode à double stimulus.

Le protocole utilise également différentes échelles, dont chacune correspond au degré de dégradation par rapport à l'image d'origine. Le tableau suivant représente une échelle à 5 niveaux et le degré de dégradation correspond à chaque niveau [3] :

Qualité	Niveau de dégradation
Excellente	Imperceptible
Bonne	Très peu perceptible et non gênante
Passable	Peu perceptible et peu gênante
Médiocre	Gênante
Mauvaise	Très gênante

TABLE 1.2 – Échelle représentative d'un protocole double stimulus.

1.3.2.3 Méthode comparative

L'objectif principal de cette méthode est de comparer deux ou plusieurs stimuli [1]. Cette méthode consiste à noter la différence entre deux versions d'une même image à l'aide d'une échelle : Deux images sont affichées en même temps et l'observateur juge quelle est la meilleure qualité [4].

1.3.3 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 [2].

a) MOS "Mean Opinion Score" : Une fois tous les tests subjectifs terminés, la note finale de chaque image appelée MOS peut être obtenue en combinant les différentes notes données par les différents observateurs. Le score représente la moyenne des scores attribués par l'observateur à chaque image. Il est donné par la relation suivante [2] :

$$MOS_{jk} = \frac{1}{N} \sum_{i=1}^N Note_{ijk} \quad (1.2)$$

Où N : Le nombre d'observateur.

$Note_{ijk}$: La note attribuée par l'observateur i à l'image originale j dégradée avec la dégradation k .

b) 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 [2] :

$$[MOS(i) - e_j, MOS(i) + e_j] \quad (1.3)$$

Où

$$e_j = 1.95 \frac{\sigma_j}{\sqrt{N}} \quad (1.4)$$

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

$$\sigma_j = \sqrt{\frac{\sum_{k=1}^N (Note_i(K) - MOS(K))^2}{N - 1}} \quad (1.5)$$

c) **DMOS "Différence de Score Optimal Moyen"** : La mesure moderne de la qualité d'image n'applique pas directement les résultats de l'évaluation, mais utilise la différence de qualité entre les images pour définir DMOS comme la différence entre l'image de référence et le score de qualité d'origine du test. Le DMOS est calculé à l'aide de l'équation suivante [5] :

$$DMOS_{ijk} = Note_{ij} - Note_{ijk} \quad (1.6)$$

Où $Note_{ijk}$: La note attribuée par l'observateur i à l'image originale j dégradée avec la dégradation k .

$Note_{ij}$: La note attribuée par l'observateur i à l'image originale j .

- **La différence entre MOS et DMOS** : MOS est la moyenne des scores donnés par l'observateur au signal dégradé, DMOS est la moyenne de la différence entre les scores donnés par le signal d'origine et le signal dégradé. La valeur de DMOS est inférieure à celle de MOS et reflète mieux la qualité du signal.

Enfin, un protocole de rejet des mauvais observateurs ayant donné un score anormal a été mis en place, pour rendre le score de qualité le plus fiable possible [5].

1.4 Evaluation objective

L'évaluation subjective de la qualité d'images reste rude et inconfortable à cause du temps et des moyens financiers [5], ils ne peuvent pas être incorporés dans des applications en temps réel. De plus, leurs résultats dépendent largement de l'état physique et émotionnel de l'observateur.

L'évaluation objective de la qualité d'image est dispensée par des algorithmes informatiques maximisant la corrélation de leurs résultats avec les résultats des tests subjectifs humains. Cette alternative automatique d'évaluation de la qualité est avantageuse pour l'industrie de l'image dans la mesure où elle comble les besoins que l'évaluation subjective ne peut atteindre en termes de temps d'évaluation, et coût financier [5].

1.4.1 Métriques pour la mesure de la qualité objective

Les métriques objectives de mesure de la qualité proposées dans la littérature sont divisées en trois catégories selon que l'image originale est disponible. Alors, on distingue les métriques avec référence complète, référence réduite et sans référence [1].

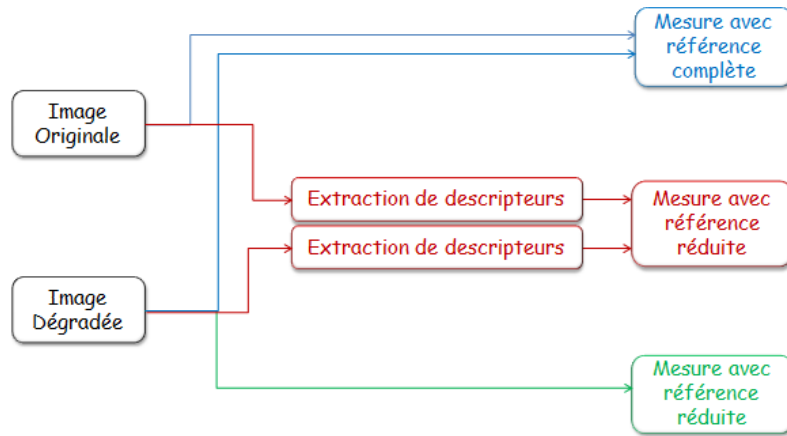


FIGURE 1.11 – Catégories d'image objective.

1.4.1.1 Métriques à référence complète

Dans lesquelles l'algorithme a un accès à une version parfaite de l'image avec lesquelles il peut comparer la version dégradée. La version parfaite vient généralement d'un dispositif de haute qualité d'acquisition, après elle est dégradée par la compression et les erreurs de transmission. Pour évaluer la quantité de distorsions introduites par la compression et la qualité de l'image en résultant. Nous explicitons dans ce qui suit plusieurs critères d'évaluation de qualité d'images présents dans la littérature [4] :

1) Métriques simples :

Ce type de critères traite l'image comme un signal purement numérique : il extrait les informations nécessaires pour obtenir un score de qualité de l'image, plutôt que sur la base de toute information liée à l'opération SVH (Système visuel humain), on trouve [4] :

- **MSE (Erreur quadratique moyenne) :** Il compare toujours l'image dégradée T avec l'original I pour déterminer sa similitude. Ce critère est le plus utilisé [2] :

$$MSE = \frac{1}{M \times N} \sum_{m=1}^M \sum_{n=1}^N (I(m, n) - T(m, n))^2 \quad (1.7)$$

Où $(M \times N)$ est la taille de l'image, et I_p et T_p sont respectivement les amplitudes des pixels sur les images originale et dégradée. Il est vraisemblable que l'œil humain compte beaucoup plus des erreurs à grandes amplitudes, ce qui favorise la mesure quadratique.

- **PSNR (Rapport crête signal sur bruit) :** Est une métrique utilisée pour évaluer la fidélité d'une image par rapport à une image de référence. Cette mesure permet de quantifier la distorsion qui existe entre deux images en utilisant la formule suivante [2] :

$$PSNR = 10 \log_{10} \left(\frac{(I_{max})^2}{MSE} \right) \quad (1.8)$$

Une valeur de PSNR supérieure à 34 dB indique une assez bonne qualité d'image, MSE et PSNR sont largement utilisés car leur implémentation est facile et leur complexité est faible. Mais plusieurs tests ont démontré qu'ils ne sont pas corrélés suffisamment avec la perception humaine [5].

- **SNR (Rapport signal sur bruit) :** On utilise parfois une autre variante du rapport signal sur bruit, qui relie le MSE à l'énergie moyenne de l'image [2] :

$$SNR = 10 \log_{10} \left(\frac{\frac{1}{N} \sum I^2}{MSE} \right) \quad (1.9)$$

Force est de constater que ces mesures de distorsion, même si elle est utilisée très fréquemment en traitement d'image pour quantifier la qualité, ne permet pas de prendre en compte la sensibilité du système visuel humain, et n'apparaît pas être performante [2].

2) Métriques avec prise en compte des caractéristiques SVH :

La plupart des modèles SVH (Système visuel humain) dans le traitement d'image utilisent trois attributs de base de la vision humaine [2] :

- La sensibilité à la fréquence détermine la sensibilité de l'œil à diverses fréquences spatiales.
- La sensibilité à la luminosité mesure l'effet du seuil de détection de bruit sur un fond constant.
- L'effet de masquage détermine la visibilité d'un signal en présence d'un autre signal.

- **SSIM (Indice de similarité structurelle) :** Est une mesure de la similitude entre deux images numériques par rapport à l'image originale, Il a été développé pour mesurer la qualité visuelle des images déformées. L'idée du SSIM est de mesurer la similitude structurelle entre deux images, au lieu de mesurer la différence entre les pixels comme le PSNR. L'hypothèse de base est que les yeux humains sont plus sensibles aux changements dans la structure de l'image [2].

La métrique SSIM est calculée sur plusieurs fenêtres de l'image. Nous désignons l'image d'origine et l'image déformée par x et y respectivement.

La similitude compare la luminosité, le contraste et la structure entre chaque paire de fenêtres :

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

$$\mu_x = \frac{1}{N} \sum_1^N x_i \quad (1.10)$$

N : le nombre de pixels de chaque fenêtre.

x_i : l'intensité d'un pixel.

-Le contraste pour chaque fenêtre est mesuré par :

$$\sigma_x = \left(\frac{1}{N-1} \sum_1^N (x_i - \mu_x)^2 \right)^{\frac{1}{2}} \quad (1.11)$$

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

$$\sigma_{xy} = \frac{1}{N-1} \sum_{i=1}^N (x_i - \mu_x)(y_i - \mu_y) \quad (1.12)$$

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 $c(x, y)$, compare 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 de $s(x, y)$. Elle est de la forme :

$$S(x, y) = f(l(x, y), c(x, y), s(x, y)) \quad (1.13)$$

Wang et Al ont proposé dans une fonction de comparaison de la luminance de la forme :

$$l(x, y) = \frac{2\mu_x\mu_y + (K_1L)^2}{\mu_x^2 + \mu_y^2 + (K_1L)^2} \quad (1.14)$$

K_1 est une constante de très faible valeur. La constante $(K_1L)^2$ évite l'instabilité de la fonction de comparaison lorsque $\mu_x^2 + \mu_y^2$ sont très proches de zéro. Dans le cas des images à niveau de gris, la constante est $L = 255$.

Notez que l'équation de $L(x, y)$ est conforme à la loi de Weber, qui est largement utilisée pour modéliser l'adaptation à la lumière en SVH. Selon la loi de Weber, le changement de luminosité est proportionnel à la luminosité de l'arrière-plan. En d'autres termes, SVH est sensible aux changements relatifs de luminosité entre deux signaux ou entre deux images [2].

- La fonction de comparaison de contraste prend une forme similaire :

$$c(x, y) = \frac{2\sigma_x\sigma_y + (K_2L)^2}{\sigma_x^2 + \sigma_y^2 + (K_2L)^2} \quad (1.15)$$

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. Notez que la corrélation entre deux vecteurs est une mesure simple et efficace de la similitude structurelle. Ensuite, la fonction de comparaison de structure est donnée par [2] :

$$s(x, y) = \frac{2\sigma_x\sigma_y + (K_2L)^2}{2\sigma_x + \sigma_y + (K_2L)^2} \quad (1.16)$$

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 :

$$SSIM(x, y) = l(x, y).c(x, y).s(x, y) \quad (1.17)$$

Encore une fois sous forme simplifiée :

$$SSIM(x, y) = \frac{2\mu_x\mu_y + (K_1L)^2}{\mu_x^2 + \mu_y^2 + (K_1L)^2} * \frac{2\sigma_x\sigma_y + (K_2L)^2}{\sigma_x^2 + \sigma_y^2 + (K_2L)^2} \quad (1.18)$$

L'indice de similarité est utilisé comme un choix approprié des constantes K1 et K2 pour généraliser l'évaluation à toute l'image.

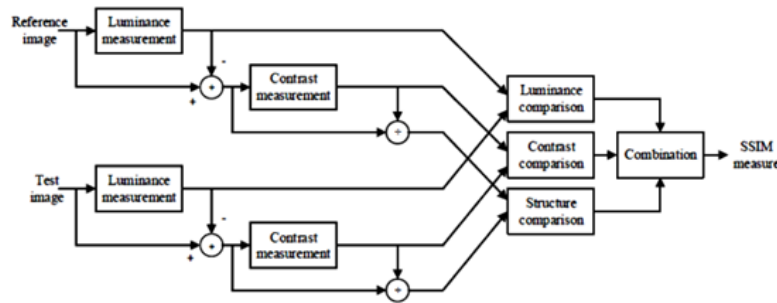


FIGURE 1.12 – Le schéma de l'algorithme SSIM.

1.4.1.2 Métriques à référence réduite

Les méthodes avec référence réduite est une mixtes entre les méthodes avec référence complète et les méthodes sans référence.

L'évaluation de la qualité avec référence réduite a pour but de mesurer la qualité d'une image dégradée en disposant d'informations partielles représentant l'image originale.

Habituellement, une extraction des caractéristiques de l'image dégradée est effectuée, ensuite une extraction des caractéristiques partielles de l'image originale est mise en oeuvre, afin de quantifier la différence entre les deux images [5].

Les métriques d'évaluation avec référence réduite sont souvent employées dans des applications de télécommunication, par exemple dans un système de visio-conférence temps réel. Ces métriques peuvent pister les dégradations affectant la qualité du

rendu visuel et contrôler les ressources liées à la transmission ; en effet, les caractéristiques partielles extraites de l'image au niveau de l'émetteur sont transmises au récepteur qui lui effectue deux opérations : extraction des caractéristiques de l'image dégradée reçue et analyse des caractéristiques de l'image originale pour effectuer l'évaluation de la qualité [5].

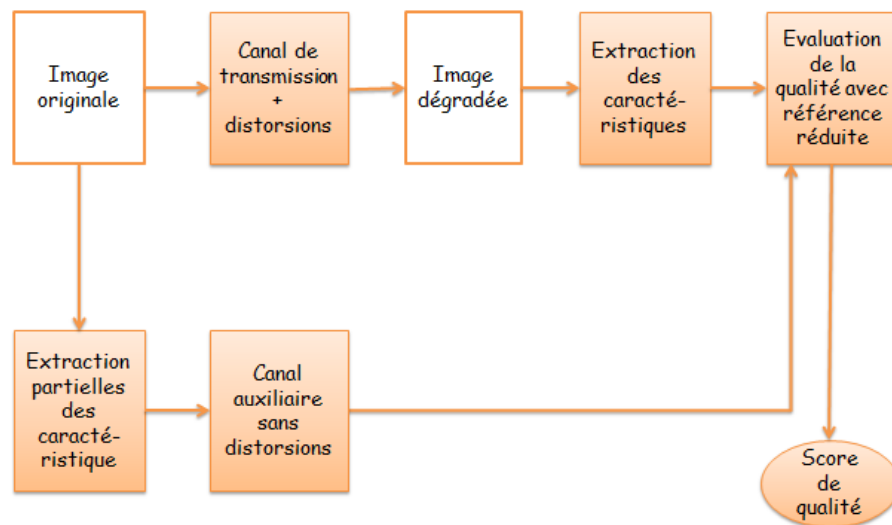


FIGURE 1.13 – Evaluation de la qualité d'image avec référence réduite.

Les caractéristiques partielles sont transmises via un canal auxiliaire avec un bas débit, afin de minimiser les erreurs de transmission, et de là les utiliser à des fins d'évaluation de la qualité d'images et de réglage des paramètres de transmission [5].

1.4.1.3 Métrique sans référence

L'évaluation de la qualité d'images sans référence est le type le plus important dans l'évaluation de la qualité parce qu'il est l'exemple idéal du processus de jugement humain.

En effet, leur domaine d'application est très vaste, Elles peuvent opérer dans des systèmes d'acquisition, de compression, de communication, d'affichage et rendu, d'impression, de restauration, d'analyse, et de tatouage. Par exemple l'ajustement dynamique de la qualité d'un moniteur ou et des paramètres d'algorithmes de traitement d'images peut s'effectuer à l'aide de ce type de métriques [5].

Les méthodes sans référence visent à atteindre les objectifs en utilisant uniquement des images de test pour estimer la qualité de l'image perceptuelle [5].

Ces méthodes sont nettement plus complexes que les méthodes avec référence et avec référence réduite en raison de l'absence d'informations sur les images originale, Dans la littérature la mise au point d'une métrique sans référence suit globalement une des 3 approches suivantes [5] :

1) Une approche basée sur des à priori liées aux dégradations affectant l'image :

L'algorithme évalue la qualité de l'image en fonction des dégradations prises indépendamment. L'impact de chaque dégradation est quantifié et est utilisé pour donner le score final de qualité.

On trouve des algorithmes spécifiques aux distorsions JPEG, JPEG2000, Flou [5].

2) Une approche basée sur les statistiques des scènes naturelles (NSS) :

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

• **La métrique BRISQUE (Blind/Referenceless image spatial quality evaluator) :**

Cette métrique [5] opère directement dans le domaine spatial de l'image, et utilise les statistiques de la luminance locale normalisée pour quantifier les éventuelles pertes du rendu naturel de l'image.

L'évaluation de la qualité se fait à partir de l'apprentissage d'une carte produite des caractéristiques et des scores de qualité en utilisant un module de régression (SVR). Les étapes de la métrique peuvent être exposées comme suit :

a) Calcul du coefficient de luminance normalisée. Ce dernier induit un effet de décorrélation entre les pixels et suit une distribution Gaussienne quand l'image est naturelle (sans distorsion) ou quand il y a peu de distorsion :

$$\hat{I}(i, j) = \frac{I(i, j) - \mu(i, j)}{\sigma(i, j) + 1} \quad (1.19)$$

Avec :

$$\mu(i, j) = \sum_{k=-K}^K \sum_{l=-L}^L \omega_{k,l} I(i+k, j+l) \quad (1.20)$$

$$\sigma(i, j) = \sum_{k=-K}^K \sum_{l=-L}^L \omega_{k,l} [I(i+k, j+l) + \mu(i, j)]^2 \quad (1.21)$$

Et μ et σ sont respectivement la moyenne et le contraste local. ω est une Gaussienne circulaire 2D (GGD).

b) Une distribution Gaussienne généralisée (GGD) permet de modéliser l'allure du coefficient MSCN, qui change en fonction de la distorsion. Les deux paramètres α (contrôle l'allure) et σ^2 (contrôle la variance) de la GGD sont les 2 premières caractéristiques de l'image.

$$f(x; \alpha, \sigma^2) = \frac{\alpha}{2\beta\Gamma(1/\alpha)} \exp\left(-\left(\frac{|x|}{\beta}\right)^\alpha\right) \quad (1.22)$$

Avec :

$$\beta = \sigma \frac{\Gamma(1/\alpha)}{\Gamma(3/\alpha)}$$

$$\Gamma(\alpha) = \int_0^\infty t^{\alpha-1} e^{-t} dt \quad \alpha > 0$$

Une distribution Gaussienne asymétrique généralisée (AGGD) permet de visualiser comment le produit des paramètres α et σ^2 varie quand il y a présence de distorsion suivant 4 orientations.

$$f(x; \alpha, \sigma_l^2, \sigma_r^2) = \begin{pmatrix} \frac{\alpha}{\beta_r + \beta_l \Gamma(1/\alpha)} \exp\left(-\left(\frac{|x|}{\beta_l}\right)^\alpha\right) & \forall x \leq 0 \\ \frac{\alpha}{\beta_r + \beta_l \Gamma(1/\alpha)} \exp\left(-\left(\frac{|x|}{\beta_r}\right)^\alpha\right) & \forall x \geq 0 \end{pmatrix} \quad (1.23)$$

Avec :

$$\beta_l = \sigma_l \sqrt{\frac{1/\alpha}{3/\alpha}}, \quad \beta_r = \sigma_r \sqrt{\frac{1/\alpha}{3/\alpha}}$$

$$\eta = (\beta_r - \beta_l) \frac{\Gamma(2/\alpha)}{\Gamma(1/\alpha)} \text{ 4ème paramètre}$$

Les paramètres α (contrôle l'allure) et σ_l, σ_r (contrôlent l'étendue de la gaussienne sur chaque côté de la gaussienne).

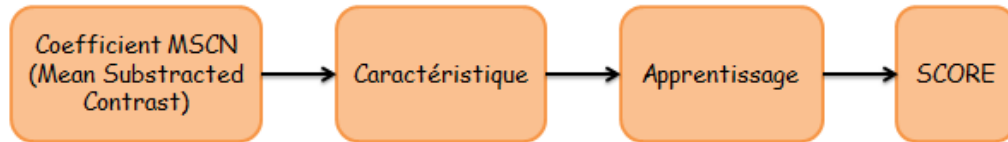


FIGURE 1.14 – Résumé du métrique BRISQUE.

Un apprentissage est effectué à partir des caractéristiques, et des scores de qualité subjectifs en utilisant un module à régression de telle façon que lorsque une image dégradée est présentée, le module à régression (SVR) entraîné, fournit la qualité de cette image.

3) Une approche basée sur l'apprentissage :

Pour chaque image, un vecteur de caractéristiques est extrait pour la représenter. Ayant accès aux scores subjectif de qualité, un processus d'apprentissage (comme les réseaux de neurones, SVM, ect) est utilisé pour prédire la qualité. Le vecteur d'entrée est le vecteur de caractéristique, le scalaire désiré est le score subjectif [5].

- **Approches neuronales :** Il est naturel que les réseaux de neurones s'intéressent à l'évaluation de la qualité d'image. Le processus d'évaluation subjective a une puissante dimension d'apprentissage. En fait, les humains peuvent facilement reconnaître, par exemple, des images floues sans utiliser l'image d'origine.

1.4.2 Mésures inspirées du Système Visuel Humain

Dans cette partie, on va voir, quelques approches de système visuel humain (SVH). Différentes caractéristiques du SVH ont été identifiées et utilisées dans le développement de nombreux algorithmes de traitement d'images et d'évaluation de la qualité.

1.4.2.1 Système visuel humain

Le SVH est un système complexe et n'est pas encore totalement maîtrisé. Malgré cela, il est toujours considéré comme un système de transcription d'informations en données que le cerveau peut utiliser (Figure 1.15). L'étape de conversion capture les informations reçues et les convertit en un signal que le cerveau peut décoder [1].

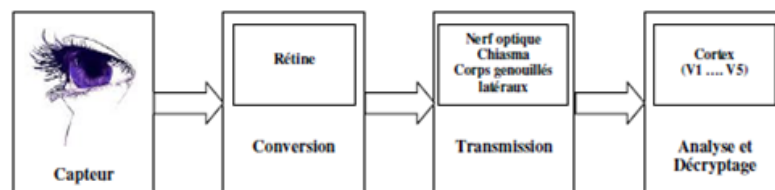


FIGURE 1.15 – Schéma simplifié de Système Visuel Humain.

Le rôle de l'œil est de convertir l'énergie lumineuse en signaux nerveux, puis de les transmettre au cortex visuel par le nerf optique et les membres inférieurs. Le cortex visuel décrypte et traite les informations reçues.

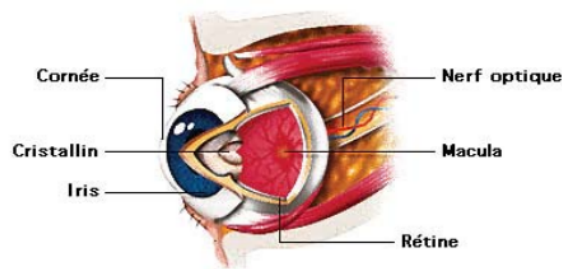


FIGURE 1.16 – Schéma d'œil humain.

Afin de mieux comprendre les flux d'informations et le fonctionnement de SVH, nous avons passé en revue le rôle important joué dans le processus de collecte et de traitement des informations visuelles. Donc, le SVH comporte quatre composantes en série décrites ci-dessous [1, 3] :

1) Capteur : La première composante est le système optique qui représente le capteur des signaux lumineux. Elle est constituée des éléments suivants [3] :

- **Cornée :** La cornée sert à préserver l'œil du milieu extérieur. Elle concentre les rayons lumineux vers la rétine.
- **Iris :** La variation de son ouverture permet de laisser passer l'intensité lumineuse adaptée.

- **Cristallin** : Il oriente le flux lumineux vers la rétine là où les cellules photoréceptrices sont disposées (les cônes et les bâtonnets).

2) **Conversion** : Les tâches de conversion sont effectuées au niveau de la rétine :

- **La rétine** : Il est utilisé pour convertir la lumière capturée par les cellules photoréceptrices en signaux neuraux (électriques) à travers le nerf optique. Avant d'atteindre le sol dans les cellules photoréceptrices de la rétine, la lumière traverse les autres couches de neurones. Chaque couche est composée de différentes cellules (cellules ganglionnaires, cellules bipolaires, cellules horizontales et cellules amacrine). La rétine possède essentiellement deux types de récepteurs [3] :

- **Cônes** : Ils sont disposés autour de la fovéa à l'extrémité du nerf optique. Ils permettent la vision des couleurs.
- **Bâtons** : Contrairement aux cônes, ils sont situés à l'extérieur et à l'intérieur de la fovéa. ils sont sensibles à une faible luminosité, mais pas sensibles à la couleur.

3) **transmission** : Les différentes informations recueillies par la rétine sont transmises par le nerf optique, qui transmet les informations au cortex visuel à travers le chiasme optique et le corps géniculé latéral [3].

- **Le chiasma** Les informations recueillies de l'oeil droit et de l'oeil gauche sont croisées, sans être pour autant mélangées, et le chiasma les transmet aux corps genouillés latéraux.

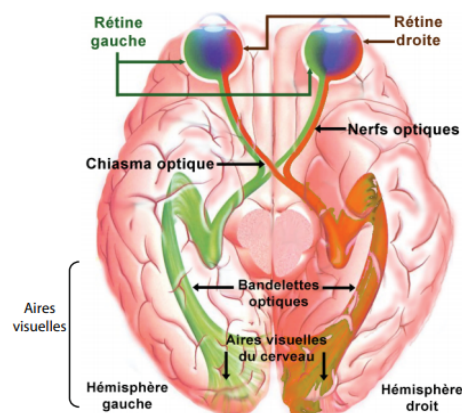


FIGURE 1.17 – Schéma des voies visuelles.

- **Les corps genouillés latéraux (CGL)** : Divers catégories de neurones dont chacun présente une rôle bien défini (M, K et P) constituent les corps genouillés latéraux. Ils sont le relais de l'information entre le chiasma et le cortex visuel primaire.

4) **Décryptage** : Une fois les signaux reçus, ils sont décryptés et analysés par le cortex visuel. Ce dernier se compose de plusieurs parties. Les signaux des CGL

sont reçus par la partie V1 qui effectue une sélection fréquentielle et directionnelle à ce niveau (sachant que l'aire V1 est la zone la plus connue jusqu'à présent). Cependant, les autres aires ont aussi chacune des tâches bien définies telle que le codage des couleurs, des formes, la direction du mouvement, etc. [3].

- **Rôle du cerveau :**

L'œil est un ensemble de capteurs recueillant des informations mais ne pouvant pas les interpréter, il transforme ces informations en messages nerveux transmis au cerveau. Les traitements de plus haut niveau sont réalisés dans le cerveau au niveau des aires visuelles que l'on trouve dans la partie arrière des deux hémisphères cérébraux (figure 1.17) [7]. Les informations dans le cerveau sont séparées en deux voies distinctes [2] :

- **Chemin ventral :** analyse les informations de forme et de couleur afin d'identifier les objets que vous voyez.
- **Chemin dorsal :** Analyse les informations relatives au mouvement et à la position dans l'espace d'une certaine manière pour permettre ou non le mouvement dans la direction de l'objet.

1.4.2.2 Caractéristiques psychophysiques de SVH

- **Fovéation :** La densité des cônes et des cellules ganglionnaires de la rétine est inégale. Ils atteignent un maximum dans la fovéa et diminuent rapidement avec l'éloignement de la fovéa. Le résultat normal est que tant que l'observateur fixe un point dans son environnement, la zone autour du point fixe sera résolue avec la résolution spatiale la plus élevée, et la résolution diminuera à mesure que la distance du point fixe diminuera. L'acuité visuelle haute résolution produite par l'observateur fixant la ligne de visée dans une certaine zone est appelée acuité visuelle fovéale, et l'acuité visuelle qui réduit progressivement la résolution est appelée acuité visuelle périphérique. La plupart des modèles d'évaluation de la qualité d'image peuvent utiliser la vision fovéale [2].

- **Phénomène d'adaptation à la lumière :** Le SVH fonctionne sur une certaine gamme de valeurs d'intensité lumineuse, des nuits de pleine lune aux jours ensoleillés, Il fait face à une gamme de couleurs aussi large grâce à un phénomène appelé adaptation à la lumière, qui régule le gain des neurones récepteurs dans la rétine en contrôlant la quantité de lumière pénétrant dans l'œil par la pupille et le mécanisme d'adaptation dans les cellules de rétine [2].

- **Fonction de sensibilité de contraste :** La fonction de sensibilité au contraste modélise les changements de sensibilité de SVH à différentes fréquences spatio-temporelles dans les stimuli visuels, donc pour déterminer s'il existe des «bords» entre des surfaces homogènes [2].

- **Effet de masquage :** Le masquage fait référence au fait que la présence d'un composant d'image réduit ou augmente la visibilité d'un autre composant d'image.

Par rapport au cas sans masque, un masque réduit généralement la visibilité du signal. Généralement, lorsque le masque et le signal ont le même contenu fréquentiel, l'effet de masque sera plus fort [2].

Par exemple, La figure suivante illustre les effets de masquage qui peuvent se produire sur l'image. La même quantité de bruit uniforme est ajoutée à la zone rectangulaire de l'image, qui est située en haut de l'image de gauche et en bas de l'image de droite. Le bruit est clairement visible sur l'image de gauche, tandis que sa visibilité est plus difficile sur l'image de droite [6].

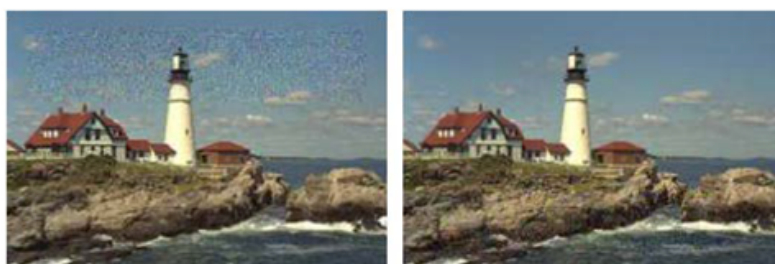


FIGURE 1.18 – Illustration de l'effet de masquage.

1.4.2.3 Méthodes basées sur SVH

Nous présentons quelques métriques d'évaluation de la qualité d'images qui sont basées sur le paradigme de sensibilité à l'erreur du SVH [2] :

- **Algorithme de Daly** : Il vise à calculer une probabilité de la carte de détection entre l'image référence et l'image déformée.
- **Algorithme de Lubin** : Il estime la possibilité de détecter des différences entre la version déformée et la version originale.
- **Métrique de Watson** : Elle est basée sur la transformée DCT, Elle est utilisée généralement dans la compression d'image.

1.5 Évaluation des performances des métriques de qualité

Les performances d'un critère objectif d'évaluation de qualité sont mesurées en comparant les notes de qualité données par ce critère, quand il est appliqué sur un groupe donné d'images, aux notes fournies par des observateurs humains lors de tests subjectifs d'évaluation de qualité [4].

La performance des mesures objectives est évaluée en ce qui concerne leur capacité d'estimer l'évaluation subjective de la qualité visuelle en trois aspects :

- **Précision de la prédiction** : la capacité de prévoir les estimations de qualité subjectives avec une erreur faible.

- **Monotonie de la prévision** : le degré auquel les prédictions du modèle sont conformes aux importances relatives d'estimations de la qualité subjective.
- **Cohérence de la prévision** : le degré auquel le modèle maintient la précision de la prédiction sur toute la gamme de la séquences de test d'image, à savoir, que sa réponse est robuste par rapport à une variété de déficience d'image.

Le procédé de comparaison entre notes subjectives et notes objectives est de plusieurs formes, dont [4] :

- **Coefficient de corrélation linéaire de Pearson (PCC) :**

Couramment employé, le terme corrélation est utilisé pour désigner la liaison ou la relation entre deux variables quelconques [5].

Le coefficient de corrélation linéaire exprime la dépendance linéaire entre les mesures objectives $Mobj$ et les notes subjectives moyennes MOS. Il est donné par :

$$PCC = \frac{1}{NbJK} \frac{\sum_{j=1}^J \sum_{k=1}^{K(j)} (MOS_{jk} - \overline{MOS})(Mobj_{jk} - \overline{Mobj})}{\sqrt{\sigma_{MOS}^2 \cdot \sigma_{Mobj}^2}} \quad (1.24)$$

Avec :

- $NbJK = \sum_{j=1}^J K(j)$ le nombre d'images testées (on suppose que J versions originales ont subi chacune K(j) dégradations différentes.
- $\overline{MOS} = \frac{1}{NbJK} \sum_{j=1}^J \sum_{k=1}^{K(j)} MOS_{jk}$ le MOS moyen.
- $\overline{Mobj} = \frac{1}{NbJK} \sum_{j=1}^J \sum_{k=1}^{K(j)} Mobj_{jk}$ la mesure objective moyenne.
- $\sigma_{MOS}^2 = \frac{1}{NbJK-1} \sum_{j=1}^J \sum_{k=1}^{K(j)} (MOS_{jk} - \overline{MOS})^2$ la variance des MOS.
- $\sigma_{Mobj}^2 = \frac{1}{NbJK-1} \sum_{j=1}^J \sum_{k=1}^{K(j)} (Mobj_{jk} - \overline{Mobj})^2$ la variance des mesures objectives.

La valeur du coefficient PCC est toujours comprise entre -1 et 1. Une valeur proche de 1 ou de -1 indique une bonne corrélation. Une valeur proche de 0 indique une mauvaise corrélation.

- **Coefficient de corrélation des rangs de Spearman (SCC) :**

Le coefficient de corrélation de rang décrit combien les mesures objectives et subjectives de qualité, après remise en ordre propre à chaque type, suivent un ordre similaire c'est-à-dire évoluent dans le même sens. Il est donné par la relation :

$$SCC = 1 - \frac{6 \cdot \sum_{j=1}^J \sum_{k=1}^{K(j)} d_{jk}^2}{NbJK^3 - NbJK} \quad (1.25)$$

Avec :

d_{jk} la différence de classement de l'image j ayant subi la dégradation k entre $Mobj$ et MOS .

Le coefficient de corrélation de rang est compris entre 1 et -1, la valeur 1 indique une forte corrélation positive, La valeur -1 indique une forte corrélation négative et la valeur 0 indique une absence de corrélation.

Une valeur de SCC proche de 1 montre que le classement des images par le critère objectif de qualité est similaire à celui fait par les observateurs humains.

- **Erreur de prédiction de la qualité (RMSE) :**

- L'écart-type de l'erreur de prédiction de la qualité est la mesure de la différence entre MOS et $Mobj_j$ pour une même image :

$$\sigma(MOS) = \sqrt{\frac{\sum_{j=1}^J \sum_{k=1}^{k(j)} (MOS_{jk} - Mobj_{jk})^2}{NbJK - 1}} \quad (1.26)$$

1.6 Conclusion

Nous avons consacré ce chapitre à présenter les méthodes d'évaluation de la qualité d'image qui sont divisent en deux grandes familles : les méthodes subjectives basées sur l'observateur humain, cette méthode est coûteuse au sens du matériel et du temps, les méthodes objectives où l'introduction de l'observateur humain n'est pas nécessaire, les résultats obtenus par ces méthodes doivent être conformes à l'évaluation subjective. Par la suite, nous avons présenté une définition au SVH, ses caractéristiques et quelques méthodes basées sur elle. En fin nous terminons ce chapitre par l'évaluation des performances des métriques de qualité des images.

Dans le chapitre suivant, nous présentons les concepts et les idées les plus importants de l'apprentissage automatique basé sur les réseaux de neurones artificiels pour l'évaluation de la qualité des images.

CHAPITRE 2

RÉSEAU DE NEURONES ARTIFICIELS

2.1 Introduction

Le domaine des réseaux de neurones n'est pas nouveau, car il est né dans les années 1940 (modèle de connexion synaptique de Hebb). Ce travail a conduit au modèle du perceptron dans les années 1960 (modèles principalement utilisés pour la reconnaissance de caractères). Mais jusqu'en 1986, avec la sortie de modèles de réseaux et d'algorithmes d'apprentissage, la recherche dans ce domaine s'est considérablement développée et son efficacité est suffisante pour résoudre des problèmes réels et complexes [9].

Les réseaux de neurones artificiels constituent une branche spécifique de la recherche en informatique et en neuro informatique. Différents types de réseaux de neurones artificiels offrent différentes possibilités de traitement des informations [10].

Le principe des réseaux de neurones se base sur une approche mathématique du fonctionnement du cerveau humain, dans ce chapitre, on étudiera le réseau de neurones artificiel, son modèle mathématique, ses architectures ainsi que ses domaines d'applications [17].

2.2 Réseau de neurone biologique

2.2.1 Le neurone biologique

Le cerveau humain contient un grand nombre de neurones fortement interconnectés (environ 100 milliards) (environ 10^{15} connexions), formant ainsi un réseau de neurones. Les neurones sont des cellules du système nerveux spécialisées dans la communication et le traitement de l'information. Le terme neurone a été proposé par Waldeyer en 1881 [17].

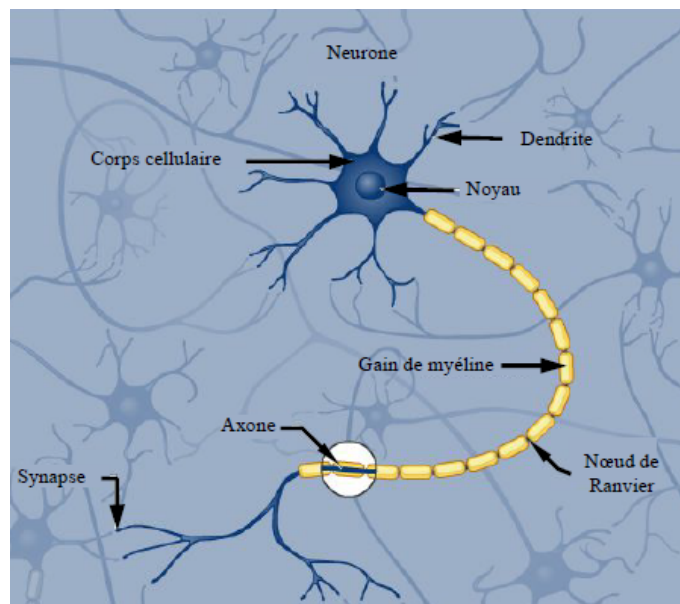


FIGURE 2.1 – Modèle d'un neurone biologique.

- **Corps cellulaire** : Habituellement appelé " Le Soma", il exécute la somme des informations qui l'atteignent, traite les mêmes informations et renvoie le résultat à d'autres neurones via des axones sous forme de signaux électriques.
- **Axone** : C'est un long prolongement fibreux du neurone, il fait la liaison entre les neurones et leurs conduit l'influx nerveux.
- **Dendrite** : Agit comme un capteur car il reçoit des informations de l'extérieur vers le corps cellulaire. Certaines dendrites peuvent avoir pour effet de favoriser la transmission d'informations dans l'axone, au contraire, d'autres dendrites ont pour effet inhibiteur d'inhiber la transmission de l'impulsion dans l'axone.
- **Synapses** : c'est une zone située entre deux neurones et assurant la transmission des informations de l'une à l'autre.
- **Influx Nerveux** : c'est une activité électrique qui parcourt les axones sous la forme d'une séquence de potentiel d'action, provoquant la libération des neurotransmetteurs au niveau des synapses. Ces potentiels d'action sont générés par des échanges d'ions entre l'intérieur et l'extérieur des neurones. La vitesse de propagation de l'influx nerveux est de 100m/s.

2.2.2 Fonctionnement du neurone biologique

Les dendrites reçoivent des informations sous forme de signaux électriques (impulsions neurales) d'autres neurones, elles seront envoyées au corps cellulaire (Soma), et plus précisément le noyau qui fera la somme des influx arrivant aux dendrites tout en affectant des poids dits "poids synaptiques" [17].

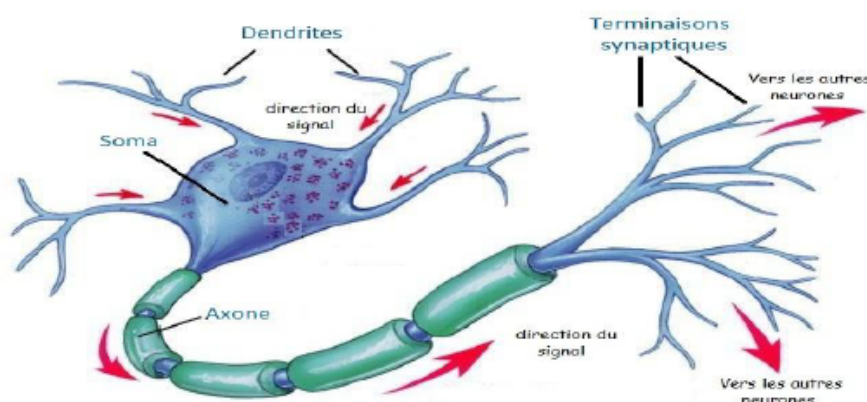


FIGURE 2.2 – Circulation de l'information dans le neurone biologiques.

Le résultat transite le long de l'axone jusqu'aux terminaisons synaptiques, et avec l'arrivé du signal, les vésicules synaptiques se fusionnent avec la membrane cellulaire pour libérer les neurotransmetteurs [17].

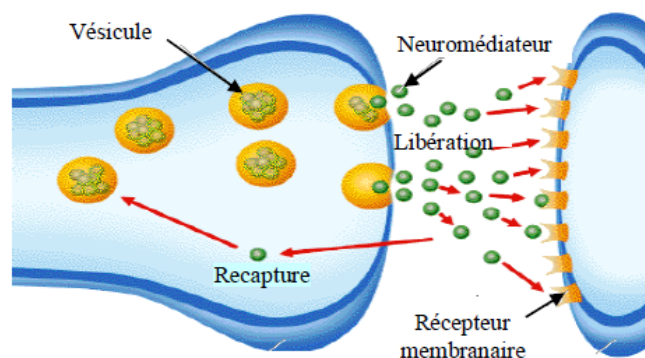


FIGURE 2.3 – Transmission de l'information entre les neurones biologiques.

2.3 Réseau de neurone artificiele

Le réseau de neurones artificiels est un système informatique basé sur la structure des neurones biologiques dans le cerveau humain, équipé d'ordinateurs dotés de fonctions d'intelligence artificielle. Les réseaux de neurones artificiels permettent aux ordinateurs de résoudre eux-mêmes les problèmes et d'améliorer leurs capacités globales. Selon la méthode d'intelligence artificielle utilisée, certains nécessitent une supervision préalable. Plus le problème à résoudre est complexe, plus le réseau neuronal artificiel doit avoir de couches. Chaque couche contient un grand nombre de neurones artificiels spécialisés [10].

2.3.1 Du neurone biologique au neurone formel

Si nous essayons d'expliquer la fonction précédente du neurone biologique d'une autre manière, nous pouvons dire que le noyau est la somme du produit de l'information obtenue par les dendrites et du poids attribué à la même information, puis le résultat de l'analyse du corps cellulaire, si ce dernier est supérieur au seuil,

il sera envoyé à l'axone, qui le subdivise pour alimenter d'autres neurones, ou pour attaquer des composants moteurs (tels que les muscles) ; Mc Culloch et Pitts ont été les premiers à mettre en œuvre un réseau de neurones, et ils ont utilisé des neurones formalisés. L'équation résume cette méthode mathématique [18] :

$$S = F \left(\sum_{i=1}^n W_i E_i \right) \quad (2.1)$$

où :

E_i : Représente l'entrée i du neurone.

W_i : Représente le poids attribué à l'entrée i .

F : Représente la fonction d'activation (Seuil) du neurone

$$\{1 \text{ si } F(x) > \theta; 0 \text{ sinon } \}, \text{ avec } \theta : \text{Seuil.}$$

S : Sortie du neurone.

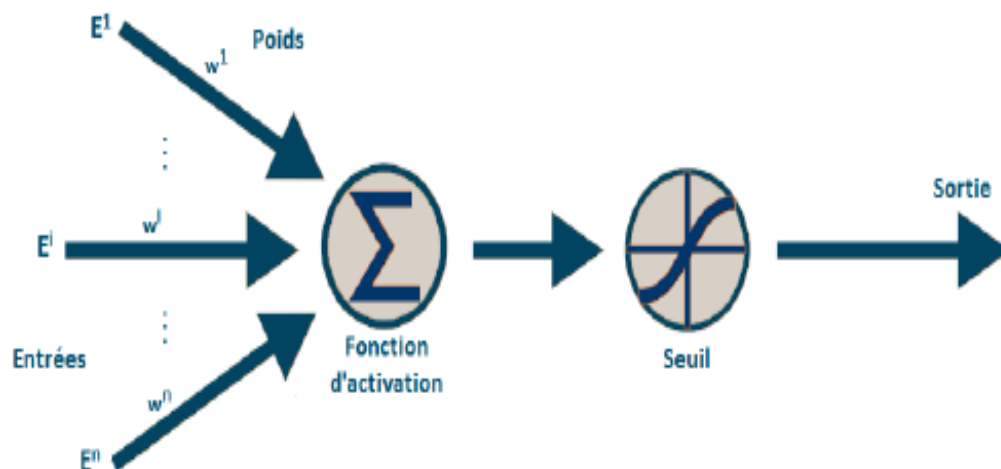


FIGURE 2.4 – Modèle d'un neurone formel.

On en déduit l'équivalence suivante :

Neurone biologique	Neurone formel
Synapses	Poids des connexions
Axones	Signal de sortie
Dendrites	Signal d'entrée
Noyau ou Somma	Fonction d'activation

TABLE 2.1 – Equivalence entre le neurone biologique et le neurone formel.

2.3.2 Historique

Les recherches sur les méthodes neuronales de traitement de l'information en vue de modéliser le comportement du cerveau humain ne sont pas récentes, OÙ en [15] :

- **1890** : La loi de fonctionnement pour l'apprentissage est présentée par W. James.
- **1943** : Warren Mc Culloch et Walter Pitts proposent le premier modèle du neurone formel.
- **1949** : Règle de Donald Hebb qui décrit que si les neurones d'une synapse sont activés d'une façon synchrone et répétée, la force de connexion synaptique est croissante.
- **1958** : Création du premier réseau de neurones par Rosenblatt, le "perceptron", ce dernier est inspiré du système visuel. Il permet d'apprendre et d'identifier des formes simples et aussi de calculer certaines fonctions logiques.
- **1969** : Marvin Minsky et Seymour Papert montrent les limites du perceptron, surtout à l'incapacité de résoudre des problèmes non linéairement séparables (exemple du XOR).
- Période noire des Réseaux de neurones (15 ans) et beaucoup de déception chez les amateurs de l'intelligence artificielle.
- **1975** : Werbos propose l'idée d'une possible utilisation d'une rétropropagation du gradient.
- **1982** : John Hopfield réactive l'intérêt de l'utilisation de ce domaine grâce à sa découverte sur l'utilisation des réseaux récurrents (feed-back) après la première classe du perceptron.
- **1985-1986** : Le Cun et Parker (1985) continuent leurs recherches du principe de Werbos (1975), mais les travaux de Rumelhart (1986) furent le vrai départ de l'apprentissage des réseaux de neurones multicouche avec la méthode de rétropropagation du gradient.

Depuis ce temps, le domaine des réseaux de neurones fournit constamment de nouvelles théories, de nouvelles structures et de nouveaux algorithmes.

2.3.3 Caractéristiques de RNA

Un réseau de neurone artificiel se compose de trois couches différentes : une couche d'entrée, une couche cachée et une couche de sortie.

- **Les entrées** : Chaque neurone possède plusieurs entrées, on note E_i les entrées du réseau, et à chacune d'entre elles on affecte un poids W_i , on y ajoute un coefficient dit "biais" W_0 supposé lié à une entrée $E_0 = 1$, la i^{eme} information qui parviendra au neurone est le produit $W_i \times E_i$.

Le neurone calculera la somme : $\sum_{i=1}^n W_i \times E_i$, celle-ci sera le potentiel du neurone, la sortie sera générée par une fonction d'activation, qui sera très importante car elle déterminera par la suite le fonctionnement du réseau [17].

- **La fonction d'activation** : Ou fonction de transfert, est une fonction qui doit renvoyer un réel proche de 1 quand les "bonnes" informations d'entrée sont données et un réel proche de 0 quand elles sont "mauvaises". On utilise généralement des fonctions à valeurs dans l'intervalle réel $[0;1]$. Quand le réel est proche de 1,

on dit que le neurone est "active" alors que quand le réel est proche de 0, on dit que le neurone est "inactive". Elle peut prendre de nombreuses formes : binaire, linéaire, sigmoïde, et bien d'autres encore [17]. La figure suivante illustre les fonctions d'activations les plus utilisées [15] :

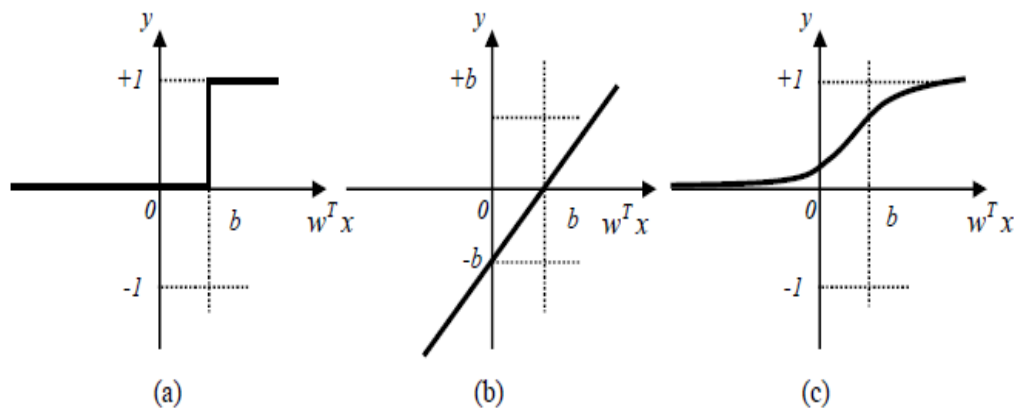


FIGURE 2.5 – Fonctions d'activations : (a) du neurone «seuil» ; (b) du neurone «linéaire», et (c) du neurone «sigmoïde».

La fonction linéaire est très simple, elle affecte directement son entrée à sa sortie :

$$y = s \quad (2.2)$$

Appliquée dans le contexte d'un neurone. Dans la figure précédente la sortie du neurone correspond à son niveau d'activation dont le passage à zéro se produit lorsque $w^T x = b$.

La fonction de transfert sigmoïde est quant à elle illustrée à la (figure 2.5) Son équation est donnée par :

$$y = \frac{1}{1 + \exp^{-s}} \quad (2.3)$$

Elle ressemble soit à la fonction seuil, soit à la fonction linéaire, selon que l'on est loin ou près de b , respectivement. La fonction seuil est très non linéaire car il y a une discontinuité lorsque $w^T x = b$. De son côté, la fonction linéaire est tout à fait linéaire. Elle ne comporte aucun changement de pente. La sigmoïde est un compromis intéressant entre les deux précédentes.

Notons finalement, que la fonction "tangente hyperbolique (tanh)" est une version symétrique de la sigmoïde [15].

- **La sortie** : peut être utiliser en tant que résultat final, ou pour alimenter d'autres neurones [17].

2.3.4 Architecture de RNA

Un réseau de neurones peut prendre différentes formes selon la finalité des données qu'il traite, sa complexité et la méthode de traitement des données. Les architectures ont leurs forces et faiblesses et peuvent être combinées pour optimiser les résultats. Le choix de l'architecture s'avère ainsi crucial et il est déterminé principalement par l'objectif [11].

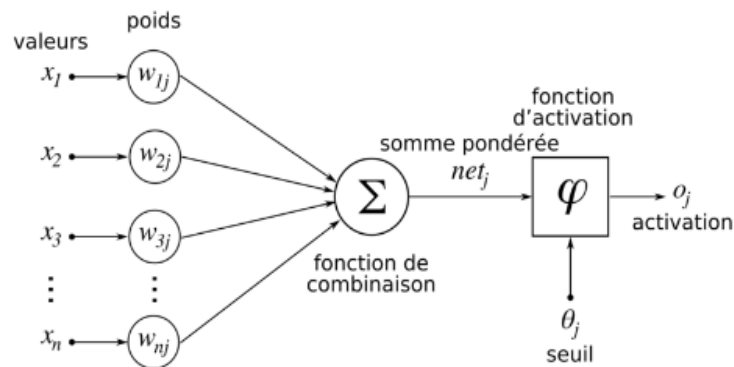


FIGURE 2.6 – L'architecture de réseau neurone.

Les réseaux de neurones classifiés selon l'architecture en deux grandes familles [17] :

1) Les réseaux Feed-forward :

Se sont des réseaux dans lesquels les informations se propagent successivement de couche en couche sans retour en arrière. On trouve :

- **Le perceptron monocouche** : Appelé aussi perceptron simple car c'est le réseau le plus simple en vue de l'architecture et au niveau de sa mise en oeuvre, il se compose d'une couche d'entrée et une couche de sortie. Ce réseau est capable de résoudre des problèmes linéairement séparables comme fonction logique 'OU' ou 'AND', tout les neurones de la couche d'entrée sont liés à tout les neurones de sortie, comme la figure suivante :

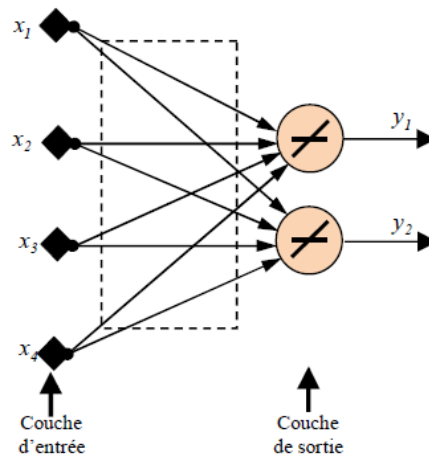


FIGURE 2.7 – Perceptron monocouche (simple).

- **Le perceptron multicouche :** Contient des couches cachées entre la couche d'entrée et celle de sortie, ce réseau résout des problèmes non-linéairement séparables comme la fonction logique 'XOR', et là aussi, chaque neurone des couches cachées est connecté à tout les neurones de la couche qui la précède et celle qui la suit.

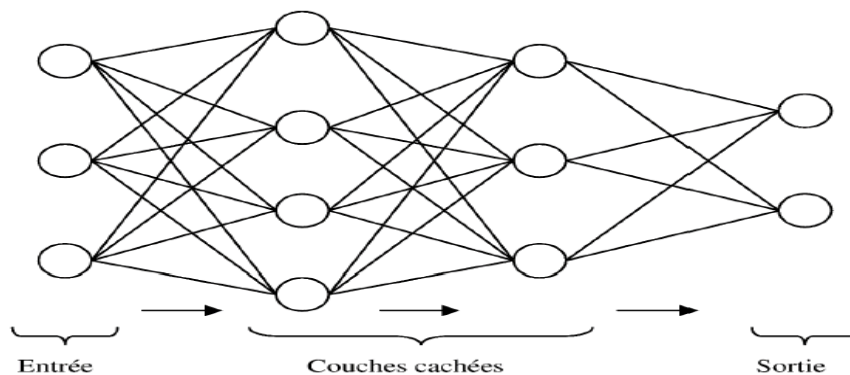


FIGURE 2.8 – Réseau neurone multicouche.

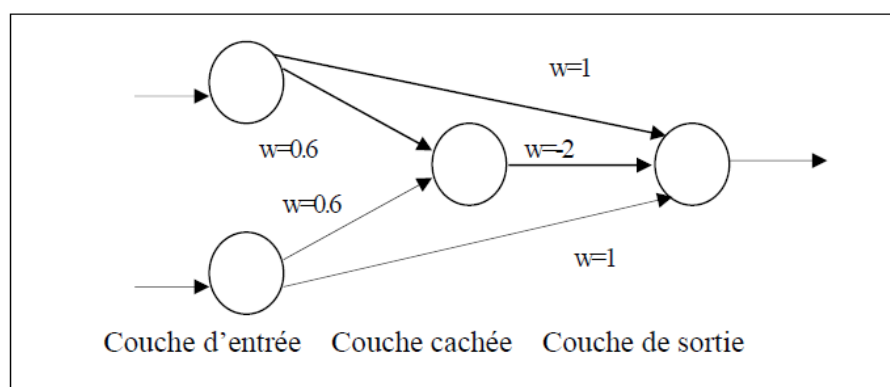


FIGURE 2.9 – Perceptron calculant la fonction XOR.

- **Le réseau à fonctions radiales :** possède la même architecture que le perceptron multicouche, la différence réside au niveau de la fonction d'activation des

neurones, car les réseaux à fonctions radiales utilisent les fonctions gaussiennes.

2) Les réseaux Feed-back :

appelés aussi "réseaux récurrents", sont des réseaux dans lesquels le retour en arrière est possible [17]. Son avantage réside dans leur capacité à prendre en compte les informations contextuelles après avoir traité à plusieurs reprises les mêmes informations. Ce réseau dynamique et autonome se compose d'une ou plusieurs couches [11].

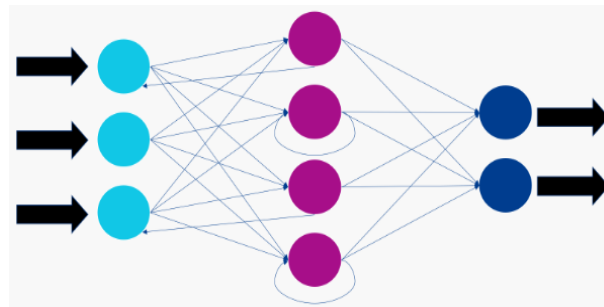


FIGURE 2.10 – Réseaux de neurones récurrents.

Ce type de réseau de neurone est utilisé par exemple en matière de reconnaissance vocale, de traduction et de reconnaissance d'écriture manuscrite [10]. On trouve :

- **Cartes auto-organisatrices de Kohonen** : C'est un réseau à apprentissage non supervisé, il est sous forme d'une grille dont chaque noeud représente un neurone associé à un vecteur de poids.

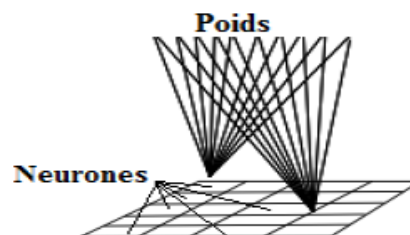


FIGURE 2.11 – Architecture d'un réseau de Kohonen.

- **Réseaux de Hopfield** : Ce sont des réseaux entièrement connectés. Dans ce type de réseau, chaque neurone est connecté à tous les autres neurones, et il n'y a pas de différence entre les neurones d'entrée et de sortie. Ils peuvent trouver des objets stockés sur la base de représentations partielles ou bruitées. L'application principale du réseau Hopfield est l'entrepôt de connaissances, et c'est la solution du problème d'optimisation. Le mode d'apprentissage utilisé à nouveau est le mode non supervisé.

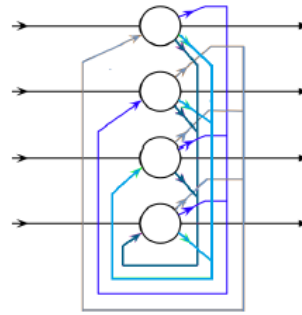


FIGURE 2.12 – Architecture d'un Réseau de Hopfield.

2.3.5 L'apprentissage

Le fait de définir une architecture spécifique d'un réseau est insuffisant pour résoudre un problème donné. Il faut déterminer une configuration de poids adaptée au problème, c'est le rôle du processus d'apprentissage.

L'apprentissage est une étape du développement du réseau de neurones. Dans cette étape, le comportement du réseau de neurones sera modifié jusqu'à ce que le comportement souhaité soit obtenu. Il peut être divisé en trois grandes classes [14] :

2.3.5.1 Apprentissage supervisé

L'apprentissage supervisé a pour but d'établir des règles de comportement à partir d'une base de données contenant des exemples de cas déjà étiquetés. La base de données est en principe un ensemble de couples entrées / sorties (X, Y) . Le but est d'apprendre à prédire pour toute nouvelle entrée X , la sortie Y [20].

Ensuite, la méthode d'apprentissage supervisé se définit comme l'objectif global de la construction à partir de la bibliothèque d'apprentissage, du classificateur ou de la fonction de classification. Une telle fonction permet d'identifier des attributs spécifiques à partir de la description de l'objet [20].

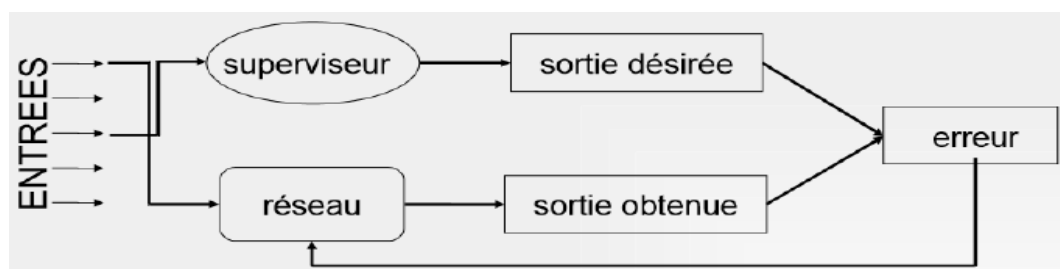


FIGURE 2.13 – Apprentissage supervisé.

Exemple : Classification des emails avec des emails déjà étiquetés [19].

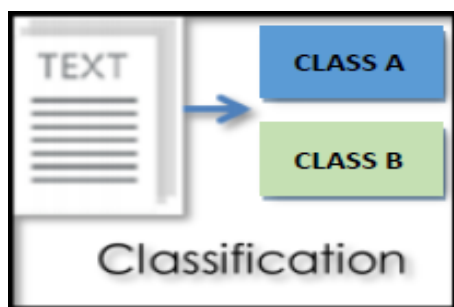


FIGURE 2.14 – Exemple du classification.

Il existe deux types de sous-problèmes en apprentissage supervisé [20] :

- **Classification** : lorsque la valeur cible à prédire est discrète.
- **Régression** : lorsque la valeur cible à prédire est continue.



FIGURE 2.15 – Illustration de la différence entre classification linéaire et régression linéaire.

2.3.5.2 Apprentissage non supervisée

L'apprentissage non-supervisé, appelé apprentissage par découverte à partir d'observations, ne demande aucun étiquetage préalable des données [27].

Le but est que le modèle réussisse à regrouper les observations disponibles en catégories par lui-même pour créer des classes d'objets selon une mesure de similarité entre les objets. Les classes sont construits de façon à maximiser la similarité intra-classes et à minimiser la similarité interclasses.

Il vise à concevoir un modèle structurant l'information. La différence ici est que les comportements (ou catégories ou encore les classes) des données d'apprentissage ne sont pas connus.

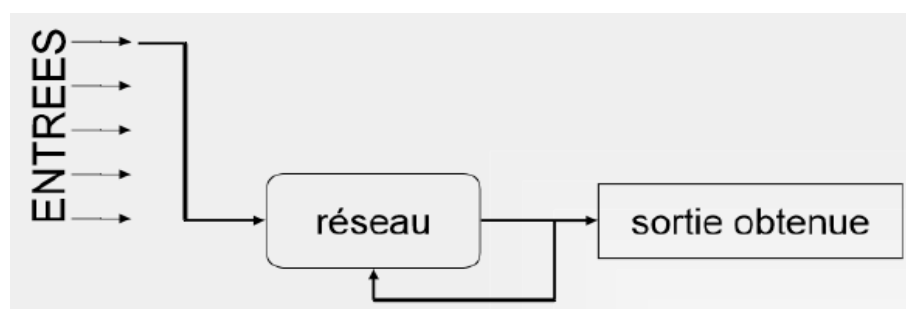


FIGURE 2.16 – apprentissage non supervisé.

L'apprentissage non supervisé est très souvent synonyme de clustering. Clustering (regroupement) construire des classes automatiquement en fonction des exemples disponible [20].

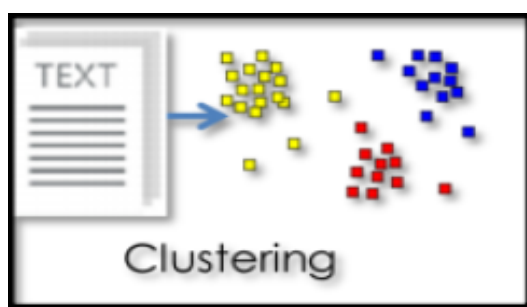


FIGURE 2.17 – Exemple Clustering.

2.3.5.3 Apprentissage Semi-Supervisé

Dans l'apprentissage supervisé, l'étiquetage des données peut être long et coûteux. Si les étiquettes sont limitées, il est possible d'utiliser des exemples non étiquetés pour améliorer l'apprentissage supervisé. Étant donné que la machine n'est pas entièrement supervisée, on emploie le terme «semi-supervisé ». En ce qui concerne l'apprentissage semi-supervisé, on utilise des exemples non étiquetés et une petite quantité de données étiquetées pour améliorer la précision de l'apprentissage [19].



FIGURE 2.18 – L'apprentissage Semi-supervisé.

Pour ces trois types d'apprentissage, vous pouvez également choisir traditionnellement entre les deux suivants [14] :

- **Apprentissage "hors ligne"** : toutes les données sont dans la base de données exemple apprentissage simultané.
- **Apprentissage "en ligne"** : les exemples sont présentés les uns après les autres dans un exemple. Quand ils sont disponibles.

2.4 Réseau de neurone convolutionnel

Les réseaux de neurones à convolution ont été créés dans les années 80. Désignés par l'acronyme CNN, de l'anglais Convolutional Neural Network. L'idée qui a mené à sa création est le fait d'essayer de faire ressortir certaines parties de l'image qui pourraient être intéressantes et les analyser, indépendamment de leur position dans l'image [22].

A ce jour sont les modèles les plus performants pour le traitement des images [13]. Utilisé pour le traitement d'informations complexes et très variées, il est envisageable de créer plusieurs réseaux de neurones distincts dédiés à traiter chacun une partie de l'information. Ces réseaux peuvent être imaginés comme une compilation d'un segment d'informations pour au final traiter l'ensemble de l'information [11]. ils comportent deux parties bien distinctes [12, 16].

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, 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 local. Au final, les cartes de convolutions sont mises à plat et concaténées en un vecteur de caractéristiques, appelé code CNN.

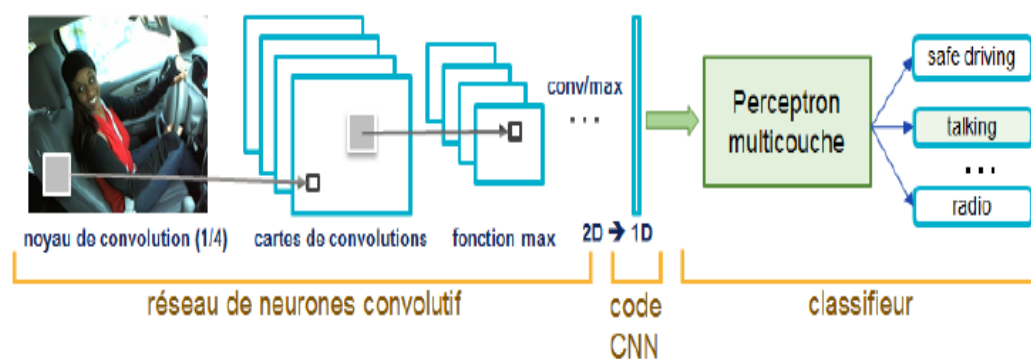


FIGURE 2.19 – Architecture standard d'un réseau de neurone convolutionnel.

Ce code CNN en sortie de la partie convolutive est ensuite branché 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 pour classer l'image. La sortie est une dernière couche comportant un neurone par catégorie. Les valeurs numériques obtenues sont généralement normalisées entre 0 et 1, de somme 1, pour produire une distribution de probabilité sur les catégories [12, 16].

2.4.1 Architecture de CNN

Les réseaux de neurones convolutionnels sont basés sur les perceptrons multi-couches, et inspirés du comportement du cortex visuel des vertébrés. Bien qu'efficaces pour le traitement d'images. Une architecture CNN est formée par un empilement de couches de traitement indépendantes [12, 16, 20, 21, 24] :

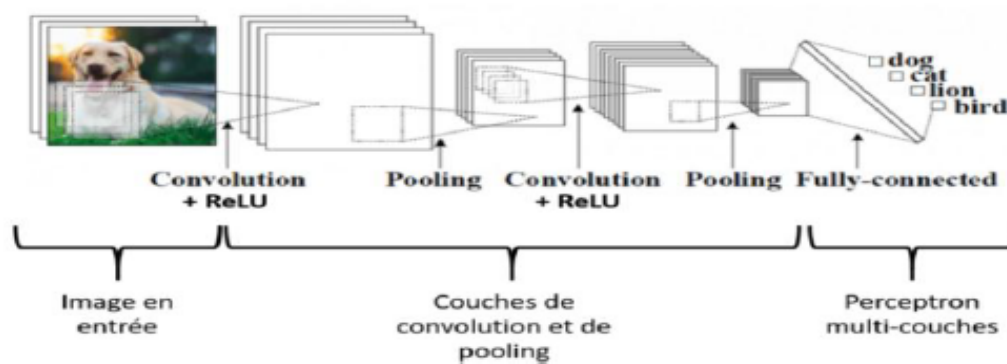


FIGURE 2.20 – Les Réseaux de Neurones Convolutifs (CNN)

2.4.1.1 Couche de convolution(CONV)

Est le bloc de construction de base d'un CNN qui traite les données d'un champ récepteur. Trois paramètres permettent de dimensionner le volume de la couche de convolution :

- **Profondeur de la couche** : nombre de noyaux de convolution (nombre de neurones associés à un même champ récepteur).
- **Le pas** : contrôle le chevauchement des champs récepteurs. Plus le pas est petit, plus les champs récepteurs se chevauchent et plus le volume de sortie sera grand.
- **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.

Si le pas et la marge appliquée à l'image d'entrée permettent de contrôler le nombre de champs récepteurs à gérer (surface de traitement), la profondeur permet d'avoir une notion de volume de sortie, et de la même manière qu'une image peut avoir un volume, si on prend une profondeur de 3 pour les trois canaux RGB d'une image couleur, la couche de convolution va également présenter en sortie une profondeur. C'est pour cela que l'on parle plutôt de "volume de sortie" et de "volume d'entrée", car l'entrée d'une couche de convolution peut être soit une image soit la sortie d'une autre couche de convolution.

La taille spatiale du volume de sortie peut être calculée en fonction de la taille du volume d'entrée W la surface de traitement K (nombre de champs récepteurs),

le pas S avec lequel ils sont appliqués, et la taille de la marge P . La formule pour calculer le nombre de neurones du volume de sortie est :

$$W_0 = \frac{WI - K + 2P}{S} + 1 \quad (2.4)$$

Si W_0 n'est pas entier, les neurones périphériques n'auront pas autant d'entrée que les autres. Il faudra donc augmenter la taille de la marge (pour recréer des entrées virtuelles).

Souvent, on considère un pas $S = 1$, on calcule donc la marge de la manière suivante :

$$p = \frac{K - 1}{2} \quad (2.5)$$

si on souhaite un volume de sortie de même taille que le volume d'entrée. Dans ce cas particulier la couche est dite "connectée localement".

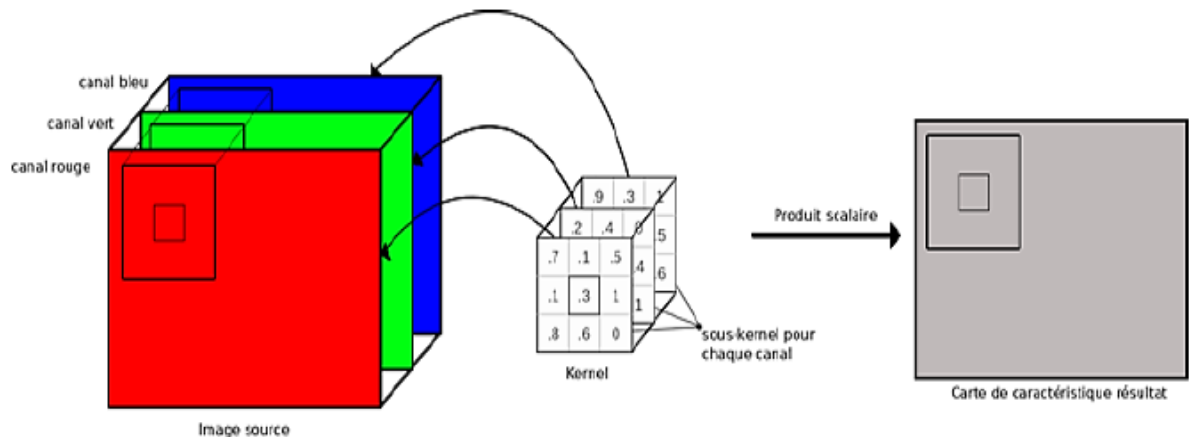


FIGURE 2.21 – Exemple d'image RGB en couche de convolution.

2.4.1.2 Couche de pooling (POOL)

Un autre concept important des CNNs est le pooling, ce qui est une forme de sous-échantillonnage de l'image. L'image d'entrée est découpée en une série de rectangles de n pixels de côté ne se chevauchant pas (pooling). Chaque rectangle peut être vu comme une tuile. Le signal en sortie de tuile est défini en fonction des valeurs prises par les différents pixels de la tuile.

Le pooling réduit la taille spatiale d'une image intermédiaire, réduisant ainsi la quantité de paramètres et de calcul dans le réseau. Il est donc fréquent d'insérer périodiquement une couche de pooling entre deux couches convolutives successives d'une architecture CNN pour contrôler l'overfitting (sur-apprentissage). L'opération de pooling crée aussi une forme d'invariance par translation.

La couche de pooling fonctionne indépendamment sur chaque tranche de profondeur de l'entrée et la redimensionne uniquement au niveau de la surface. La forme la plus courante est une couche de mise en commun avec des tuiles de taille 2×2 (largeur/hauteur) et comme valeur de sortie la valeur maximale en entrée. On parle

dans ce cas de "Max-Pool 2x2".

Il est possible d'utiliser d'autres fonctions de pooling que le maximum. On peut utiliser un "average pooling" (la sortie est la moyenne des valeurs du patch d'entrée), du "L2-norm pooling". Dans les faits, même si initialement l'average pooling était souvent utilisé il s'est avéré que le max-pooling était plus efficace car celui-ci augmente plus significativement l'importance des activations fortes. En d'autres circonstances, on pourra utiliser un pooling stochastique.

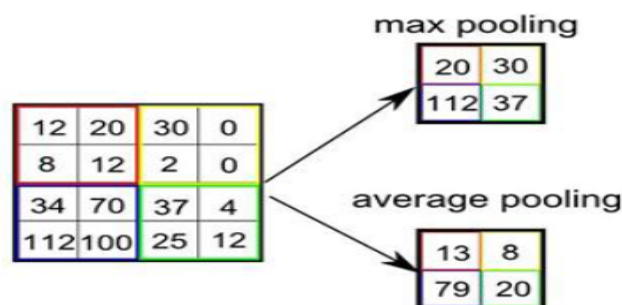


FIGURE 2.22 – Pooling avec un filtre 2x2 et un pas de 2.

Le pooling permet de gros gains en puissance de calcul. Cependant, en raison de la réduction agressive de la taille de la représentation (et donc de la perte d'information associée), la tendance actuelle est d'utiliser de petits filtres (type 2x2). Il est aussi possible d'éviter la couche de pooling mais cela implique un risque sur-apprentissage plus important.

2.4.1.3 Couches de correction (RELU)

Il est possible d'améliorer l'efficacité du traitement en intercalant entre les couches de traitement, une couche qui va opérer une fonction mathématique (fonction d'activation) sur les signaux de sortie.

La couche de correction ReLU remplace donc toutes les valeurs négatives reçues en entrées par des zéros. En ne modifiant pas les données positives, ReLU n'impacte pas les caractéristiques mises en évidence par la convolution, au contraire : elle les met davantage en évidence en creusant l'écart (valeurs négatives) « entre » deux caractéristiques. La fonction ReLU (Unités Rectifié linéaires) force les neurones à retourner des valeurs positives [16, 21] :

$$F(x) = \max(0, x) \quad (2.6)$$

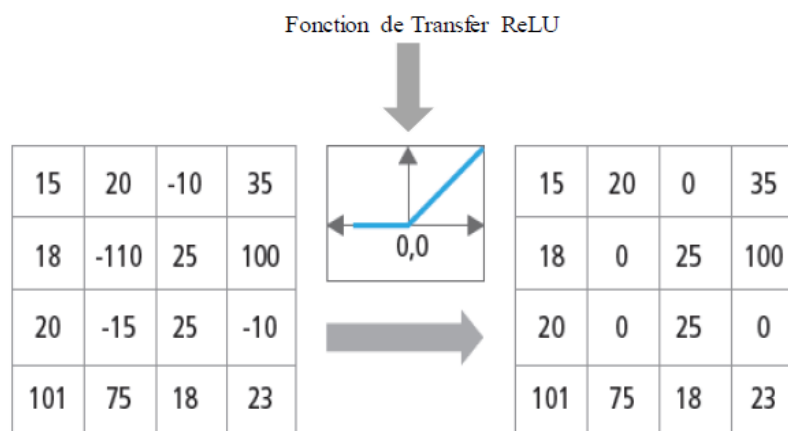


FIGURE 2.23 – Représentation graphique de la fonctionnalité Relu.

2.4.1.4 Couche entièrement connectée (FC)

Les cartes des caractéristiques de résultat de la couche de convolutions, pooling et ReLu fournissent au final un ensemble de caractéristiques données sous forme de carte 2D. Ces cartes sont concaténées en un vecteur de caractéristiques, appelé code CNN.

Ce code CNN en sortie de la partie convolutive est ensuite branché en entrée d'une deuxième partie, constituée d'une ou de plusieurs couches entièrement connectées (FC ou fully connected) qu'on peut assimiler au perceptron multicouches. Les neurones dans une couche entièrement connectée ont des connexions vers les sorties de la couche précédente.

Leurs fonctions d'activation peuvent donc être calculées avec une multiplication matricielle entre le vecteur d'entrée H et la matrice des poids F auquel on ajoute le vecteur des biais des neurones C :

$$A = FH + C \quad (2.7)$$

Le nombre de neurones de la dernière couche entièrement connectée est égal au nombre de classes. Le rôle de cette partie est donc de combiner les caractéristiques du code CNN pour classer une image [16, 21].

2.4.1.5 Couche de perte (LOSS)

La couche de perte spécifie comment l'entraînement du réseau pénalise l'écart entre le signal prévu et réel. Elle est normalement la dernière couche dans le réseau. Diverses fonctions de perte adaptées à différentes tâches peuvent y être utilisées. La fonction « Softmax » permet de calculer la distribution de probabilités sur les classes de sortie [16, 21].

2.5 Classification SVM

Le SVM (Support Vector Machine) est la méthode qui consiste à trouver le classifieur linéaire $f(x; \theta^*)$ (s'il existe) avec la marge maximale [26].

Le classificateur de marge maximale est simple, mais il ne peut pas être appliqué à la majorité des bases de données. Il peut résoudre des problèmes linéaires et non linéaires et fonctionne bien pour de nombreux problèmes pratiques [24].

L'idée de SVM très simple est que l'algorithme crée une ligne ou un hyperplan qui sépare les données en classes. Il peut être utilisé pour la classification binaire ou multiclass.

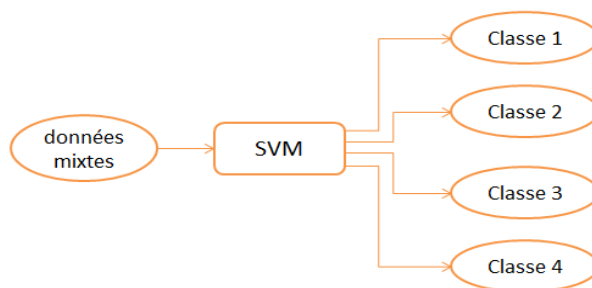


FIGURE 2.24 – Schéma de principe d'un SVM multiclass.

SVM est une extension du classifieur de vecteurs de support résultant de l'élargissement de l'espace des fonctions à l'aide de noyaux (Kernels) [24].

L'approche par noyau est simplement une approche informatique efficace pour tenir compte d'une limite non linéaire entre les classes. Le noyau est une fonction qui quantifie la similarité de deux observations. Le noyau peut être de n'importe quel degré. L'utilisation d'un noyau avec un degré supérieur à un conduit à une limite de décision plus flexible.

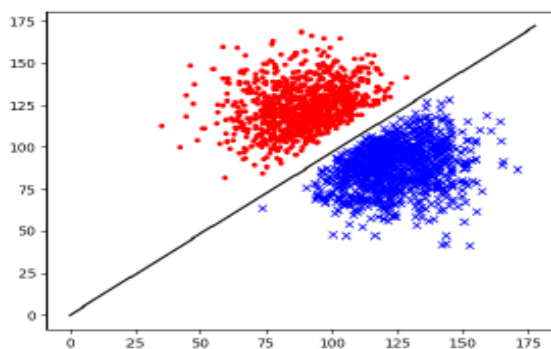


FIGURE 2.25 – Exemple d'un problème de discrimination à deux classes, avec une séparatrice linéaire.

La conception de l'architecture de classification SVM est très simple et nécessite principalement le choix du noyau. SVM trouve l'hyperplan laissant la plus grande

fraction possible de points de la même classe du même côté, tout en maximisant la distance entre l'une ou l'autre classe et l'hyperplan. Cet hyperplan minimise le risque d'exemples de classification erronés de l'ensemble de tests [24].

Pour un ensemble de points linéairement séparables, il existe une infinité d'hyperplans séparateurs.

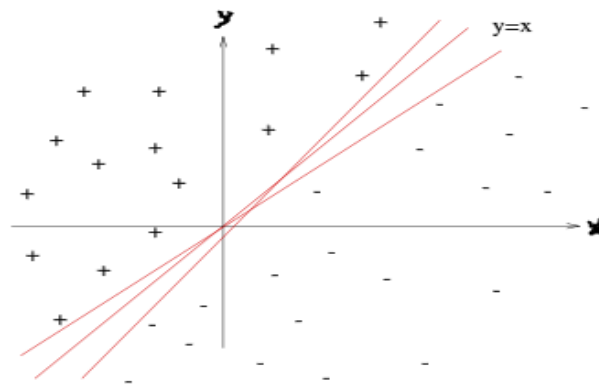


FIGURE 2.26 – Exemple d'hyperplans séparateurs.

- **SVR :**

Le SVR (Support Vector Regression) utilise le même principe que SVM, mais pour les problèmes de régression [29].

Le problème de la régression est de trouver une fonction qui rapproche le mappage d'un domaine d'entrée à des nombres réels sur la base d'un échantillon d'apprentissage.

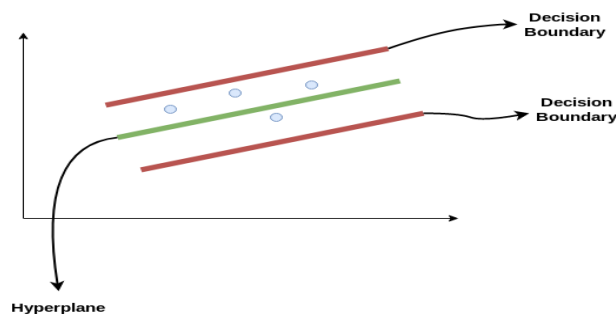


FIGURE 2.27 – Exemple d'hyperplans séparateurs.

Ces deux lignes rouges sont la limite de décision et la ligne verte est l'hyperplan. Donc l'objectif, est de considérer essentiellement les points qui se trouvent à l'intérieur de la ligne de démarcation de décision. Notre meilleure ligne d'ajustement est l'hyperplan qui a un nombre maximum de points.

Le SVR utiliser des noyaux linéaires, polynomiaux et RBF [31] :

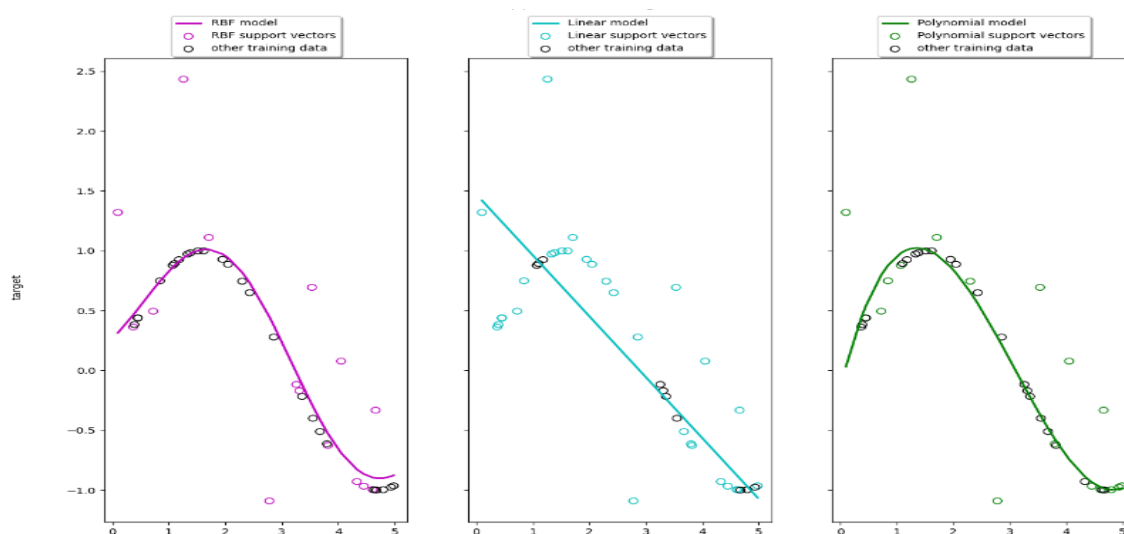


FIGURE 2.28 – noyaux linéaires et non linéaires de SVR.

2.6 Domaines d’application des RNA

Aujourd’hui, les réseaux de neurones artificiels ont de nombreuses applications dans de nombreux domaines [15] :

- **Traitement d’images** : reconnaissance de caractères et de signatures, compression d’image, reconnaissance de forme, cryptage, classification, etc.
- **Traitement du signal** : filtrage, classification, reconnaissance de source, traitement de la parole, etc.
- **Contrôle** : contrôle de processus, diagnostic, contrôle qualité, contrôle de robot, systèmes de guidage automatique pour voitures et avions, etc.
- **Défense** : guidage de missile, suivi de cible, reconnaissance faciale, radar, sonar, lidar, compression de données, suppression du bruit, etc.
- **Optimisation** : planification, allocation des ressources, gestion et finances..
- **Simulation** : simulation de vol, simulation de boîte noire, prévision météo, copie de modèle..

Les applications les plus importantes des réseaux de neurones artificiels d’apprentissage automatique incluent des exemples bien connus d’Alexa, Siri et l’assistant vocal de Google [10].

2.7 Avantages et inconvénients des RNAs

La raison pour laquelle les gens s’intéressent aux réseaux de neurones est qu’ils ont des propriétés intéressantes. Ils ont également de nombreuses limites ou d’inconvénients [18].

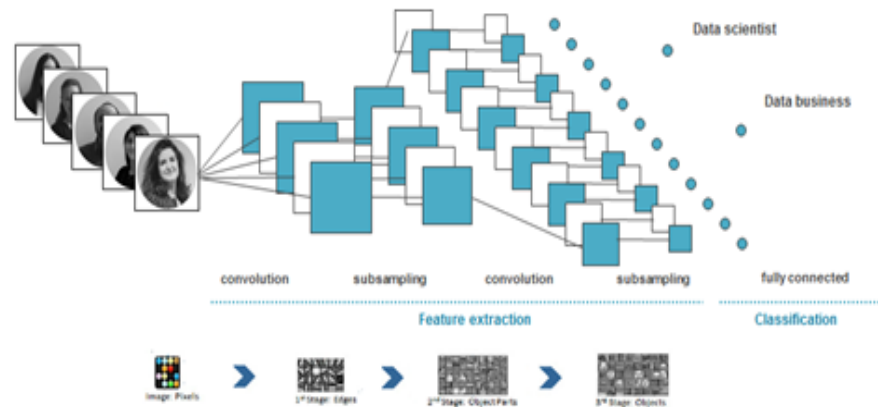


FIGURE 2.29 – Réseau neurone classique pour reconnaissance d'image.

- **Avantages :**

- **L'exploitation de connaissances empiriques :** L'apprentissage à partir d'exemples se fait d'une façon assez simple et permet d'obtenir de bons résultats par rapport aux autres techniques d'apprentissage.
- **La robustesse :** Dans le réseau neuronal, la mémoire est distribuée, ce qui correspond à la carte d'activation neurale. La carte est dans une certaine mesure un encodage de faits stockés. L'avantage attribué à ces réseaux réside dans l'avantage de aux pannes , car la perte d'éléments ne correspond pas à la perte de faits mémorisés. Par conséquent, même si l'appareil tombe en panne, le réseau peut fonctionner normalement. De plus, même si le bruit d'entrée est important, de nombreux modèles de réseau peuvent fournir de bons résultats.
- **La prise en compte de la non-linéarité et du temps :** L'avantage des réseaux de neurones artificiels est qu'ils peuvent prendre en compte les non-linéarités. Certains réseaux peuvent également prendre en compte l'aspect temporel (réseau récurrent).

- **Inconvénients :**

- **La difficulté de choix de l'architecture et des paramètres :** Pour un problème donné, il n'y a pas de moyen automatique de choisir la meilleure architecture. Il est très difficile de trouver la topologie de réseau correcte et les paramètres corrects pour ajuster l'algorithme d'apprentissage. L'évolution du processus d'apprentissage est largement affectée par ces deux facteurs (architecture du réseau et paramètres d'ajustement), et dépend dans une large mesure du type de problème traité. Le simple fait de changer la base d'apprentissage utilisée peut nous obliger à reconfigurer l'ensemble du réseau.
- **Le problème d'initialisation et de codage :** Les algorithmes d'apprentissage connexionniste reposent généralement fortement sur l'état initial du réseau (initialisation aléatoire des poids) et la configuration de la bibliothèque

d'apprentissage. Un mauvais choix des poids utilisés pour initialiser le réseau, la méthode de codage des données, voire l'ordre des données, peut empêcher l'apprentissage ou faire converger le réseau vers une bonne solution.

2.8 Conclusion

Nous avons présenté dans ce chapitre des notions de base sur le neurone biologique et le neurone formel, ceci dans le but de bien connaître les caractéristiques d'un neurone qui nous intéresse, de même le principe de son fonctionnement. Par la suite nous avons exposé les différents architectures et les différents modes d'apprentissage existants des réseaux de neurones, ainsi que les réseaux de neurone convolutionnel. Enfin, nous avons cité ses différents domaines d'application et leurs avantages et inconvénients.

Dans le chapitre suivante, on va montrer la partie expérimentale de notre travail et on discute les différents résultats obtenus

CHAPITRE 3

IMPLÉMENTATION ET RÉSULTATS

3.1 Introduction

Après avoir présenté, les notions générale d'évaluation de la qualité des images fixés et les techniques d'apprentissage automatique. Dans ce chapitre, nous présentons les outils utilisées dans notre application, l'architecture de notre approche, l'implémentation et les résultats obtenus.

3.2 Ressources matérielles

Notre système est développé dans deux ordinateurs (acer, DELL) dont les même caractéristiques techniques, sont les suivantes :

- Processeur : Intel « I3 »
- RAM 4 Go
- Système d'exploitation : Windows 10 professionnel 64bit

3.3 Ressources logicielles

3.3.1 Python

Un langage de programmation de niveau supérieur, caractérisé par une écriture et une lecture simples, facile à apprendre, utiliser un style de programmation ouvert et évolutif.

Python est un langage polyvalent utilisé dans de nombreux domaines, tels que la création de programmes autonomes utilisant des interfaces graphiques connues et l'exécution de programmes Web, et son utilisation comme langage de script pour contrôler les performances des programmes les plus populaires.

En général, Python peut être utilisé pour programmer des programmes simples pour les débutants et pour réaliser de grands projets en même temps que tout autre langage de programmation. Il est souvent conseillé aux débutants en programmation d'apprendre cette langue car c'est l'une des langues de logiciel d'apprentissage les plus rapides [13].



FIGURE 3.1 – Logo Python.

3.3.2 PyCharm

Un environnement de développement intégré utilisé pour programmer en Python. Il permet l'analyse de code et contient un débogueur graphique. Il permet également la gestion des tests unitaires, l'intégration de logiciel de gestion de versions, et supporte le développement web .

Développé par l'entreprise tchèque JetBrains, c'est un logiciel multi-plateforme qui fonctionne sous Windows, Mac OS X et Linux. Il est décliné en édition professionnelle, diffusé sous licence propriétaire et en édition communautaire diffusé sous licence Apache [13].

3.3.3 PyTorch

PyTorch est une bibliothèque logicielle Python open source d'apprentissage machine qui s'appuie sur Torch développée par Facebook [28].

PyTorch permet d'effectuer les calculs tensoriels nécessaires notamment pour l'apprentissage profond (deep learning). Ces calculs sont optimisés et effectués soit par le processeur (CPU) soit, lorsque c'est possible, par un processeur graphique (GPU) supportant CUDA.

PyTorch permet de :

- Manipuler des tenseurs (tableaux multidimensionnels), de les échanger facilement avec Numpy et d'effectuer des calculs efficaces sur CPU ou GPU (par exemple, des produits de matrices ou des convolutions).
- Calculer des gradients pour appliquer facilement des algorithmes d'optimisation par descente de gradient. PyTorch utilise la bibliothèque autograd.



FIGURE 3.2 – Logo PyTorch.

3.3.4 Anaconda

Une distribution libre et open source des langages de programmation Python et R appliqué au développement d'applications dédiées à la science des données et à l'apprentissage automatique, qui vise à simplifier la gestion des paquets et de déploiement. Les versions de paquetages sont gérées par le système de gestion de paquets conda.

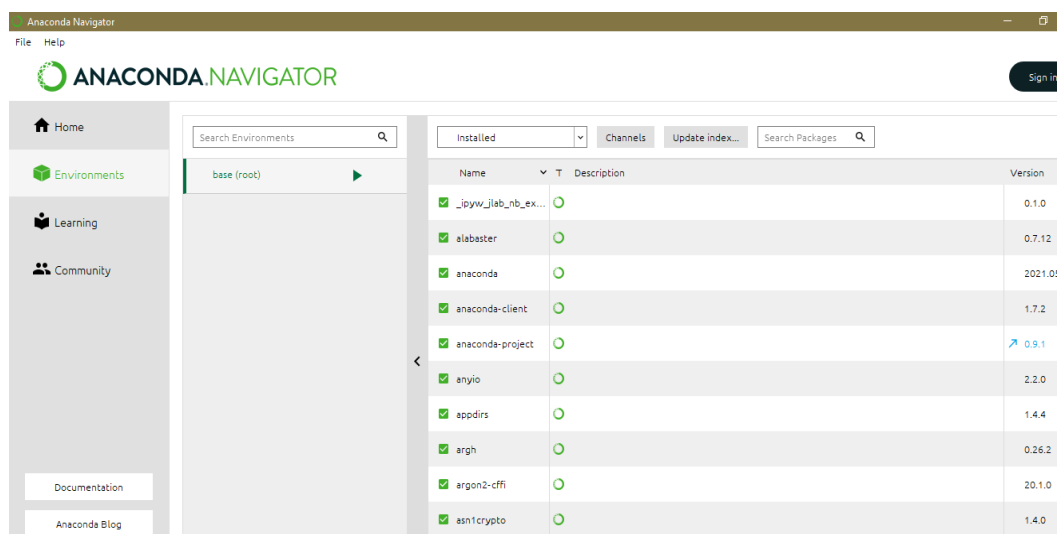


FIGURE 3.3 – Interface Anaconda.

3.3.5 Scikit-Learn

Scikit-learn est une bibliothèque libre Python destinée à l'apprentissage automatique. Elle est développée par de nombreux contributeurs [32].

La bibliothèque sklearn contient de nombreux outils efficaces pour l'apprentissage automatique et la modélisation statistique, notamment la classification, la régression, le clustering et la réduction de la dimensionnalité.

Sklearn est utilisé pour créer des modèles d'apprentissage automatique. Il ne doit pas être utilisé pour lire les données, les manipuler et les résumer. Il existe de meilleures bibliothèques pour cela (par exemple NumPy, Pandas, etc.)



FIGURE 3.4 – Logo Scikit-Learn.

3.3.6 Google Colab

Google Colaboratory ou Colab [27], un outil Google simple et gratuit pour collaborer avec vos collègues sur des projets en science des données. Colab permet :

- d'améliorer vos compétences de codage en langage de programmation Python.
- de développer des applications en Deep Learning en utilisant des bibliothèques Python populaires telles que TensorFlow et PyTorch .
- d'utiliser un environnement de développement (Jupyter Notebook) qui ne nécessite aucune configuration.

Mais la fonctionnalité qui distingue Colab des autres services est l'accès à un processeur graphique GPU, totalement gratuitement !.

Comme son nom l'indique « collaboration ». Colab exploite les mêmes fonctionnalités de collaboration des autres éléments de la G Suite : Sheet, Slide, Docs, etc. Il fonctionne sur les serveurs Google et vous n'avez rien à installer. De plus, les documents Colab (Jupyter Notebook) sont enregistrés directement votre compte Google Drive.

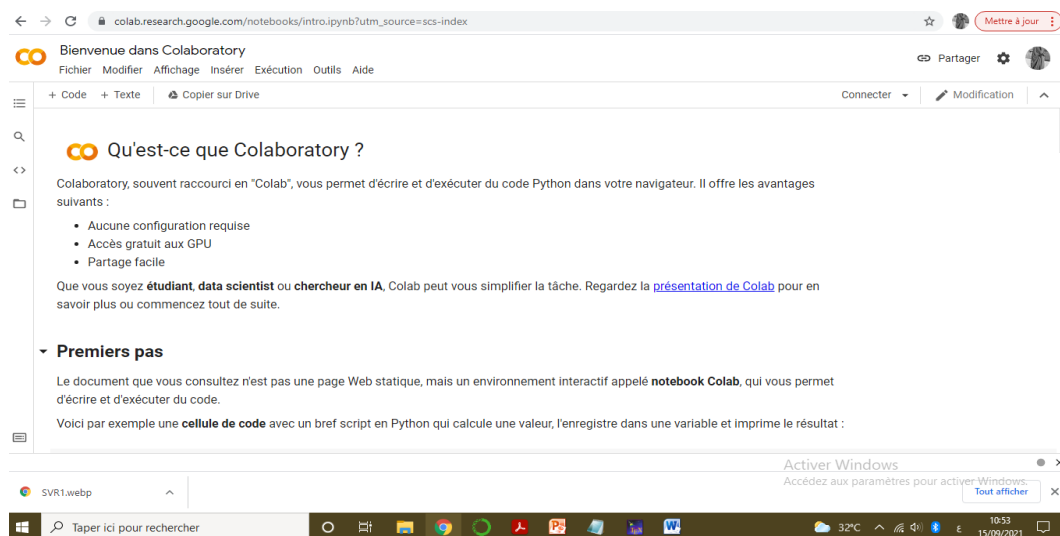


FIGURE 3.5 – Interface Anaconda.

3.4 Base des images

Dans le domaine de la qualité des images, Après conception et développement d'une métrique, il est nécessaire de la tester sur des images associées à des scores subjectifs humains, en vue de mesurer le taux de corrélation entre leurs prédictions et ces derniers [5].

La conception des méthodes génériques d'évaluation de la qualité d'image exige un test avec un grand ensemble d'images ayant été déformées par une variété de dégradation. Plusieurs bases d'image ont été suggérées. Il existe six d'elles plus couramment utilisées dans la littérature [2]. Dans notre travail nous utilisons que :

- **La base LIVE :**

C'est base a été développée au laboratoire d'ingénierie de l'image et de la vidéo collaboration avec le Center pour Perceptual Systems de l'Université du Texas à Austin ETATS-UNIS, la première version de c'est base a été mise en ligne en 2003 et la deuxième version a été publiée en 2005 [8].

Pour notre travail la base LIVE contient 259 images divisé sur 5 dégradation (Table 3.1). Nous utilisons 80% images de chaque classe pour l'apprentissage et 20% image pour le test.

Dégradation	Nombre des images
JP2K	50
JPEG	60
WN	49
BLUR	50
FF	50

TABLE 3.1 – Type et nombre des images utilisé .

La figure ci-dessus illustre des images apprenant dans la base LIVE :

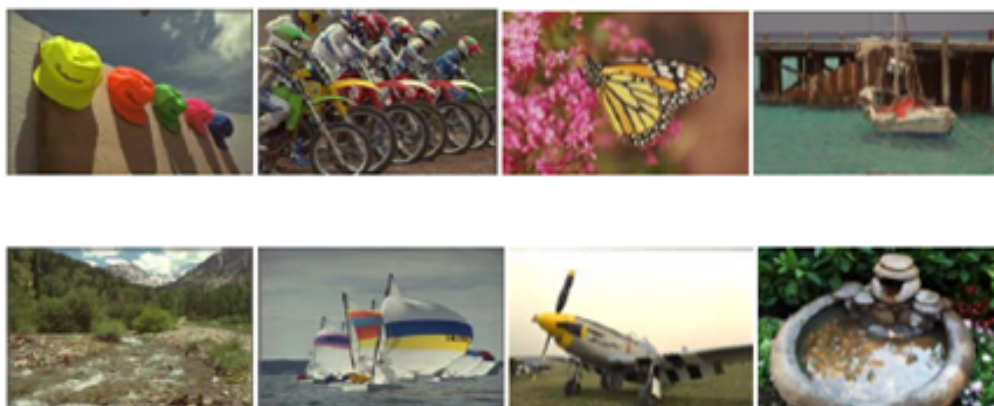


FIGURE 3.6 – Echantillon des images de la base LIVE.

3.5 Interfaces Graphiques Utilisateurs

Pour notre application, nous avons créé un interface graphique d'utilisateur comme il est montré dans la figure suivant :



FIGURE 3.7 – La fenêtre principale d'application.

La fenêtre principale contient les fonctionnalités suivantes :

1. Pour choisir le dossier LIVE.
2. Choisir un image pour le traiter.
3. Pour quitter et sortir de fenêtre.
4. Afficher les filters utilisé pour faire l'extraction des caractéristique.
5. Afficher les cartes des caractéristiques (feature maps).
6. Fonctionnement de AGGD (Asymmetric Generalized Gaussian Distribution).
7. Dessiner le courbe de MSCN (mean subtracted contrast normalized).
8. Afficher le model d'apprentissage de SVR.
9. Evaluation des performances PCC et SCC.

3.6 L'architecture d'application

Voici l'architecture de notre application qui présente les principales méthodes utilisé :

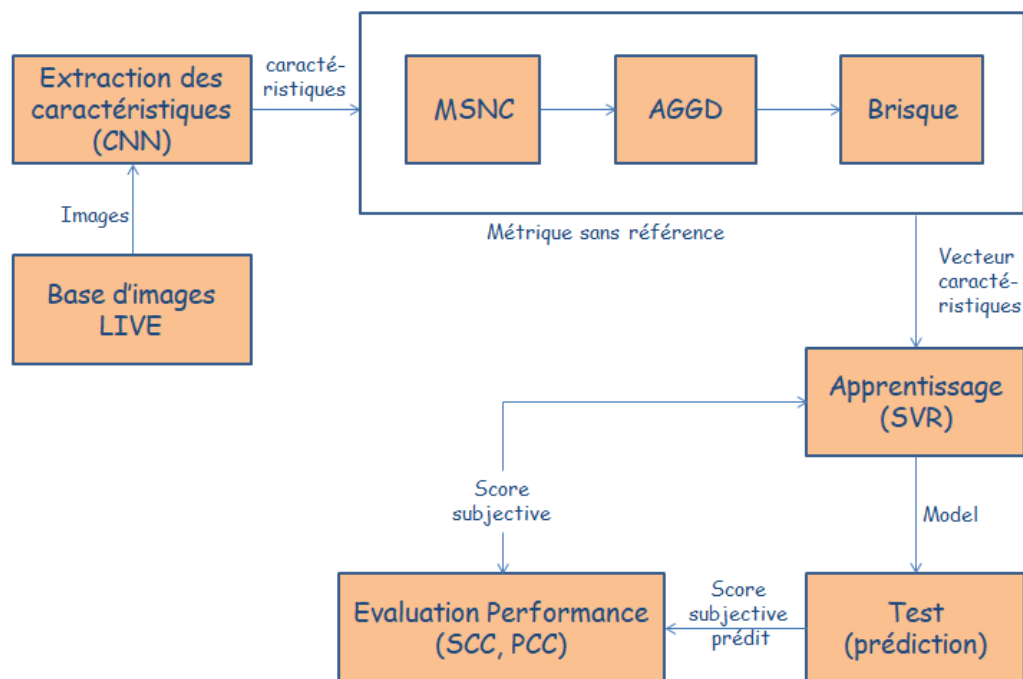


FIGURE 3.8 – Architecture d'application.

3.6.1 Extraction des caractéristiques (CNN)

Un réseau neuronal convolutif (CNN) est conçu pour prédire avec précision la qualité de l'image sans image de référence, c'est-à-dire l'approche sans référence.

- Dans la couche de convolution, Nous convolués les image avec 64 filtres de type 3x3, ces dernier illustré dans la figure suivante :

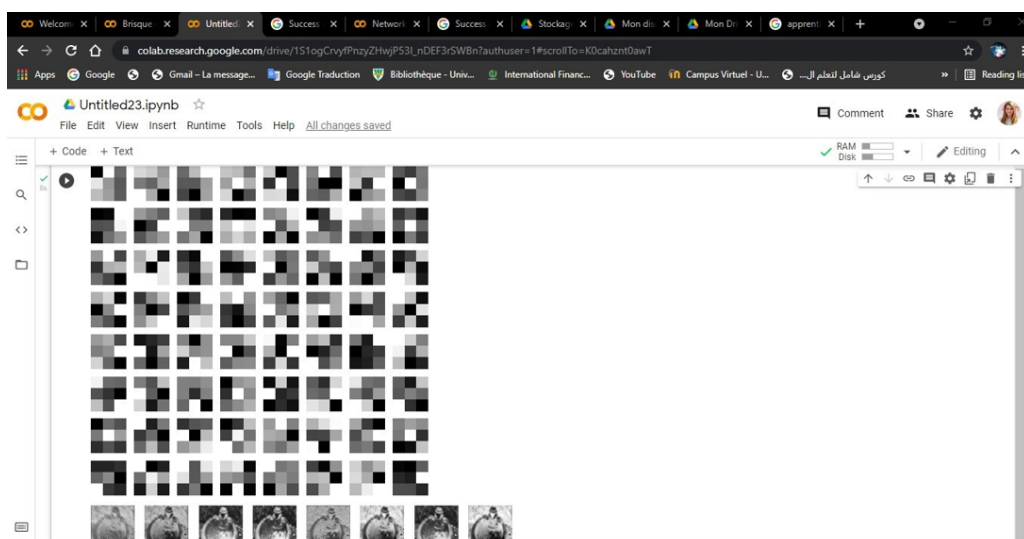


FIGURE 3.9 – Filter de model VGG16.

- L'affichage des cartes de caractéristique (feature maps) qui sont extrait par les réseaux de neurone convolutionnel :

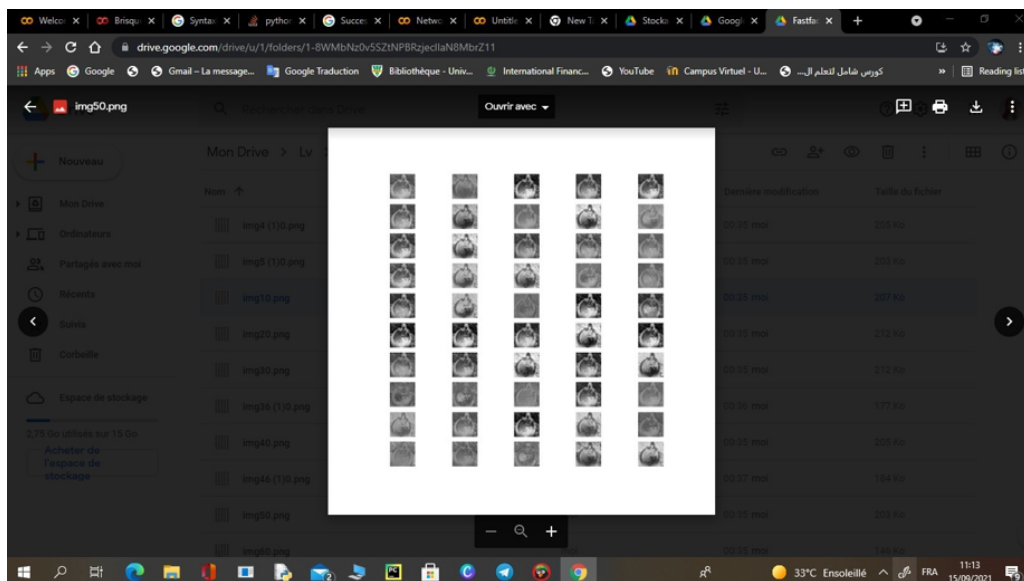


FIGURE 3.10 – Cartes des caractéristiques (feature maps).

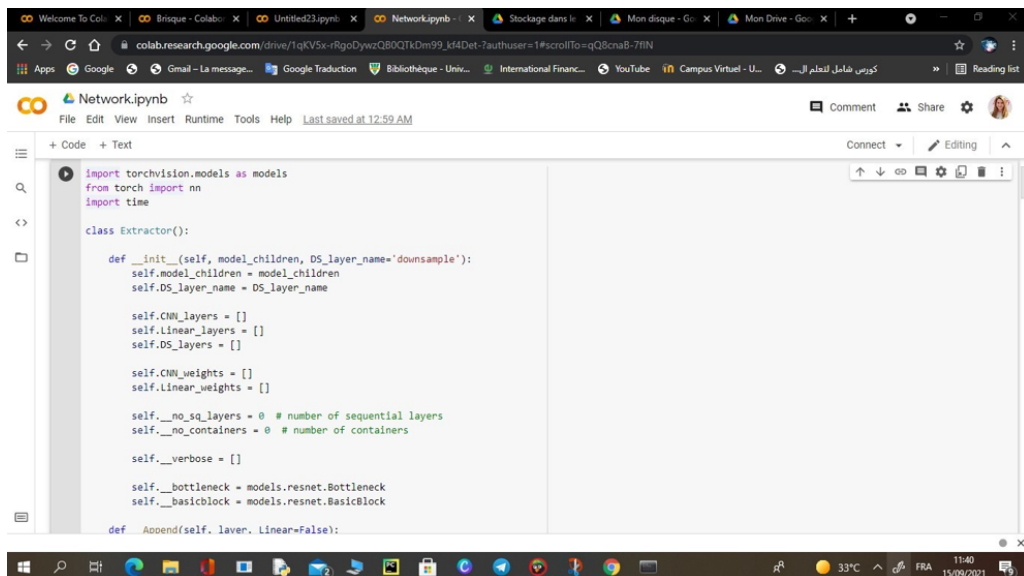


FIGURE 3.11 – Code source de méthode d'extraction des caractéristiques.

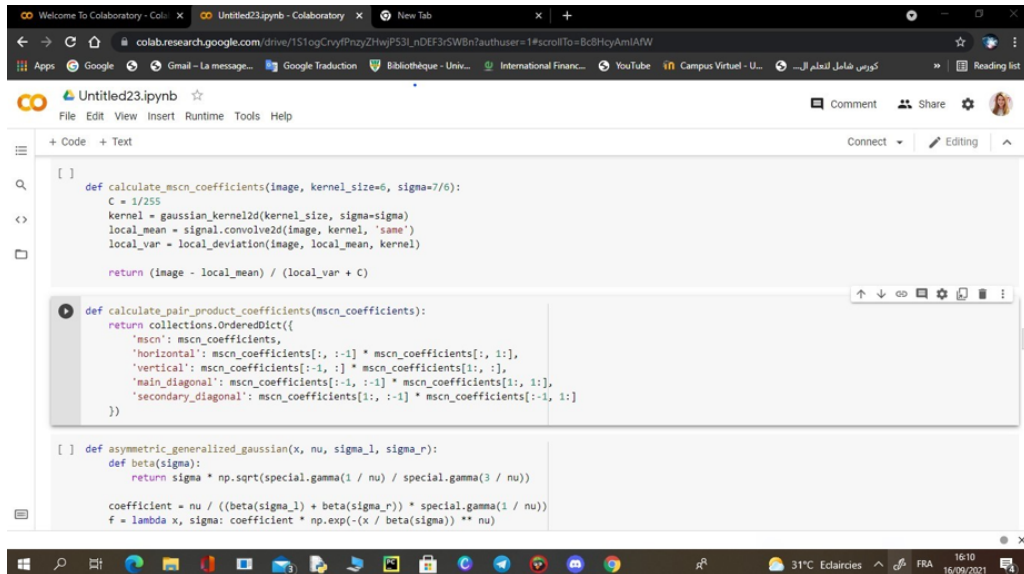
3.6.2 Evaluation de la qualité d'image sans référence

L'évaluation de la qualité d'image sans référence ne nécessite pas d'image de base pour évaluer une qualité d'image, la seule information que l'algorithme reçoit est l'image déformée dont la qualité est évaluée.

- MSNC :

Les signes des coefficients adjacents présentent également une structure régulière, qui est perturbée en présence de distorsion. Le modèle de produits propose par paires de coefficients MSCN voisins le long de quatre directions horizontale, verticale, principale-diagonale et secondaire-diagonale.

Les caractéristiques nécessaires pour calculer la qualité de l'image sont le résultat de l'ajustement des coefficients MSCN et des produits décalés aux distributions gaussiennes généralisées. Tout d'abord, nous devons ajuster les coefficients MSCN au GDD, puis les produits par paires à l'AGGD.



```

def calculate_mscn_coefficients(image, kernel_size=6, sigma=7/6):
    C = 1/255
    kernel = gaussian_kernel2d(kernel_size, sigma=sigma)
    local_mean = signal.convolve2d(image, kernel, 'same')
    local_var = local_deviation(image, local_mean, kernel)
    return (image - local_mean) / (local_var + C)

def calculate_pair_product_coefficients(mscn_coefficients):
    return collections.OrderedDict({
        'mscn': mscn_coefficients,
        'horizontal': mscn_coefficients[:, :-1] * mscn_coefficients[:, 1:],
        'vertical': mscn_coefficients[:-1, :] * mscn_coefficients[1:, :],
        'main_diagonal': mscn_coefficients[:-1, :-1] * mscn_coefficients[1:, 1:],
        'secondary_diagonal': mscn_coefficients[1:, :-1] * mscn_coefficients[:-1, 1:]
    })

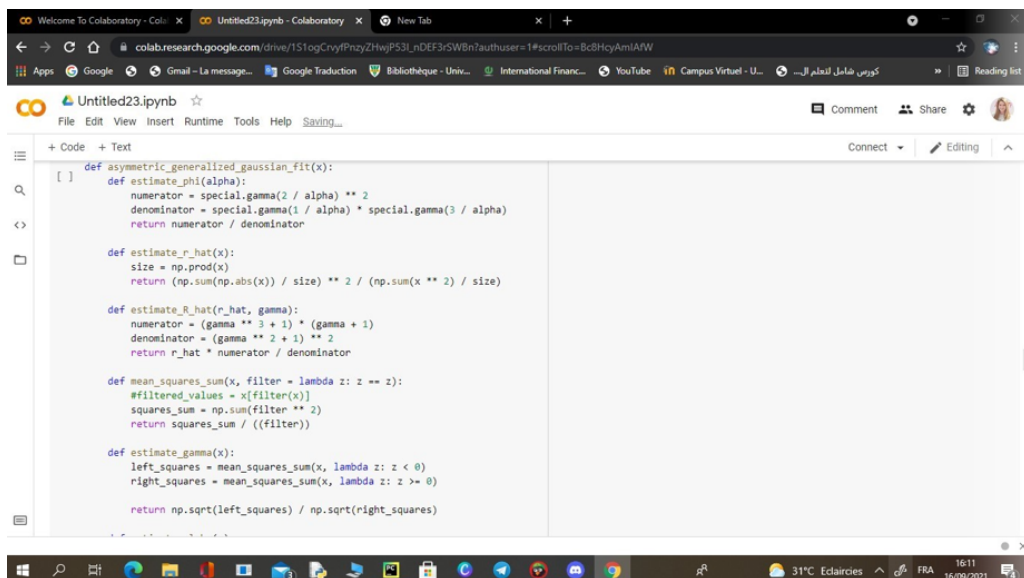
def asymmetric_generalized_gaussian(x, nu, sigma_l, sigma_r):
    def beta(sigma):
        return sigma * np.sqrt(special.gamma(1 / nu) / special.gamma(3 / nu))
    coefficient = nu / ((beta(sigma_l) + beta(sigma_r)) * special.gamma(1 / nu))
    f = lambda x, sigma: coefficient * np.exp(-(x / beta(sigma)) ** nu)

```

FIGURE 3.12 – Code source de MSCN.

- **AGGD :**

La distribution gaussienne généralisée ne fournit pas un bon ajustement aux histogrammes empiriques des produits de coefficients. Ainsi, nous utilisons le modèle de distribution gaussienne généralisée asymétrique (AGGD).



```

def asymmetric_generalized_gaussian_fit(x):
    def estimate_phi(alpha):
        numerator = special.gamma(2 / alpha) ** 2
        denominator = special.gamma(1 / alpha) * special.gamma(3 / alpha)
        return numerator / denominator

    def estimate_r_hat(x):
        size = np.prod(x)
        return (np.sum(np.abs(x)) / size) ** 2 / (np.sum(x ** 2) / size)

    def estimate_r_hat(r_hat, gamma):
        numerator = (gamma ** 3 + 1) * (gamma + 1)
        denominator = (gamma ** 2 + 1) ** 2
        return r_hat * numerator / denominator

    def mean_squares_sum(x, filter = lambda z: z == 2):
        #filtered_values = x[filter(x)]
        squares_sum = np.sum(filter ** 2)
        return squares_sum / ((filter))

    def estimate_gamma(x):
        left_squares = mean_squares_sum(x, lambda z: z < 0)
        right_squares = mean_squares_sum(x, lambda z: z >= 0)
        return np.sqrt(left_squares) / np.sqrt(right_squares)

```

FIGURE 3.13 – Code source AGGD.

- AGGD d'apprentissage (80%) :

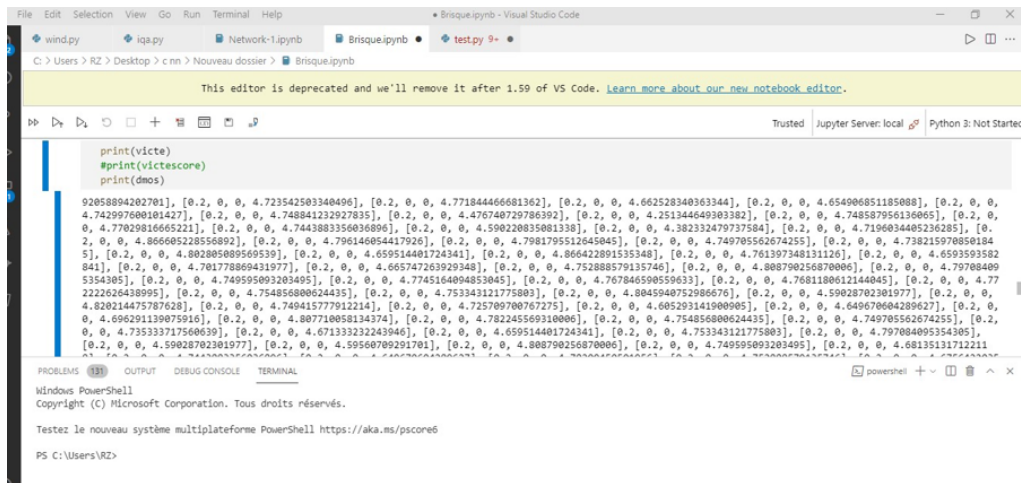


FIGURE 3.14 – Résultat AGGD d’apprentissage (Vecteur de caractéristique).

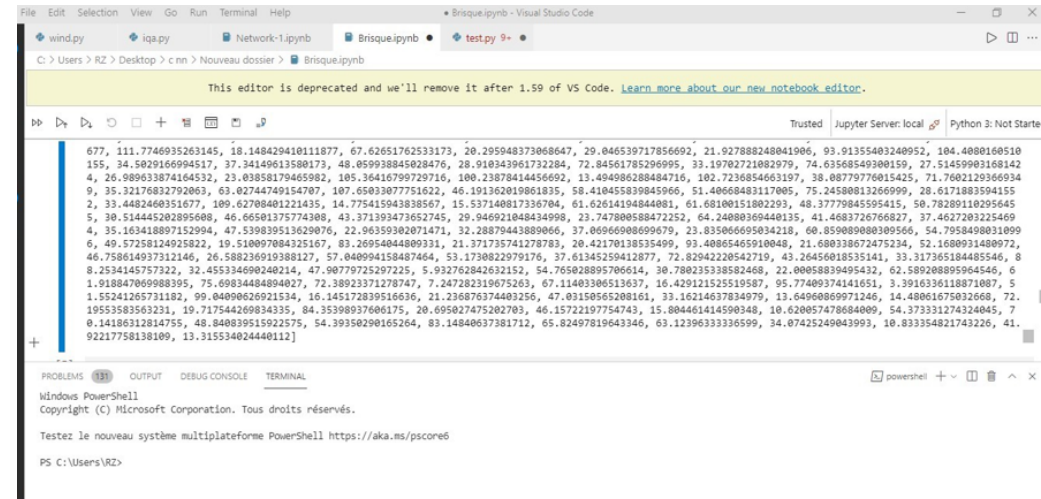


FIGURE 3.15 – Résultat AGGD d’apprentissage (complète la figure 3.14).

- AGGD de test (20%) :

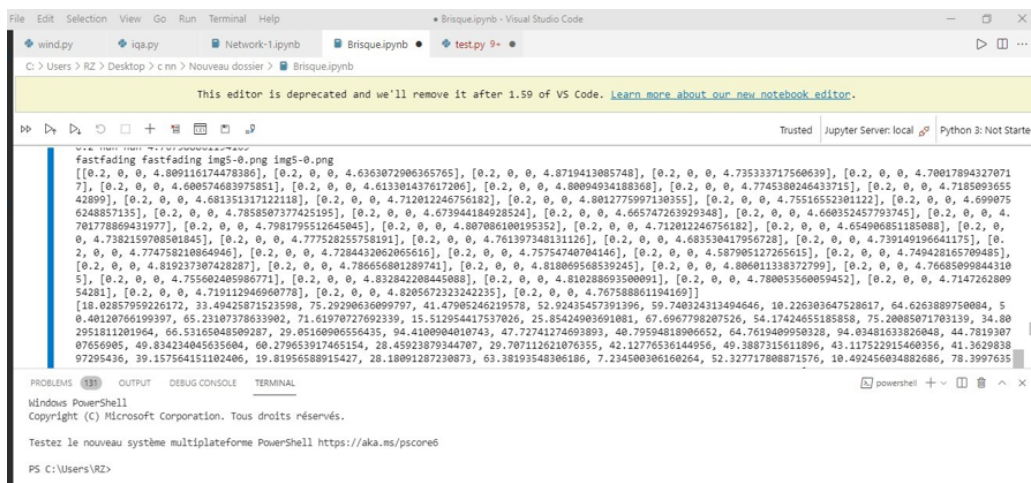


FIGURE 3.16 – Résultat AGGD de test.

- **Brisque :**

Brisque est un modèle qui utilise uniquement les pixels de l'image pour calculer les caractéristiques (d'autres méthodes sont basées sur la transformation de l'image vers d'autres espaces comme les ondelettes ou la DCT). Il est démontré qu'il est très efficace car il n'a besoin d'aucune transformation pour calculer ses caractéristiques.

Il s'appuie sur le modèle spatial Natural Scene Statistics (NSS) des coefficients de luminance normalisés localement dans le domaine spatial, ainsi que sur le modèle des produits par paires de ces coefficients.

```

[24] constant = np.sqrt(special.gamma(1 / alpha) / special.gamma(3 / alpha))
      mean = estimate_mean(alpha, sigma_l, sigma_r)
      return alpha, mean, sigma_l, sigma_r

[25] def calculate_brisque_features(image, kernel_size=7, sigma=7/6):
      def calculate_features(coefficients_name, coefficients, accum=np.array([])):
          alpha, mean, sigma_l, sigma_r = asymmetric_generalized_gaussian_fit(coefficients)

          if coefficients_name == 'mscn':
              var = (sigma_l ** 2 + sigma_r ** 2) / 2
              return [alpha, var]

          return [alpha, mean, sigma_l ** 2, sigma_r ** 2]

      mscn_coefficients = calculate_mscn_coefficients(image, kernel_size, sigma)
      coefficients = calculate_pair_product_coefficients(mscn_coefficients)

      features = [calculate_features(name, coeff) for name, coeff in coefficients.items()]
      flatten_features = list(chain.from_iterable(features))
      return np.array(flatten_features)
  
```

FIGURE 3.17 – Code source Brisque.

3.6.3 Apprentissage (SVR) et Test (prédit)

Voici notre code source du SVR avec les paramètres (score subjective, vecteur de caractéristique) pour l'apprentissage et du test pour la prédiction de score subjective prédite :

```

[ ] from sklearn.svm import SVC
from sklearn import preprocessing
from sklearn import utils
lab_enc = preprocessing.LabelEncoder()
#victest
x = victe[0:144]
print(len(x),x)
y = dmos
print(len(dmos),dmos)
training = lab_enc.fit_transform(dmos)
print(utils.multiclass.type_of_target(training))
print(utils.multiclass.type_of_target(training.astype('int')))
print(utils.multiclass.type_of_target(training))
#instanciation
model_SVR = SVC()
#training
model_SVR.fit(x, training)
#model_SVR.show
#Prédiction(test)
#model_SVR.predict( [victest])
    
```

FIGURE 3.18 – Code source d’apprentissage et de test.

Voici le résultat de test que nous avons obtenu qui représentent le score subjective prédite :

```

179 [14.34477530839105, 4.495168881480339, 34.315394671086715, 87.8267449661924, -0.194608687639995, 48.5281572537932, 79.95944789086505, 28.9449556052722
5, 50.1222403479616, 4.3208561200949565, 35.4156346289015, 57.0885763912255, 33.58553766172936, 30.55413159493454, 34.439397588231305, -2.040615610409613,
47.80218047163155, 30.180770593920424, 53.951334760512935, 32.754563027118, 33.283198045948446, 72.05750190482459, 72.8144214022269, 12.56494613089226, 6
3.6907204698986, 47.2858717156399, 45.8298134752761, 42.61472207624115, 33.240480634782365, 32.009866412702586, 6.681376948835168, 41.36730112939379, 59.
87396522983406, 12.65410627820995, 43.42259860497224, 70.94908661426099, 48.4017211733363, 53.578731920146005, 24.09025497677283, 11.714566786308875, 46.68
4096124058215, 35.59966912005858, 76.28674666680452, 52.7856364511723, 8.613511703300915, 65.85757417524141, 74.14564153374138, 75.781128476878, 48.088946
90810563, 69.2292141708657, 43.696253511551454, 22.461157863083844, 13.501078103650578, 3.8890621428035583, 74.39794796497739, 12.692083542828538, 23.258787
47358680, 77.32511548366257, 45.679496649039394, 68.99086164929872, 39.37667556675135, 26.9693839367697, 63.64493591720355, 29.392249180595478, 71.2121820
2942677, 111.7746935263145, 18.14842941011877, 67.62651762533173, 20.295048373966647, 29.046539717856692, 21.92788824804396, 93.91355403240952, 104.408016
0510155, 34.5029166994517, 37.34149613580173, 48.059938045028476, 28.910343961732284, 72.84561785296995, 33.19702721082979, 74.63568549300159, 27.514590316
81424, 26.989633874164532, 23.03858179465982, 105.36416799729716, 100.23878414456692, 13.404986288484716, 102.7236854663197, 38.08779776015425, 71.760212936
69349, 35.32176832792063, 63.02744749154707, 107.65033077751622, 46.191362019861835, 58.410455839845966, 51.40668483117005, 75.24580813266999, 28.6171883594
1552, 33.4482460351677, 109.62708401221435, 14.775415943838567, 15.573740817336704, 61.62614194844081, 61.68100151802293, 48.377984595415, 50.782891102956
455, 30.514445302895608, 46.66501375774308, 43.371393473652745, 29.946521048434998, 23.747808588472252, 64.24080369440135, 41.4683726766827, 37.462720322546
94, 35.163418897152994, 47.539839513629076, 22.96359302071471, 32.28879443890666, 37.06966980899679, 23.83586695934218, 60.85089808309566, 54.795849893109
96, 49.57258124925822, 19.510097084325167, 83.26954044809331, 21.371735741278783, 20.421170138535499, 93.40865465910048, 21.680338672475234, 52.168091420097
5, AK.7684160373151AK.76.588376010308137. 67.AB0A041C8A87A6A. 63.173083307017A. 37.A13AC304113077. 73.87047330643710. A2.36AC6A816E3C81A1. 33.3173AC16A8A8E3CA
    
```

FIGURE 3.19 – Résultat de test.

```

69349, 35.32176832792063, 63.02744749154707, 107.65033077751622, 46.191362019861835, 58.410455839845966, 51.40668483117005, 75.24580813266999, 28.6171883594
1552, 33.4482460351677, 109.62708401221435, 14.775415943838567, 15.573740817336704, 61.62614194844081, 61.68100151802293, 48.377984595415, 50.782891102956
455, 30.514445302895608, 46.66501375774308, 43.371393473652745, 29.946521048434998, 23.747808588472252, 64.24080369440135, 41.4683726766827, 37.462720322546
94, 35.163418897152994, 47.539839513629076, 22.96359302071471, 32.28879443890666, 37.06966980899679, 23.83586695934218, 60.85089808309566, 54.795849893109
96, 49.57258124925822, 19.510097084325167, 83.26954044809331, 21.371735741278783, 20.421170138535499, 93.40865465910048, 21.680338672475234, 52.168091420097
2, 46.758614937312146, 26.588236919388129, 67.040994150487464, 53.1730822979176, 37.01345259121877, 72.824222054719, 43.2645601833141, 33.31736184484854
6, 88.253414575322, 32.45534669024014, 47.9077925297225, 5.932762842632152, 54.765028895706614, 30.780235338582468, 22.00858398945432, 62.5892889596454
6, 61.91884706988395, 75.69834484094027, 72.3892371278747, 7.247282319675263, 67.11403306513637, 16.429121525519587, 95.77409374141651, 3.39163611887108
7, 51.55241265731182, 99.04090626921534, 16.145172839516636, 21.236876374403256, 47.03158565208161, 33.16214637834979, 13.64908689971246, 14.48061675032668,
72.1955383563231, 19.717544269834335, 84.35398937606175, 20.695027475202703, 46.15722197754743, 15.804461414590348, 16.620057478604009, 54.373331274324045,
70.14186312814755, 48.040838515922575, 54.39350290165204, 83.14840637361712, 65.82497819643346, 63.12396333365999, 34.07425249043993, 10.833354822174226, 4
1.9221778138109, 13.315534024440112]
multiclass
multiclass
multiclass
C:\r-r-1.0_hnask_kie-falca_racha_cfa-300_r1asc_wahnt-hona_rnafa-0.A
    
```

FIGURE 3.20 – Résultat de test (complète la figure 3.19).

```

6, 61.918847069988395, 75.69834484894027, 72.38923371278747, 7.247282319675263, 67.11403306513637, 16.429121525519587, 95.77409374141651, 3.391633611887108
7, 51.55241265731182, 99.04090626921534, 16.145172839516636, 21.236876374403256, 47.03150565208161, 33.16214637834979, 13.64908069971246, 14.48061675832668,
72.19553583353231, 19.717544269834335, 84.35398937686175, 20.695027475202703, 46.15721197754743, 15.804461414590340, 10.620057478664809, 54.373331274324045,
70.14186312814755, 48.840839515922575, 54.39350290165264, 83.14040637381712, 65.02497819643346, 63.12396333336599, 34.07425240043993, 10.83354821743226, 4
1.9217758138109, 13.315534024440112]
multiclass
multiclass

SVC(C=1.0, break_ties=False, cache_size=200, class_weight=None, coef0=0.0,
    decision_function_shape='ovr', degree=3, gamma='scale', kernel='rbf',
    max_iter=-1, probability=False, random_state=None, shrinking=True,
    tol=0.001, verbose=False)

```

FIGURE 3.21 – Résultat de test (complète la figure 3.20).

3.6.4 Evaluation des performance

```

[4] y = dmos
print(len(dmos),dmos)
training = lab_enc.fit_transform(dmos)
print(utils.multiclass.type_of_target(training))
print(utils.multiclass.type_of_target(training.astype('int')))
print(utils.multiclass.type_of_target(training))
#instanciation
model_SVR = SVC()
#training
r=model_SVR.fit(x, training)
dmost.append(r)
#model_SVR.show
#Prédiction(test)
#model_SVR.predict([victest])

srooc=[]
plcc=[]

SRCC = srooc.spearmanr(dmosTest,dmost)[0]
PLCC = plcc.pearsonr(dmosTest,dmost)[0]

```

FIGURE 3.22 – Code source de SCC et PCC.

Nous étions censés obtenir un résultat environ 1 (0,70 par exemple) mais malheureusement nous n’y sommes pas arrivés.

3.7 Conclusion

Dans ce chapitre nous avons expliqué l’approche utilisée et les résultats obtenues. Dans la suite, nous présentons notre conclusions et perspectives.

CONCLUSION GÉNÉRALE

Notre société devient de plus en plus intelligente a l'heure actuelle, et comme tous les autres domaines de l'apprentissage, le domaine de l'évaluation de qualité d'image a connu une évolution majeure depuis l'apparition du réseaux neurones.

Les réseaux de neurones éliminent la nécessité d'effectuer un évaluation de qualité d'image subjectivement qui est généralement une tâche fastidieuse. Les caractéristiques utilisées ne sont pas pré entraînés, c'est à dire que le réseau les apprend lui-même en s'entraînant sur des images.

Pour réaliser notre travail de l'évaluation de la qualité des images on a utilisé les réseaux de neurones, la méthode d'apprentissage qui a montré ses performances ces dernières années et nous avons choisi la méthode CNNs, ce choix est justifié par la simplicité et l'efficacité de la méthode.

"Le résultat obtenu lors de la phase de test confirme l'efficacité de notre approche.

Notre travail n'est que dans sa version initiale, on peut dire que ce travail reste ouvert pour des travaux de comparaison et/ou d'hybridation avec d'autres méthodes d'évaluation .

Nous proposons comme perspectives, l'utilisation de GPU (processeur graphique) avec reseaux de neurones convolutionnels pour améliorer les résultats d'évaluation de qualité des images.

BIBLIOGRAPHIE

- [1] Aladine Chetouani, "*Vers un système d'évaluation de la qualité d'image multi-critères*", thèse de Doctorat, Université Paris 13, 29 Novembre 2010.
- [2] AHMED SEGHIR Zianou, "*Evaluation de la qualité d'image*", thèse de Doctorat, Université de Mentouri-Constantine , 4/06/2012.
- [3] CHERIFI Daikha, "*Amélioration et Évaluation de la qualité des images*", thèse de Doctorat, Université des sciences et de technologie d'Oran Mohamed Boudiaf, 2015.
- [4] Fadi Boulos, "*Etat de l'art des critères objectifs de qualité d'images et de vidéos*", Recherche de Master, Ecole Polytechnique de l'Université de Nantes, 21 avril 2006.
- [5] Anass NOURI, "*Mise en place d'un Benchmark pour les métriques sans références d'évaluation de la qualité d'image*", Mémoire de Stage recherche M2, Université Ibn TOFAIL de Kénitra MAROC, Septembre 2013.
- [6] C. CHARRIER, C. LARABI, H. SAADANE, "*Evaluation de la qualité des images*".
- [7] Alexandre BENOIT, "*Le système visuel humain au secours de la vision par ordinateur*", thèse Doctorat, Institut National Polytechnique de Grenoble ,14 février 2007.
- [8] Atidel LAHOULOU, "*Machine learning techniques for image quality evaluation*", thèse de Doctorat, Laboratoire des Dispositifs de Communications et de Conversion Photovoltaïque ,Algérie, 03 Décembre 2012.
- [9] Gilles BUREL, "*RESEAUX de NEURONES en TRAITEMENT d'IMAGES :Des Modèles Théoriques aux Applications Industrielles*", Thèse de doctorat, Univer-

- sité de Bretagne Occidentale, 6 décembre 1991.
- [10] <https://www.ionos.fr/digitalguide/web-marketing/search-engine-marketing/quest-ce-quun-reseau-neuronal-artificiel/>.
- [11] <https://www.juripredis.com/fr/blog/id-19-demystifier-le-machine-learning-partie-2-les-reseaux-de-neurones-artificiels>.
- [12] Boughaba Mohammed, Boukhris Brahim, *"L'apprentissage profond (Deep Learning) pour la classification et la recherche d'images par le contenu"*, Mémoire de Master, Université Kasdi Merbah-Ouargla , 2017.
- [13] Abdelaziz HABBA, Omar ISHAK, *"La classification des images satellitaires par l'apprentissage profonde (deep learning)"*, Mémoire de Master, Université Ahmed Draia - Adrar, 2019.
- [14] Nassima Mezhoud, *"Une approche hybride pour la segmentation d'images"*, thèse de Doctorat, Université de Mentouri-Constantine, 22/01/2012.
- [15] Youcef Djeriri, *"Les Réseaux de Neurones Artificiels"*, Université de Sidi-Bel-Abbes, September 2017.
- [16] Rabah IDOUI, Ahmed HIMEUR, *"Reconnaissance d'images par les réseaux de neurones convolutionnels (CNN)"*, Université Mohammed Seddik BenYahia-Jijel, 2020.
- [17] GANA Massine, *"Implémentation d'un réseau de neurones dans un microcontrôleur"*, Mémoire de master, Université Mouloud Mammeri TIZI-OUZOU, 2016.
- [18] Baza Halima Saadia, *"La reconnaissance d'écriture arabe manuscrite"*, Mémoire de master, Université Mohamed Boudiaf- M'SILA, 2016.
- [19] Fatma Zahra Amrani, *"Extraction les concepts textuels en utilisant la technique de deep learning"*, Mémoire de master, Université Mohamed Khider – BISKRA.
- [20] GHENNAM Bochra. SMARA Safia, *"Les réseaux de neurone convolutionnel (CNN) pour la classification des images associées aux places de stationnement d'un parc de véhicule"*,Mémoire de master ,Université DJILALI BOUNAAMA-KHEMIS MILIANA, 2019.
- [21] Mr Mokri Mohammed Zakaria, *"Classification des images avec les réseaux de neurones convolutionnels"*, Mémoire de Master, Université Abou Bakr Belkaid Tlemcen, 03/07/2017.

-
- [22] Flávio BARREIRO LINDO, " *Interprétation d'images basée sur la technologie des réseaux de neurones*", Haute École de Gestion-Genève, 2 juillet 2018.
- [23] Jérone Pasquet, " *Modélisation, détection et classification d'objets urbains à partir d'images photographiques aériennes*", thèse de Doctorat, Université de Montpellier, 03/11/2016.
- [24] LOUAM ABDELHAK BILAL, " *Deep Learning basé sur les méthodes de réduction pour la reconnaissance de visage*", Mémoire de Master, Université Mohamed Khaider de Biskra, 6 juillet 2019.
- [25] Fentazi Mohamed Lamine, Bouzit Omar, " *Evaluation aveugle de qualité des images fixes*", Mémoire de Master, Université Mohammed Seddik BenYahia-Jijel, 2020.
- [26] Aboubacar Konate, " *Un aperçu sur quelques méthodes en apprentissage automatique supervisé* ", 5/12/2018.
- [27] Bakha boubkar, Kaiba Houssam, " *Un Système OCR basé Apprentissage Profond pour la Reconnaissance des Vignettes de Médicaments*", Mémoire de Master, Université Mohammed Seddik BenYahia-Jijel, 2019.
- [28] Julien Bergounhox, " *Avec PyTorch 1.0, Facebook cherche à créer la boîte à outils ultime pour l'intelligence artificielle*", l'usine digitale, 4 mai 2018.
- [29] <https://towardsdatascience.com/an-introduction-to-support-vector-regression-svr>.
- [30] <https://medium.com/swlh/reading-kang-cvpr14-convolutional-neural-networks-for-no-reference-image-quality-assessment>.
- [31] https://scikit-learn.org/stable/auto_examples/svm/plot_svm_regression.html.
- [32] <https://www.analyticsvidhya.com/blog/2015/01/scikit-learn-python-machine-learning-tool/>.