

People's Democratic Republic of Algeria
Ministry of Higher Education and Scientific Research
University of 8 May 1945-Guelma-
Faculty of Mathematics, Computer Science and
Science of Matter Department of Computer Science



Master Thesis

Specialty: Computer Science

Option:

Science and technology of information and communication

Theme

**Binarisation d'images de documents dégradés en
utilisant l'apprentissage automatique**

Presented by: Ilhem Ladaimia

Jury Members:

N	Full name	Quality
1	Dr. Chemesseennehar Bencheriet	Chairman
2	Dr. Abderrahmane Kefali	Supervisor
3	Dr. Salah Derdar	Examiner

June 2023.

ملخص

تعتبر عملية التحويل الثنائي خطوة مهمة في أي عملية لمعالجة الصور وتحليلها ، وبشكل أكثر تحديداً ، صور المستندات . في الواقع ، تم اقتراح عدد كبير من التقنيات في الأدبيات لترتيب الصور ذات التدرج الرمادي أو الصور الملونة ، كل منها مناسب لنوع معين من الصور ، ولكن للأسف لا يظهر أي منها فعالاً في التحويل الثنائي للصور الوثائقية القديمة المتدهورة . هذه الأخيرة معروفة بجودتها الرديئة بسبب التدهورات المختلفة التي حدثت خلال دورة حياة الوثيقة والوسائل البدائية المستخدمة حتى الآن لحمايتها . إنَّ غالبية الطرق المقترحة في الأدبيات تستند إلى حساب عتبة واحدة أو أكثر لإجراء عملية التحويل الثنائي . ولهذا يطلق عليها: طرق التحويل الثنائي بالعتبة .

في هذه المذكرة ، نقترح طريقة جديدة للتحويل الثنائي لصور المستندات المتدهورة ، والتي لا تعتمد على العتبة (مثل معظم خوارزميات التحويل الثنائي) ، ولكن على التعلّم الآلي ، وبالأخص على شبكة عصبية اصطناعية . تُعرف هذه الأخيرة بقدرتها انطلاقاً من مجموعة محدودة من البيانات تسمى بيانات التعلّم ، على التعميم على بيانات أخرى التي لم يتم تعلّمها . الطريقة المقترحة تم اختبارها على مجموعة من الصور العامّة والنتائج التي تم الحصول عليها مشجعة .

الكلمات المفتاحية:

صور الوثائق ، المستندات المتدهورة ، التحويل الثنائي ، التعلّم الآلي ، الشبكات العصبية

Abstract

Binarization is an important step in any image processing and analysis process and more particularly, document images. Indeed, a large number of techniques have been proposed in the literature for the binarization of grayscale or color images, each of which is appropriate for a particular type of images, but unfortunately none of them show effective for binarization of degraded old document images. The latter are known for their poor quality due to the various deteriorations undergone during the life cycle of the document and the provisions hitherto employed for their safeguard. However, the majority of the methods proposed in the literature are based on the calculation of one or more thresholds to perform the binarization. They are called as follows: binarization methods by thresholding. In this thesis we propose a new method of binarization of images of degraded documents which is not based on thresholding (like most binarization algorithms), but on machine learning, a multilayer Perceptron type artificial neural network. The latter is known for its ability to generalize from a limited set of so-called learning behavior data for other data that have not been learned. The proposed method is tested on a collection of public images and the results obtained are encouraging.

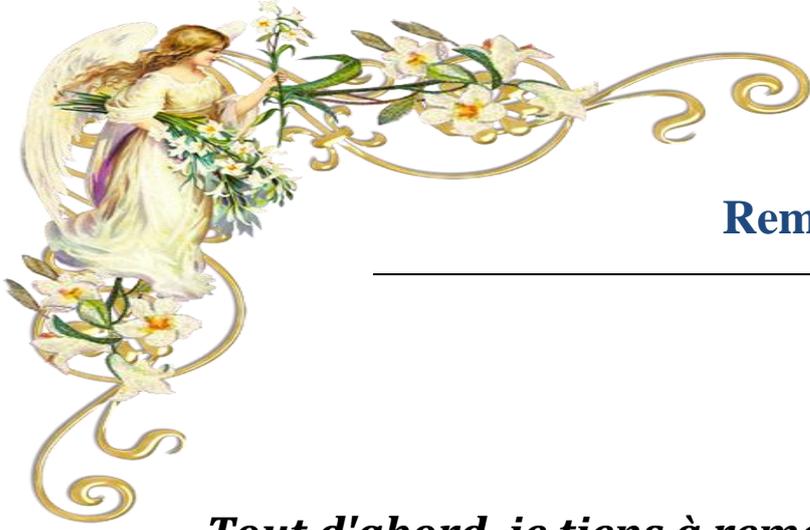
Keywords: Document images, degraded documents, binarization, machine learning, neural networks.

Résumé

La binarisation est une étape importante dans tout processus de traitement et d'analyse d'images et plus particulièrement, d'images de documents. En effet, un grand nombre de techniques ont été proposées dans la littérature pour la binarisation d'images en niveaux de gris ou en couleurs, dont chacune est appropriée à un type particulier d'images, mais malheureusement aucune d'entre elle ne montre efficace pour la binarisation d'images de documents anciens dégradées. Ces derniers sont connus pour leur mauvaise qualité due aux différentes détériorations subies au cours du cycle de vie du document et les dispositions jusque-là employées pour leur sauvegarde. Cependant, la majorité des méthodes proposées dans la littérature sont basées sur le calcul d'un ou plusieurs seuils pour effectuer la binarisation. Elles sont dites ainsi : *méthodes de binarisation par seuillage*.

Dans ce mémoire nous proposons une nouvelle méthode de binarisation d'images de documents dégradés qui ne repose pas sur le seuillage (comme la grande partie des algorithmes de binarisation), mais sur l'apprentissage automatique, réseau de neurones artificiel de type *Perceptron multicouches*. Ce dernier est connu par son capacité de généraliser à partir d'un ensemble limité de données dite d'*apprentissage* le comportement pour d'autres données n'ont pas été apprises. La méthode proposée est testée sur une collection d'images publiques et les résultats obtenus sont encourageants.

Mots-clés : Images de documents, documents dégradés, binarisation, apprentissage automatique, réseaux de neurones.



Remerciement

Tout d'abord, je tiens à remercier Dieu de nous avoir donné la santé, la volonté et la patience pour terminer notre formation et pouvoir mener à bien ce travail de recherche.

je remercie également mes parents qui m'ont encouragé et aidé à atteindre cette étape de ma formation. je tiens à remercier mon directeur de thèse, Dr., Abderrahmane Kefali qui m'a guidé dans la réalisation de ce projet, et mes plus grands remerciements vont à notre collègue H. Bensalah



Dédicace

Je commence par rendre grâces à DIEU et à sa bonté pour la patience, la compétence et le courage qu'il m'a donné pour arriver à ce stade.

*À mes chers parents : Rachid et Fella (rabi yerhmeha)
Sources de mes joies, secrets de ma force Papa, dans ta détermination, ta force et ton honnêteté Maman dans sa bonté, sa patience et son dévouement pour C'est à vous que je dois offrir cette réussite Et je suis fière de vous l'offrir
Dédicaces pour mon mari ,mes chères sœurs, mes frères,
Merci mon collègue Hazem ...*

Table de figures

Figure 1.1. Résultat de la binarisation d'une image de document, (a) Image originale, (b) Image binarisée	7
Figure 2.1: Types d'apprentissage automatique [Sai,15].....	24
Figure 2.2. La classification/La régression[Nib,86].....	25
Figure 2.3. Exemple d'apprentissage automatique non-supervisé (Clustering) [Nib,86].....	26
Figure 2.4. Exemple d'un PMC [Che,93].....	29
Figure 2.5. Principe de l'apprentissage par rétropropagation de l'erreur.....	30
Figure 2.6. Carte auto-organisatrice rectangulaire de 63 neurones [Gar,04].....	33
Figure 3.1. Exemples de défauts présents sur des documents dégradés : (a),(b) et (f) montrent des taches d'encre et d'usures, (c) transparence du verso de la page, (d) montre un trou dans la page, (e) écriture n'est pas clair.....	37
Figure 3.2 : Schéma général de notre approche.....	39
Figure 3.3. Image de document dégradé avec son image de vérité terrain.....	40
Figure 3.4. PMC pour classifier un pixel p dans une fenêtre de 3×3	40
Figure 3.5. Vecteur d'entrée pour un pixel dans une fenêtre 3×3	42
Figure 3.6. Exemple d'un fichier d'apprentissage.....	46
Figure 3.7. Exploitation du PMC entraîné pour la binarisation d'une image.....	49
Figure 4.1. Sigle de Netbeans [W2].....	51
Féfigure 4.2. Exemples documents de notre collection.....	52
Figure 4.3. Interface principale de notre application.....	52
Féfigure 4.4. Fenêtre de préparation des données d'apprentissage.....	53
Figure 4.5. Zones de textes permettant de choisir le nombre de clusters et la taille de la fenêtre.....	53
Figure 4.6. Affichage du regroupement.....	54
Figure 4.7. Affichage du regroupement avec ACP.....	55
Figure 4.8. Boite de dialogue «Enregistrer les vecteurs».....	56
Féfigure 4.9. Affichage de message a été enregistrés.....	56
Féfigure 4.10. Fenêtre de l'apprentissage.....	57
Figure 4.11. Lancement de l'apprentissage.....	57
Féfigure 4.12. Fenêtre du test.....	58
Figure 4.13. Affichage du résultat de binarisation.....	59

Figure 4.14. Mise en correspondance de l'image binarisée et l'image de vérité terrain..... 59
Figure 4.15. Affichage du résultat de comparaison en termes de F-measure. 60

Liste de tableaux

Tableau 1.1. Quelques méthodes de binarisation basées sur l'apprentissage profond	18
Tableau 4.1. Répartition des images de notre collection.....	52
Tableau 5.1. Résultats obtenus par différents fenêtres en termes de Fmeasure.	61

Table de matières

ملخص	i
Abstract	ii
Résumé	iii
Remerciement	iv
Dédicace	v
Table de figures	vi
Liste de tableaux	viii
Table de matières	1
Introduction générale	4
Chapitre 1. Binarisation d'images de documents	6
1. Introduction.....	6
2. La Binarisation.....	6
2.1. Définition.....	6
2.2. Binarisation vs Segmentation.....	7
3. Le seuillage.....	7
4. Les méthodes de binarisation basées sur le seuillage.....	8
4.1. Les méthodes globales.....	8
4.1.1. Seuillage globale fixe.....	8
4.1.2. Méthode de Kapur et al.....	9
4.1.3. Méthode d'Otsu.....	9
4.1.4. Méthode Isodata.....	10
4.2. Les Méthodes locales.....	10
4.2.1. Méthode de Niblack.....	10
4.2.2. Méthode Sauvolaet Pietikainen.....	11
4.2.3. Méthode Brensen.....	11
4.3. Les méthodes hybrides.....	11
5. Les méthodes de binarisation basées sur l'apprentissage automatique.....	11
5.1. Méthodes de binarisation basées sur l'apprentissage supervisé.....	12
5.1.1. Méthodes de binarisation neuronales.....	12
5.1.2. Méthodes de binarisation basées sur les SVM.....	13
5.2. Méthodes de binarisation basées sur l'apprentissage non-supervisé.....	14
5.2.1. Méthodes basées sur les cartes de Kohonen.....	14
5.2.1. Méthodes basées sur l'algorithme K-means.....	15

5.3.	Techniques d'apprentissage en profondeur.....	16
6.	Conclusion.....	18
	Chapitre 2. L'apprentissage automatique.....	20
1.	Introduction.....	20
2.	L'apprentissage automatique.....	20
2.1.	Définition de l'Apprentissage automatique.....	20
2.2.	Objectifs de l'Apprentissage automatique.....	21
2.3.	Applications de l'Apprentissage automatique.....	21
2.3.1.	Classification d'images.....	22
2.3.2.	Reconnaissance de la parole.....	22
2.3.3.	Reconnaissance optique de caractères.....	22
2.3.4.	Autres.....	22
3.	Approches de l'Apprentissage automatique.....	23
3.1.	Apprentissage supervisé.....	24
3.2.	Apprentissage non supervisé.....	25
3.2.1.	Regroupement ou Clustering.....	25
3.2.2.	Association.....	26
3.3.	Apprentissage par renforcement.....	26
4.	Algorithmes d'apprentissage automatique.....	27
4.1.	Algorithmes d'apprentissage supervisé.....	27
4.1.1.	Réseaux de neurones artificiels.....	27
4.1.2.	SVM.....	30
4.1.3.	Arbre de décision.....	31
4.1.4.	La régression.....	31
4.1.5.	KPPV (KPlus Proche Voisins).....	31
4.2.	Apprentissage non supervisé.....	32
4.2.1.	K-means.....	32
4.2.2.	Classification hiérarchique.....	32
4.2.3.	Fuzzy kmeans(K-moyennes flou).....	32
4.2.4.	Les cartes de Kohonen.....	33
5.	Le deep learning (apprentissage en profondeur).....	33
5.1.	Présentation.....	33
5.2.	Réseaux de neurones profond.....	34
5.3.	Réseaux de neurones convolutionnel.....	34
6.	Conclusion.....	35
	Chapitre 3. Conception.....	36
1.	Introduction.....	36

2.	Caractéristiques des documents dégradés.....	36
3.	Motivations.....	37
4.	Description de l'approche proposée.....	38
4.1.	Préparation des données d'apprentissage.....	39
4.1.1.	Transformation de l'image en niveaux de gris.....	41
4.1.2.	Extraction des pixels d'apprentissage.....	41
4.1.3.	Enregistrement des données d'apprentissage.....	45
4.2.	Définition et apprentissage du PMC.....	46
4.2.1.	Définition du PMC.....	46
4.2.2.	Apprentissage du PMC.....	48
4.3.	Évaluation et exploitation du PMC.....	49
5.	Conclusion.....	49
	Chapitre 4. Implémentation et résultats.....	50
1.	Introduction.....	50
2.	Langage de programmation, Environnement de développement et bibliothèques.....	50
2.2.	Environnements de développement.....	50
2.2.1.	Environnement matériel.....	50
2.2.2.	Environnement logiciel.....	51
3.	Collection d'images utilisée pour l'apprentissage et le test.....	51
4.	Présentation de l'application.....	52
4.1.	Fenêtre de préparation des données d'apprentissage.....	53
4.2.	Fenêtre de l'apprentissage du PMC.....	56
4.3.	Fenêtre du test.....	58
5.	Résultats obtenus.....	60
6.	Conclusion.....	61
	Conclusion générale.....	62
	Références.....	63

Introduction générale

L'utilisation de l'image numérique dans notre vie quotidienne est devenue une nécessité car elle constitue un support et une source d'informations objectives. Pour cela, le domaine très important du traitement de l'image connaît une progression importante liée à l'évolution des technologies de l'information et de la communication. Le traitement d'image est né de l'idée de la nécessité de remplacer l'observateur humain par la machine.

Dans la plupart de temps, le traitement automatique d'images passe par une étape de binarisation. Cette dernière est une étape fondamentale ayant pour objectif d'isoler dans l'image, les objets à quantifier ou à reconnaître. Elle permet ainsi d'obtenir une information sur la composition ou sur la structure de l'image, et peut en fournir une analyse en termes de zone ou de régions qui la constituer. La technique la plus usuelle de binarisation est le seuillage. Ce dernier consiste à transformer l'image codée sur 6, 8 ou 16 bit, en une image binaire (en noir et blanc uniquement) où les pixels noirs correspondent aux objets et les pixels blancs forment le fond de l'image.

A cause de son importance, la binarisation d'images a attiré l'attention des chercheurs depuis longtemps. Un grand nombre de méthodes de binarisation, spécialement basées sur le seuillage, ont été proposées dans la littérature et ce nombre augmente continuellement chaque année. Chacune des méthodes proposées est spécifique à un type particulier d'images .

L'objectif du présent travail s'inscrit dans cette démarche. Nous proposons dans ce mémoire une nouvelle méthode de binarisation qui ne se base pas sur le seuillage comme la plupart des méthodes de la littérature, mais sur l'apprentissage automatique (la méthode Kmeans et les réseaux de neurones artificiels). Nous nous intéressons dans ce travail aux images de documents dégradées le type de document le plus difficile à traiter à cause de caractéristiques de dégradations et de détériorations rendant leur traitement difficile à plusieurs niveaux.

Le présent mémoire est organisé en quatre chapitres comme suit :

Chapitre 1. Binarisation d'images de documents

Ce chapitre est consacré à la présentation du problème de la binarisation d'images. Il survole les différentes approches de binarisation et expose les techniques les plus fréquemment citées dans la littérature.

Chapitre 2 : Apprentissage automatique

Le deuxième chapitre introduit les notions de base de l'apprentissage automatique et décrit les techniques d'apprentissage automatique les plus connues tout en focalisant sur les réseaux de neurones artificiels.

Chapitre 3 : Conception

Le troisième chapitre énumère les motivations auxquelles nous avons partie, présente l'architecture générale de notre méthode de binarisation et détaille les différentes étapes incluses.

Chapitre 4 : Implémentation et résultats

Ce chapitre illustre l'implémentation de notre application, la collection de test et les résultats obtenus.

Chapitre 1. Binarisation d'images de documents

1. Introduction

La plupart des documents papier sont convertis en version numérique à l'ère numérique. De grandes quantités de papiers sont conservées pendant le processus de transformation à l'aide de la numérisation électronique. Ces documents peuvent être obtenus à partir de diverses sources, notamment des documents anciens, des archives judiciaires historiques, des rapports médicaux, des partitions musicales, des feuilles de palmier et des rapports liés à la sécurité. Les écrits anciens et historiques, en particulier, sont difficiles à lire en raison de la dégradation en termes de mauvais contraste et de la présence d'artefacts corrompus.

En effet, la numérisation des archives contribue à la protection de documents historiques, à les conserver et à les diffuser, mais elle n'est pas suffisante pour pouvoir les exploiter convenablement. Des outils spécifiques d'analyse, de recherche et de reconnaissance doivent être mis en place. Ces outils intègrent plusieurs étapes de traitement dont la binarisation est certainement une des étapes les plus importantes.

Cependant, la binarisation des documents dégradés a fait l'objet de beaucoup d'attention ces dernières décennies, et de nombreuses façons de traiter les problèmes et les obstacles liés à la binarisation des documents ont été créés [Sul,19]. Cela est l'objet de ce premier chapitre.

2. La Binarisation

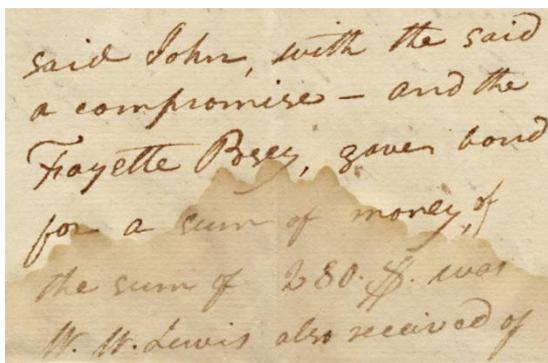
Une image binaire est une image numérique qui utilise seulement deux valeurs de couleur, généralement noir et blanc, pour représenter les pixels de l'image. Chaque pixel est assigné une valeur binaire, soit 1 (noir) ou 0 (blanc), et l'ensemble des pixels forment l'image binaire.

2.1. Définition

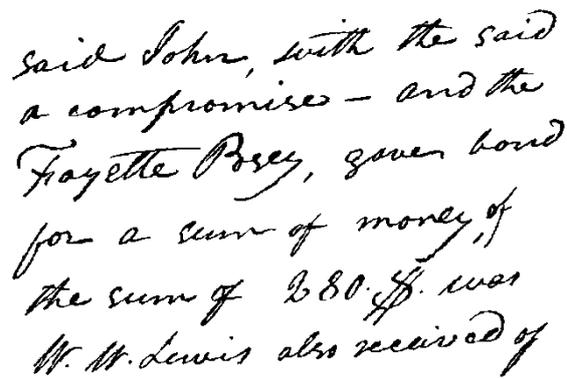
La binarisation d'une image est le processus de conversion d'une image en niveaux de gris ou en couleurs, en une image binaire, où chaque pixel est assigné à une des deux valeurs possibles (généralement noir et blanc) [Sul,19]. La figure 1.1. illustre un exemple du résultat de binarisation.

Selon Saini et al. [Sai,15], la binarisation des images est un prétraitement essentiel pour de nombreux systèmes de reconnaissance de caractères automatique. Les méthodes couramment utilisées pour la binarisation incluent les méthodes basées sur le seuillage (seuillage global ou local) et les méthodes basées sur l'apprentissage automatique.

Pour le cas des images de documents dégradés, la binarisation est un défi technique en raison de la présence de bruit, de la décoloration et de la déformation dans l'image. Les méthodes couramment utilisées pour surmonter ce défi incluent l'utilisation de filtres, de seuillage adaptatif et de méthodes de traitement d'image basées sur l'apprentissage automatique.



(a)



(b)

Figure 1.1. Résultat de la binarisation d'une image de document, (a) Image originale, (b) Image binarisée

2.2. Binarisation vs Segmentation

La binarisation et la segmentation sont deux techniques couramment utilisées en traitement d'image pour extraire des informations utiles à partir des images. La segmentation est le processus de découper une image en différentes régions ou parties, chacune ayant une signification ou une caractéristique distincte.

La binarisation est souvent utilisée comme prétraitement pour la segmentation d'images, notamment pour les images dégradées. Cela est dû au fait que la binarisation permet de simplifier l'image en noir et blanc.

Donc la distinction entre la binarisation et la segmentation est que la binarisation transforme une image en niveaux de gris en une image dont les valeurs de pixel ne peuvent être que de 1 ou 0, ce qui donne deux classes, alors que la segmentation peut créer plusieurs classes [Sam,15].

3. Le seuillage

Le seuillage est une technique couramment utilisée en traitement d'image pour la binarisation d'images. Il consiste à définir une ou plusieurs valeurs de seuil pour convertir une image en noir et blanc en fonction de la luminosité des pixels.

Il existe de nombreuses méthodes de seuillage, qui peuvent être réparties en trois grandes classes: seuillage global, seuillage adaptatif, seuillage multi-niveaux.

Le seuillage global consiste à utiliser une seule valeur de seuil pour convertir tous les pixels de l'image. Cette méthode est simple à mettre en œuvre mais peut ne pas être efficace pour les images avec des niveaux de luminosité variables.

Le seuillage adaptatif, d'autre part, utilise une valeur de seuil différente pour chaque pixel ou groupe de pixels, en fonction de la région de l'image. Ce qui permet de s'adapter aux variations de luminosité dans l'image [Ba,16].

Le seuillage multi-niveaux est une technique qui consiste à utiliser plusieurs seuils pour convertir une image en plusieurs niveaux de gris. Cette technique est utilisée pour les images avec des niveaux de luminosité variables, elle permet une meilleure séparation des différents objets dans l'image [Ba,16].

4. Les méthodes de binarisation basées sur le seuillage

Nous présentons dans cette section quelques méthodes de binarisation, et plus spécifiquement les méthodes de seuillage, les plus connues dans la littérature.

4.1. Les méthodes globales

Les méthodes globales de binarisation sont basées sur l'utilisation d'une seule valeur de seuil pour convertir tous les pixels d'une image en noir et blanc. Ces méthodes sont généralement simples à mettre en œuvre car elles ne nécessitent pas de calculer des valeurs de seuil pour chaque pixel ou groupe de pixels individuellement.

Cependant, diverses techniques ont été employées pour choisir le seuil global, ce qui nous permet de distinguer différents groupes de méthodes de seuillage global, en fonction de l'approche employée. Nous décrivons dans cette partie quelques méthodes de seuillage global des plus connues.

4.1.1. *Seuillage globale fixe*

Il s'agit d'une approche et d'une méthodologie simplifiée de binarisation qui consiste à comparer le niveau de gris de chaque pixel d'image x_i à un seuil global fixe T (par exemple 127).

4.1.2. Méthode de Kapur et al.

La méthode de Kapur et al. [Kap,85] exploite l'entropie de distribution des niveaux de gris pour l'estimation d'un seuil global de binarisation. Cette méthode prend en compte la distribution de probabilité de l'objet P_f et la distribution de probabilité du fond P_b dans la détermination de l'entropie de division. Le seuil de binarisation T est choisi dont l'entropie $H = H_f + H_b$ soit maximale, avec :

$$H_f(t) = -\sum_{i=0}^t \frac{p_i}{P_f(t)} \times \log\left(\frac{p_i}{P_f(t)}\right) \quad \text{et} \quad H_b(t) = -\sum_{i=t+1}^{255} \frac{p_i}{P_b(t)} \times \log\left(\frac{p_i}{P_b(t)}\right)$$

4.1.3. Méthode d'Otsu

La méthode d'Otsu [Ot,79] est une méthode globale de seuillage basée sur le regroupement des niveaux de gris. Cette méthode classe les pixels en deux catégories : premier-plan et arrière-plan en fonction d'un seuil unique. La méthode d'Otsu sélectionne alors le meilleur seuil pour séparer l'image en deux classes. La valeur du seuil est choisie de façon à réduire la variance intra-classe tout en maximisant la variance inter-classe. Le calcul de la variance interclasses ou intra-classes est basé sur l'histogramme normalisé M de l'image

$$\text{d'où } \sum_{i=1}^{255} M_i = 1.$$

La variance pondérée inter-classe de deux classes est calculée à l'aide de l'équation suivante:

$$\sigma_{inter}^2 = q_f(t) \times q_b(t) \times [\mu_f(t) - \mu_b(t)]^2$$

Tel que:

$$\mu_f(t) = \frac{1}{q_f(t)} \sum_{i=0}^{t-1} i \times M_i \quad \text{et} \quad \mu_b(t) = \frac{1}{q_b(t)} \sum_{i=t}^{255} i \times M_i$$

$$q_f(t) = \sum_{i=0}^{t-1} M_i \quad \text{et} \quad q_b(t) = \sum_{i=t}^{255} M_i$$

Seules les images avec un histogramme bimodal fonctionnent bien avec la méthode d'Otsu. En dehors de cela, elle échoue avec des images dégradées, ayant un éclairage inégal, ... comme le cas des images de documents anciens [Ot,79].

4.1.4. Méthode Isodata

Le seuillage ISODATA [Vel,80] consiste à déterminer un seuil en découpant itérativement l'histogramme en deux classes avec connaissance préalable des valeurs associées à chaque classe.

Cette approche commence par diviser la plage de valeurs non nulles de l'histogramme en deux segments de taille égale. Il utilise ensuite m_1 et m_2 comme moyennes arithmétiques pour chaque classe. Mettre à jour les deux moyennes m_1 et m_2 , et répéter le calcul du seuil jusqu'à convergence, le seuil T calculé comme l'entier le plus proche de $(m_1+m_2)/2$ [Ten,20].

4.2. Les Méthodes locales

Les méthodes locales de seuillage sont basées sur l'analyse des propriétés des pixels individuels ou de petits groupes de pixels. Ces méthodes examinent les caractéristiques des pixels dans une région spécifique de l'image et utilisent ces informations pour déterminer la valeur binaire appropriée pour chaque pixel. Tout comme les méthodes globales, les méthodes de seuillage local utilisent différentes techniques et exploitent diverses informations afin de calculer les valeurs de seuils. Nous présentons brièvement dans cette section quelques méthodes locales les plus populaires.

4.2.1. Méthode de Niblack

La méthode de Niblack [Nib,86] est une méthode de seuillage local basée sur la variation locale d'illumination dans l'image. Ainsi, un seuil est calculé pour chaque pixel à l'aide des données locales sur l'image. Dans la méthode de Niblack, une fenêtre rectangulaire est glissée sur l'image en niveaux de gris pour estimer le seuil de pixel. Le seuil est estimé à l'aide de la moyenne et de l'écart type des statistiques locales de la fenêtre. Le seuil dans le pixel (i,j) , noté $T(i,j)$ est approché comme indiqué dans l'équation suivante :

$$T(i, j) = \mu + k \times \sigma$$

Tel que μ représente la moyenne de la fenêtre et σ représente l'écart type de la fenêtre. La valeur constante k est un paramètre déterminant la qualité de la binarisation.

Étant donné que cette méthode dépend des caractéristiques locales de l'image, elle est influencée par les zones vides de l'image et est inefficace pour les images avec un bruit de fond [Cha,16].

4.2.2. Méthode Sauvolaet Pietikainen

La méthode de Sauvolaet Pietikainen [Sau,97] est une amélioration de la méthode de Niblackpour donner plus de performance dans les documents présentant trop de variations et illumination inégal. Ainsi, le seuil local $T(i, j)$ est calculé par l'équation suivante.

$$T(i, j) = \mu * [1 + k(\sigma / R) - 1]$$

Où μ est la moyenne de la fenêtre et σ son écart type. Les valeurs proposées par les auteurs pour k et R sont respectivement 0,5 et 128.

La qualité de l'image est affectée par la taille de la fenêtre et la valeur de k , mais R a peu d'effet. Cette procédure convient aux documents présentant un éclairage irrégulier, une texture claire ou qui sont décolorés. Cependant, la méthode de Sauvola et Pietikainen affine le texte après son application[Cha,16].

4.2.3. Méthode Bensen

La méthode de Bensen [Ber,86] utilise le contraste de l'image. Le seuil est calculé comme la moyenne des valeurs d'intensité les plus élevées et les plus faibles de la fenêtre. Cette équation détermine le contraste local de la fenêtre.

$$C(i,j) = I_{max} - I_{min}$$

En comparant le contraste local avec la valeur de seuil, les pixels sont classés en avant-plan ou en arrière-plan. Si le contraste local s'avère inférieur au seuil, le pixel sera étiqueté comme arrière-plan, et vice versa. La technique de Bensen échoue avec des images avec des arrière-plans plus compliqués[Cha,16].

4.3. Les méthodes hybrides

Il existent également des techniques de binarisation hybrides qui combinent des méthodes de binarisation locales et globales.

Par exemple, Wu et al. [Wu,03] propose une méthode de binarisation en deux étapes qui consiste à appliquer d'abord un seuil global puis à appliquer un seuil local modifié en fonction des informations spatiales incluses dans le document.

De plus, certains travaux utilisent un seuillage global pour annuler les effets d'un seuillage local précédent[Afr,15].

5. Les méthodes de binarisation basées sur l'apprentissage automatique

Comme nous avons avancé, le seuillage constitue l'approche de binarisation la plus répandue dans la littérature mais elle n'est pas la seule. Il en existent d'autres. En effet, la binarisation des images avec fond bruité et irréguliers est difficile en utilisant des techniques

conventionnelles de seuillage global et local. Une approche alternative pour surmonter ces défis consiste à utiliser des techniques d'apprentissage automatique. De nombreuses tâches de binarisation de documents ont été largement utilisées avec une variété de techniques d'apprentissage automatique.

Les méthodes basées sur l'apprentissage automatique ont été introduites pour résoudre les difficultés rencontrées par les méthodes traditionnelles de binarisation qui ne sont pas toujours efficaces pour les images complexes avec des variations de luminosité et des niveaux de gris irréguliers. Nous présentons dans cette section quelques unes d'entre elles.

5.1. Méthodes de binarisation basées sur l'apprentissage supervisé

5.1.1. Méthodes de binarisation neuronales

Les méthodes de binarisation neuronales sont des techniques qui utilisent des réseaux de neurones artificiels (RNA) pour classer les pixels d'une image en deux catégories : *fond* et *objet*. Les réseaux de neurones utilisés pour la binarisation peuvent être des perceptrons multi-couches (MLP pour *Multi Layer Perceptron*), des cartes de Kohonen (SOFM pour *Self-Organized Feature Map*), des réseaux de neurones profonds (DNN), ...

Chen et Takagi [Che,93] ont proposé de déterminer un seuil global optimal en utilisant un MLP entraîné par un algorithme de rétro-propagation modifié. La technique commence par le calcul de l'histogramme de l'image contrastée. L'histogramme est ensuite mis à l'échelle dans [0, 1] et divisé en 128 parties. Le MLP utilisé a une couche d'entrée de 128 neurones recevant les 128 parties de l'histogramme normalisé, 2 couches cachées de 256 et 512 neurones respectivement, et une couche de sortie ayant 128 neurones.

Dans [Kha,07], les auteurs ont combiné une méthode de seuillage global, notamment le seuillage MD (pour Mass-Difference thresholding) et un RNA supervisé dans le but de sélectionner la valeur du seuil global la plus optimale. Chaque image d'apprentissage est d'abord subdivisée en plusieurs sous-images de taille 32×32 pixels. Ensuite, l'algorithme MD est appliqué sur chacune des sous-images produisant plusieurs valeurs de seuils locales, et sur l'image entière résultant un seul seuil global. Après, le RNA supervisé est entraîné en utilisant les valeurs des seuils locaux comme entrées et la valeur du seuil global comme sortie. Le RNA entraîné est utilisé par la suite pour trouver le seuil global le plus optimal pour d'autres images.

Làzaro et al. [Laz,10] ont proposé d'obtenir un seuil optimal pour chaque image en utilisant un RNA à régression générale et une description sémantique de son histogramme.

L'histogramme de l'image d'entrée est d'abord lissé afin d'éliminer les faux minima et maxima et sa dérivée discrète est obtenue. En utilisant une version polygonale de la dérivée et l'histogramme lissée, une nouvelle description de l'histogramme est calculée, à partir de laquelle une description sémantique est inférée. Les descriptions sémantiques des images d'apprentissage constituent les entrées du RNA et les valeurs de seuils optimales correspondantes représentent les sorties désirés.

5.1.2. Méthodes de binarisation basées sur les SVM

Les méthodes de binarisation basées sur les SVM (Support Vector Machines) utilisent les propriétés des machines à vecteurs de support pour séparer les pixels d'une image en deux catégories : les pixels de fond et les pixels de texte. Les SVM sont des outils d'apprentissage supervisé qui peuvent être utilisés pour résoudre des problèmes de classification binaire.

Plusieurs méthodes basées sur les SVM ont été proposées pour résoudre les problèmes de binarisation d'images, telles que les méthodes basées sur les noyaux [Su,07] et les méthodes basées sur les données de texture [Fan,08].

Une autre méthode basée sur les SVMs pour la binarisation d'images de documents a été décrite dans [Cho,10]. La méthode proposée divise l'image en plusieurs régions et décide comment binariser chaque région séparément. Les règles de décision sont déterminées par un SVM entraîné en utilisant des caractéristiques extraites de chaque région. Ainsi, trois caractéristiques sont utilisées : le seuil d'Otsu, la moyenne, et l'écart-type.

Kita et Wakahara [Koh,10] ont combiné l'algorithme k-means avec un SVM pour la binarisation de caractères multi-couleurs dans les images de scènes contenant diverses dégradations et fonds complexes. La méthode proposée est composée de trois étapes. La première étape est la génération provisoire des images binarisées via chaque dichotomisation des K classes obtenues de la classification en utilisant l'algorithme K-means des pixels dans l'espace de couleurs HSI (Teinte, Saturation, Intensité). Le nombre de classes K est déterminé en avance. La deuxième étape consiste à utiliser un SVM pour déterminer si chaque image binarisée provisoire représente ou non un caractère et à quel degré. Deux caractéristiques extraites de chaque image binarisée provisoire ont été utilisées pour entraîner le SVM, et le SVM estime le degré d' « apparence de caractère ». Dans la dernière étape, on sélectionne l'image binarisée ayant le maximum degré d' « apparence de caractère » comme le résultat de binarisation optimal.

Il est important de noter que les méthodes de binarisation basées sur les SVM peuvent nécessiter un grand nombre de paramètres pour être ajustées correctement et peuvent

également souffrir de biais de données si les données d'apprentissage ne sont pas suffisamment variées.

5.2. Méthodes de binarisation basées sur l'apprentissage non-supervisé

Les méthodes de binarisation basées sur l'apprentissage non-supervisé sont des techniques qui utilisent des algorithmes d'apprentissage automatique, notamment des algorithmes de clustering, pour apprendre à classer les pixels d'une image en deux catégories : *fond* et *objet*, sans utiliser d'étiquettes préalablement définies.

5.2.1. Méthodes basées sur les cartes de Kohonen

Les cartes auto-organisatrices de Kohonen (SOFM), sont une catégorie de réseaux de neurones non-supervisés dédiées à la classification automatique (clustering).

N. Papamarkos et al. dans [Pap,00] ont proposé une approche de multi-seuillage utilisant l'analyse en composantes principales (ACP) et un RNA de type carte de Kohonen (SOFM pour Self-Organized Feature Map). La couche d'entrée du SOFM code les 255 éléments de l'histogramme de niveaux de gris et la couche de sortie est une carte 1D dans laquelle un seul neurone gagnant est activé pour chaque vecteur d'entrée.

Dans [Pap,03], l'auteur a proposé une technique neuronale floue basée sur la combinaison d'une carte de Kohonen (SOFM) et de l'algorithme C-moyens flous (FCM pour fuzzy C-means) pour la binarisation d'images de documents en niveaux de gris et en couleurs. La technique proposée utilise les couleurs des pixels en plus d'autres caractéristiques spatiales locales extraites à partir du voisinage de chaque pixel pour entraîner une SOFM. Après l'apprentissage, deux classes de neurones de sortie sont obtenues. En utilisant le contenu de ces classes, deux fonctions d'appartenance floues sont définies et utilisées par la suite par l'algorithme FCM afin de déterminer la valeur binaire finale de chaque pixel.

[Ham,05] a introduit trois approches de binarisation neuronales. Les trois approches utilisent une carte de Kohonen (SOFM) entraînée sur une partie de l'image afin d'extraire les niveaux de gris ou les couleurs les plus représentatives, suivie d'un autre classifieur. Les entrées de la SOFM sont les pixels de la sous-image. Dans la première approche, les neurones de la SOFM sont regroupés en 2 régions en utilisant l'algorithme Kmeans. Une fois les pixels sont regroupés, la SOFM est utilisé pour classifier les pixels de l'image entière. Chaque pixel est donné comme entrée à la SOFM en vue d'être classifié. La deuxième approche utilise un MLP conjointement à la SOFM pour la binarisation d'images couleurs. La SOFM est d'abord entraîné sur une partie de l'image et ses neurones sont regroupés en utilisant l'algorithme Kmeans comme auparavant. Le MLP est ensuite entraîné en se basant sur les neurones

étiquetés. Les entrées du MLP correspondent aux neurones de la SOFM alors que les sorties sont les classes des mêmes neurones. Après l'apprentissage, le MLP est appliqué sur l'image entière en vue de classifier tous les pixels. Dans la troisième approche, les seuils de Sauvola ou Niblack sont appliqués sur les neurones de la SOFM entraînée, et un seuil global est extrait. Ce dernier est appliqué pour binariser l'image entière.

Padekas et Papamarkos [Bed,07] ont proposé un système pour la binarisation d'images de documents imprimés normales et dégradés basé sur la combinaison des résultats de plusieurs algorithmes de binarisation en utilisant une carte de Kohonen (SOFM). Premièrement, les valeurs les plus optimales de paramètres des techniques de binarisation sont estimées. Ensuite, la SOFM prend en entrée les résultats de binarisation obtenus par les différentes techniques en utilisant les valeurs de paramètres estimées. Après l'apprentissage, la SOFM est utilisée pour classifier chaque pixel de l'image en Objet ou Fond.

5.2.1. Méthodes basées sur l'algorithme K-means

L'algorithme K-means a été utilisé principalement pour l'extraction de l'avant-plan dans les images de documents couleurs.

Bottou et al. [Bot,98] ont implémenté une séparation entre l'avant-plan et l'arrière-plan dans le cadre d'une nouvelle méthode de compression d'images de documents baptisée « DjVu ». L'approche proposée est basée sur l'algorithme k-means en considérant plusieurs grilles de résolutions croissantes. Dans cette méthode, chaque grille délimite des blocs carrés de pixels et la taille des blocs d'une grille est une fraction de la taille de blocs de la grille précédente. L'algorithme k-means est appliqué sur les blocs de la première grille pour trouver les deux couleurs dominantes (une de l'avant-plan et l'autre de l'arrière-plan) dans chaque bloc. Les blocs de la grille suivante sont ensuite traités par un algorithme de regroupement de couleurs légèrement modifié et ce traitement continue jusqu'à la convergence. Finalement, plusieurs filtres ont été appliqués en se basant sur divers critères dans le but de nettoyer les composantes de l'avant-plan et d'éliminer les erreurs de classification.

Leydier et al. [Ley,04] ont proposée une méthode de séparation avant-plan/ arrière-plan basée sur une sérialisation de l'algorithme k-means appliqué séquentiellement sur l'image dans une fenêtre glissante. Cependant, l'approche implique une intervention manuelle pour décider le nombre de classes et de nuées ainsi que de leurs centres initiaux. Lors du déplacement de la fenêtre, l'algorithme réutilise des informations provenant des nuées calculées dans la fenêtre précédente et les ajuste automatiquement pour adapter le classifieur à toute variation locale des couleurs.

[Gar,06] décrit une méthode basée sur l'algorithme K-means pour la séparation avant-plan/ arrière-plan appliquée sur des images en niveaux de gris et en couleurs. La méthode proposée commence par l'étiquetage des composantes connexes de l'image pour capturer les pixels de couleurs similaires spatialement. Après, les composants du fond dominant sont déterminés en se basant sur leur taille et l'image entière est divisée en blocs rectangulaires de différentes tailles. Ces blocs représentent l'uniformité locale d'illumination, fond, etc. et les parties d'avant-plan respectives sont traitées contre ces uniformités locales. Puis, les blocs sont arrangés sous forme d'arbre et connectés hiérarchiquement. Des relations père-fils sont établies entre les blocs en se basant sur leur taille et leur disposition géométrique. Les blocs dans l'arbre sont par la suite traités d'une manière descendante. Les composantes connexes dans chaque bloc subissent une classification bi-couleurs par l'algorithme k-means dont l'initialisation est faite en utilisant les résultats de classification obtenus du traitement de son parent. La combinaison des résultats du traitement de tous les blocs fournit l'image binarisée finale.

5.3. Techniques d'apprentissage en profondeur

Dans les approches traditionnelles de binarisation d'image, le(s) seuil(s) approprié(s) est(sont) généralement trouvé(s) à l'aide de l'histogramme ou de statistiques locales.

En apprenant des caractéristiques dépendant du contexte qui sont moins sujettes aux erreurs de dégradation qui se produisent souvent dans les images de documents, les techniques d'apprentissage en profondeur sont capables de produire, théoriquement, des versions binarisées des images meilleurs que celles obtenues par les techniques classiques de binarisation [Suk,22]. Ainsi, Ntirogiannis, K. et al. [Nti,12] ont fait une étude qui soutient cette supposition. Il a comparé différentes méthodes de binarisation pour les images de documents historiques et ont montré que les méthodes neuronales ont obtenu les meilleurs résultats pour la binarisation de ces images.

Parmi ces techniques, les réseaux de neurones convolutionnels (CNN) qui ont montré des résultats prometteurs dans la binarisation d'images.

Par exemple, dans [Be,11], les auteurs ont proposé un descripteur binaire profond pour la reconnaissance d'objets basé sur un réseau de neurones profond qui a montré des résultats de reconnaissance d'objet comparables à ceux obtenus par les descripteurs de point de référence.

Saidane et Garcia. [Sai,07] ont utilisé un réseau de neurones de convolution [Gar,04] pour la binarisation d'images de texte couleurs. Le réseau de neurones est entraîné sur un

ensemble d'images couleurs fournies avec leur image binaire correspondante en utilisant l'algorithme de rétro-propagation.

D'autres méthodes de binarisation basées sur l'apprentissage profond sont récapitulées dans le tableau suivant:

Référence	Idée principale	Technique	Inconvénients
[Wu,20]	Dans cette méthode, des échantillons de documents et leur vérité terrain associée sont utilisés pour former un réseau neuronal multicouche pour la classification de documents au niveau des pixels. Le réseau de neurones est capable de produire des résultats de binarisation adaptative et extrait des informations des pixels en tenant compte de leur voisinage.	Simple réseau neurone multicouche	Nécessite néanmoins davantage de données d'entraînement et un temps d'entraînement adéquat.
[Wes,18]	Ils ont séparé l'image du document d'entrée de 64×64 pixels en parties non superposées. Le modèle RNN est alors pensé pour utiliser ces blocs comme séquence d'entrée. La séquence de caractéristiques de niveau intermédiaire est créée en combinant et en alignant la sortie de quatre couches LSTM de grille indépendantes (L 1) après la lecture de chaque bloc. La carte des fonctionnalités de haut niveau est créée en introduisant ces fonctionnalités de niveau intermédiaire dans deux couches LSTM de grille distinctes (L 2) et	RNN	temps d'exécution élevé

	un LSTM bidirectionnel. Le résultat de la binarisation a ensuite été créé en appliquant une couche de connexion complète (L 3) à la carte des fonctionnalités de haut niveau.		
[Vo,18]	L'architecture du réseau a deux objectifs principaux. L'objectif principal de ce réseau est d'utiliser des fonctionnalités de haut niveau pour séparer le premier plan (texte) de l'arrière-plan bruité. Son objectif secondaire est de maintenir l'excellente qualité visuelle des informations de premier plan (texte) tout en traitant l'arrière-plan bruité et d'autres problèmes.	Deep supervised network (DSN)	Architecture complexe et peu exhaustive
[Ten,17]	Un modèle a été développé qui tente d'effectuer une tâche de classification au niveau des pixels en minimisant les fonctions de perte P-FM et FM combinées. Le modèle prend en entrée des caractéristiques d'obscurité relative en niveaux de gris et calculées localement et applique plusieurs couches d'encodage (fonction convolutive et sigmoïde) pour générer un masque de binarisation.	FCNN fully convolutional neural network	/

Tableau 1.1. Quelques méthodes de binarisation basées sur l'apprentissage profond

6. Conclusion

La binarisation d'images est un processus important en traitement d'images et en analyse et reconnaissance de documents, car elle permet de simplifier l'image en la rendant binaire, c'est-à-dire composée seulement deux niveaux de gris ou de couleurs. En effet, il existe plusieurs

méthodes de binarisation, chacune ayant ses propres avantages et inconvénients. Les méthodes de binarisation les plus populaires sont les méthodes basées sur le seuillage. Ce dernier peut être global, local, ou hybride. Cependant, l'inefficacité des techniques classiques de seuillage pour la binarisation d'images de documents dégradés, à fond complexe, et illumination inégale, a ouvert la porte vers des nouvelles techniques de binarisation qui reposent sur l'apprentissage automatique.

Dans le présent chapitre nous avons essayé de survoler le domaine de la binarisation d'images de documents tout en passant en revue les principales techniques et méthodes proposées dans la littérature.

En conclusion, il est important de choisir la méthode de binarisation appropriée en fonction des besoins de l'application et des caractéristiques de l'image.

Chapitre 2. L'apprentissage automatique

1. Introduction

Un sous-domaine de l'informatique appelé "apprentissage automatique" permet aux ordinateurs d'apprendre à partir de données sans avoir à être explicitement programmés. L'apprentissage supervisé et l'apprentissage non supervisé sont les deux principales sous-catégories de l'apprentissage automatique. L'apprentissage en profondeur est une méthode très appréciée pour effectuer des tâches telles que la reconnaissance de la parole et des images. Les différents techniques de l'apprentissage supervisé et non supervisé ainsi que les principes fondamentaux de l'apprentissage en profondeur sont examinés dans ce chapitre.

2. L'apprentissage automatique

L'apprentissage automatique, est un domaine de l'informatique qui a évolué à partir de l'étude de la reconnaissance des formes et de la théorie de l'apprentissage informatique (computational learning) en intelligence artificielle. Il s'agit de l'apprentissage et de la construction d'algorithmes capables d'apprendre à partir de jeux de données et de faire des prédictions sur ces derniers. Ces procédures fonctionnent par construction d'un modèle à partir d'exemples d'entrées afin de faire des prédictions ou des choix basés sur des données plutôt que de suivre des instructions de programme statiques et fermes.

2.1. Définition de l'Apprentissage automatique

Le terme "apprentissage automatique" a été inventé en 1959 par Arthur Samuel, un pionnier dans le domaine de l'intelligence artificielle et des jeux vidéo. Il a défini l'apprentissage automatique comme "*un domaine d'étude qui donne aux ordinateurs la capacité d'apprendre sans être explicitement programmés*".

L'apprentissage automatique ou machine learning (ML) désigne la capacité d'un système à acquérir et à intégrer des connaissances par le biais d'observations à grande échelle, ainsi qu'à s'améliorer et à s'étendre en apprenant de nouvelles connaissances plutôt qu'en étant programmé avec ces connaissances [Kid,16].

Une définition plus moderne a été fournie par Tom Mitchell dans son livre "Machine Learning": "On dit d'un programme informatique qu'il apprend de l'expérience E en ce qui

concerne une certaine tâche T et une certaine mesure de performance P, si sa performance sur T, telle que mesurée par P, s'améliore avec l'expérience E." [Mit,07].

Ainsi, si nous voulons que notre programme prévoie, par exemple, les formes de trafic à un nœud occupé (tâche T), nous pouvons le soumettre à un processus d'apprentissage automatique avec des données sur les formes de trafic précédentes (expérience E) et, s'il a "appris" avec succès, il sera alors plus performant pour prédire les formes de trafic à venir (mesure de performance P).

2.2. Objectifs de l'Apprentissage automatique

L'apprentissage automatique qui est l'un des sous-domaines de l'intelligence artificielle, a pour objectif principal d'apprendre automatiquement à généraliser à partir d'exemples observés afin de pouvoir faire de l'inférence sur de nouveaux exemples jamais été observé auparavant.

Les autres objectifs de l'apprentissage automatique sont de fournir une plus grande précision de solution, une plus grande couverture des problèmes, une plus grande économie dans l'obtention de solutions et une plus grande simplicité dans la représentation des connaissances [McD,89].

Cependant, l'apprentissage automatique a plusieurs finalités, qui peuvent varier en fonction du domaine d'application et du contexte d'utilisation. Voici quelques-unes des finalités les plus courantes :

- **Prédiction:** consiste à prédire des valeurs à partir de données d'entrée.
- **Classification:** consiste à attribuer une étiquette à un objet ou à une observation.
- **Regroupement:** consiste à identifier des groupes d'observations similaires dans un ensemble de données.
- **Réduction de dimensionnalité:** consiste à réduire le nombre de variables dans un ensemble de données.
- **Optimisation:** tels que la maximisation des profits ou la minimisation des coûts.

2.3. Applications de l'Apprentissage automatique

L'apprentissage automatique est une technologie en pleine expansion, qui a de nombreuses applications dans différents domaines. Voici quelques exemples d'applications les plus courantes de l'apprentissage automatique :

2.3.1. Classification d'images

La classification d'images est l'une des applications les plus courantes de l'apprentissage automatique. Il s'agit de la tâche consistant à attribuer une étiquette à une image en fonction de son contenu. Par exemple, une image peut être classée comme étant un chien, un chat, un arbre ou une voiture.

La classification d'images est utilisée dans de nombreuses applications, telles que la reconnaissance faciale, la reconnaissance de plaques d'immatriculation, la détection de maladies à partir d'images médicales, etc.

L'apprentissage automatique peut être utilisé pour entraîner des algorithmes de classification d'images à partir d'un grand ensemble de données d'images annotées. Il permet d'améliorer considérablement la précision des applications en permettant aux algorithmes de classification d'images de reconnaître des motifs complexes dans les images et d'apprendre à partir de grandes quantités de données.

2.3.2. Reconnaissance de la parole

La reconnaissance de la parole ou la reconnaissance vocale est une application importante de l'apprentissage automatique qui permet aux machines de comprendre et d'interpréter la parole humaine. La reconnaissance de la parole est utilisée dans de nombreuses applications, notamment les assistants personnels tels que Siri d'Apple, Google Assistant, les systèmes de dictée, les centres d'appels, les systèmes de contrôle vocal pour les personnes handicapées, etc. L'apprentissage automatique joue un rôle crucial dans la reconnaissance de la parole en permettant aux machines de comprendre la parole humaine avec une grande précision et de s'améliorer continuellement grâce à l'apprentissage à partir de données.

2.3.3. Reconnaissance optique de caractères

L'apprentissage automatique est utilisé pour entraîner des algorithmes de reconnaissance optiques de caractères. La reconnaissance optique de caractères (OCR) est le processus permettant de convertir une image de texte en format de texte lisible par une machine. Par exemple, si vous scannez un formulaire ou un ticket de caisse, votre ordinateur enregistre la numérisation en tant que fichier image. Vous ne pouvez pas utiliser un éditeur de texte pour modifier le fichier, y faire une recherche ou en compter les mots. Par contre, vous pouvez utiliser l'OCR pour convertir l'image en document texte, dont le contenu sera stocké en tant que données textuelles.

2.3.4. Autres

Il existe en fait d'autres applications de l'apprentissage automatique.

Il est utilisé pour analyser de grandes quantités de données et en extraire des connaissances utiles. Par exemple, l'apprentissage automatique peut être utilisé pour analyser les données de ventes d'une entreprise et identifier les tendances de marché, les produits les plus rentables, les segments de clientèle les plus rentables, etc. Il est utilisé ainsi pour recommander des produits, des services, des films, des livres, etc.

Il est également utilisé pour entraîner des algorithmes de traduction automatique capables de traduire des textes d'une langue à une autre. Ces algorithmes sont utilisés dans des applications telles que les services de traduction en ligne, les logiciels de traduction assistée, etc.

3. Approches de l'Apprentissage automatique

L'apprentissage automatique, en anglais *machine learning*, est une forme d'intelligence artificielle (IA) qui permet à un système d'apprendre à partir des données et non à l'aide d'une programmation explicite. Cependant, l'apprentissage automatique n'est pas un processus simple. Au fur et à mesure que les algorithmes ingèrent les données de formation, il devient possible de créer des modèles plus précis basés sur ces données. Un modèle de machine learning est le résultat généré lorsque vous entraînez votre algorithme d'apprentissage automatique avec des données. Après la formation, lorsque vous fournissez des données en entrée à un modèle, vous recevez un résultat en sortie [Jud,18].

En effet, il existe plusieurs approches d'apprentissage automatique. Ces différentes approches se distinguent généralement par les types de problèmes qu'elles tentent de résoudre, ainsi que par le type et la quantité de retour d'information fourni. En gros, nous pouvons séparer trois approches d'apprentissage automatique : 1) l'apprentissage supervisé ; 2) l'apprentissage non supervisé ; 3) l'apprentissage par renforcement.

La figure suivante résume les trois approches d'apprentissage avec les problèmes connexes à résoudre :

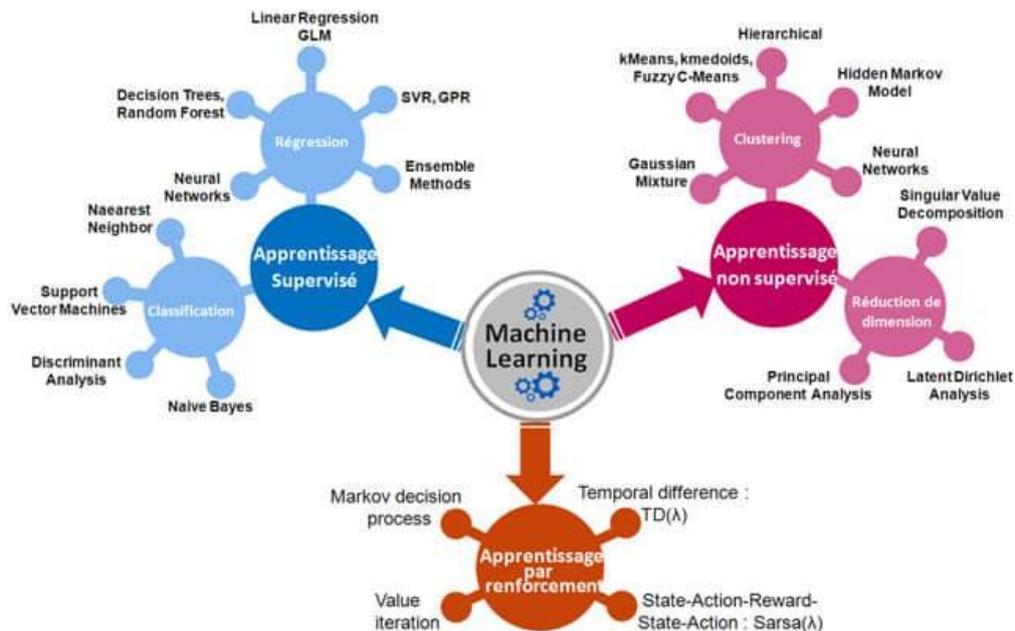


Figure 2.1: Types d'apprentissage automatique [W5]

3.1. Apprentissage supervisé

L'apprentissage supervisé est basé sur un certain nombre d'exemples pré-classifiés, dans lesquels est connu à priori la catégorie à laquelle appartient chacune des entrées utilisées comme exemples. Dans ce cas, la question cruciale est le problème de généralisation, après l'analyse d'un échantillon d'exemples, le système devrait produire un modèle qui devrait fonctionner pour toutes les entrées possibles.

L'ensemble de données pour l'entraînement, est constitué de données étiquetées, c'est-à-dire d'objets et de leurs classes associées. Cet ensemble d'exemples étiquetés constitue donc l'ensemble d'apprentissage.

Afin de mieux comprendre ce concept, prenons un exemple : un utilisateur reçoit chaque jour un grand nombre d'e-mails, certains sont des e-mails d'entreprises importants et d'autres sont des e-mails indésirables non sollicités ou des spam. Un algorithme supervisé sera présenté avec un grand nombre d'e-mails qui ont déjà été étiquetés par l'utilisateur comme *spam* ou *non-spam*. L'algorithme fonctionnera sur toutes les données étiquetées, faire des prédictions sur l'e-mail et voir si c'est un spam ou non. Cela signifie que l'algorithme examinera chaque exemple et fera une prédiction pour chacun pour savoir si l'e-mail est un spam ou pas. La première fois, l'algorithme fonctionne sur toutes les données non étiquetées, la plupart des emails seront mal étiquetés car il peut fonctionner assez mal au début. Cependant, après chaque exécution, l'algorithme compare sa prédiction au résultat souhaité (l'étiquette). Au fur et à mesure, l'algorithme apprendra à améliorer ses performances et sa précision [Mar,17].

Dans certains cas, le résultat n'est pas nécessairement discret et il se peut que nous n'ayons pas un nombre fini de classes dans lesquelles classer nos données. Par exemple, nous essayons peut-être de prédire l'espérance de vie d'un groupe de personnes en fonction de paramètres de santé préétablis. Dans ce cas, comme le résultat est une fonction continue (nous pouvons spécifier une espérance de vie comme un nombre réel exprimant le nombre d'années que la personne devrait vivre), nous ne parlons pas d'une tâche de classification mais plutôt d'un problème de régression [Djo,15].

Dans un problème de régression, l'ensemble d'apprentissage est une paire formée par un objet et une valeur numérique associée. Il existe plusieurs algorithmes d'apprentissage supervisé qui ont été développés pour la classification et la régression. Parmi tous, les arbres de décision, les règles de décision, les réseaux de neurones et les réseaux bayésiens [Kal,18].

La figure suivante illustre une comparaison entre le résultat d'un algorithme de classification et un algorithme de régression.

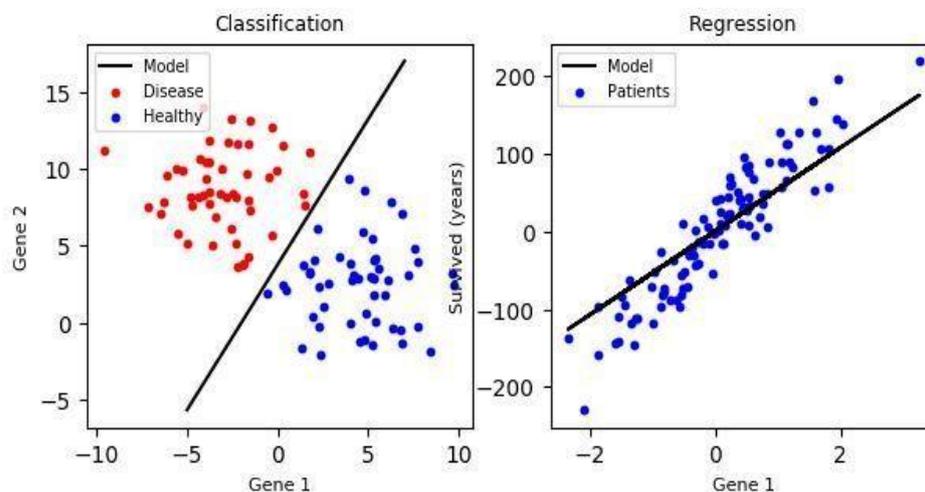


Figure 2.2. La classification/La régression[Acc,21]

3.2. Apprentissage non supervisé

Dans cette classe d'algorithmes d'apprentissage automatique, nous n'étiquetons pas les données au préalable, nous laissons plutôt l'algorithme arriver à sa conclusion.

Ce type d'apprentissage est important car il est beaucoup plus commun dans le cerveau humain que l'apprentissage supervisé [Mar,17].

L'apprentissage non supervisé comprend deux catégories d'algorithmes: Algorithmes de regroupement (clustering) et d'association.

3.2.1. Regroupement ou Clustering

Les algorithmes d'apprentissage non supervisé sont particulièrement utilisés dans les problèmes de clustering, dans lesquels, étant donné une collection d'objets, nous voulons être

en mesure de comprendre et de montrer leurs relations. Une approche standard consiste à définir une mesure de similarité entre deux objets, puis à rechercher tout groupe d'objets plus similaires les uns aux autres, par rapport aux objets des autres clusters. Par exemple, dans le cas des e-mails spam/ non-spam, l'algorithme peut être capable de trouver des éléments communs à tous les spam (par exemple, la présence de mots mal orthographiés) [Djo,15].

Bien que cela puisse fournir une classification meilleure qu'aléatoire, il n'est pas clair que les spam/non-spam puissent être facilement séparés [Kal,18].

Dans la figure 3.3, on reconnaît trois clusters : (un en rouge, un bleu et un vert).

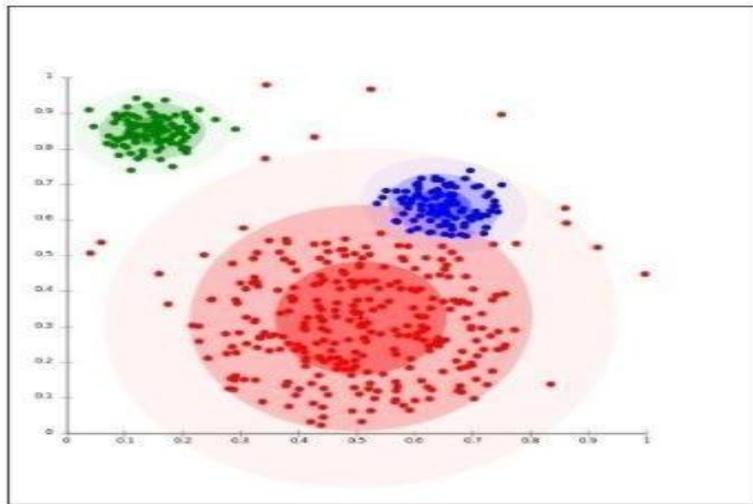


Figure 2.3. Exemple d'apprentissage automatique non-supervisé (Clustering) [Acc,21]

3.2.2. Association

L'association consiste à découvrir des relations intéressantes entre des variables dans de grandes bases de données. Par exemple, les personnes qui achètent une nouvelle maison ont aussi tendance à acheter de nouveaux meubles. Il découvre la probabilité de co-occurrence d'éléments dans une collection [Ac,21].

3.3. Apprentissage par renforcement

Selon [Ray,19] l'apprentissage par renforcement est une classe de problèmes d'apprentissage automatique dont le but est d'apprendre par expérience ce qu'il faut faire dans diverses circonstances afin de maximiser progressivement une récompense monétaire.

L'apprentissage par renforcement est une approche de l'intelligence artificielle qui met l'accent sur l'apprentissage du système à travers ses interactions avec l'environnement. Avec l'apprentissage par renforcement, le système adapte ses paramètres en fonction des réactions reçues de l'environnement, qui fournit ensuite un retour d'information sur les décisions prises [Mar,17].

Par exemple, un système qui modélise un joueur d'échecs qui utilise le résultat des étapes précédentes pour améliorer ses performances, est un système qui apprend avec le renforcement.

L'agent doit déterminer par lui-même ce qu'il doit faire en réponse aux conditions environnementales. Ses activités sont dictées par une récompense, bonne ou négative, qui l'amènera à essayer différents comportements pour voir lesquels rapportent le plus.

Lors de tests répétés, l'agent recherche le meilleur comportement décisionnel (appelé stratégie ou politique, qui est une fonction reliant l'état présent à l'action à entreprendre), visant à maximiser la somme des récompenses dans le temps [Wie,12].

La recherche actuelle sur l'apprentissage avec renforcement est hautement interdisciplinaire et comprend des chercheurs spécialisés dans les algorithmes génétiques, les réseaux de neurones, la psychologie et les techniques de contrôle [Kal,18].

4. Algorithmes d'apprentissage automatique

Les algorithmes d'apprentissage automatique sont utilisés à diverses fins telles que l'exploration de données, le traitement d'images, l'analyse prédictive, etc.

Comme il est avancé dans la section précédente, il existe trois classes d'algorithmes d'apprentissage. Nous allons explorer dans cette section les deux principales approches d'apprentissage automatique : supervisé et ses méthodes de classification, et non supervisé et ses méthodes de regroupement.

4.1. Algorithmes d'apprentissage supervisé

Si les classes sont prédéterminées et les exemples connus, le système apprend à classer selon un modèle de classification ou de classement, c'est ce qu'on appelle l'apprentissage supervisé [Ber,86]. Parmi les algorithmes d'apprentissage supervisé les plus connus on trouve:

4.1.1. Réseaux de neurones artificiels

Les réseaux de neurones artificiels (RNA) sont des systèmes informatiques inspirés à partir des réseaux de neurones biologiques et apprenant à partir d'énormes quantités de données en modifiant les poids et les biais en réponse aux signaux d'entrée [Kum,21].

L'idée de base derrière un RNA est de simuler le comportement d'un grand nombre d'unités de traitement simples, ou neurones, qui sont connectés les uns aux autres dans un réseau. Chaque neurone reçoit des signaux d'entrée d'autres neurones du réseau, traite ces informations à l'aide d'une fonction d'activation, puis envoie des signaux de sortie aux autres neurones du réseau. En ajustant la force de ces connexions entre les neurones, un RNA peut

apprendre à reconnaître des modèles dans les données et à faire des prédictions basées sur de nouvelles entrées [Kro,08].

L'apprentissage d'un RNA implique généralement de lui présenter un ensemble d'exemples étiquetés et d'ajuster les poids entre les neurones afin que le réseau produise des sorties précises pour ces entrées. Ce processus est souvent itératif, les pondérations étant ajustées sur plusieurs tours jusqu'à ce que le réseau atteigne des performances satisfaisantes sur un ensemble de validation [Kro,08].

Les RNA ont été appliqués à un large éventail de problèmes, notamment la reconnaissance d'images, la reconnaissance de la parole, le traitement du langage naturel et la modélisation prédictive [Kro,08].

En effet, il existe plusieurs types de réseaux de neurones artificiels, dont le Perceptron Multi-Couches (PMC) est le plus populaire. Dans cette section, nous nous focalisons sur la présentation de ce dernier, en décrivant son fonctionnement et son architecture. Avant de présenter le fonctionnement et l'architecture du Perceptron Multi-Couches, il est important de commencer par l'introduction du neurone formel, qui est l'unité de base de tout réseau de neurones.

a) Le Neurone Artificiel

Un neurone est une unité de calcul de base dans un réseau neuronal artificiel. Le neurone formel est un modèle mathématique qui reprend les principes fondamentaux du fonctionnement des neurones biologiques, notamment la sommation des entrées. Il reçoit des signaux d'entrée d'autres neurones ou de sources externes, traite ces informations à l'aide d'une fonction d'activation, puis envoie des signaux de sortie à d'autres neurones du réseau. [Cho,10]. Sachant que les synapses n'ont pas toutes la même "valeur" au niveau biologique (les connexions entre neurones sont plus ou moins fortes), les auteurs ont imaginé un algorithme qui pondère la somme de ses entrées par les poids synaptiques (coefficients de pondération).

b) Les couches

Un réseau neuronal multicouche est constitué de plusieurs couches de neurones interconnectés. La première couche est généralement la couche d'entrée, qui reçoit des entrées de données provenant de sources externes. La dernière couche est la couche de sortie, qui produit des prédictions ou des décisions basées sur les entrées. Entre ces couches se trouvent une ou plusieurs couches cachées, chacune constituée de plusieurs neurones qui traitent les informations et les transmettent aux autres neurones de la couche suivante [Kro,08].

c) *Le Perceptron*

Un perceptron est un type de réseau neuronal artificiel constitué d'une seule couche de neurones, dont chacun calcule une somme pondérée de ses entrées et applique une fonction d'activation pour produire une sortie. Les perceptrons ont été introduits pour la première fois dans les années 1950 comme moyen de modéliser des processus décisionnels simples, et ils ont depuis été utilisés dans diverses applications, notamment la reconnaissance d'images, le traitement du langage naturel et la modélisation prédictive [Kro,08].

d) *Le perceptron multi-couches*

Un perceptron multicouche (PMC), également connu sous le nom de réseau de neurones à anticipation, est une extension du modèle de base du perceptron qui comprend une ou plusieurs couches cachées entre les couches d'entrée et de sortie. Chaque couche cachée est constituée de plusieurs neurones qui calculent des sommes pondérées et appliquent des fonctions d'activation à leurs entrées. Les PMC sont capables d'apprendre des modèles de données plus complexes que les perceptrons à une seule couche, et ils ont été utilisés dans de nombreuses applications, notamment la reconnaissance vocale, les prévisions financières et le diagnostic médical [Kro,08].

La figure 2.4. illustre un exemple d'un PMC à deux couches cachées.

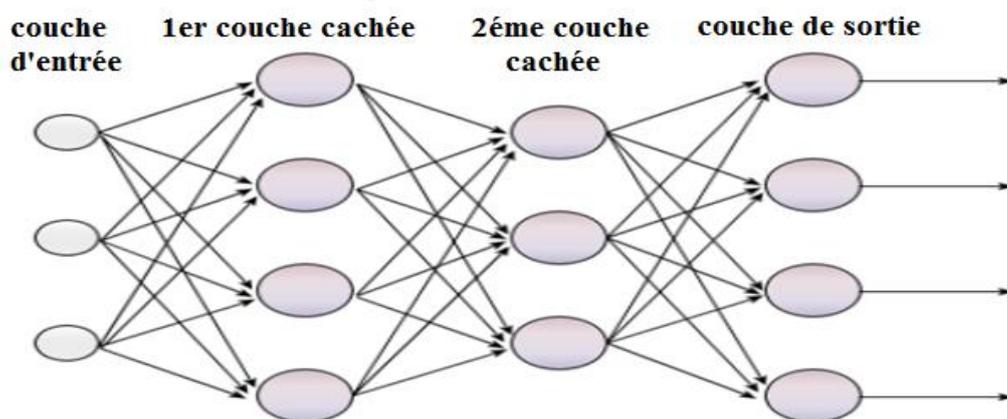


Figure 2.4. Exemple d'un PMC [Nor,05]

e) *Apprentissage d'un perceptron multi-couches*

L'apprentissage est une étape du développement d'un réseau neuronal au cours de laquelle le comportement du réseau est modifié jusqu'à ce qu'il atteigne le comportement souhaité. Cependant, il existe plusieurs modes et plusieurs algorithmes d'apprentissage en fonction de type de réseau de neurones utilisé. Pour le perceptron multi-couches, l'algorithme d'apprentissage utilisé est *l'algorithme d'apprentissage par rétropropagation du gradient*.

La rétropropagation est le processus par lequel l'erreur d'un neurone remonte vers ses synapses et les neurones qui lui sont liés. La rétropropagation par gradient d'erreur est couramment utilisée dans les réseaux de neurones et plus précisément les réseaux de type PMC, et elle consiste à corriger les erreurs en fonction de l'importance des éléments qui ont effectivement participé à la réalisation de ces erreurs : les poids synaptiques qui contribuent à la génération d'une erreur importante auront été modifiés de manière plus significative que les poids qui ont contribué à la génération d'une erreur mineure.

L'algorithme de rétropropagation du gradient se passe en deux phases alternées.

- **La première phase** : on présente un exemple d'entrée au réseau et on propage les signaux de couche en couche jusqu'à la sortie à travers les poids connectant les sorties de neurones d'une couche à ceux de la couche suivante (flèches noires dans la figure 2.5). Alors on peut mettre en évidence les erreurs apparaissant sur les neurones de sortie.
- **La seconde phase** : on applique l'algorithme de mise à jour dans la dernière couche de poids puis on propage les erreurs du sortie à travers cette couche de poids sur l'avant-dernière couche ce qui permet alors de réitérer l'algorithme en mettant à jour l'avant-dernière couche de poids et ainsi de suite (flèches rouges dans la figure 2.5).

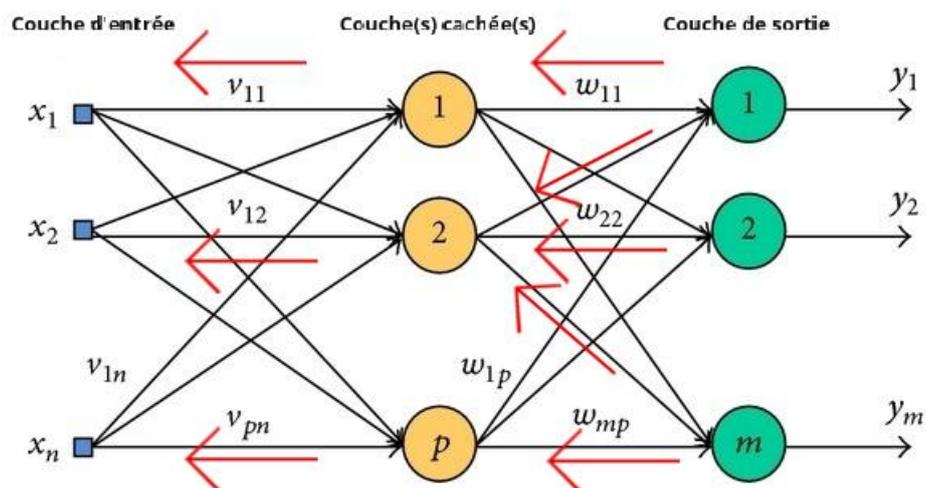


Figure 2.5. Principe de l'apprentissage par rétropropagation de l'erreur.

4.1.2. SVM

Les machines à vecteurs de support (SVM) visent à classer correctement les objets en fonction de l'ensemble de données d'apprentissage. Le SVM peut gérer à la fois des données semi-structurées et structurées, il peut gérer des fonctions complexes si la fonction noyau appropriée peut être dérivée. Les SVM ont été utilisés dans des applications pratiques de

diagnostic du cancer, de détection des fraudes, dans la reconnaissance de l'écriture manuscrite, la détection des visages et de texte, la classification, etc...[Ray,19].

Les SVM ont également été utilisés pour la binarisation d'images de documents. L'avantage d'utiliser un SVM pour la binarisation d'images de documents est qu'elle peut segmenter efficacement le texte (objet) à partir de l'arrière-plan d'un document historique endommagé [Xio,18]. Le classificateur SVM est utilisé pour déterminer le meilleur seuil global de binarisation de chaque bloc d'image, ce qui aide à classer avec précision chaque pixel en premier plan ou en arrière-plan [Xio,18].

4.1.3. Arbre de décision

La technique la plus efficace et la plus appréciée pour la catégorisation et la prédiction est l'arbre de décision [Ros,96].

Un arbre de décision est une structure arborescente qui ressemble à un organigramme, dans laquelle chaque nœud feuille (nœud terminal) a une étiquette de classe, chaque nœud interne représente un test sur un attribut, chaque branche indique un résultat de test [Ros,96].

4.1.4. La régression

La régression est une approche d'apprentissage supervisé. Elle peut être utilisée pour modéliser et prédire des variables continues.

Voici des exemples d'applications d'algorithmes de régression linéaire : la prévision des prix de l'immobilier, la prévision des ventes, la prévision des résultats des examens des étudiants et la prévision des mouvements des cours boursiers.

Cependant, nous étiquetons des ensembles de données en régression, et la valeur de la variable de sortie est déterminée par les valeurs de la variable d'entrée - il s'agit d'une approche d'apprentissage supervisé.

Le type de régression le plus élémentaire est la régression linéaire, qui tente d'ajuster une ligne droite (hyperplan droit) à l'ensemble de données lorsque la relation entre les variables de l'ensemble de données est linéaire [Ray,19].

4.1.5. KPPV (K Plus Proche Voisins)

Un des algorithmes populaires de classification est l'algorithme de *K Plus Proche Voisins* ou en anglais *K Nearest Neighbor* (KNN). Il utilise une base de données avec des points de données classés en différentes classes, et l'algorithme tente de classer le point de données échantillon qui lui est fourni comme un problème de classification. L'algorithme *K Nearest Neighbors* peut nécessiter beaucoup de mémoire ou d'espace pour stocker toutes les données [Ray,19].

4.2. Apprentissage non supervisé

Quand le système ou l'opérateur ne dispose que d'exemples, mais non d'étiquettes, et que le nombre de classes et leur nature n'ont pas été prédéterminés, on parle d'apprentissage non-supervisé [Mah,20]. Nous décrivons dans cette section les algorithmes d'apprentissage non-supervisés les plus connus.

4.2.1. *K-means*

L'algorithme des *K-moyennes* de McQueen [Ba,16] est l'un des algorithmes d'apprentissage non supervisé, et plus précisément de clustering, les plus simples, est connu sous le nom d'algorithme du centre mobile [Mar,17].

Selon [Djo,15], il attribue chaque point d'un cluster au point dont le centre (barycentre) est le plus proche. Le centre est la moyenne de tous les points du cluster et ses coordonnées sont la moyenne arithmétique de tous les points du cluster pour chaque dimension séparément, c'est-à-dire que chaque cluster est représenté par son centroïde.

K signifie que l'algorithme de clustering utilise en général le nombre K de clusters pour fonctionner sur un ensemble de données donné. De cette manière, la sortie contient K clusters avec les données d'entrée partitionnées entre les clusters[Ray,19].

Algorithme:

L'algorithme K-means fonctionne comme suit:

- Tout d'abord, nous initialisons k points, appelés moyennes, de manière aléatoire.
- Nous catégorisons chaque élément à sa moyenne la plus proche et nous mettons à jour les coordonnées de la moyenne, qui sont les moyennes des éléments classés dans cette moyenne jusqu'à présente.
- Nous répétons le processus pour un nombre donné d'itérations et à la fin, nous avons nos clusters.

4.2.2. *Classification hiérarchique*

La classification hiérarchique est une méthode de catégorisation des objets dans un ordre logique. Dans le domaine de l'apprentissage automatique, la classification hiérarchique est également connue sous le nom de décomposition de l'espace d'instance, car elle divise un grand problème multi-classes en problèmes de classification plus petits [Ara,96].

4.2.3. *Fuzzy kmeans(K-moyennes flou)*

Un objet de données peut appartenir à deux clusters ou plus à l'aide de la méthode de clustering *Fuzzy K-means*. Cette méthode est fréquemment utilisée en reconnaissance de

formes et est dérivée de l'algorithme *K-means* [Bal,67]. Il a été développé par Dunn [Bez,81] en 1973 et amélioré par Bezdek [Dun,73] en 1981.

Brièvement *K-means flou* est en fait une généralisation du *K-means* dans laquelle chaque observation possède une probabilité d'appartenir à chaque classe.

4.2.4. Les cartes de Kohonen

Bien que les cartes de Kohonen (SOM pour *Self Organizing Maps*) font partie des réseaux de neurones artificiels, elles sont qualifiées comme des techniques d'apprentissage non-supervisés.

Une carte auto-organisée est constituée d'une grille de neurones de faible dimension. Chaque neurone a deux voisins lorsque la grille est unidimensionnelle. Lorsque la grille est bidimensionnelle, les neurones sont disposés de manière rectangulaire, chaque neurone ayant quatre voisins (topologie rectangulaire), ou de manière hexagonale, chaque neurone ayant six voisins (topologie hexagonale) [Haj,14]

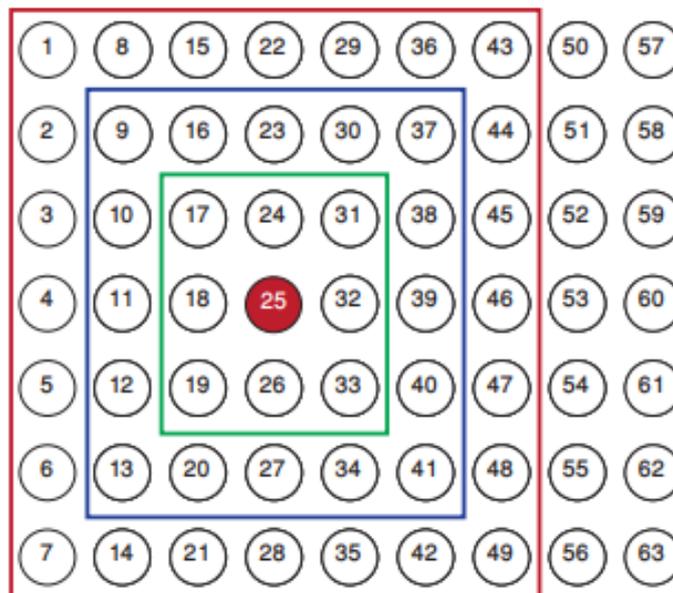


Figure 2.6. Carte auto-organisatrice rectangulaire de 63 neurones [Haj,14]

Leur fonction principale est de projeter des données de grande dimension sur un espace de faible dimension de manière non linéaire. Dans la classification des données, les cartes auto-organisées sont couramment utilisées [Haj,14].

5. Le deep learning (apprentissage en profondeur)

5.1. Présentation

Un sous-ensemble de l'apprentissage automatique est connu sous le nom d'apprentissage en profondeur (deep learning), qui est entièrement pris en charge par des

réseaux de neurones artificiels. Cette méthode d'apprentissage est inspirée par la façon dont le cerveau humain traite et analyse l'information. Les réseaux de neurones artificiels imitent le fonctionnement du cerveau, en utilisant des couches de nœuds interconnectés pour traiter des données complexes. L'apprentissage en profondeur permet d'automatiser la découverte de représentations de données utiles, sans qu'il soit nécessaire de les programmer explicitement.

Le terme "profondeur" ou "deep" dans l'appellation fait référence à la profondeur des architectures de réseaux de neurones utilisées pour résoudre des problèmes de reconnaissance de formes. Cependant, la principale différence entre l'apprentissage classique et l'apprentissage en profondeur est que ce dernier utilise des réseaux de neurones artificiels profonds pour créer des représentations de caractéristiques hiérarchiques à partir des données, ce qui permet de résoudre des problèmes plus complexes de reconnaissance de formes.

Cette technique est utilisée dans de nombreux domaines, notamment la vision par ordinateur. Grâce à l'apprentissage en profondeur, les ordinateurs peuvent désormais effectuer des tâches auparavant réservées aux êtres humains, avec une précision accrue et une vitesse supérieure. En conséquence, l'apprentissage en profondeur est devenu un domaine de recherche très actif, avec de nombreux chercheurs qui travaillent sur l'amélioration des algorithmes existants et le développement de nouveaux modèles [Bou,22].

5.2. Réseaux de neurones profond

Un réseau de neurones profond est une architecture de réseau de neurones artificiels qui comporte plusieurs couches de neurones interconnectées. Ces réseaux sont dits "profonds" en raison du grand nombre de couches qui les composent, pouvant aller de quelques dizaines à plusieurs centaines de couches [Bou,22].

Les réseaux de neurones profonds sont utilisés dans le domaine de l'apprentissage automatique pour résoudre des problèmes de classification, de reconnaissance d'images, de reconnaissance de la parole, de traduction automatique, de synthèse de texte, etc. Leur puissance de calcul et leur complexité leur permettent d'extraire des caractéristiques très fines des données d'entrée, ce qui les rend particulièrement efficaces pour la reconnaissance de motifs complexes dans les données.

Les méthodes d'apprentissage en profondeur nécessitent une grande quantité de données d'entraînement ainsi qu'un temps d'entraînement approprié.

5.3. Réseaux de neurones convolutionnel

Ce sont des réseaux de neurones de reconnaissance d'images dans lesquels chaque neurone répond à des régions qui se chevauchent dans l'espace visuel appelées champs récepteurs [Ac,21].

Un réseau de neurones convolutionnel (CNN) est un type de réseau de neurones profond spécialement conçu pour la reconnaissance d'images et la vision par ordinateur. C'est une version avancée du perceptron multicouche, où chaque couche est composée de filtres de convolution [Ac,21].

Les filtres de convolution sont appliqués de manière répétée à l'image pour créer une carte de caractéristiques, qui est ensuite réduite de taille en utilisant une opération de pooling pour préserver les caractéristiques les plus importantes.

La convolution n'est rien de plus qu'une simple méthode de filtrage permettant l'activation. La position et la force d'une caractéristique détectée sont fournies par ce processus de filtrage lorsqu'il est appliqué à plusieurs reprises. Ces réseaux sont fréquemment utilisés dans le traitement d'images, le traitement du langage naturel et les systèmes de recommandation en raison de leur capacité à produire efficacement des résultats pour la caractéristique importante qui a été reconnue [Ros,96].

6. Conclusion

Enfin, ce chapitre présente un examen approfondi de l'apprentissage automatique, en commençant par sa définition et ses objectifs, puis en explorant ses nombreuses applications. Les nombreuses techniques d'apprentissage automatique, telles que l'apprentissage supervisé, non supervisé et par renforcement, sont abordées en détails, ainsi que leurs algorithmes respectifs. Une section est également consacrée à l'apprentissage en profondeur, qui couvre la présentation de la technologie et les différents types des réseaux de neurones profonds.

Chapitre 3. Conception

1. Introduction

Le présent chapitre est dédié à la présentation de notre approche basée sur l'apprentissage automatique pour la binarisation d'images de documents dégradés. Comme nous avons déjà mentionné, la binarisation est un processus crucial dans le domaine de l'analyse et la reconnaissance de documents, permettant de convertir une image couleurs en une version binaire où les pixels sont classés comme étant soit noirs, soit blancs. Dans le cas spécifique des images de documents dégradés, la binarisation devient d'autant plus complexe en raison des divers artefacts et imperfections présents.

Nous commençons ce chapitre par la présentation des caractéristiques des images de documents dégradés, l'objet de notre étude. Après, nous décrivons la conception de notre approche et nous détaillons les différentes étapes impliquées dans notre méthodologie, avant de conclure.

2. Caractéristiques des documents dégradés

Dans notre travail, nous nous intéressons à la binarisation d'images de documents anciens, il est donc intéressant de présenter même brièvement les caractéristiques de ce type de documents, pour montrer la difficulté de leur traitement.

Les documents anciens possèdent plusieurs caractéristiques qui viennent de leurs natures, leurs états de stockage...etc. Ces caractéristiques rendent les algorithmes classiques de traitement d'images non fiables. La première catégorie de dégradation regroupe tous les défauts qui proviennent essentiellement de l'état document. Dans la plupart des cas, ces dégradations ne peuvent être atténuées. À titre d'exemple la (Figure 3.1) illustre des dégradations associées aux ouvrages.

Nous pouvons citer les défauts suivants :

- **Les défauts touchant les caractères** : L'encre faible ou de mauvaises qualités peut dégrader le contraste entre l'écriture et le fond. Dans d'autres cas, l'encre peut se diffuser de façon trop importante dans le papier et créer des caractères épais avec des formes non uniformes et des trous bouchés.
- **La transparence du verso** : certains papiers, trop fins, laissent apparaître l'encre présente au verso du document à travers le recto. Selon la quantité d'encre qui est diffusée, la transparence peut être plus ou moins visible. Un exemple est présent sur la (Figure 3.1.c)
- **Les taches** : sont souvent dues à l'humidité, mais peuvent aussi être la cause de mauvais usages. Les (Figures 3.1.a et 3.1.b) montrent des exemples de taches pouvant avoir plusieurs origines.
- **Les pliures et ondulations peuvent avoir plusieurs origines** : Par exemple, le coin d'un document peut être plié la reliure peut aussi créer une ondulation locale sur une partie du document.
- **Les déchirures et trous** : (Figure 3.1.d) sont des dégradations plus rares qui peuvent avoir des origines diverses (usure, mauvaise utilisation, insectes, etc.).

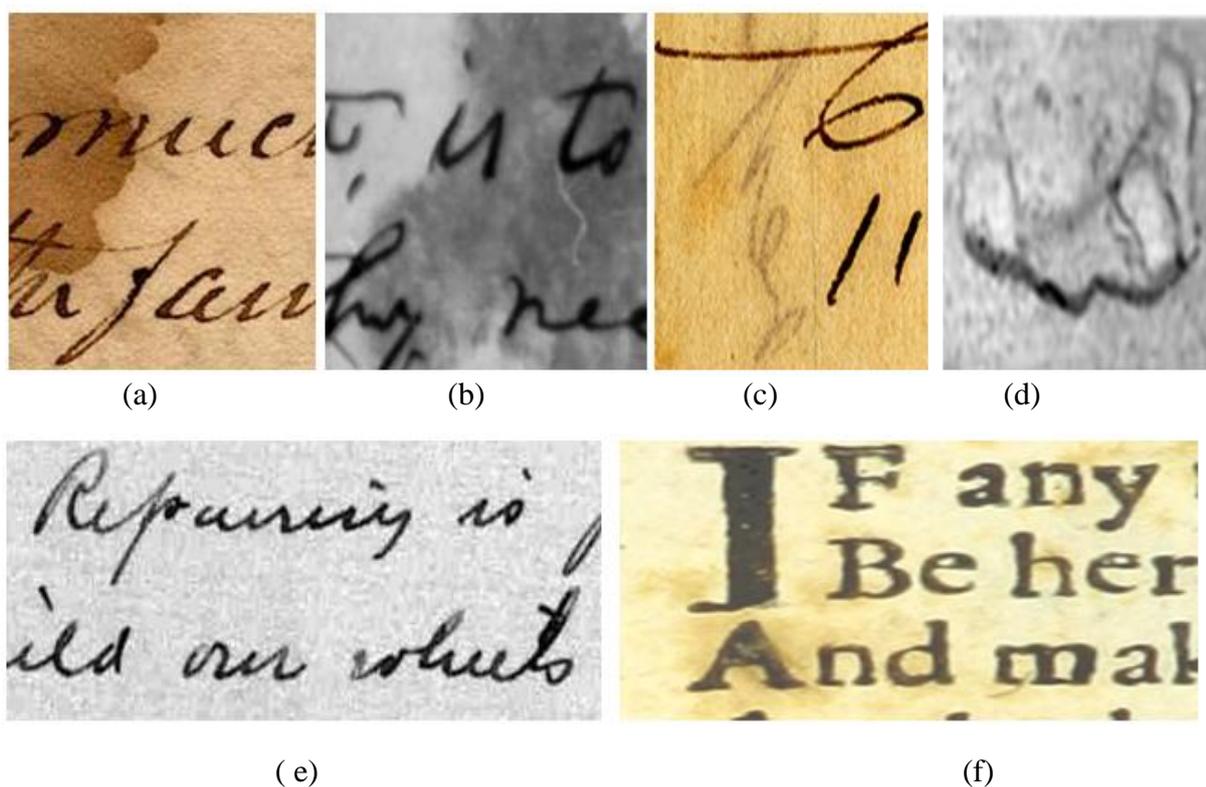


Figure 3.1. Exemples de défauts présents sur des documents dégradés : (a),(b) et (f) montrent des taches d'encre et d'usures, (c) transparence du verso de la page, (d) montre un trou dans la page, (e) écriture n'est pas clair

3. Motivations

L'objectif principal de notre travail est de proposer une approche efficace pour la binarisation des images de documents dégradés présentant différents types de dégradations. Pour atteindre cet objectif, nous avons opté pour l'utilisation de techniques d'apprentissage automatique, qui se sont révélées prometteuses dans de nombreux problèmes de traitement d'images et de vision par ordinateur.

Les méthodes traditionnelles de binarisation des images de documents dégradés rencontrent souvent des limites en raison de la complexité des artefacts et des imperfections présents dans ces images. Les approches basées sur le seuillage, que ce soit global, local, ou hybride ont du mal à s'adapter de manière efficace et précise aux variations et aux dégradations spécifiques rencontrées dans les documents dégradés. Par conséquent, il est nécessaire d'explorer de nouvelles approches plus flexibles et adaptatives.

L'apprentissage automatique offre une solution prometteuse à ces défis. En utilisant des techniques d'apprentissage, il est possible d'extraire des caractéristiques pertinentes et d'apprendre des modèles complexes à partir des données d'entraînement. Cela permet au modèle d'adaptation de manière plus précise aux variations et aux défauts spécifiques des images de documents dégradés.

De plus, l'apprentissage automatique permet une approche plus générale et flexible, car il n'est pas limité par des règles ou des seuils préétablis. Le modèle peut apprendre à partir de grandes quantités de données, ce qui lui permet de capturer des motifs et des structures subtiles qui

échappent souvent aux méthodes traditionnelles. En utilisant des techniques d'apprentissage automatique, nous avons l'opportunité d'améliorer significativement la précision et la qualité de la binarisation des images de documents dégradés.

4. Description de l'approche proposée

La méthode proposée ne repose pas sur le seuillage comme la plupart des méthodes de binarisation proposées dans la littérature, mais sur la classification directe des pixels en noir et blanc. Dans cette méthode, nous utilisons une combinaison de l'algorithme K-means et d'un réseau de neurones artificiels (ANN pour Artificial Neural Network) de type PMC (Perceptron multi-couches) pour effectuer la transformation d'une image en niveaux de gris en une image binaire. La méthode proposée ne calcule ou apprend aucun seuil mais effectue une binarisation directe en classifiant les pixels de l'image en deux classes. Comme nous l'avons déjà dit, la binarisation est une sorte de problème de classification en deux classes.

L'idée est de faire apprendre au PMC la valeur binaire de certains pixels (dits d'apprentissage) en se basant sur des informations locales et le PMC sera par la suite capable de généraliser le comportement pour d'autres pixels non vus dans l'apprentissage, et donc de déterminer leur valeur binaire.

L'approche proposée est composée de trois parties essentielles : préparation des données d'apprentissage, création et apprentissage du PMC, et test et exploitation. Ces étapes sont résumées par la figure 3.2.

Nous détaillons dans ce qui suit chacune de ces étapes.

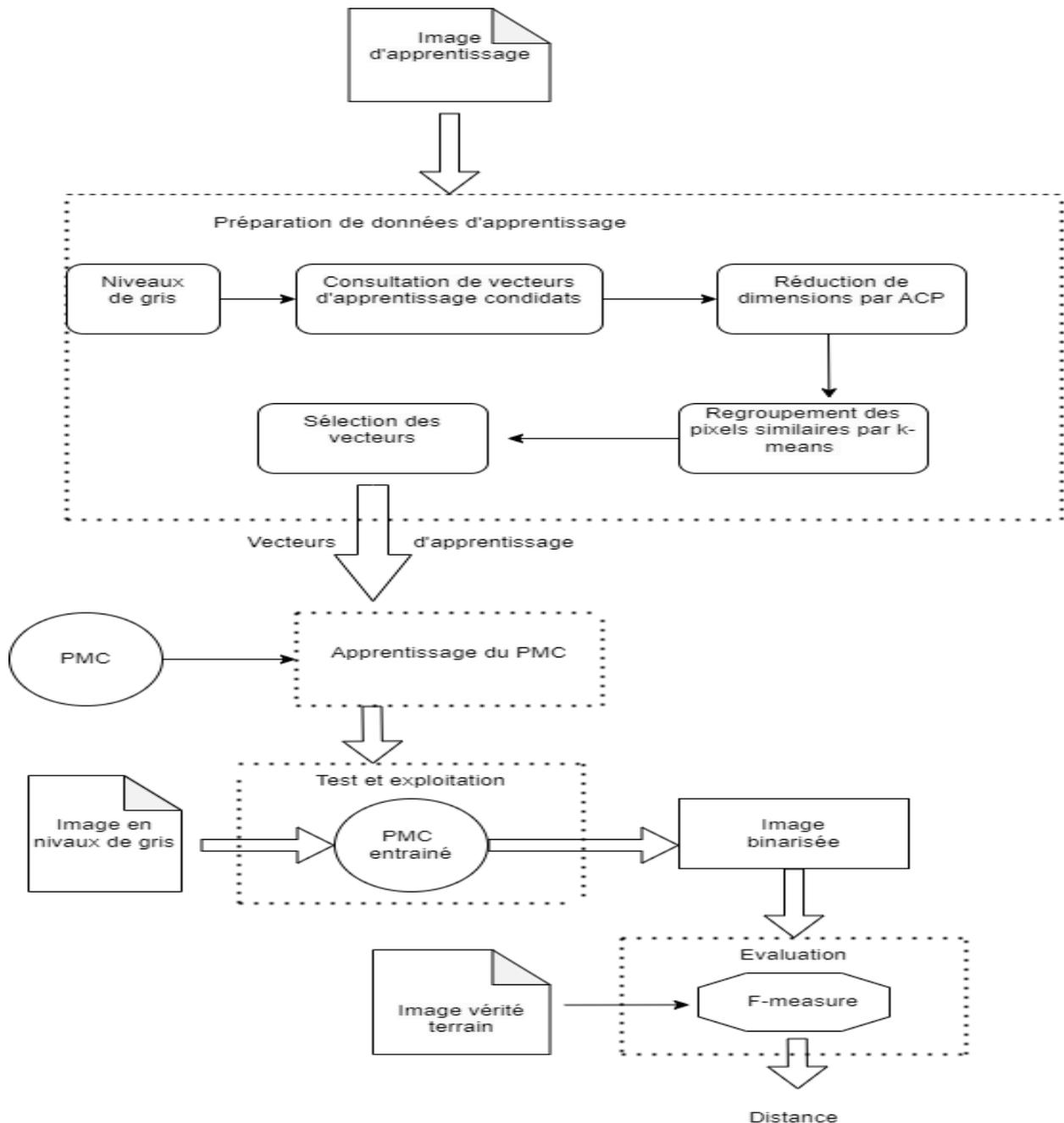


Figure 3.2 : Schéma général de notre approche

4.1. Préparation des données d'apprentissage

La première phase dans notre approche est la préparation des données d'apprentissage à partir d'une base d'images d'apprentissage. C'est à partir de ces données d'apprentissage que le PMC va être entraîné.

Le PMC est une technique d'apprentissage supervisée c'est à dire que les exemples d'entrée doivent être accompagnés des sorties désirées correspondantes. Cependant, l'apprentissage du PMC est un processus de mise à jour de poids, qui se fait à chaque itération en se basant sur l'erreur calculée entre les sorties estimées par le PMC et les sorties désirées ou attendues

accompagnant les entrées. Cette erreur est rétro propagée de couche en couche pour ajuster les paramètres (les poids) du PMC, d'où vient l'appellation apprentissage par rétro propagation du gradient.

Le PMC va apprendre à binariser des pixels en niveau de gris c'est pourquoi il doit être entraîné sur une base large de pixels et de leur valeur binaire (noir ou blanc).

De ce fait, la base d'images d'apprentissage contient plusieurs images de documents dégradées et des images de vérité terrain correspondantes. Ces dernières sont des images binaires de références décrivant le résultat idéale de binarisation qu'on souhaiterait avoir. La figure 3.3 illustre un exemple d'une image de documents dégradé et son image de vérité terrain.

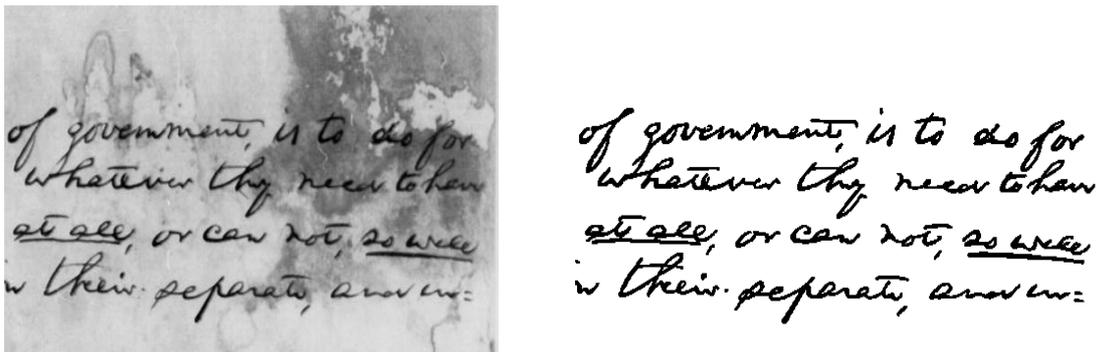


Figure 3.3. Image de document dégradé avec son image de vérité terrain

De même, les données d'apprentissage sont composées d'un ensemble de vecteurs d'entrées et les sorties désirées correspondantes. Dans notre proposition, chaque vecteur d'entrée contient la valeur de gris d'un pixel p_i de l'échantillon d'images d'apprentissage, avec celles de ses voisins dans une fenêtre $w \times w$ centrée sur p_i . La sortie désirée correspondante est la valeur binaire b_i du pixel p_i dans l'image de référence.

Veillez voir la figure suivante pour plus d'éclaircissements.

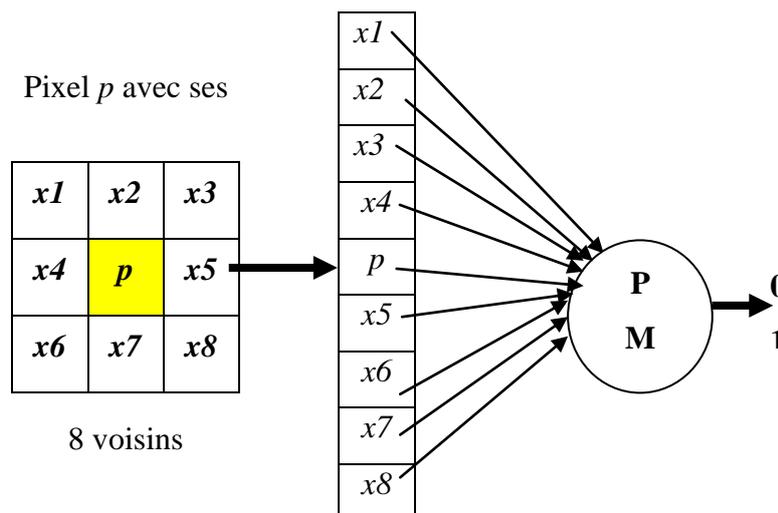


Figure 3.4. PMC pour classifier un pixel p dans une fenêtre de 3×3 .

Le choix d'inclure des informations locales sur le voisinage du pixel en question est justifié par le fait que les images de documents dégradés peuvent présenter des structures complexes et des irrégularités qui nécessitent une prise en compte du contexte local pour une

binarisation précise. Les pixels voisins peuvent contenir des informations pertinentes sur la texture, les contours et d'autres caractéristiques locales qui contribuent à la compréhension globale de l'image. En incluant ces informations locales, nous permettons au PMC de capturer des relations spatiales et des motifs locaux importants.

Les relations spatiales entre les pixels sont également cruciales dans le processus de binarisation. En considérant les pixels voisins, nous prenons en compte les variations locales et les dégradations spécifiques qui peuvent se propager sur une certaine distance spatiale. Par exemple, une tache ou une dégradation peut affecter plusieurs pixels adjacents, et en incluant ces pixels voisins dans l'ensemble d'apprentissage, le modèle peut apprendre à reconnaître ces motifs et à les différencier des régions de texte ou de fond.

Les étapes impliquées dans cette phase pour préparer les données d'apprentissage sont les suivantes:

4.1.1. Transformation de l'image en niveaux de gris

La transformation d'une image couleur en une image de niveau de gris est nécessaire pour effectuer la binarisation. Cette étape peut être résumée par le pseudo-code suivant :

Algorithme de transformation en niveaux de gris

Entrée : image en couleurs

Sortie : image en niveaux de gris

Début

Pour chaque pixel p de l'image faire :

$r \leftarrow$ la quantité de la couleur rouge du pixel p

$v \leftarrow$ la quantité de la couleur vert du pixel p

$b \leftarrow$ la quantité de la couleur bleu du pixel p

$g \leftarrow (r+v+b)/3$

Fin Pour

Fin.

Cet algorithme parcourt chaque pixel de l'image en couleurs. Pour chaque pixel, il extrait les valeurs de rouge (r), de vert (v) et de bleu (b) du pixel. Ensuite, il calcule la moyenne de ces trois valeurs $(r + v + b) / 3$, et cette moyenne est utilisée comme la valeur du pixel dans l'image en niveaux de gris.

4.1.2. Extraction des pixels d'apprentissage

Comme mentionné précédemment, l'entraînement du PMC repose sur un ensemble de pixels associés à leurs valeurs binaires correspondantes. Toutefois, l'utilisation de tous les pixels de chaque image de la base d'apprentissage n'est pas réalisable en pratique. Par exemple, si nous disposons de 100 images de taille 1000×1000 pixels chacune, cela représenterait un total de 10^8 vecteurs d'entrée. Par conséquent, il est nécessaire de sélectionner judicieusement les

pixels à inclure dans l'ensemble d'apprentissage? La question se pose alors : comment effectuer cette sélection ?

La solution la plus simple consiste à choisir les pixels de manière aléatoire. Bien que cette approche soit facile à implémenter, elle présente le risque de sélectionner des pixels non représentatifs, ce qui pourrait introduire un biais dans l'apprentissage du PMC. Afin de remédier à ce problème, il est préférable d'adopter une approche plus réfléchie pour la sélection des pixels.

Dans notre approche, nous avons envisagé de regrouper les pixels de chaque image en utilisant des techniques de regroupement basées sur leurs caractéristiques visuelles, puis de sélectionner un nombre spécifié de pixels à partir de chaque groupe formé. Pour réaliser cette tâche de regroupement, nous avons choisi d'adopter l'algorithme bien connu de clustering: K-means. L'idée principale derrière cette approche est de regrouper les pixels similaires en termes de valeurs de gris et de sélectionner de représentant de chaque groupe. En sélectionnant un certain nombre de pixels de chaque groupe, nous nous assurons d'avoir une représentation équilibrée des différentes caractéristiques présentes dans les images. Par exemple, si nous avons des régions contenant du texte, des images présentant des dégradations spécifiques ou des coins d'image, il est essentiel d'avoir des exemples de chaque de ces caractéristiques dans notre ensemble d'apprentissage. En choisissant des pixels de chaque groupe, nous évitons de nous concentrer uniquement sur une partie spécifique des données, ce qui pourrait entraîner un apprentissage binarisé.

a) Construction des vecteurs d'entrée candidats

Après avoir transformé les images en niveaux de gris, on procède à l'extraction de tous les pixels pour préparer les vecteurs d'entrée candidats. Cependant, à chaque pixel on fait correspondre un vecteur. Ce dernier est formé du niveau du pixel en question en plus de ceux de ces voisins dans une fenêtre $w \times w$ centrée sur lui. Ainsi le vecteur d'entrée d'un pixel p_i est de la forme suivante :

$$X_i = \{g_i^1, g_i^2, \dots, g_i^{w \times w}\},$$

Avec la valeur de gris du j ème pixel dans la fenêtre $w \times w$ centrée sur le pixel p_i .

De ce fait la taille des vecteurs d'entrée est relative à la taille de fenêtre utilisée. Elle égale exactement w^2 . La Figure 3.5 présente un exemple de tel vecteur pour un pixel (de niveau de gris 173) dans une fenêtre 3×3 .

242	173	173
245	173	173
223	233	36

$$X = \{242, 173, 173, 245, \underline{173}, 173, 223, 233, 36\}$$

Figure 3.5. Vecteur d'entrée pour un pixel dans une fenêtre 3×3 .

b) Séparation des vecteurs en fonction de leur appartenance à l'image de vérité terrain

L'étape suivante dans la phase de préparation des données d'apprentissage consiste à séparer les vecteurs de tous les pixels en deux classes distinctes en fonction de leur correspondance avec l'image de vérité terrain. Cette séparation s'effectue en considérant deux catégories : les vecteurs des pixels noirs dans l'image de vérité terrain et les vecteurs des pixels blancs dans l'image de vérité terrain.

Cette séparation en deux classes distinctes nous permet de former un ensemble d'apprentissage contenant des exemples représentatifs des deux catégories de pixels. En séparant les vecteurs en fonction de leur appartenance à l'image de vérité terrain, nous créons des ensembles de données équilibrés et pertinents pour l'apprentissage de notre PMC.

c) Réduction de la dimensionnalité avec l'ACP

Lorsque la taille de la fenêtre utilisée pour former les vecteurs d'entrée est grande, cela peut entraîner une augmentation significative de la dimension des vecteurs. Cette augmentation de la dimension peut rendre les processus de clustering (à l'aide de l'algorithme Kmeans) et d'apprentissage (du MLP) plus complexe et peut également avoir un impact négatif sur les performances du système.

Pour résoudre ce problème, nous avons proposé de procéder à une réduction de la dimensionnalité des données en appliquant la célèbre technique d'Analyse en Composantes Principales (ACP). L'ACP est une méthode couramment utilisée pour réduire la dimensionnalité des données en identifiant les composantes les plus importantes et en les sélectionnant pour former un nouvel espace de caractéristiques. Cette étape peut être résumée par le pseudo-code suivant :

Algorithme ACP

Entrées : matrice D avec M vecteurs de taille N

Début

1. Calculer la matrice de covariance C à partir de la matrice D.
 - Créer une matrice C de taille $N * N$, initialisée à zéro.
 - Pour chaque colonne i dans D:
 - Calculer la moyenne μ_i de la colonne i de D.
 - Pour chaque paire d'éléments (x, y) dans la colonne i de D:
 - Ajouter $(x - \mu_i) * (y - \mu_i)$ à l'élément correspondant de C.
 - Diviser chaque élément de C par (M - 1) pour obtenir la covariance.
 2. Calculer les vecteurs propres et les valeurs propres de la matrice de covariance C.
 - calculer les vecteurs propres V et les valeurs propres λ de C.
 3. Trier les vecteurs propres en fonction des valeurs propres correspondantes, de la plus grande à la plus petite.
 - Associer chaque vecteur propre V_i avec sa valeur propre λ_i .
 4. Calculer la variance totale Var_total de la matrice D.
 - Calculer la somme des valeurs propres λ_i pour tous les i.
 5. Calculer la variance cumulative Var_cumul de la matrice D.
 - Initialiser Var_cumul à zéro.
-

-
- Pour chaque i de 1 à N :
 - Ajouter λ_i à Var_cumul .
 - Si Var_cumul est supérieur ou égal à $0,8 * \text{Var_total}$, sortir de la boucle.
 - Fin Pour
6. Sélectionner les $N1$ premiers vecteurs propres correspondant à la variance cumulative requise.
- Extraire les $N1$ premiers vecteurs propres $V1$ à $VN1$.
7. Calculer la matrice de projection P en multipliant la matrice D par les vecteurs propres sélectionnés.
- La matrice de projection P aura la taille $M * N1$.
 - Pour chaque vecteur V_i dans $V1$ à $VN1$:
 - Calculer le produit matriciel de D et V_i pour obtenir la i -ème colonne de P .
8. Le résultat est la matrice $D1$, qui est égale à P .
- Sorties:**
- $D1$ est une matrice de taille $M * N1$ contenant les vecteurs projetés réduits en dimensionnalité avec $N1 < N$.
- Fin.
-

d) Regroupement des vecteurs similaires (Clustering)

Une fois que les vecteurs ont été séparés en deux classes en fonction de leur appartenance à l'image de vérité terrain, nous passons au regroupement des vecteurs similaires en groupes.

Ce regroupement a pour objectif de créer des ensembles homogènes de pixels qui partagent des caractéristiques visuelles communes de sorte que les pixels d'un même cluster présentent des similitudes visuelles, tandis que les pixels de clusters différents présentent des différences marquées.

En effet, le regroupement des vecteurs similaires présente un avantage significatif pour la sélection des pixels à inclure dans l'apprentissage. Plutôt que de considérer tous les pixels individuellement, ce regroupement nous permet de choisir quelques pixels représentatifs de chaque cluster formé.

Pour effectuer ce regroupement, nous avons opté pour l'utilisation de l'algorithme K-means. Ce dernier qui est un algorithme de clustering bien connu. K-means vise à partitionner un ensemble de données en un nombre prédéfini de clusters en minimisant la distance entre les vecteurs et les centroïdes des clusters correspondants.

Nous avons choisi d'utiliser l'algorithme K-means en raison de plusieurs raisons. Tout d'abord, il est efficace en termes de temps de calcul, ce qui est important lorsque nous traitons de grandes quantités de données. De plus, il est largement étudié et bien compris dans la communauté de l'apprentissage automatique, ce qui facilite son implémentation et son utilisation.

L'algorithme K-means est appliqué séparément sur les vecteurs des pixels noirs et sur les vecteurs de pixels blancs. Cela signifie que nous effectuons deux regroupements distincts pour chaque catégorie de pixels.

Le processus de regroupement à l'aide de l'algorithme K-means peut être résumé par le pseudo-code suivant:

Algorithme de K-means

Entrées:

- K : nombre de groupes
- Ensemble_de_données : contient les vecteurs de caractéristiques de chaque pixel

Début

- Choisir K vecteurs aléatoirement à partir de l'ensemble de données et les assigner comme centres (centroids) des clusters (c_1, c_2, \dots, c_k)
- Répéter jusqu'à convergence:
 - Pour chaque vecteur de données
 - Assigner le vecteur au cluster du centroid le plus proche
 - Pour chaque cluster
 - Calculer le nouveau centroid comme la moyenne de tous les vecteurs assignés à ce cluster
 - Mettre à jour les centroids avec les nouveaux centroids
- Retourner les clusters et les centroids

Fin.

e) Sélection des vecteurs d'apprentissage

A partir de chaque groupe obtenu du regroupement, nous effectuons une sélection aléatoire de pixels représentatifs. Cette sélection aléatoire consiste à choisir un certain nombre de pixels de manière aléatoire à partir de chaque groupe. La sélection aléatoire des pixels permet

d'éviter tout biais potentiel dans le choix des échantillons. Elle assure une représentativité plus équitable des pixels dans chaque groupe, en s'assurant que tous les pixels ont une chance égale d'être sélectionnés.

Nous proposons de sélectionner:

- 10 vecteurs à partir de chaque cluster de vecteurs noirs.
- 20 vecteurs à partir de chaque cluster de vecteurs blancs.

A la sortie de cette étape, nous obtenons une base d'apprentissage composée d'un grand nombre de vecteurs représentant différents échantillons de pixels. Chaque vecteur représente un pixel de l'image d'origine, accompagné de ses caractéristiques et de sa valeur binaire correspondante (noir ou blanc).

4.1.3. Enregistrement des données d'apprentissage

Chaque vecteur créé auparavant qui correspond à un pixel dans une image d'apprentissage, en plus de la valeur binaire correspondante dans l'image de vérité terrain est enregistré comme une ligne dans un fichier CSV. Ce dernier sert de fichier d'apprentissage.

Le fichier d'apprentissage est ensuite éclaté en deux fichiers : un contient l'ensemble d'apprentissage, et l'autre contient l'ensemble de test.

Pour ce faire, on interroge l'utilisateur une autre fois pour déterminer la portion de l'ensemble de test et de l'ensemble d'apprentissage. Comme résultat nous obtiendrons 2 fichiers.

La figure suivante présente un exemple de tel fichier.

	A	B	C	D	E	F	G	H	I	J	K
1	177	182	185	179	163	162	146	89	65	0	
2	182	185	180	163	162	179	89	65	119	0	
3	179	179	163	178	146	89	161	96	123	0	
4	179	163	162	146	89	65	96	123	136	0	
5	163	162	179	89	65	119	123	136	64	0	
6	162	179	181	65	119	174	136	64	145	0	
7	178	146	89	161	96	123	110	85	156	0	
8	146	89	65	96	123	136	85	156	173	0	
9	89	65	119	123	136	64	156	173	72	0	
10	65	119	174	136	64	145	173	72	112	0	
11	119	174	180	64	145	179	72	112	175	0	
12	183	183	182	173	158	130	107	93	76	0	
13	183	182	185	158	130	130	93	76	79	0	
14	182	185	182	130	130	159	76	79	90	0	
15	185	182	182	130	159	170	79	90	117	0	
16	178	184	183	175	153	154	123	84	76	0	
17	184	183	180	153	154	167	84	76	117	0	
18	180	180	168	187	149	119	160	137	84	0	
19	168	175	181	119	168	180	84	131	145	0	
20	176	161	96	134	110	85	71	71	128	0	
21	161	96	123	110	85	156	71	128	178	0	
22	96	123	136	85	156	173	128	178	174	0	
23	123	136	64	156	173	72	178	174	96	0	
24	136	64	145	173	72	112	174	96	100	0	
25	64	145	179	72	112	175	96	100	167	0	
26	176	173	158	128	107	93	76	85	114	0	
27	173	158	130	107	93	76	85	114	146	0	

Figure 3.6. Exemple d'un fichier d'apprentissage

4.2. Définition et apprentissage du PMC

Dans cette section, nous abordons la définition et l'apprentissage de notre classifieur utilisé pour la binarisation d'images de documents dégradés, à savoir le PMC.

Tout d'abord, nous définissons la structure du PMC utilisé (nombre de couches, nombre de neurones dans chaque couche,...). Ensuite, nous décrivons le processus d'apprentissage du PMC.

4.2.1. Définition du PMC

Comme nous avons mentionné précédemment, nous avons choisi d'utiliser un réseau de type PMC (Perceptron Multicouche) pour notre approche de binarisation d'images de documents dégradés. Cette décision repose sur plusieurs raisons:

- Le PMC est une architecture de réseau de neurones largement utilisée et bien établie dans le domaine de l'apprentissage automatique. Il est capable de modéliser des fonctions complexes en utilisant des couches de neurones interconnectés, ce qui en fait un choix approprié pour des problèmes de traitement d'images.

- Le PMC offre une grande flexibilité en termes de conception et de configuration. Nous pouvons ajuster la taille et le nombre de couches, ainsi que le nombre de neurones dans chaque couche, pour adapter le réseau aux spécificités de notre problème de binarisation. Cela nous permet d'explorer différentes architectures et de trouver celle qui offre les meilleures performances.
- Le PMC est adapté à l'apprentissage supervisé, qui est la méthode que nous avons choisie pour notre approche. En fournissant des données d'entrée accompagnées de leurs sorties binaires désirées, nous pouvons entraîner le PMC à apprendre la relation entre les caractéristiques des pixels et leur état binaire correspondant.
- Le PMC est capable de généraliser à partir des exemples d'apprentissage pour traiter de nouveaux cas. Une fois que le réseau est entraîné sur un ensemble de pixels, il peut être utilisé pour prédire les valeurs binaires des pixels dans des images de documents dégradés non vues précédemment. Cela nous permet d'obtenir une approche robuste et applicable à une grande variété de documents.

La création du PMC regroupe le choix du nombre de ses couches, le nombre de neurones dans chaque couche, le nombre d'entrées et de sorties, et la fonction d'activation de chaque neurone.

Effectivement, le choix de la topologie ou de l'architecture optimale d'un réseau PMC est d'une importance capitale, mais il n'est pas facile et il n'est pas toujours justifié par des règles ou des théorèmes clairs. Cependant, il existe quelques indications heuristiques pour guider ce choix.

L'une de ces indications est le célèbre théorème de Cybenko, qui affirme que "avec une seule couche cachée, on peut approximer n'importe quelle fonction continue". Cela suggère qu'une architecture à une seule couche cachée pourrait être suffisante pour notre problème de binarisation d'images de documents dégradés.

Une autre heuristique couramment utilisée est de choisir le nombre de neurones dans la couche cachée égal à la racine carrée du produit du nombre d'entrées et du nombre de sorties du réseau. Cela peut servir de point de départ pour déterminer la taille de la couche cachée.

Cependant, compte tenu de la complexité et de la spécificité de notre problème, nous avons opté pour une approche expérimentale dans la conception de notre réseau PMC. Nous avons réalisé des expérimentations itératives en ajustant différents paramètres tels que le nombre de couches, le nombre de neurones par couche et les fonctions d'activation, afin de trouver une architecture qui offre de bonnes performances de binarisation pour nos images de documents dégradés.

Finalement, nous avons conçu un réseau PMC avec une architecture spécifique qui nous semblait adaptée à notre problème. Cette architecture est la suivante:

- La taille de fenêtre adoptée est 3×3 et donc les vecteurs d'entrée ont 9 valeurs et par conséquent le nombre de neurones dans la couche d'entrée est 9.
- Une seule couche cachée de 9 neurones.
- La couche de sortie utilise une fonction de classification binaire pour produire les prédictions de classe (0 ou 1) 0 noir et 1 blanc.

- La fonction d'activation choisie est la fonction Sigmoidé définie par :

$$f(x) = \frac{1}{1 + e^{-ax}}$$

4.2.2. Apprentissage du PMC

Dans cette étape, le PMC est entraîné en utilisant les données d'apprentissage préparées précédemment. Nous utilisons pour ça l'algorithme de *rétro-propagation du gradient* qui est un algorithme d'apprentissage supervisé où le système est fournie avec des entrées et des sorties désirées correspondantes. La mise à jours des poids se fait dans le but de minimiser l'erreur entre les sorties du PMC (sorties estimées) et les données réelles (sorties désirées).

Avant de lancer l'apprentissage, les vecteurs de la base d'apprentissage sont divisées en deux ensembles : un ensemble d'apprentissage et un ensemble de test. Par défaut, 80% des vecteurs sont utilisées pour l'apprentissage et 20% pour la test, et l'utilisateur a la possibilité de modifier cette répartition.

Le processus d'apprentissage à l'aide du rétropropagation du gradient peut être résumé par le pseudo-code suivant :

Algorithme d'apprentissage

Entrée : Vecteurs d'apprentissage, PMC initial, nombre d'époques (N_epochs), nombre de neurones par couche N= 9,25.. ,40 (taille vecteur)

Sortie : PMC entraîné

1. Initialiser le PMC avec 9 couches cachées contenant N neurones chacune et une couche de sortie utilisant une fonction de classification binaire.
 2. Pour chaque époque de 1 à N_epochs :
 - a. Pour chaque vecteur d'apprentissage dans les vecteurs d'apprentissage :
 - i. Introduire le vecteur d'apprentissage à la couche d'entrée du PMC.
 - ii. Propager l'activation à travers les couches cachées du PMC en utilisant la fonction d'activation appropriée (par exemple, la fonction sigmoïde).
 - iii. Calculer la sortie prédite du PMC en utilisant la fonction d'activation appropriée pour la couche de sortie.
 - iv. Calculer l'erreur entre la sortie prédite et l'étiquette réelle du vecteur d'apprentissage.
 - v. Mettre à jour les poids et les biais du PMC en utilisant la rétropropagation du gradient pour minimiser l'erreur d'apprentissage.
 - b. Répéter l'étape a pour tous les vecteurs d'apprentissage.
 3. Retourner le PMC entraîné.
-

Le PMC obtenu avec les poids finaux est enregistré afin d'être utilisé dans la dernière phase d'exploitation.

4.3. Évaluation et exploitation du PMC

La dernière phase dans l'approche proposée est l'évaluation et l'exploitation du PMC après son entraînement. Il est d'abord crucial de tester le PMC sur une base de données distincte de celle utilisée pour l'apprentissage. Ce test permet d'évaluer les performances du PMC et de détecter les types de données qui posent problème.

Pour effectuer la binarisation d'une image I contenant R pixels, nous utilisons le PMC entraîné. Ainsi, pour chaque pixel p_i de l'image I , avec $1 \leq i \leq R$, nous fournissons au PMC entraîné le vecteur de caractéristiques correspondant à ce pixel, tout comme lors de la phase d'apprentissage. Le PMC renverra une valeur numérique correspondant à la valeur binaire prédite pour ce pixel. Le vecteur de caractéristiques d'un pixel est constitué du niveau de gris du pixel lui-même, ainsi que des niveaux de gris de ses voisins. Cette représentation riche permet d'incorporer à les informations locales du pixel.

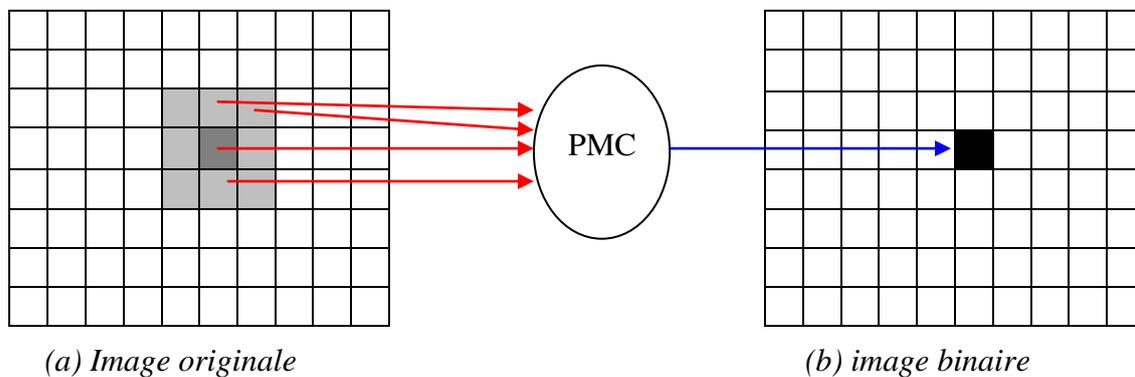


Figure 3.7. Exploitation du PMC entraîné pour la binarisation d'une image

Si les performances du PMC ne sont pas satisfaisantes lors de l'évaluation sur la base de test, plusieurs actions peuvent être envisagées. Il est possible de modifier l'architecture du réseau en ajustant le nombre de neurones, les couches cachées, ou les fonctions d'activation. Une autre possibilité consiste à revoir la base d'apprentissage en ajoutant de nouveaux exemples ou en éliminant des exemples problématiques.

5. Conclusion

Dans ce chapitre, nous avons présenté notre approche pour la binarisation d'images de documents dégradés en utilisant des techniques d'apprentissage automatique. Nous avons décrit en détail les différentes étapes impliquées dans notre méthode, notamment la construction des vecteurs d'entrée à partir des pixels et de leurs voisins, le regroupement des vecteurs similaires à l'aide de l'algorithme K-means, la sélection de pixels représentatifs de chaque groupe, et la définition et l'apprentissage du PMC.

Nous avons souligné l'importance de choisir une architecture appropriée pour le réseau de neurones, en prenant en compte des facteurs tels que la taille de fenêtre et le nombre de neurones dans la couche cachée. Nous avons également justifié notre choix d'utiliser l'ACP pour réduire la dimension des vecteurs d'entrée.

Chapitre 4. Implémentation et résultats

1. Introduction

Dans ce chapitre, et après avoir présenté dans le chapitre précédent la conception détaillée de notre application, nous expliquerons tout ce qui concerne la réalisation de cette application depuis l'environnement de développement, les outils et les bibliothèques utilisées. Ensuite nous présentons la collection d'images utilisée le long de nos expérimentations. Après nous décrivons les étapes de fonctionnement de l'application et ses différentes fenêtres. Dans la dernière section, nous présentons des résultats quantitatifs obtenus.

2. Langage de programmation, Environnement de développement et bibliothèques

2.1. java

La technologie Java définit à la fois un langage de programmation orienté objet et une plateforme informatique. Créée par l'entreprise Sun Microsystems (souvent juste appelée "Sun") en 1995, et reprise depuis par la société Oracle en 2009, la technologie Java est indissociable du domaine de l'informatique et du Web. On la retrouve donc sur les ordinateurs, mais aussi sur les téléphones mobiles, les consoles de jeux, etc. L'avènement du smartphone et la puissance croissante des ordinateurs, ont entraîné un regain d'intérêt pour ce langage de programmation [W1].

2.2. Environnements de développement

L'environnement de travail est constitué par deux parties : environnement matériel et environnement logiciel.

2.2.1. Environnement matériel

L'application a été développée sur un PC ayant les caractéristiques suivantes:

- **Ordinateur** : Dell
- **Système d'exploitation** : Windows 10 Professionnel
- **Processeur** : Intel(R) Core(TM) i5-4300U CPU @ 1.90GHz 2.50 GHz
- **Mémoire RAM** : 4,00 Go

- **Type de système** : Système d'exploitation 64 bits, processeur x64

2.2.2. Environnement logiciel

Notre application a été développée en langage de programmation Java, avec l'environnement de développement Netbeans IDE-8.2-windows (voir la Figure 4.1) :



Figure 4.1. Sigle de Netbeans [W2].

Netbeans est un environnement de développement intégré (EDI), placé en open source par Sun en juin 2000 sous licence CDDL (Common Development and Distribution License) et GPLv2. En plus de Java, NetBeans permet la prise en charge native de divers langages tels le C, le C++, le JavaScript, le XML, le Groovy, le PHP et le HTML, ou d'autres (dont Python et Ruby) par l'ajout de greffons. Il offre toutes les facilités d'un IDE moderne (éditeur avec coloration syntaxique, projets multi-langage, refactoring, éditeur graphique d'interfaces et de pages Web) [W3].

2.3. bibliothèque utilisée

2.3.1. Weka

Weka [W4] est une bibliothèque logicielle open source largement utilisée pour l'apprentissage automatique et l'exploration de données. Elle offre une gamme complète d'outils et d'algorithmes pour la préparation des données, la classification, la régression, le clustering, l'association de règles, la visualisation et l'évaluation des modèles. Weka est programmé en plusieurs langages y compris Java et permet aux utilisateurs de construire des pipelines d'apprentissage automatique en combinant différentes étapes et techniques. Il est également accompagné d'une interface graphique conviviale, ce qui facilite son utilisation, notamment pour les débutants en apprentissage automatique.

3. Collection d'images utilisée pour l'apprentissage et le test

Pour l'apprentissage, et le test de notre application, nous avons utilisé les collections d'images fournies dans le cadre des compétitions DIBCO/H-DIBCO et plus précisément DIBCO 2009 [GAT 09]. Ce sont des compétitions internationales de binarisation d'images de documents. Les images de la collection de test sont réparties comme suit :

	Nombre d'images
Apprentissage	15
Test	7
Total	22

Tableau 4.1. Répartition des images de notre collection.

Les figures suivantes présente quelques exemples documents de notre collection :

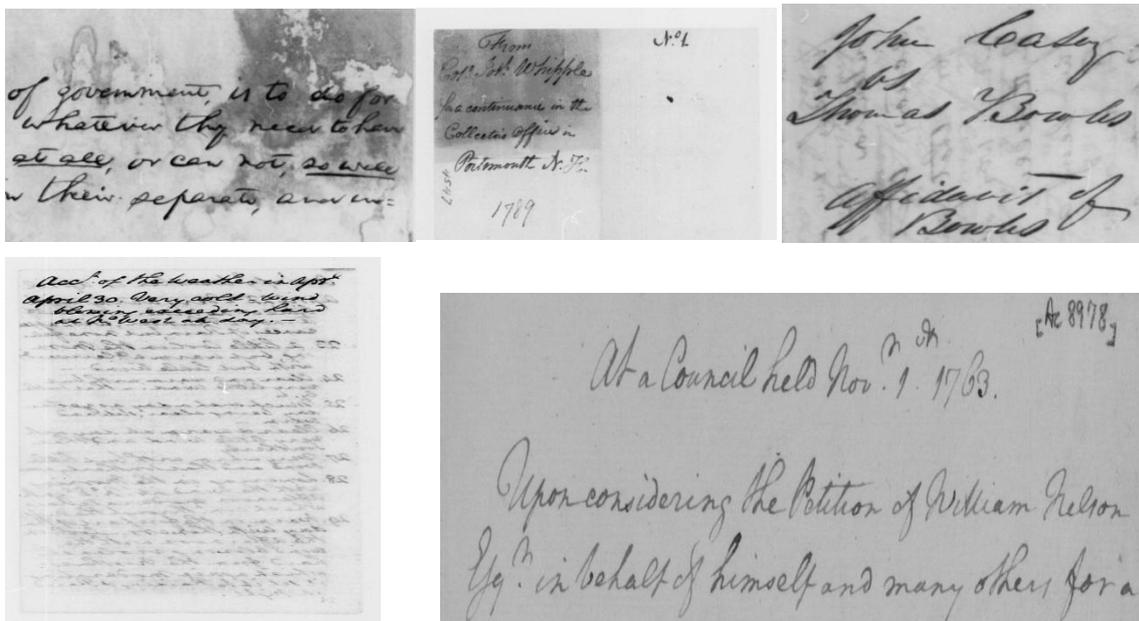


Figure 4.2. Exemples documents de notre collection

4. Présentation de l'application

Au démarrage de l'application, la fenêtre principale de notre application s'affiche à l'écran comme le montre la figure suivante :

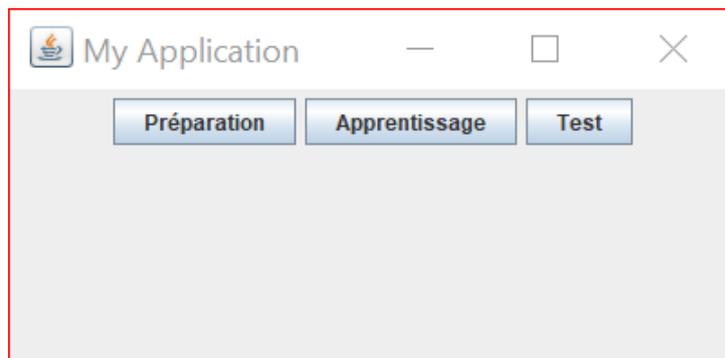


Figure 4.3. Interface principale de notre application.

Cette interface nous permet d'accéder aux différentes fonctionnalités de l'application. Ainsi, l'utilisateur peut choisir la fonctionnalité qu'il veut en cliquant sur le bouton correspondant.

Ainsi, le système contient trois fenêtres, chacune d'elles a une fonction spéciale et nous allons maintenant présenter et expliquer les fonctions et caractéristiques de ces fenêtres.

4.1. Fenêtre de préparation des données d'apprentissage

En cliquant sur le bouton "Préparation", la fenêtre illustrée par la (Figure 4.4) s'affiche. C'est la fenêtre permettant à l'utilisateur de préparer des données d'apprentissage.

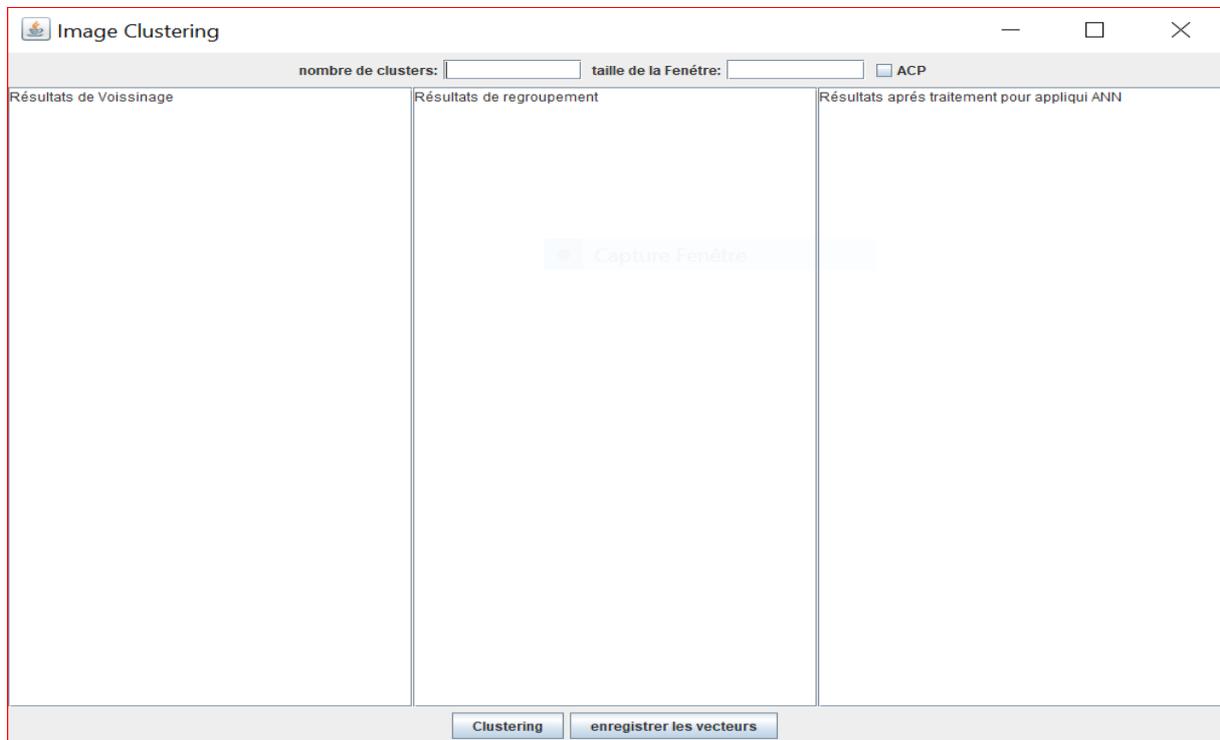


Figure 4.4. Fenêtre de préparation des données d'apprentissage.

Cette fenêtre nous donne la possibilité de traiter une seule image, ou bien plusieurs images à partir des fichiers spécifiés dans le code (images en niveaux de gris avec son image binaire).

Comme nous avons déjà dit dans le chapitre précédent, la préparation des données d'apprentissage passe par plusieurs étapes. Toutes ces étapes sont réunies dans la présente fenêtre.

Le premier chose à faire est de déterminer le nombre de clusters de l'algorithme Kmeans, et la taille de la fenêtre dans le voisinage de pixels, en saisissant leur valeur dans les zones de texte dédiées. Voir la figure 4.5.

Pour la taille de la fenêtre, nous avons testé dans notre application 3 tailles différentes de fenêtre : 3×3 , 5×5 , et 7×7 . Pour le nombre de clusters, nous avons fixé le nombre 10 clusters .



Figure 4.5. Zones de textes permettant de choisir le nombre de clusters et la taille de la fenêtre.

Dans cette fenêtre nous disposons également d'une case à cocher "ACP". Si on la sélectionne, l'application applique une analyse en composantes principales (ACP) aux instances de pixels blancs et noirs pour réduire leur dimensionnalité.

En validant les choix, nous cliquons sur le bouton « Clustering » pour effectuer le regroupement des pixels similaires des images spécifiées en utilisant l'algorithme K-means (Figure 4.6) ou K-means avec ACP (Figure 4.7).

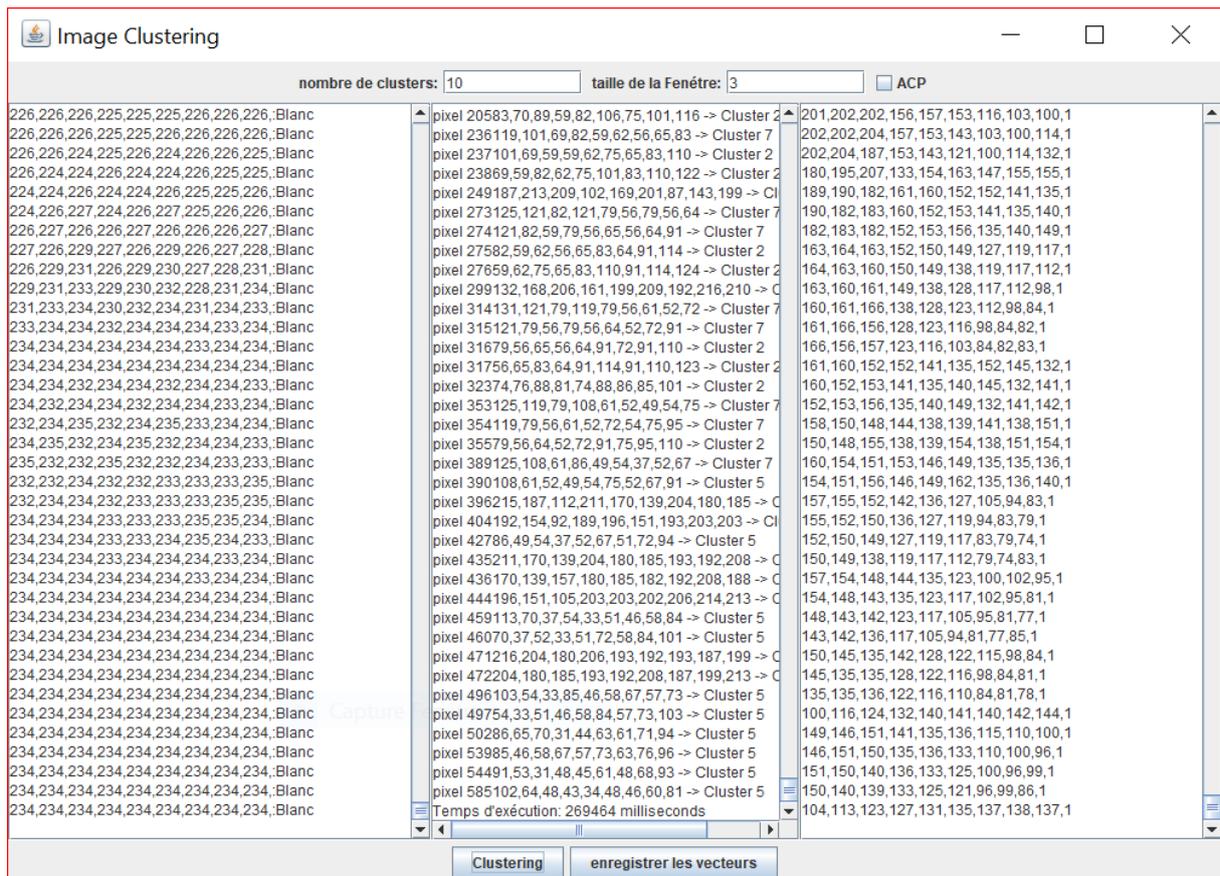


Figure 4.6. Affichage du regroupement .

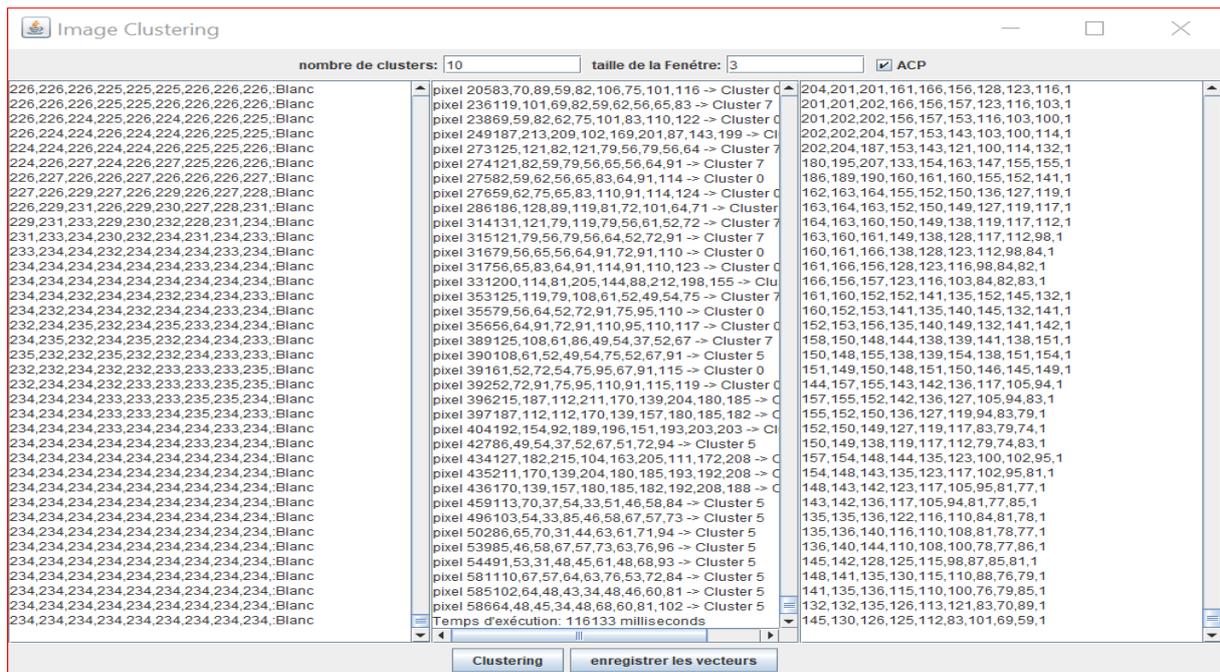


Figure 4.7. Affichage du regroupement avec ACP

Nous expliquons les affichages obtenus:

1. La zone de gauche affiche les résultats de voisinage d'une images du document.
2. La zone de mélieu affiche les résultats de regroupement du pixels similaires par k-means , Il affiche les vecteurs noir et les vecteurs blanc .
3. La zone de droit affiche les instances ou les vecteurs sélectionnés aléatoirement) partir des clusters (10 vecteurs de chaque cluster) .

Le bouton « Enregistrer les vecteurs », sauvegarde les données générées des images traitées dans un fichier CSV.

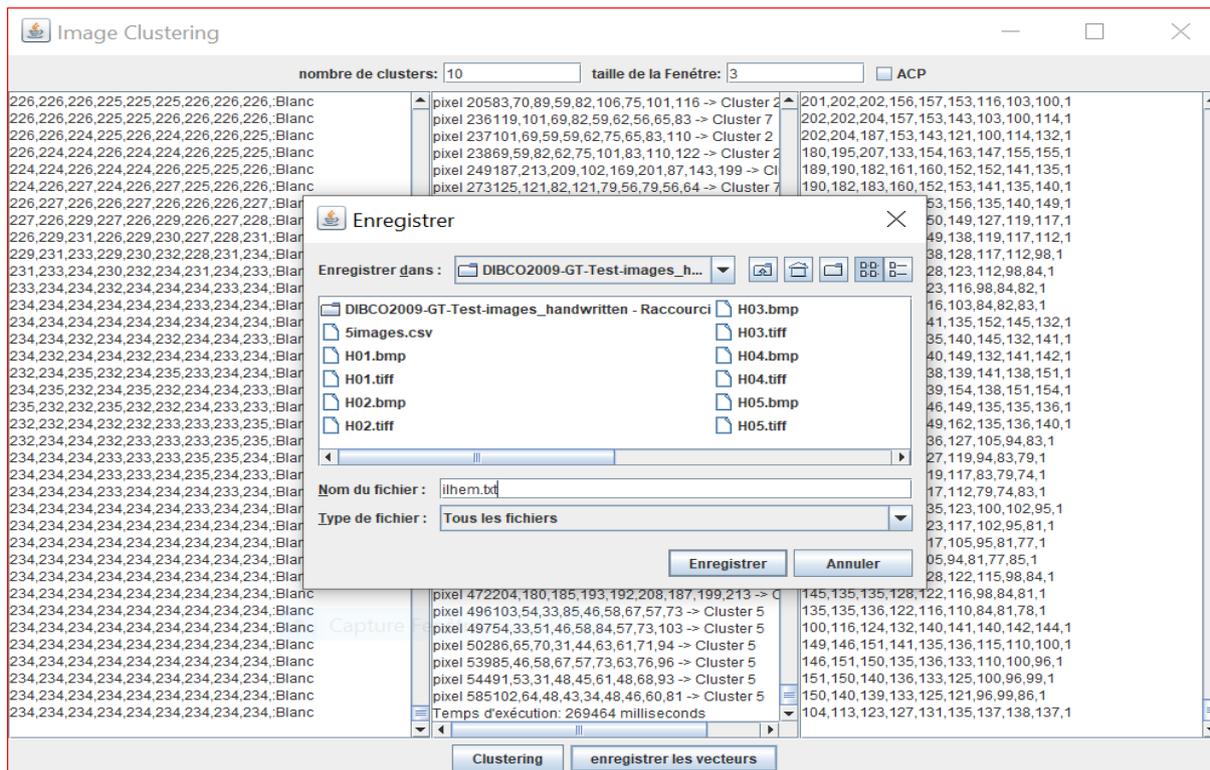
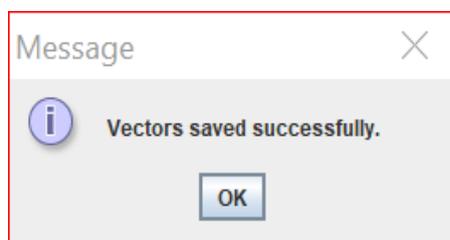


Figure 4.8. Boîte de dialogue «Enregistrer les vecteurs ».



Fégre 4.9. Affichage de message a été enregistré

4.2. Fenêtre de l'apprentissage du PMC

La fenêtre d'apprentissage peut être activée on cliquant sur le bouton "Apprentissage" dans l'interface principale (Figure 4.3). La fenêtre de l'apprentissage est la suivante :

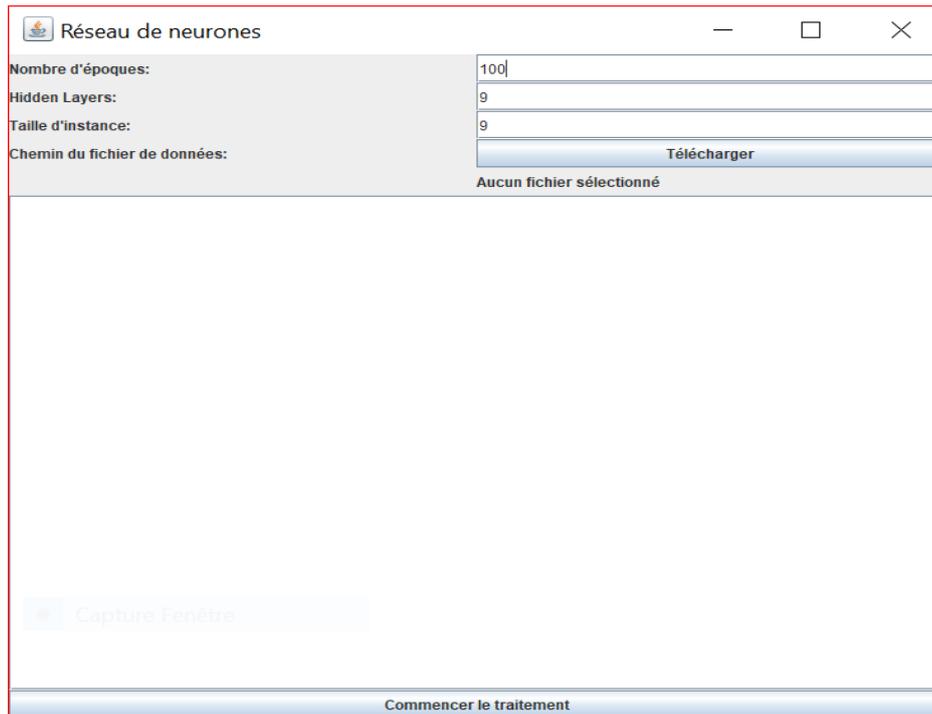


Figure 4.10. Fenêtre de l'apprentissage.

A partir de cette fenêtre, l'utilisateur choisit les paramètres tels que le nombre d'époques, le nombre de couches cachées et la taille. En cliquant sur le bouton « Télécharger » pour subdiviser la base d'apprentissage (sous forme d'un fichier CSV) déjà construite en deux : un fichier d'apprentissage et un fichier de test.

Après, on lance l'apprentissage par le bouton « Commencer le traitement » (Figure 4.11).

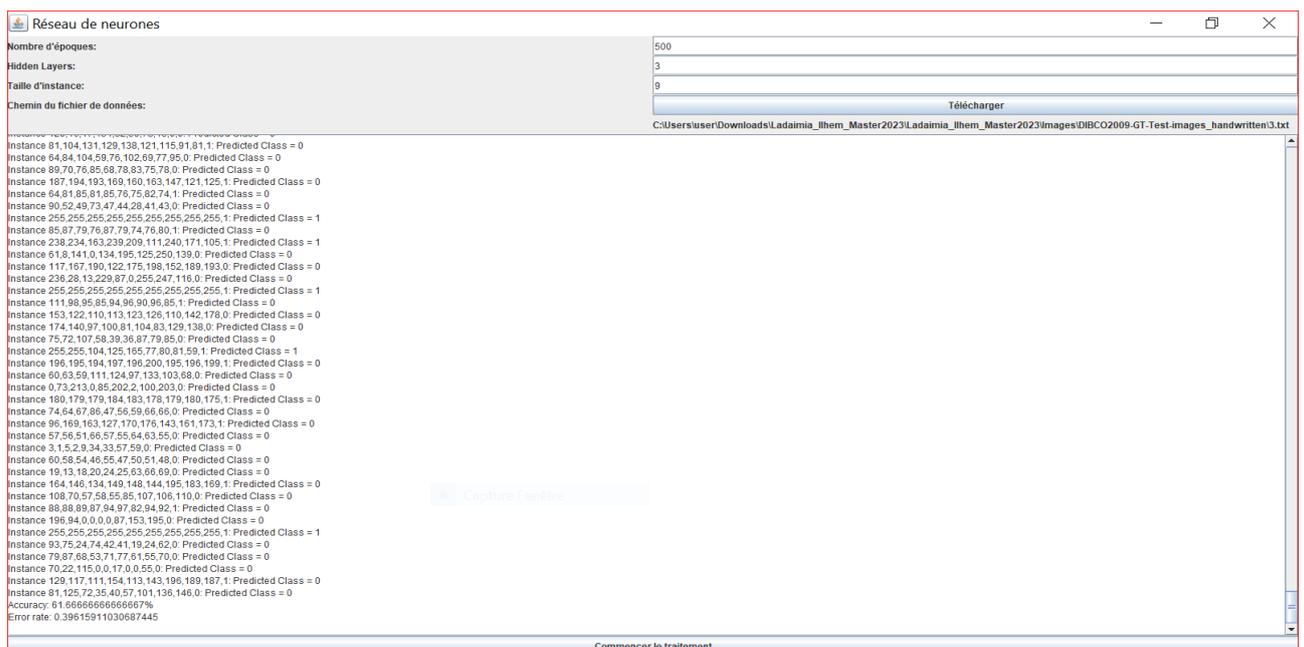
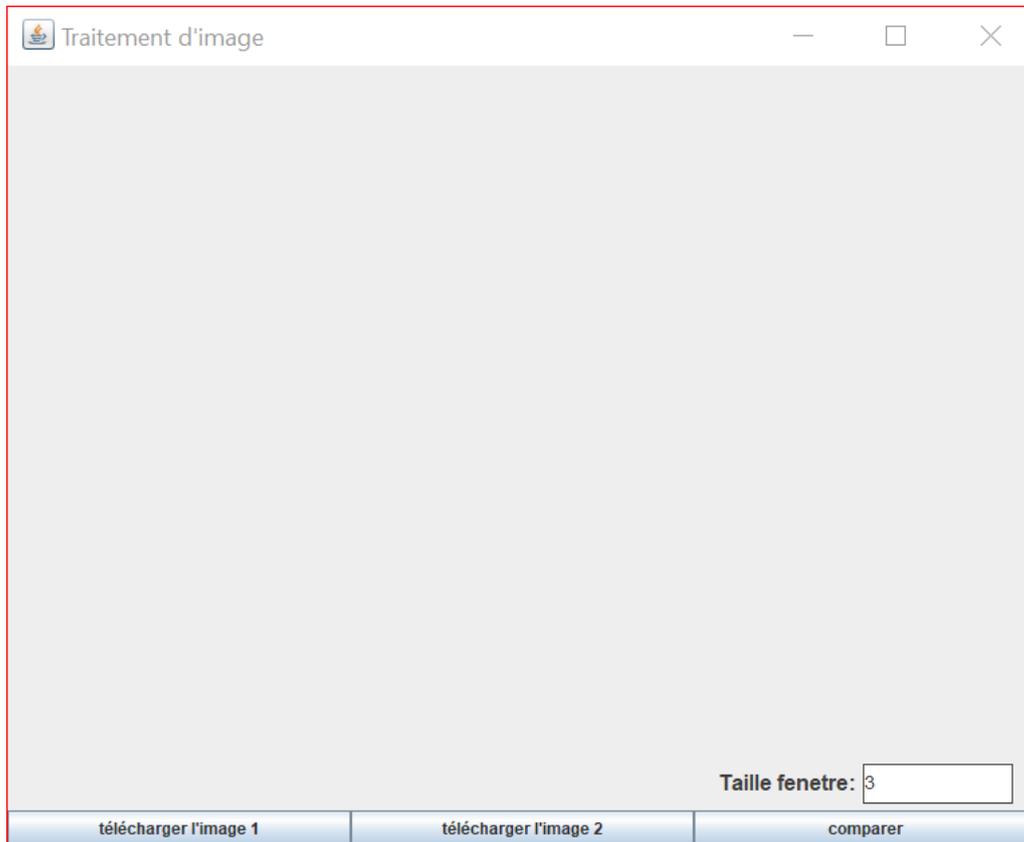


Figure 4.11. Lancement de l'apprentissage.

Une fois l'apprentissage est terminé, le PMC entraîné est enregistré automatiquement.

4.3. Fenêtre du test

La fenêtre du test est le produit final de notre application. C'est l'interface à partir de laquelle on peut binariser des nouvelles images en utilisant le PMC entraîné précédent. Cette fenêtre s'affiche comme suit :



Fégre 4.12. Fenêtre du test .

A partir de cette fenêtre, on doit d'abord sélectionner l'image qu'on veut binariser et le PMC qui va faire cette binarisation. On commence par importer le réseau de neurones enregistré auparavant, et on clique sur le bouton « télécharger l'image 1 ». Le résultat de binarisation s'affiche dans le panneau 1 comme le montre la figure 4.13.

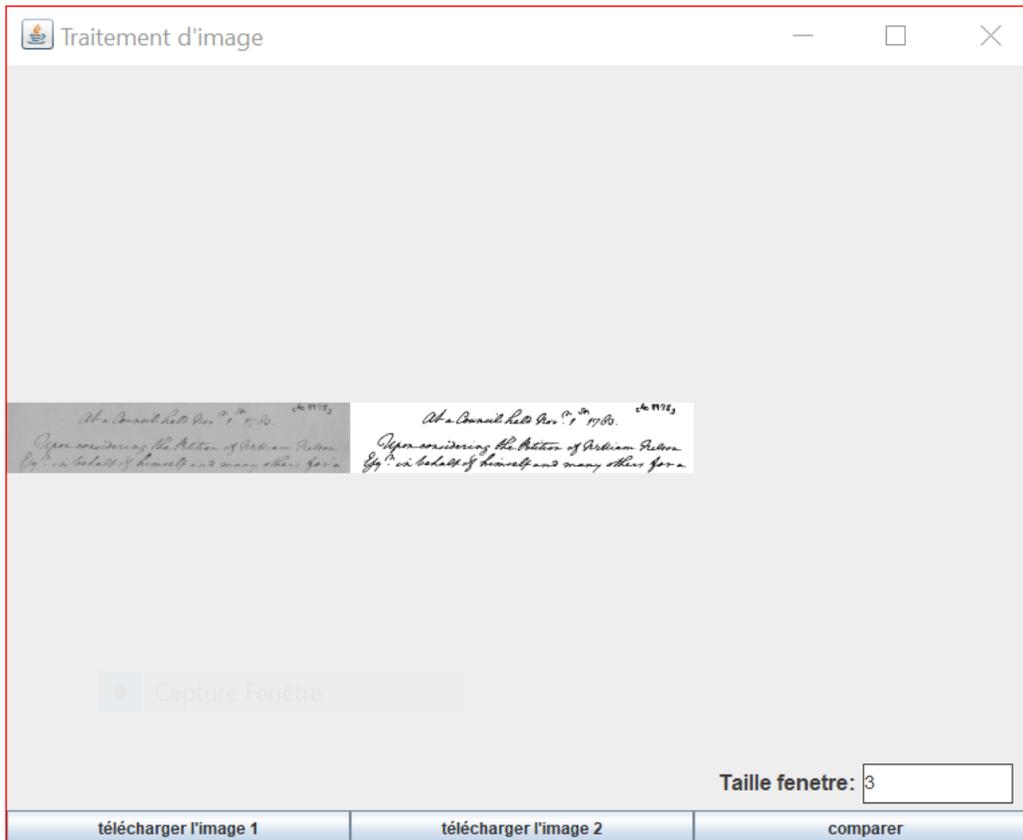


Figure 4.13. Affichage du résultat de binarisation.

Le bouton «télécharger l'image 2» permet de charger l'image de vérité terrain correspondante à l'image traitée pour des besoins de comparaison. L'image de vérité terrain chargée s'affiche à côté de l'image traitée dans le panneau principal (Figure 4.14).



Figure 4.14. Mise en correspondance de l'image binarisée et l'image de vérité terrain.

Finalement, pour évaluer la qualité de l'image binarisée, nous cliquons sur le bouton « Comparer ». Ce dernier permet de comparer l'image binarisée et l'image de vérité terrain en termes de Fmeasure et affiche le résultat de comparaison comme le montre la figure suivante:

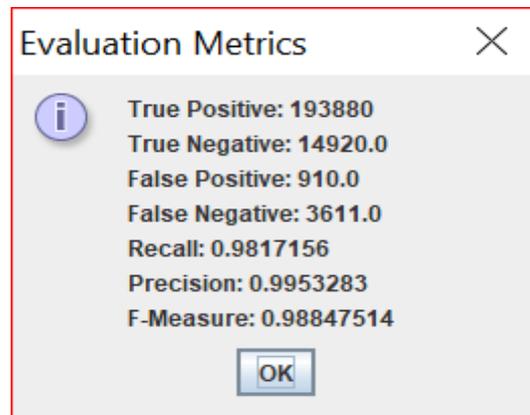


Figure 4.15. Affichage du résultat de comparaison en termes de F-measure.

5. Résultats obtenus

Comme nous avons déjà dit, le choix de la topologie adéquate de notre réseau de neurones artificiels est effectué par expérimentations. Cependant, nous avons testé plusieurs topologies, en les appliquant sur l'ensemble de test composé de 7 images de documents dégradés, comparé leurs résultats avec les images de vérité terrain, et choisi la topologie qui fournit les meilleurs résultats. La comparaison a été effectuée en termes de Fmeasure défini par :

$$FMeasure = \frac{2 \times Recall \times Precision}{Recall + Precision}$$

$$\text{Tel que : } Recall = \frac{TP}{TP+FN}, Precision = \frac{TP}{TP+FP}$$

TP, FP, FN dénotent le vrai positif, faux positif, et faux négative respectivement.

- Un pixel noir est classé comme vrai positif s'il est présent dans les deux images : de vérité terrain et celle résultante de la binarisation.
- Un pixel est classé comme faux positif s'il est présent seulement dans l'image binarisée.
- Un pixel est un faux négatif s'il est présent seulement dans l'image de vérité terrain.

Les résultats obtenus par 3 tailles différentes de fenêtre sur chaque image de test, ainsi que le résultat moyen sont résumées par le (Tableau 5.1).

Image Taille	1	2	3	4	5	6	7	Moyenne
3*3	0,99	0,97	0,99	0,97	0,97	0,99	0,98	0,98
5*5	0,98	0,97	0,99	0,98	0,98	0,97	0,98	0,97
7*7	0,98	0,95	0,98	0,98	0,97	0,96	0,98	0,96

Tableau 5.1. Résultats obtenus par différents fenêtres en termes de Fmeasure.

A partir du tableau précédent, les meilleures performances (*Fmeasure* max) ont été obtenues avec la taille (3*3) ce que signifie que c'est la taille de fenêtre la plus adéquate parmi les topologies testées.

Nous avons également expérimenté des topologies différents du réseau de neurones et choisi celle optimale mentionnée dans le chapitre 3.

6. Conclusion

A travers ce dernier chapitre nous avons présenté un guide d'utilisateur de notre application basée sur la méthode Kmeans, ACP et les réseaux de neurones artificiels pour la binarisation d'images de documents dégradés.

Conclusion générale

Le travail adressé dans ce mémoire a pour but le développement d'un outil pour la binarisation d'images de documents dégradés. Comme nous avons dit précédemment, la binarisation est toujours une des premières étapes utilisées avant l'étape de reconnaissance. Elle permet d'extraire le fond du texte du document. Plusieurs techniques de binarisation ont été proposées dans la littérature. Ces méthodes calculent un ou plusieurs seuils pour effectuer la binarisation.

Dans ce mémoire, nous avons proposé une nouvelle méthode de binarisation basée sur la classification directe des pixels en une des deux classes « noir » et « blanc » et non pas sur le seuillage comme la plupart des techniques de la littérature. Ainsi, la classification a été effectuée à l'aide d'un réseau de neurones artificiel (RNA) de type PMC entraîné en utilisant l'algorithme de rétro-propagation du gradient. Le principe de la méthode proposé est de faire apprendre au PMC la valeur binaire de quelques pixels dits d'apprentissage et le PMC soit capable de trouver la valeur binaire pour le reste des pixels. L'objectif de l'utilisation des RNAs, et plus précisément le PMC, dans la binarisation d'images est de combler le manque de l'utilisation des techniques d'apprentissage automatique dans tel problème, et de profiter de la capacité de généralisation du PMC.

Plusieurs expérimentations ont été effectuées afin de déterminer la topologie adéquate du PMC. Nous avons ainsi testé plusieurs topologies du MLP, en les appliquant sur l'ensemble de test composé de 7 images de documents dégradés, comparé leurs résultats avec les images de vérité terrain, et choisir la topologie qui fournit les meilleurs résultats.

Plusieurs extensions sont envisageables :

- Enrichir de plus la base d'apprentissage qualitativement et quantitativement
- Modifier la procédure du choix des pixels d'apprentissage.
- Penser à expérimenter d'autres classifieurs (par exemple HMM ou SVM) ou encore à combiner plusieurs classifieurs pour la binarisation d'images de documents.

Finalement, notre souhait ultime est que cette étude soit un modèle pour d'autres.

Références

- [Ac,21] «apprentissage supervisevs non supervise,» (Enligne). Available: <https://analyticsinsights.io/apprentissage-supervise-vs-non-supervise/>.consulté le 07/04/2023.
- [Acc,21] «machine-learning-classification-vs-regression», 19 Juillet 2019.(En Ligne). Available:<https://dev.to/petercour/machine-learning-classification-vs-regression-1gn>. Consulté le 16/03/2023.
- [Afr,15] Afzal, M. Z., Pastor-Pellicer, J., Shafait, F., Breuel, T. M., Dengel, A., & Liwicki, M. (2015, August). Document image binarization using lstm: A sequence learning approach. In Proceedings of the 3rd international workshop on historical document imaging and processing (pp. 79-84).
- [Ara,96] Arabie,P.,Hubert, L., De Soete, G., & Gordon, A. (1996). Hierarchical classification. P. Arabie, L. Hubert, G. De Soete, & A. Gordon, Clustering and classification, 65-121.
- [Ba,16] Baka, A., & Fillali, H. (2016). Traitement et reconnaissance des caractères, Mémoire de Master, Université de M'hamed Bougara - Boumerdes.
- [Bal,67] Ball, G. H. et Hall, D. J. ISODATA, an Iterative Method of Multivariate Analysis and Pattern Recognition. Behavior Science, 153, 1967
- [Be,11] Bo, L., Ren, X., & Fox, D. (2011, September). Depth kernel descriptors for object recognition. In 2011 IEEE/RSJ International Conference on Intelligent Robots and Systems (pp. 821-826). IEEE.
- [Bed,07] E. Badeskas, N. Papamarkos (2007). Optimal Combination of Document Binarization Techniques Using a Self Organizing Map Neural Network.*Engineering Applications of Artificial Intelligence*, Vol. 20, No. 1, pp. 11-24.
- [Ber,86] J. Bernsen, « Dynamic thresholding of grey-level images », in Proc. 8th International Conference on Pattern Recognition, p. 1251-1255, Paris, France, 1986
- [Bez,81] J.C. Bezdek (1981):"Pattern Recognition with Fuzzy Objective Function Algorithms", Plenum Press, New York. 1981.
- [Bou,22] BOUKERTOUTA, M. A. (2022). Detection des intrusions basée sur l'apprentissage automatique dans les systèmes IdO (Internet des Objets).
- [Bot,98] L. Bottou, P. Haffner, and P.G. Howard, "High Quality Document Image Compression with DjVu," Journal of Electronic Imaging, vol 7, no 3, pp 410-425, SPIE, 1998.
- [Cha,16] Chauhan, S., Sharma, E., & Doegar, A. (2016, September). Binarization techniques for degraded document images—A review. In 2016 5th international conference on reliability, infocom technologies and optimization (Trends and Future Directions)(ICRITO) (pp. 163-166). IEEE.
- [Che,93] T. Chen, M. Takagi (1993). Image Binarization by Back Propagation Algorithm.*International Archives of Photogrammetry and Remote Sensing*, Vol.29, pp. 345-345.
- [Cho,10] Chou, C. H., Lin, W. H., & Chang, F. (2010). A binarization method with learning-built rules for document images produced by cameras. Pattern Recognition, 43(4), 1518-1530.

- [Djo,15] A.E.DJOKHRAB,«Planification et Optimisation de Trajectoire d'un Robot Manipulateur à 6 D.D.L.par des Techniques Neuro-Floues,» Mastersthesis ,UniversitéMohamed Khider, Biskra ,2015.
- [Dun,73] J. C. Dunn (1973) : "A Fuzzy Relative of the ISODATA Process and Its Use in Detecting Compact Well-Separated Clusters", *Journal of Cybernetics* ,no 3, pp 32-57. 1973
- [Fan,08] Fang, X. (2008). Texture-based image binarization using support vector machines. *Image and Vision Computing*, 26(11), pp. 1465-1472.
- [Gar,04] C. Garcia and M. Delakis. Convolutional face finder: A neural architecture for fast and robust face detection. *IEEE Transactions on pattern analysis and machine intelligence*, 26(11), November 2004.
- [Gar,06] Garain, U., Paquet, T., & Heutte, L. (2006). On foreground—background separation in low quality document images. *International Journal of Document Analysis and Recognition (IJ DAR)*, 8(1), 47-63.
- [GAT 09] Gatos, Basilis, Konstantinos Ntirogiannis, and Ioannis Pratikakis. "ICDAR 2009 document image binarization contest (DIBCO 2009)." *2009 10th International conference on document analysis and recognition*. IEEE, 2009.
- [Haj,14] Hajjar, C. S. (2014). Cartes auto-organisatrices pour la classification de données symboliques mixtes, de données de type intervalle et de données discrétisées (Doctoral dissertation, Supélec).
- [Ham,05] H. Hamza, E. Smigiel, A. Belaid (2005). Neural based binarization techniques.*In Proceedings of the 8th International Conference on Document Analysis and Recognition ICDAR*, pp. 317-321.
- [Jud,18] D.k.JudithHurwitz,MachineLearning,IBM,2018.
- [Kal,18] O.M.KALOUNESalim,«Classification d'objet save cle Deep Learning,» Université Akli Mohand Oulhadj de Bouira Faculté des Sciences et des Sciences Appliquées, Bouira, 2018.
- [Kap,85] J.N. Kapur, P.K. Sahoo, A.K.C. Wong, "A New method for gray-level picture threshold using the entropy of the histogram", *Computer Vision, Graphics, and Image Processing*, vol.29, No. 3,
- [Kha,07] A. Khashman, B. Sekeroglu (2007). Global Binarization of Document Images Using a Neural Network. *Third International IEEE Conference on Signal-Image Technologies and Internet-Based System SITIS*, Shanghai, pp. 665-672.
- [Kid,16] Andrea Kidd TAYLOR, Kyle ESDAILLE et Jennifer AMES. « Integrated Pest Management Policies in America's Schools : Is Federal Legislation Needed ? » In : *The Toxic Schoolhouse* (2016).
- [Koh,10] Kohei Kita, Toru Wakahara, "Binarization of Color Characters in Scene Images Using k-means Clustering and Support Vector Machines", *International Conference on Pattern Recognition*, pp. 3183 - 3186, 2010.
- [Kro,08] Krogh, A. (2008). What are artificial neural networks?. *Nature biotechnology*, 26(2), 195-197.
- [Kum,21] Kumar, S. (2021). Comparison of deep reinforcement learning algorithms in a self-play setting (Doctoral dissertation).

- [Laz,10] J. Lazaro, J.L. Martin, J. Arias, A. Astarloa, C. Cuadrado (2010). Neuro Semantic Thresholding for High Precision OCR Applications. *Image and Vision Computing*, Vol. 28, No. 4, pp. 571-578.
- [Ley,04] Y. Leydier, F.L. Bourgeois, and H. Emptoz, "Serialized k-Means for Adaptative Color Image Segmentation Application to Document Images and Others," in 6th International Workshop on Document Analysis Systems (DAS), Florence, Italy, September, 2004, Lecture Notes in Computer Science (LNCS), Springer-Verlag, Germany, vol. 3163 pp. 252-263.
- [Mah,20] Mahesh, B. (2020). Machine learning algorithms-a review. *International Journal of Science and Research (IJSR)*. [Internet], 9, 381-386.
- [Mar,17] M.R.K.A.M. Giancarlo Zaccone, *Deep Learning with TensorFlow: Explore neural networks with Python*, Packt, 24 avril 2017.
- [McD,89] McDonald, C. "Machine learning: a survey of current techniques". *Artificial Intelligence Review*, vol. 3, no 4, pp. 243-280, 1989.
- [Mit,07] Mitchell, T. M. (2007). *Machine learning* (Vol. 1). New York: McGraw-hill.
- [Nib,86] W. Niblack, « An Introduction to Digital Image Processing », Ed. Prentice Hall, Englewood Cliffs, 1986.
- [Nor,05] Noriega, L. (2005). Multilayer perceptron tutorial. School of Computing. Staffordshire University, 4, 5.
- [Nti,12] Ntirogiannis, K., Gatos, B., & Pratikakis, I. (2012). Performance evaluation methodology for historical document image binarization. *IEEE Transactions on Image Processing*, 22(2), 595-609.
- [Ot,79] Otsu, N. (1979). A threshold selection method from gray-level histograms. *IEEE Transactions on Systems, Man, and Cybernetics*, 9(1), 62-66.
- [Pap,00] Papamarkos, N., Strouthopoulos, C. and Andreadis, I. (2000) 'Multithresholding of color and gray-level images through a neural network technique' *Image and Vision Computing*, Vol. 18 No. 3, pp. 213-222.
- [Pap,03] N. PAPAMARKOS, « A neuro-fuzzy technique for document binarisation », *Neural Computing & Applications*, vol. 12, no 3-4, p. 190-199, 2003.
- [Ray,19] Ray, S. (2019, February). A quick review of machine learning algorithms. In 2019 International conference on machine learning, big data, cloud and parallel computing (COMITCon) (pp. 35-39). IEEE
- [Ros,96] J Ross QUINLAN. « Learning decision tree classifiers ». In : *ACM Computing Surveys (CSUR)* 28.1 (1996), p. 71-72
- [Sai,07] Saidane, Z., & Garcia, C. (2007, September). Robust binarization for video text recognition. In *Document Analysis and Recognition, 2007. ICDAR 2007. Ninth International Conference on* (Vol. 2, pp. 874-879). IEEE.
- [Sai,15] Saini, R. (2015). Document image binarization techniques, developments and related issues: a review. *International Journal of Computer Applications*, 116(7), 0975-8887.
- [Sam,15] Samia, B. (2015). Segmentation d'images par seuillage non paramétrique d'histogrammes Couleur, Mémoire de Master , Université Mouloud Mammeri.

- [Sau,97] J. Sauvola, T. Seppänen, S. Haapakoski, M. Pietikänen, « Adaptive document binarisation », Fourth International Conference Document Analysis and Recognition (ICDAR), p. 147-152, Ulm, Germany, August 1997.
- [Su,07] Su, L. (2007). Image binarization using support vector machines. *Pattern Recognition*, 40(3), pp. 968-976
- [Suk,22] Sukesh, R., Seuret, M., Nicolaou, A., Mayr, M., & Christlein, V. (2022). A Fair Evaluation of Various Deep Learning-Based Document Image Binarization Approaches. In *International Workshop on Document Analysis Systems* (pp. 771-785). Springer, Cham.
- [Sul,19] Sulaiman, A., Omar, K., & Nasrudin, M. F. (2019). Degraded historical document binarization: A review on issues, challenges, techniques, and future directions. *Journal of Imaging*, 5(4), 48.
- [Ten,17] Tensmeyer, C., & Martinez, T. (2017, November). Document image binarization with fully convolutional neural networks. In *2017 14th IAPR international conference on document analysis and recognition (ICDAR)* (Vol. 1, pp. 99-104). IEEE.
- [Ten,20] Tensmeyer, C., & Martinez, T. (2020). Historical document image binarization: a review. *SN Computer Science*, 1(3), 1-26.
- [Vel,80] F.R.D. Velasco, « Thresholding using the ISODATA clustering algorithm », *IEEE Transaction on system, Man and Cybernetics*, 10, p. 771-774, 1980.
- [Vo,18] Vo, Q. N., Kim, S. H., Yang, H. J., & Lee, G. (2018). Binarization of degraded document images based on hierarchical deep supervised network. *Pattern Recognition*, 74, 568-586.
- [Wes,18] Westphal, F., Lavesson, N., & Grahn, H. (2018, April). Document image binarization using recurrent neural networks. In *2018 13th IAPR International Workshop on Document Analysis Systems (DAS)* (pp. 263-268). IEEE.
- [Wie,12] Wiering, M. A., & Van Otterlo, M. (2012). Reinforcement learning. *Adaptation, learning, and optimization*, 12(3), 729.
- [Wu,03] Wu, S., & Amin, A. (2003, August). Automatic thresholding of gray-level using multistage approach. In *Seventh International Conference on Document Analysis and Recognition, 2003. Proceedings.* (pp. 493-497). IEEE.
- [Wu,20] Wu, Q., Sun, Y., Yan, H., & Wu, X. (2020). ECG signal classification with binarized convolutional neural network. *Computers in biology and medicine*, 121, 103800.
- [W1] <https://www.journaldunet.fr/web-tech/dictionnaire-du-webmastering/1203555-java-definition/> consulté le 20/04/2023
- [W2] <https://d-3elm.com/%D8%A8%D8%B1%D9%86%D8%A7%D9%85%D8%AC-netbeans-ide/> consulté le 03/03/2023
- [W3] <https://www.oracle.com/ch-fr/tools/technologies/netbeans-ide.html> consulté le 14/04/2023
- [W4] <https://www.cs.waikato.ac.nz/ml/weka/> consulté le 30/03/2023
- [W5] <https://www.groupe-hli.com/machine-learning-dans-industrie/>. consulté le 25/03/2023
- [Xio,18] Xiong, W., Xu, J., Xiong, Z., Wang, J., & Liu, M. (2018). Degraded historical document image binarization using local features and support vector machine (SVM). *Optik*, 164, 218–223. doi:10.1016/j.ijleo.2018.02.072