**REPUBLIQUE ALGERIENNE DEMOCRATIQUE ET POPULAIRE**
**Ministère de l'Enseignement Supérieur et de la Recherche Scientifique**
**Université Mohamed Khider – BISKRA**
**Faculté des Sciences Exactes, des Sciences de la Nature et de la Vie**
**Département d'informatique**

**N° d'ordre : …./M2/2021**

## Mémoire

Présenté pour obtenir le diplôme de master académique en

## Informatique

Parcours : **Intelligence Artificielle (IA)**

---

# La détection d'anomalie pour la lutte contre la fraude documentaire

---

**Par :**
**DJEFFAL AMANI**

Soutenu le 26/06/2022 devant le jury composé de :

| | | |
|---|---|---|
| Belouaar Houcine | MCA | Président |
| Tibermacine Ahmed | MCA | Rapporteur |
| Belaich Hamza | MCB | Examinateur |

Année universitaire 2021-2022

# Dédicace

## *Je dédie ce travail :*

## A ma chère mère,

## A mon cher père,

**Qui n'ont jamais cessé, de formuler des prières à mon égard, de me soutenir et de m'épauler pour que je puisse atteindre mes objectifs qu'Allah leur bénisse.**

**Au Dr BENAMEUR Sabrina ma chère tante pour *son aide précieus et pour le temps qu'elle m'a consacré pour m'aider, qui n'a pas cessé a me conseiller, encourager et soutenir tout au long de mes études. Que dieu la protège pour sa petite famille et lui offre chance et bonheur et santé.***

***Un grand merci à mon ami Tibermacine Imad de m'avoir guidé tout au long de mon travail avec ses précieux conseils et son aide.***

***A mes très chers amies Aicha, Marthe,Sarah, kaouthar, Dounia, je serais toujours reconnaissante de vous avoir croisé et d'avoir partager de beaux moments avec vous, je vous souhaite beaucoup de réussite et de bonheur.***

***A ma famille Djaffel et Benameur cousins et cousines (.....), j'espère de tout cœur qu'on sera toujours solidaire les uns pour les autres.***

***A tous ceux qui ont contribué de pres ou de loin pour accomplir ce travail.***

# Remerciement

***Je remercie DIEU le tout puissant de m'avoir donné la santé et la volonté et le courage de terminer ce travail.***

**Tout d'abord je tiens à remercier Dr TIBERMACINE Ahmed pour la qualité de son encadrement et sa rigueur et sa disponibilité durant ma préparation de ce mémoire.**

**Enfin, j'adresse mes plus sincères remerciements à ma famille : Mes parents, tous mes proches et amis, qui m'ont accompagné, aidé, soutenu et encouragé tout au long de la réalisation de ce mémoire.**

**Merci.**

# Résumé

Avec la généralisation de l'utilisation des documents numériques dans les administrations, la fabrication et l'utilisation de faux documents sont devenus un problème sérieux. Dans ce travail, Nous avons développé un système de détection d'anomalie, adapté à la détection de la fraude documentaire en utilisant une approche basée sur d'apprentissage profond pour la détection de fraude. Nous avons implémenté un autoencodeur convolutif pour réaliser notre objectif. L'extraction des caractéristiques, la reconstruction d'image et le calcul de la fonction de perte sont les principales étapes de la détection de fraude dans notre système. Les résultats expérimentaux démontrent la performance de notre algorithme



**Mots-clés :** Détection du fraude, Extractions des caractéristiques, reconstruction, fonction de perte, Apprentissage, Réseaux de neurones, Autoencodeur convolutif.



## Abstract

With the wide spread of digital document use in administrations, fabrication and use of forged documents have become a serious problem. In this work, we have developed an anomaly detection system, adapted to the detection of documentary fraud using a deep learning based approach for fraud detection. We implemented a convolutional autoencoder to achieve our goal. Feature extraction, image reconstruction and loss function calculation are the main steps of fraud detection in our system. The experimental results demonstrate the performance of our algorithm.



**Key-words**: Fraud Detection, Feature Extractions, Reconstruction, Loss Function, Learning, Neural Networks, Convolutional Autoencoder.

# Table des matières

# Liste des figures

# Liste des Tableaux

# Introduction générale

La lutte contre la fraude est un enjeu économique très important, que ce soit pour les entreprises, pour les administrations ou pour les particuliers. Plusieurs études tendent à montrer que les entreprises cherchent de plus en plus à se doter d'outils performants de détection des fraudes, quel que soit leur domaine d'activité [1].

Les faux documents qui nous ont particulièrement intéressés sont les documents de la vie de tous les jours. Ils peuvent être de toutes sortes : factures, bulletins de salaire, avis d'imposition, certificats médicaux, justificatifs de scolarité, diplômes, quittances de loyer, curriculum vitae... Falsifier un document, c'est donc chercher à obtenir davantage, chercher à prouver quelque chose qui n'est pas forcément vrai, afin, généralement, de gagner du temps ou de l'argent. Aujourd'hui, la transformation numérique accélère le partage de documents et donne les moyens à tout un chacun de créer, numériser, modifier, copier-coller, effacer, éditer ou imprimer toutes sortes de documents. Il y a donc moins d'obstacles matériels à la fraude documentaire. Les entreprises et les administrations, prennent de plus en plus conscience de cet enjeu et cherchent donc à se prémunir de pertes en cherchant des solutions matérielles ou logicielles pour l'automatisation de la détection des faux documents [1].

Nous nous somme intéressé à la détection d'anomalie pour la lutte contre la fraude documentaire pour cela nous avons conçu et implémenté un système basé sur une méthode de l'apprentissage profond qui est l'autoencodeur convolutif. Notre système fait un apprentissage sur une base d'images authentiques, ceci lui permet de détecter toute autre image falsifié.

## Organisation du manuscrit

En plus de l'introduction générale, notre mémoire est organisé en 3 chapitres,

Le premier chapitre est consacré à l'apprentissage automatique. Dont nous commençons par l'objectif de ce domaine, par la suite nous détaillons ces différents types ainsi que les modèles existant. A la fin du chapitre nous citons les différents algorithmes d'apprentissage.

Dans le chapitre 2, nous essayons de donner une vue générale sur la vision par ordinateur. Au début nous présentons ce domaine tout en mettant l'accent sur deux composants principaux de la vision par ordinateur à savoir le traitement d'image et l'extraction de

caractéristique. Enfin, nous détaillons la détection d'anomalie avec ses différentes techniques, ainsi que la détection de fraudes et les approches utilisées dans ce domaine.

Le troisième chapitre est réservé à la conception et l'implémentation de notre système. Dont nous commençons par la présentation de la conception générale et détaillée de notre système, ensuite nous détaillons les étapes d'implémentation. Les résultats obtenus ainsi qu'une évaluation sont illustrés à la fin du chapitre.

Une conclusion ainsi que des perspectives à ce travail sont présentées à la fin de ce mémoire.

# Chapitre 01 : Apprentissage machines.

## 1.1 Introduction

L'apprentissage automatique (en anglais machine Learning, littéralement « l'apprentissage machine ») ou apprentissage statistique est un sous domaine de l'intelligence artificielle qui se fonde sur des approches statistiques pour donner aux ordinateurs la capacité « apprendre » à partir de données, c'est-à-dire d'améliorer leurs performances à résoudre des tâches sans être explicitement programmés pour chacune. Plus largement, cela concerne la conception, l'analyse, le développement et l'implémentation de telles méthodes.

L'apprentissage automatique est une tentative de comprendre et reproduire cette faculté d'apprentissage dans des systèmes artificiels. Il s'agit, très schématiquement, de concevoir des algorithmes capables, à partir d'un nombre important d'exemples (les données correspondant à l'expérience passée), d'en assimiler la nature afin de pouvoir appliquer ce qu'ils ont ainsi appris aux cas futurs.

## 1.2 Objectif de L'Apprentissage automatique

Le machine Learning, ou apprentissage statistique, est un domaine de la modélisation statistique et de l'intelligence artificielle

L'objectif du machine Learning est de reconnaître parmi des données des structures souvent trop difficiles à détecter ou à mesurer manuellement. À partir de ces structures, on peut chercher à classifier des individus, des objets, à prédire la valeur d'une variable à un certain horizon, à expliquer l'apparition ou non d'une caractéristique.

Le machine Learning est par exemple utilisé pour :

Prédire la quantité de ventes d'un produit sur les trois prochains mois, en fonction des ventes observées sur le même produit dans le passé

Créer une classification de la clientèle en fonction de caractéristiques sociodémographiques et des achats passés

Attribuer un score à une personne indiquant si celle-ci est sur le point de se désabonner d'un produit ou non

Reconnaître automatiquement un chiffre manuscrit ou un visage

Au-delà d'une ou deux variable, il devient difficile de voir à l'œil nu des structures dans les données, c'est à ce niveau que l'apprentissage statistique, ou machine Learning, trouve tout son intérêt. Il en est de même lorsque le nombre d'individus (d'observations) augmente [2].

## 1.3 Différents types d'apprentissage

Les algorithmes d'apprentissage peuvent se catégoriser selon le type d'apprentissage qu'ils emploient :

### 1.3.1 Apprentissage supervisé

Si les classes sont prédéterminées et les exemples connus, le système apprend à classer selon un modèle de classement ; on parle alors d'apprentissage supervisé (ou d'analyse discriminante).

Un expert (ou oracle) doit préalablement correctement étiqueter des exemples. L'apprenant peut alors trouver ou approximer la fonction qui permet d'affecter la bonne « étiquette » à ces exemples. Parfois il est préférable d'associer une donnée non pas à une classe unique, mais une probabilité d'appartenance à chacune des classes prédéterminées (on parle alors d'apprentissage supervisé probabiliste). L'analyse discriminante linéaire ou les SVM sont des exemples typiques. Autre exemple : en fonction de points communs détectés avec les symptômes d'autres patients connus (les « exemples »), le système peut catégoriser de nouveaux patients au vu de leurs analyses médicales en risque estimé (probabilité) de développer telle ou telle maladie. [3]

### 1.3.2 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 (ou clustering). Aucun expert n'est disponible ni requis. L'algorithme doit découvrir par lui-même la structure plus ou moins cachée des données. Le système doit ici dans l'espace de description (la somme des données) cibler les données selon leurs attributs disponibles, pour les classer en groupe homogènes d'exemples.

La similarité est généralement calculée selon la fonction de distance entre paires d'exemples. C'est ensuite à l'opérateur d'associer ou déduire du sens pour chaque groupe. Divers outils mathématiques et logiciels peuvent l'aider. On parle aussi d'analyse des données en régression. Si l'approche est probabiliste (c'est à dire que chaque exemple au lieu d'être classé dans une seule classe est associé aux probabilités d'appartenir à chacune des classes), on parle alors de « soft clustering » (par opposition au « hard clustering ») [22]. Exemple : Un épidémiologiste pourrait par exemple dans un ensemble assez large de victimes de cancers du foie tenter de faire émerger des hypothèses explicatives, l'ordinateur pourrait différencier différents groupes, qu'on pourrait ensuite associer par exemple à leur provenance géographique, génétique, à l'alcoolisme ou à l'exposition à un métal lourd ou à une toxine telle que l'aflatoxine. [3]

### 1.3.3 Apprentissage semi-supervisé

Effectué de manière probabiliste ou non, il vise à faire apparaitre la distribution sous-jacente des « exemples » dans leur espace de description. Il est mis en œuvre quand des données (ou « étiquettes ») manquent Le modèle doit utiliser des exemples non-étiquetés pouvant néanmoins renseigner [3].

Exemple : En médecine, il peut constituer une aide au diagnostic ou au choix des moyens les moins onéreux de tests de diagnostics [3].

### 1.3.4 Apprentissage par renforcement

L'algorithme apprend un comportement étant donné une observation. L'action de l'algorithme sur l'environnement produit une valeur de retour qui guide l'algorithme [3].

## 1.4 Généralisation

### 1.4.1 Sur-apprentissage

En statistique, le sur-apprentissage, ou sur-ajustement, ou encore surinterprétation (en anglais « *overfitting* »)

L'overfitting est le risque pour un modèle d'apprendre "par cœur" les données d'entraînement. De cette manière, il risque de ne pas savoir généraliser à des données inconnues.
Par exemple, un modèle qui retourne l'étiquette pour les données d'entraînement et une variable aléatoire pour les données non connues aurait de très bonnes performances pendant l'entraînement, mais pas pour de nouvelles données [4].

### 1.4.2 Régularisation

Cette technique clé de machine Learning vise à limiter le « sur-apprentissage » (*overfitting*) et à contrôler l'erreur de type variance pour aboutir à de meilleures performances [5].

Lors de l'apprentissage d'un modèle, la régularisation permet d'imposer une contrainte pour favoriser les modèles simples au détriment des modèles complexes. Autrement dit, cela permet de réduire l'erreur de type variance et d'améliorer la généralisation de la solution. Il existe de nombreuses formes de régularisation, qui dépendent de l'objectif recherché et des hypothèses fixées sur le problème.

Ainsi, une régularisation de type euclidienne dans une régression des moindres carrés favorisera des coefficients faibles tandis qu'une régularisation de type lasso (utilisée lorsque le nombre de variables d'entrée est élevée) favorisera la « sparsité » de la représentation en poussant l'algorithme à ne prendre en compte qu'une petite partie des données, ignorant les autres. Pour les réseaux de neurones, les méthodes de régularisation les plus populaires sont le *Dropout* (les poids – paramètres du réseau de neurones – sont remplacés par zéro de manière

aléatoire pendant l 'entrainement), l'*Early Stopping* (l'apprentissage s'arrête plus tôt pour favoriser les modèles simples) ou la régularisation euclidienne évoquée plus haut [5].

De plus, la régularisation permet généralement de garantir certaines propriétés théoriques des algorithmes, assurant ainsi leurs bonnes performances, comme la stabilité ou de meilleures bornes de généralisation dans le cas des méthodes à noyaux [5].

### 1.4.3 Malédiction de la dimensionnalité

On peut observer empiriquement – et dans certains cas justifier mathématiquement –que plus la dimension $d$ de l'entrée $x$ est élevée, plus les tâches d'apprentissage machine ont tendance à être difficiles à résoudre. C'est ce qu'on appelle *la malédiction de la dimensionalité* [1]. Il existe plusieurs manifestations de cette malédiction. La plus importante dans le contexte de cette thèse est le fait que le nombre de combinaisons possibles des entrées augmente exponentiellement avec la dimensionnalité $d$ : en notant $x_{ij}$ la valeur associée à la j-ème dimension de l'entrée $x_i$, si l'on suppose que ces entrées ne peuvent prendre qu'un nombre fini $k$ de valeurs, alors le nombre de combinaisons possibles est égal à $k^d$.

Un algorithme qui apprend "bêtement" à associer une valeur à chaque combinaison sans partager d'information entre les différentes combinaisons n'a aucune chance de fonctionner en haute dimension, car il ne pourra pas généraliser aux multiples combinaisons qui n'ont pas été vues dans l'ensemble d'entraînement. Dans le cas où $x_{ij}$ n'est pas contraint dans un ensemble fini de valeurs, l'intuition reste la même pour certains algorithmes qui consistent à "partitionner" $\mathbb{R}^d$ en régions

Indépendantes (possiblement de manière implicite) : si le nombre de ces régions augmente exponentiellement avec $d$, alors un tel algorithme aura de la difficulté à généraliser pour de grandes valeurs de $d$. La figure 1.1 illustre ce phénomène en une et deux dimensions, et il faut garder à l'esprit que la situation peut s'avérer encore bien pire lorsque l'on manipule des entrées à plusieurs centaines de dimensions [6].

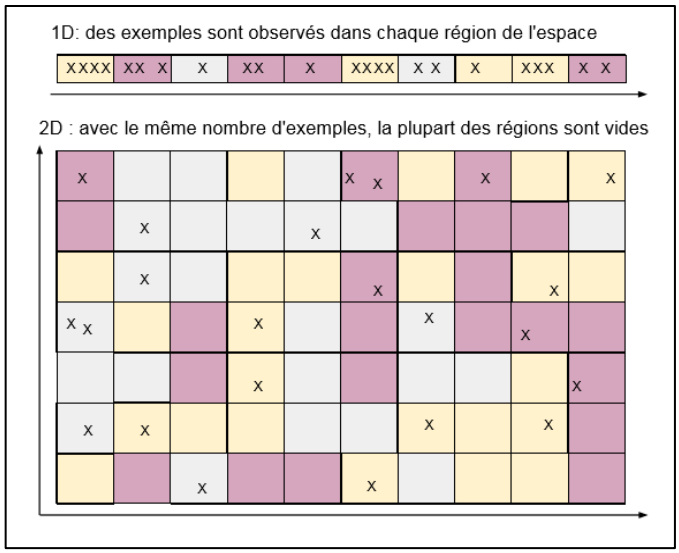



**Figure 1. 1** Malédiction de la dimensionalité : si l'algorithme partitionne l'espace en régions indépendantes, le nombre d'exemples nécessaires pour remplir ces régions augmente de manière exponentielle avec la dimension. Ici, la couleur d'une région représente la classe majoritaire dans cette région, et un tel algorithme pourrait bien généraliser à partir de 23 exemples d'entraînement pour le problème du haut (1D), mais pas pour celui du bas (2D) *[6]*.

## 1.5 Différents types de modèles

### 1.5.1 Modèles paramétriques

En apprentissage machine, un modèle paramétrique est défini par un ensemble Θ de paramètres de dimension fini, et l'algorithme d'apprentissage associé consiste à trouver la meilleure valeur possible de Θ. La figure 1.2 montre un exemple de régression linéaire en une dimension, sans régularisation [6].

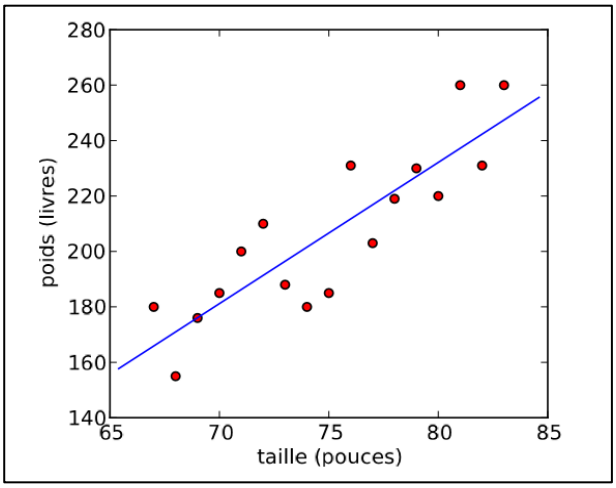



**Figure 1. 2** Modèle paramétrique : la régression linéaire. Les données (points rouges) sont un sous-ensemble des mêmes données que dans la figure 1.2, et la tâche est ici de prédire le poids d'un joueur de baseball en fonction de sa taille *[6]*.

Les modèles paramétriques peuvent également être statistiquement inefficaces. S'il y a moins d'exemples d'apprentissage, alors le problème est sur-paramétré et on risque le sur-apprentissage : le modèle pourrait apprendre des paramètres taillés "sur mesure" pour les données d'entraînement, mais qui mèneront à une mauvaise généralisation. La conséquence de cette observation est qu'en général, un modèle avec un grand nombre de paramètres est statistiquement inefficace [6].

### 1.5.2 Modelés non paramétriques

Un modèle non paramétrique n'a au contraire pas d'ensemble exacte de paramètres : le nombre de variables utilisées par le modèle augmente généralement avec le nombre d'exemples dans l'ensemble d'entrainement. Un exemple de modèle non paramétrique pour résoudre le même problème de régression que celui décrit en 4.1 est l'algorithme des fenêtres de Parzen, aussi appelé régression de Nadaraya-Watson [7] [8]. La figure 1.3 montre un exemple de régression par fenêtres de Parzen en une dimension [6].

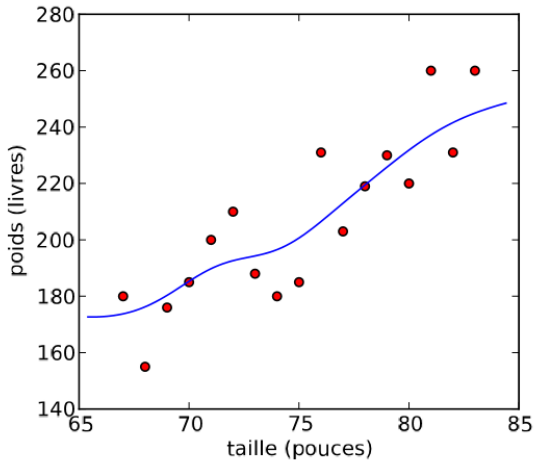



**Figure 1. 3** Modèle non paramétrique : régression par fenêtres de Parzen. Les données sont les mêmes que dans l'exemple de la régression linéaire (La figure 1.3) *[6]*.

## 1.6 Algorithmes d'apprentissage

### 1.6.1 Réseaux de neurones

Comme leur nom l'indique, les réseaux de neurones sont inspirés de l'architecture du cerveau, étant organisés en couches de neurones connectées entre elles. La première couche, appelée la couche d'entrée, est de la même dimension que les entrées $x$ et l'activation de son j-ème neurone (c'est-à-dire la valeur qu'il calcule) est égale à $x_{ij}$ lorsque l'on calcule la prédiction du réseau sur l'exemple $x_i$. La dernière couche, appelée la couche de sortie, est de la même dimension que l'étiquette dans le cas d'une tâche supervisée. Par exemple, pour la

classification, le j-ème neurone de la couche de sortie va calculer $P(Y = j|x_i)$ lorsque $x_i$ est dans la couche d'entrée. Les couches intermédiaires sont appelées les couches cachées. Il existe de nombreuses variations dans les architectures de réseaux de neurones.

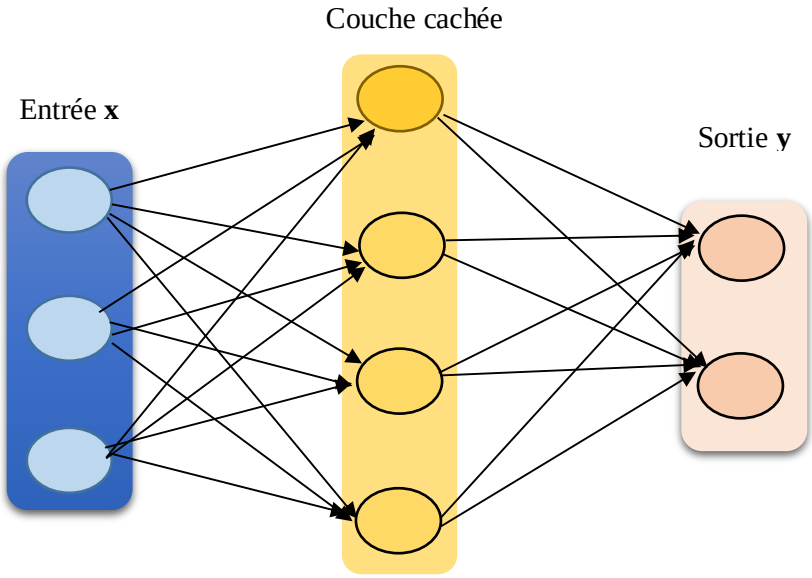



**Figure 1. 4** Réseau de neurones à une couche cachée. Une flèche du neurone i vers le neurone j indique que l'activation de j dépend directement de celle de i *[6]*.

La machine de Boltzman, dans sa forme originale, est un réseau de neurones qui possède la particularité de connecter l'ensemble des neurones entre eux. Il s'agit d'un réseau à usage principalement théorique, du fait que l'information y est distribuée de façon parallèle et pour sa proximité avec les concepts du connexionnisme. Dans sa version dite « restreinte » (qui possède quant à elle un certain nombre d'applications pratiques), on trouve un système fait uniquement de deux couches : une couche avec des neurones en entrée, et une couche avec des neurones cachés. L'ensemble de ces neurones sont interconnectés entre eux. De façon assez similaire aux réseaux *autoencoders*, la couche cachée comporte souvent moins de dimensions que la couche visible (bien que ce ne soit pas toujours le cas). Pour illustrer le concept, voici un schéma d'une machine de Boltzmann restreinte [9].

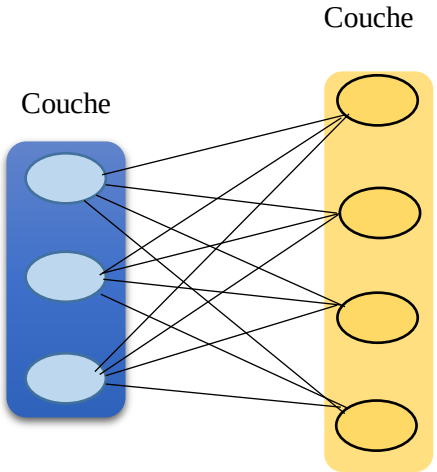



**Figure 1. 5** Machine de Boltzmann restreinte *[6]*.

### 1.6.3 Architectures profondes (Deep Learning)

Le terme d'architecture profonde désigne toute une famille de modèles inspirés des réseaux de neurones, dont le point commun est la composition de transformations successives, permettant de calculer une fonction complexe de l'entrée. Par exemple, un réseau de neurones avec une seule couche intermédiaire peut être considéré comme une architecture profonde si l'on rajoute des couches cachées : chaque couche additionnelle augmente la profondeur du réseau, et déterminer la profondeur idéale en fonction des données fait partie de la problématique des algorithmes d'apprentissage pour architectures profondes. Les réseaux de ce type ont longtemps été ignorés, d'une part parce qu'ils se sont avérés beaucoup plus difficiles à optimiser que les réseaux à une seule couche cachée, d'autre part parce qu'il a été démontré que les réseaux à une seule couche sont des approximateurs universels [10] [6]. L'intérêt pour les réseaux profonds est récemment réapparu après avoir découvert qu'une initialisation non supervisée des poids du réseau peut mener à de bien meilleures performances que l'initialisation aléatoire utilisée jusqu'à présent.

### 1.6.4 K-plus proches voisins

L'algorithme KNN figure parmi les plus simples algorithmes d'apprentissage artificiel. Dans un contexte de classification d'une nouvelle observation x, l'idée fondatrice simple est de faire voter les plus proches voisins de cette observation. La classe de x est déterminée en fonction de la classe majoritaire parmi les k plus proches voisins de l'observation x. Donc la méthode du plus proche voisin est une méthode non paramétrique où une nouvelle observation est classée dans la classe d'appartenance de l'observation de l'échantillon d'apprentissage qui lui est la plus proche, au regard des covariables utilisées. La détermination de leur similarité est basée sur des mesures de distance [11].

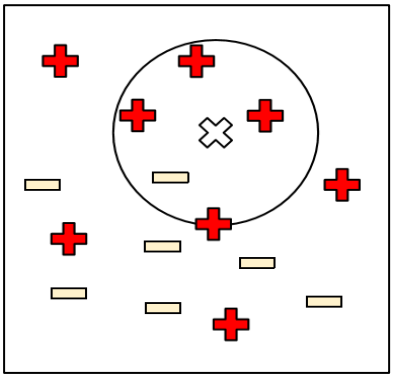


**Figure 1. 6** K-plus proches voisins (k = 5, tâche de classification) *[6]*

### 1.6.5 Fenêtres de Parzen

L'algorithme des fenêtres de Parzen a déjà été présenté dans le contexte de la régression non paramétrique, où on l'appelle parfois la régression à noyau ou la régression de Nadaraya et Watson [6][9]. On peut également utiliser une approche similaire en apprentissage non supervisé pour l'estimation de densité [12] [13]

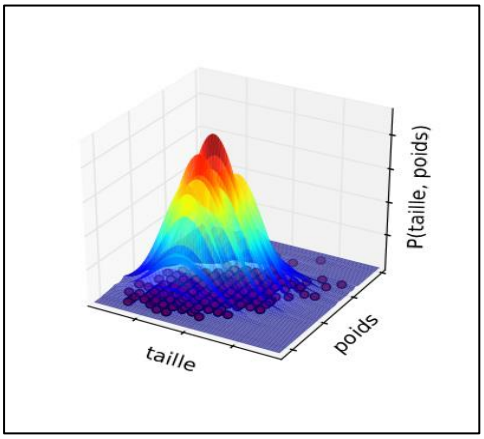



**Figure 1. 7** Fenêtres de Parzen pour l'estimation de densité *[6] ¨6]*.

### 1.6.8 Mélanges de Gaussiennes

Un modèle de mélange gaussien (désigné couramment par l'acronyme anglais GMM pour Gaussian Mixture Model) est un modèle statistique exprimé selon une densité mélange. Il sert d'habitude à estimer de façon paramétrique la distribution de variables aléatoires en les modélisant comme somme de plusieurs gaussiennes, appelées *noyaux*. Il s'agit alors de déterminer la variance, la moyenne et l'amplitude de chaque gaussienne. Ces paramètres sont optimisés selon le critère de maximum de vraisemblance afin d'approcher le plus possible la distribution recherchée. Cette optimisation est souvent effectuée en utilisant la procédure itérative appelée espérance-maximisation (*EM*).

Les modèles de mélange gaussien permettent de reconstruire de manière efficace les données manquantes dans un jeu de données expérimentales [14].

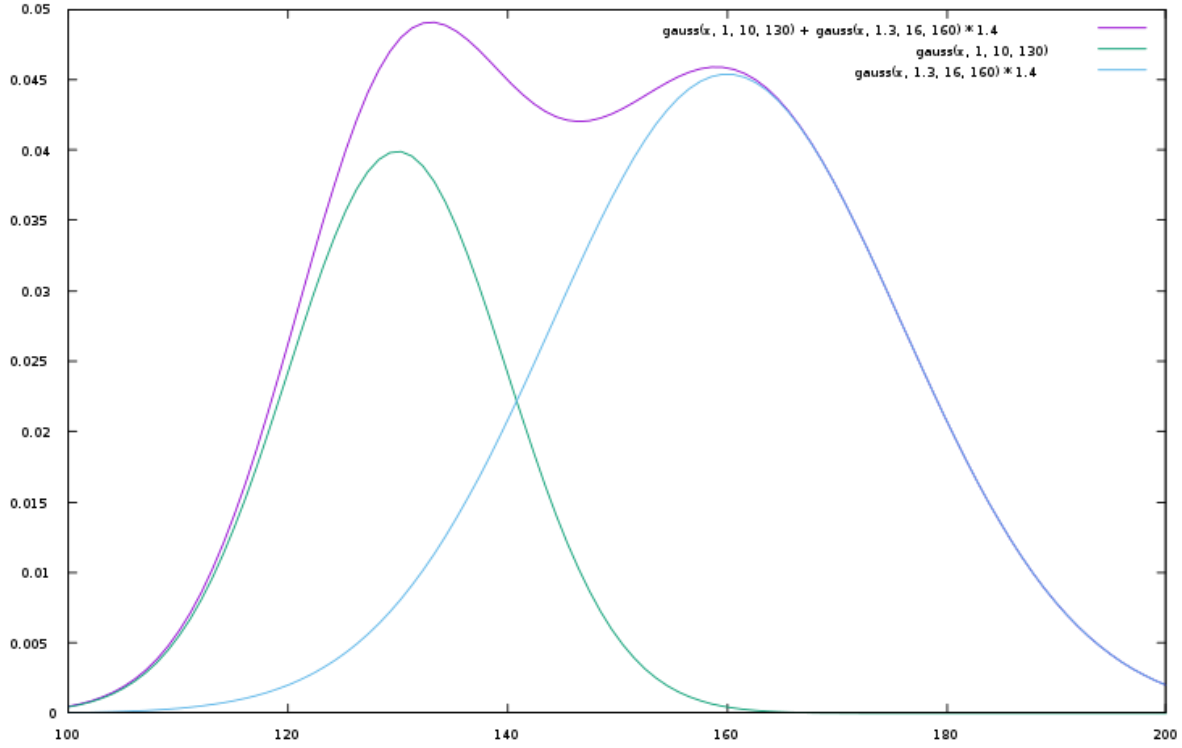



**Figure 1. 8** Deux gaussiennes forment une distribution bimodale *[14]*

### 1.6.7 Méthodes à noyau

En apprentissage automatique, l'astuce du noyau, ou kernel trick en anglais, est une méthode qui permet d'utiliser un classifieur linéaire pour résoudre un problème non linéaire. L'idée est de transformer l'espace de représentation des données d'entrées en un espace de plus grande dimension, où un classifieur linéaire peut être utilisé et obtenir de bonnes performances [15]. La discrimination linéaire dans l'espace de grande dimension (appelé aussi espace de redescription) est équivalente à une discrimination non linéaire dans l'espace d'origine.

### 1.6.8 Arbres de décision

Les arbres de décision forment une famille d'algorithmes d'apprentissage utilisés pour la classification et la régression [16]. Un arbre de décision est un arbre où chaque nœud interne représente un test sur l'entrée $x_i$. L'exemple typique d'un arbre de décision est un arbre binaire où le test effectué à chaque nœud k est de la forme $x_{ij}<\theta_k$, c'est-à-dire. Qu'on compare la $j^{éme}$ coordonnée de $x_i$ à un seuil $\theta_k$ : si elle est plus petite, on continue de parcourir l'arbre en suivant la première branche du nœud, sinon on suit la seconde branche. Lorsqu'on atteint finalement une feuille de l'arbre (un nœud sans enfants), on dit que l'exemple $x_i$ appartient à cette feuille. La figure 1.9 montre un tel arbre de décision [6].

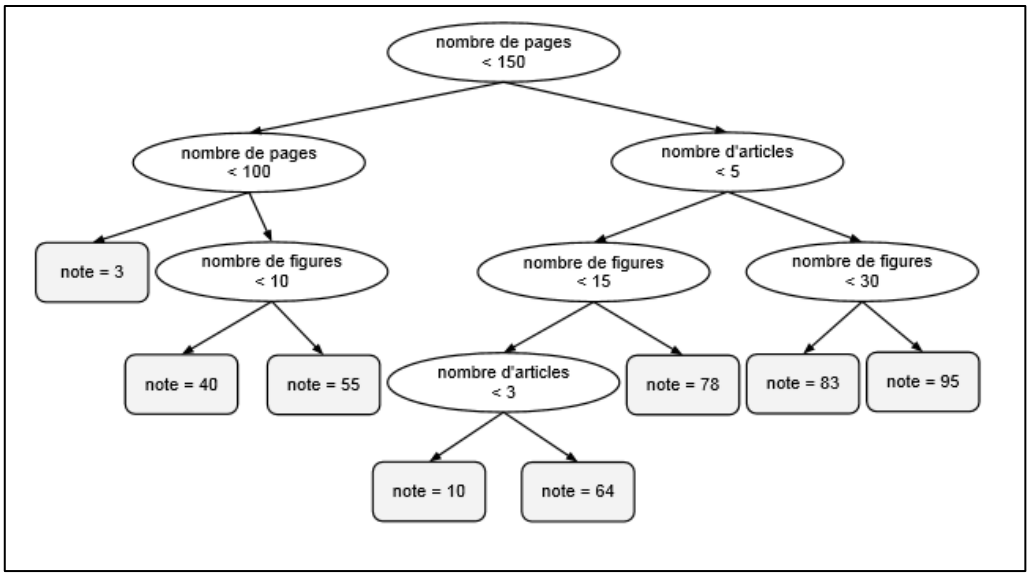



**Figure 1. 9** Arbre de décision typique *[6]*.

L'apprentissage de ce type d'arbre de décision consiste à choisir les variables testées à chaque nœud, les seuils de comparaison, la profondeur de l'arbre, ainsi que la fonction de décision associée à chaque feuille. Il existe plusieurs algorithmes pour cela, mais leur principe de base est de faire en sorte que le résultat du test effectué à chaque nœud donne de l'information supplémentaire sur $P(Y|x)$. Idéalement, la distribution $P(Y|x)$pour un nouvel exemple *x* doit être bien approximée par la distribution empirique des étiquettes des exemples d'entraînement appartenant à la même feuille que *x* [6].

## 1.6.9 Méthodes Bayésiennes

Les méthodes bayésiennes sont des méthodes par lesquelles on évalue des causes hypothétiques à partir d'un ensemble de données mesurées sous la forme d'une distribution de probabilités dite loi a priori. Dans les modèles d'inférence bayésienne, cette distribution est réévaluée à chaque nouvelle observation. Ces techniques reposent sur la formule de Bayes

$$PX/Y\left(\frac{x}{y}\right) = \frac{PX(x)PY/X(\frac{y}{x})}{PY(y)}$$

Qui donne la loi de probabilité ℘ x|y de la variable aléatoire inconnue X connaissant Yen fonction de la loi de probabilité des grandeurs mesurées ℘y|x, de la loi ℘y de y et de la loi ℘x que l'on s'est fixé a priori pour x. La statistique bayésienne est utilisée lorsque les échantillons de données sont peu nombreux, contrairement à la statistique classique ou fréquentiste qui travaille sur de grands échantillons [17].

## 1.7 Conclusion

Dans ce chapitre nous avons défini la notion de l'apprentissage artificiel et ses caractéristiques et nous avons introduit ses composants, Nous présenterons dans le chapitre suivant le domaine de vision par ordinateur

# Chapitre2 : vision par ordinateur

# 2.1 Introduction

Dans ce chapitre, nous présentons une vue générale sur la vision par ordinateur et leur domaine d'application. Ensuite, nous définissons les notions nécessaires aux traitements d'image, où l'on utilise une brève description pour chaque notion. En particulier, nous considérons la définition de ce qu'est une image et ses caractéristiques, les types existants, etc.

Enfin, nous allons parler sur la détection d'anomalie, sa définition, son principe, ses différents types, ainsi que ses caractéristiques, et nous allons discuter sur la technique de détection.

# 2.2 Vision par ordinateur

## 2.2.1 Définition

La vision par ordinateur est un domaine scientifique interdisciplinaire qui traite de la façon dont les ordinateurs peuvent acquérir une compréhension de haut niveau à partir d'images ou de vidéos numériques. Du point de vue de l'ingénierie, il cherche à comprendre et à automatiser les tâches que le système visuel humain peut effectuer. Les tâches de vision par ordinateur comprennent des procédés pour acquérir, traiter, analyser et « comprendre » des images numériques, et extraire des données afin de produire des informations numériques ou symboliques, par ex. sous forme de décisions Ainsi, il a pu surpasser les humains dans certaines tâches liées à la détection et à l'étiquetage des objets [18].

## 2.2.2 Historique

Le développement de la vision par ordinateur a commencé dans les universités pionnières de l'intelligence artificielle à la fin des années 1960. L'objectif était d'imiter le système visuel humain, première étape pour doter les robots d'un comportement intelligent. En 1966, on croyait que cela pouvait être réalisé grâce à un projet d'été, en attachant une caméra à un ordinateur et en lui faisant "décrire ce qu'il voyait.

Ce qui distinguait la vision par ordinateur du domaine prédominant du traitement d'images numériques à cette époque était le désir d'extraire une structure tridimensionnelle d'images dans le but de parvenir à une compréhension complète de la scène. Des études dans les années 1970 ont formé les premières bases de nombreux algorithmes de vision par ordinateur qui existent aujourd'hui, y compris l'extraction des bords d'images, l'étiquetage des lignes, la modélisation non polyédrique et polyédrique, la représentation d'objets sous forme d'interconnexions de structures plus petites, le flux optique et estimation de mouvement.

La décennie suivante a vu des études basées sur une analyse mathématique plus rigoureuse et des aspects quantitatifs de la vision par ordinateur. Ceux-ci incluent le concept

d'espace d'échelle, l'inférence de la forme à partir de divers indices tels que l'ombrage, la texture et la mise au point, et les modèles de contour connus sous le nom de serpents. Les chercheurs ont également réalisé que bon nombre de ces concepts mathématiques pouvaient être traités dans le même cadre d'optimisation que la régularisation et les champs aléatoires de Markov.

Dans les années 1990, certains des thèmes de recherche précédents sont devenus plus actifs que les autres. La recherche sur les reconstructions projectives 3D a permis de mieux comprendre l'étalonnage de caméras. Avec l'avènement des méthodes d'optimisation pour la calibration des caméras, on s'est rendu compte que de nombreuses idées avaient déjà été explorées dans la théorie de l'ajustement des faisceaux dans le domaine de la photogrammétrie. Cela a conduit à des méthodes pour des reconstructions 3D éparses de scènes à partir de plusieurs images. Des progrès ont été réalisés sur le problème de la correspondance stéréo dense et d'autres techniques stéréo à vues multiples. Dans le même temps, des variations de coupe graphique ont été utilisées pour résoudre la segmentation d'image [18].

Cette décennie a également marqué la première fois que des techniques d'apprentissage statistique ont été utilisées dans la pratique pour reconnaître les visages dans les images (voir Eigenface). Vers la fin des années 90, un changement important s'est produit avec l'interaction accrue entre les domaines de l'infographie et de la vision par ordinateur. Cela comprenait le rendu basé sur l'image, l'interpolation de vue, l'assemblage d'images panoramiques et le premier rendu de champ lumineux [18].

Des travaux récents ont vu la résurgence des méthodes basées sur les fonctionnalités, utilisées en conjonction avec des techniques d'apprentissage automatique et des cadres d'optimisation complexes, Les progrès des techniques d'apprentissage en profondeur ont donné une nouvelle vie au domaine de la vision par ordinateur. La précision des algorithmes d'apprentissage en profondeur sur plusieurs ensembles de données de vision par ordinateur de référence pour des tâches allant de la classification, de la segmentation et du flux optique a surpassé les méthodes antérieures [18].

## 2.3 Vision humaine et vision artificielle

Une des particularités des êtres vivants est de pouvoir acquérir des images via l'œil, comme une information, puis de pouvoir l'interpréter via le cerveau. L'enjeu de la vision artificielle est de permettre à un ordinateur de "voir" c'est-`a-dire, comme l'homme, de récupérer l'information par l'intermédiaires d'un dispositif d'acquisition d'image puis d'exploiter.

### 2.3.1 Vision humaine

Comprendre le contenu d'une image est automatique pour l'humain mais difficile pour la machine. La raison en est la complexité du processus de vision humaine qui effectue un traitement parallèle des données dont une partie seulement est purement visuelle œil (= 1 million de cellules sensibles), cerveau (= des milliards de neurones). Nommer un objet est un processus d'inférence (de d´éduction) qui n´nécessite des connaissances non contenues dans

l'image. Le cerveau humain n'est pas une machine précise, Il nous est difficile de mesurer exactement la taille des objets. Cependant, nos approximations nous permettent d'aller plus loin dans la reconnaissance. Les avantages de l'être humain c'est de faculté d'abstraction les modèles des objets que nous utilisons sont très abstraits, Faculté de d'éduction face à un nouvel objet, nous pouvons le deviner grâce à sa sémantique, pas seulement l'apparence, Faculté d'apprentissage par exemple un bébé connait peu le monde `a sa naissance, mais il apprend peu à peu à le reconnaitre, Immersion dans le milieu : nous ne faisons pas que voir les objets, nous les vivons (vision active) et informations a priori : nous prenons pour acquis plusieurs informations qui simplifie l'interprétation (horizon, structure logique d'un objet, bon sens, ...)

### 2.3.2 Vision artificielle

Une image peut montrer la vue partielle ou totale d'une scène, contenir beaucoup de détails et montrer des objets de différentes ´échelles (environnement, galaxies, bactéries, ...). La vision artificielle permet de faire certaines choses que la vision humaine ne permet pas et elle permet de voir l'invisible.

## 2.4 Traitement d'image

### 2.4.1 Définition de l'image

Une image est une représentation numérique en mémoire d'un sujet imprimé sur une rétine artificielle (matricielle comme le capteur d'un appareil photographique numérique ou la scène virtuelle d'une image de synthèse ou bien linéaire comme le capteur du télécopieur, du photocopieur ou du scanner) [19].

### 2.4.2 L'image numérique

L'image numérique est l'image dont la surface est divisée en éléments de taille fixes appelée cellules ou pixels, ayant chacun comme caractéristique un niveau de gris ou une couleur [19].

### 2.4.3 Caractéristiques de l'image numérique

L'image est un ensemble structuré d'informations caractérisé par les paramètres suivants**:**

#### 2.4.3.1 Pixels :

Le pixel représente le plus petit élément constitutif d'une image numérique. L'ensemble de ces pixels est contenu dans un tableau à deux dimensions constituant l'image [19]:

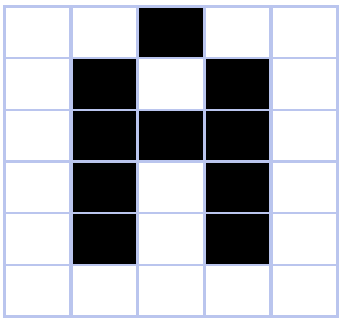


**Figure 2. 1** . Les pixels des images *[19]*

La quantité d'information que véhicule chaque pixel donne des nuances entre images monochromes et images couleurs. Dans le cas d'une image monochrome, chaque pixel est codé sur un octet [19].

### 2.4.3.2 Dimension

Le nombre total de pixels affichables horizontalement et verticalement sur un moniteur, définit la taille de l'image ou bien sa dimension. C'est le produit de la hauteur et de la largeur de l'image définit en pixels [19].


**Figure 2. 2** L'effet du changement de la dimension sur l'image *[19]*.

### 2.4.3.3 La résolution

C'est la clarté ou la finesse de détails atteinte par un moniteur ou une imprimante dans la production d'images. Sur les moniteurs d'ordinateurs, la résolution est exprimée en nombre de pixels par unité de mesure (pouce ou centimètre). On utilise aussi le mot résolution pour désigner le nombre total de pixels affichables horizontalement ou verticalement sur un moniteur ; plus grand est ce nombre, meilleure est la résolution [20].

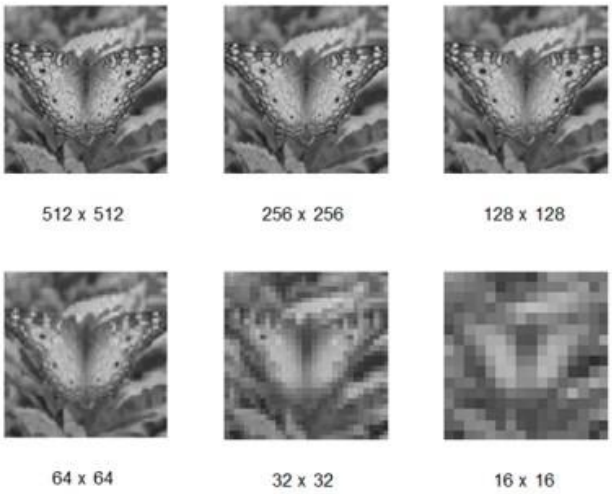



**Figure 2. 3** Résolution d'une image *[47]*.

### 2.4.3.4. Contraste

C'est l'opposition marquée entre deux régions d'une image, plus précisément entre les régions sombres et les régions claires de cette image. Le contraste est défini en fonction des luminances de deux zones d'images.

Si L1 et L2 sont les degrés de luminosité respectivement de deux zones voisines A1 et A2 d'une image, le contraste C est défini par le rapport [19] :

$$C = \frac{L1 - L2}{L1 + L2}$$

### 2.4.3.5 Luminance

C'est le degré de luminosité des points de l'image. Elle est définie aussi comme étant le quotient de l'intensité lumineuse d'une surface par l'aire apparente de cette surface, pour un observateur lointain, le mot luminance est substitué au mot brillance, qui correspond à l'éclat d'un objet. Une bonne luminance se caractérise par :

- Des images lumineuses (brillantes) ;
- Un bon contraste : il faut éviter les images où la gamme de contraste tend vers le blanc ou le noir ; ces images entraînent des pertes de détails dans les zones sombres ou lumineuses.

- L’absence de parasites.

#### 2.4.3.6 Bruit

Un bruit (parasite) dans une image est considéré comme un phénomène de brusque variation de l’intensité d’un pixel par rapport à ses voisins, il provient de l’éclairage des dispositifs optiques et électroniques du capteur [19].

#### 2.4.3.7 Histogramme

L’histogramme des niveaux de gris ou des couleurs d’une image est une fonction qui donne la fréquence d’apparition de chaque niveau de gris (couleur) dans l’image. Il permet de donner un grand nombre d’information sur la distribution des niveaux de gris (couleur) et de voir entre quelles bornes est repartie la majorité des niveaux de gris (couleur) dans le cas d’une image trop claire ou d’une image trop foncée [20].

Il peut être utilisé pour améliorer la qualité d’une image (Rehaussement d’image) en introduisant quelques modifications, pour pouvoir extraire les informations utiles de celle-ci.

Pour diminuer l’erreur de quantification, pour comparer deux images obtenues sous des éclairages différents, ou encore pour mesurer certaines propriétés sur une image, on modifie souvent l’histogramme correspondant [20].

#### 2.4.3.8 Contour

Le contour est la frontière qui s’épare des objets dans une image qui ont des pixels dont les niveaux de gris différents, ou la limite des objets qui marquant des changement d’intensité [21].

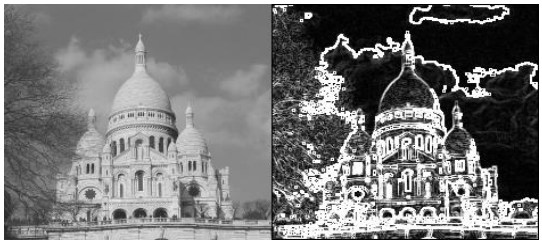


**Figure 2. 4** le contour d’une image.

### 2.4.4 Types d’images numériques

Les images sont de deux types: les images bitmap et les images "vectorielles" :

#### 2.4.4.1 Images bitmap (pixellisées, matricielles)

Ce sont des images représentées par une matrice de pixels d'où leur appellation, une

matrice est en quelques sortes un tableau. Les pixels constituants une forme rectangulaire, et chacun d'entre eux possède une couleur quelconque, et en rassemblant ces pixels on obtient notre image matricielle. Ces derniers ne sont pas visibles à l'œil nu, sauf si l'image est agrandie [19].

#### 2.4.4.2 Images vectorielles

Les images vectorielles sont composées d'entités géométriques telles qu'un cercle, une ligne, un rectangle, une courbe ou un segment etc. (donc pas de pixels). Ces entités géométriques sont représentées par des formules mathématiques (un rectangle est défini par deux points, un cercle par un centre et un rayon, une courbe par plusieurs points et une équation) [19].

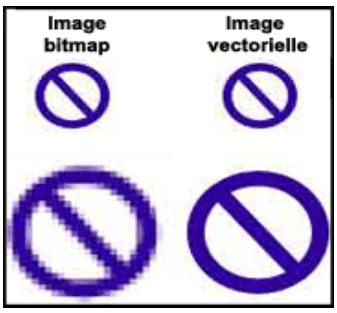



**Figure 2. 5** Image Bitmap et image vectorielle *[19]*.

### 2.4.5 Codages des couleurs

Nous avons vu qu'une image apparait comme une matrice ou` chaque case contient des nombres associés à une couleur. Usuellement on distingue 3 grands types de couleurs pour une image numérique :

- Le noir et blanc.
- Les niveaux de gris
- La couleur.

Ces types sont généralement à choisir lors d'une numérisation par scanner ou lors de la configuration d'un appareil photographique.

#### 2.4.5.1 Image noir et blanc (binaire)

Le noir et blanc est le plus simple. Le contenu de chaque case de la matrice est soit un 0 (noir) soit 1 (blanc). Le nombre de couleurs n'est que de 2 et le rendu de l'image le moins performant mais parfois suffisant dans le cadre par exemple de documents scripturaux (Figure 2.6) [22].

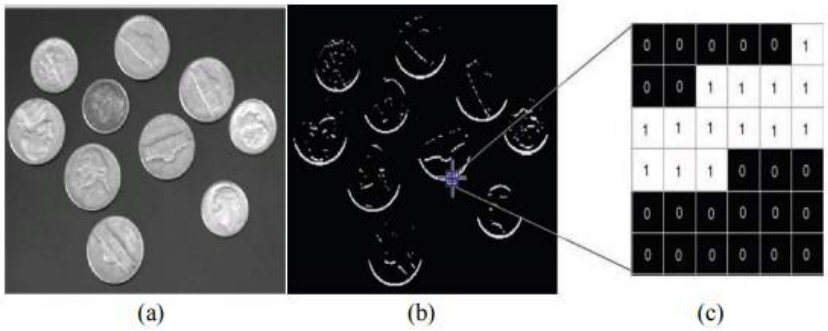



**Figure 2. 6** Image binaire.

a) Image originale, b) Une image binaire, c) Le tableau de valeurs correspondant [23].

#### 2.4.5.2 Niveaux de gris

Le codage dit en niveaux de gris permet d'obtenir plus de nuances que le simple noir et blanc. Il offre des possibilités supplémentaires pour coder le niveau de l'intensité lumineuse. La couleur est codée souvent sur un octet soit 8 bits ce qui offre la possibilité d'obtenir 256 niveau de gris (0 pour le noir et 255 pour le blanc) [22].

#### 2.4.5.3 Image couleur

**Principe** : La couleur d'un pixel est obtenue, comme le ferait un peintre, par le mélange de couleurs fondamentales. Il ne s'agit pas ici de décrire toutes les techniques utilisées. Nous allons décrire un des principes les plus couramment utilisé qui est celui de la synthèse additive [22].

**Codage RVB** : Le principe consiste à mélanger les 3 couleurs : rouge, vert et bleu (noté RVB ou RGB en anglais). A l'aide de ces 3 couleurs, on obtient toute une palette de nuances allant du noir au blanc. A chaque couleur est associé un octet (donc 256 niveaux de luminosité) de chacune des couleurs fondamentales.

Un pixel "couleur" est alors codé avec 3 octets et on a alors la possibilité d'obtenir 224 possibilités de couleurs soit de l'ordre de 16 millions de couleurs différentes [22].

### 2.4.6 Traitement d'images numérique

Le traitement d'image est une discipline issue de deux domaines qui se croisent et qui sont l'informatique et les mathématiques appliquées, cette dernière étudie les caractéristiques et les propriétés des images numériques et leurs transformations dans le but soit d'améliorer leur qualité ou d'en extraire de l'information. Dans ce qui va suivre, nous allons présenter trois opérations de traitement d'image visant à améliorer la qualité ou extraire de l'information de cette dernière [24].

### 2.4.6.1 Acquisition

L'acquisition d'images constitue un des maillons essentiels de toute chaine de conception et de production d'images. Pour pouvoir manipuler une image sur un système informatique, il est avant tout nécessaire de lui faire subir une transformation qui la rendra lisible et manipulable par ce système. Cette opération se fait par des systèmes de saisie qui peuvent être classés en deux catégories principale : Les caméras numériques et les scanners [25].

### 2.4.6.2 Segmentation

La segmentation consiste en la répartition de l'ensemble des pixels de l'image en groupe, chacun d'eux formera une région de l'image selon un critère d'homogénéité (chaque région est caractérisée par un critère d'homogénéité). Cette technique de traitement d'image vise à séparer le plus précisément possible les différents objets se trouvant dans l'image traitée et ainsi extraire l'information présente dans cette dernière. Il existe plusieurs m´méthodes et approches pour la segmentation d'images [24] .

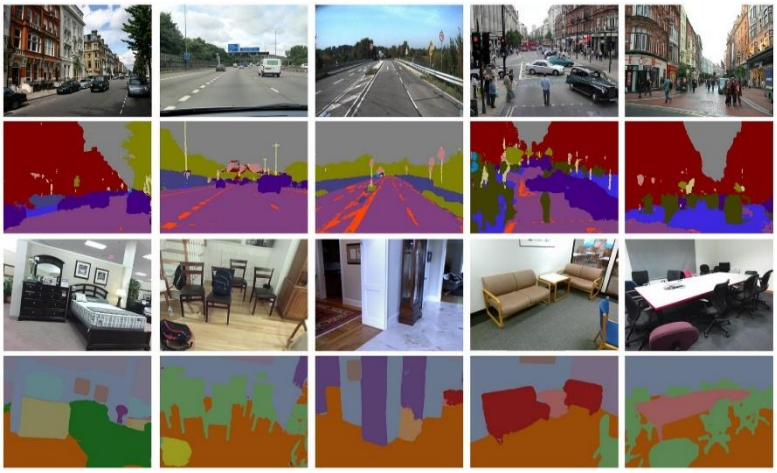


**Figure 2. 7** Illustration de la segmentation d'images (le haute: image d'origine, ci-dessous : L'image segmentée) [24].

Il y a deux types de segmentation :

#### 2.4.6.2.1 Segmentation de texte en lignes

En général, le sous-titres ou le texte graphique ce compose de plusieurs lignes. En reconnaissance de texte, le texte se segmentant en lignes séparées. Il existe certaines méthodes utilisées à cet effet, telles que la projection horizontale.

#### 2.4.6.2.2 Segmentation de lignes en caractères

Il s'agit ici de la segmentation de lignes en caractères individuels. Les points de segmentation sont identifiés à la fin d'un caractère et au début de la suivante.

#### 2.4.6.3 Filtrage

Le principe du filtrage est de modifier la valeur des pixels d'une image, généralement dans le but d'améliorer son aspect. En pratique, il s'agit de créer une nouvelle image en se servant des valeurs des pixels de l'image d'origine. Il existe plusieurs filtres effectuant différents traitements selon le besoin de l'analyse, on cite l'exemple de filtrage d'une image bruité par un filtre opérant de telle sorte a` supprimer les pixels bruits présents dans l'image [24].

**Figure 2. 8** Exemple de filtrage d'une image bruitée (à gauche l'image d'origine, à droite l'image filtrée).

## 2.5 Extraction des caractéristiques

Dans l'apprentissage automatique, la reconnaissance des formes et le traitement des images, l'extraction des caractéristiques commence à partir d'un ensemble initial de données mesurées et construit des valeurs dérivées (caractéristiques) qui visent à être informatives et non redondantes, facilitant ainsi les étapes subséquentes d'apprentissage et de généralisation. et dans certains cas conduisent à de meilleures interprétations humaines. L'extraction des caractéristiques est associée à la réduction dimensionnelle.

Lorsque les données d'entrée de l'algorithme sont trop importantes pour être traitées et qu'on soupçonne qu'elles sont redondantes (p. ex., la même mesure en pieds et en mètres, ou la répétition d'images présentées en pixels), elles peuvent être converties en un ensemble réduit de caractéristiques (également appelé vecteur de caractéristiques). La sélection d'un sous-ensemble de fonctionnalités initiales est appelée sélection de fonctionnalités. [2] On s'attend à ce que les fonctions sélectionnées contiennent des renseignements pertinents provenant des données d'entrée, de sorte que la tâche souhaitée puisse être effectuée en utilisant cette représentation réduite plutôt que des données brutes complètes.

## 2.6 Détection d’anomalie

### 2.6.1 introduction

La détection d’anomalies est une stratégie utilisée pour reconnaître des exemples étranges qui ne correspondent pas à la conduite prévue, appelés anomalies.

Généralement, les anomalies indiquent un problème tel qu'une fraude bancaire, un défaut structurel, un problème médical ou une erreur dans un texte. Les anomalies sont également appelées des valeurs aberrantes, du bruit, des écarts ou des exceptions [26].

### 2.6.2 Définition

La détection d’anomalie consiste à identifier les points de données dans les données qui ne correspondent pas aux schémas normaux.

Le but de la détection d’anomalie est repérer des données qui ne sont pas conforme à ce à quoi l’on peut s’attendre par rapport au autre données [27].

### 2.6.3 Techniques du détection d’anomalie

Il existe trois grandes catégories de techniques de détection d'anomalies [28] :

- Les techniques de détection d'anomalies non supervisées :

Détectent les anomalies dans un ensemble de données non étiquetées en supposant que la majorité des instances de l'ensemble de données sont normales et en recherchant les instances qui ne correspondent pas au reste des données ;

- Les techniques de détection d'anomalies supervisées :

Nécessitent un ensemble de données où les données sont étiquetées normales ou anormales et impliquent l'entrainement d'un classificateur (la principale différence par rapport à de nombreux autres problèmes de classification statistique réside dans la nature déséquilibrée de la détection des valeurs aberrantes) ;

- Les techniques de détection d'anomalies semi-supervisées :

Construisent un modèle représentant le comportement normal d'un ensemble de données normales, puis testent la probabilité qu'une instance de test soit compatible avec le modèle.

La détection d'anomalies est applicable dans divers domaines, tels que la détection d'intrusions, la détection de défauts, la surveillance de l'état du système, la détection d'événements dans des réseaux de capteurs et la détection de perturbations d'un écosystème , la détection de fraudes [28].

## 2.6.4 Détection des fraudes

La fraude est généralement découverte après une vérification minutieuse des documents. « Il arrive parfois que des documents paraissent suspects. La suspicion est souvent la raison d'un contrôle plus approfondi » [29].

### 2.6.4.1 La fraude documentaire

La fraude documentaire est un terme générique qui permet de désigner une fraude, une falsification d'un document, Modification d'un document, Remplissage d'un formulaire de manière erronée ou Changement d'une photo [30].

Il existe cinq type de fraudes documentaires [30]:

- **La contrefaçon** : La contrefaçon est une reproduction complète d'un document d'identité. Difficile à réaliser, la contrefaçon d'un document d'identité se détecte rapidement, car les contrefaçons sont majoritairement de mauvaises reproductions qui se détectent à l'œil nu. Il est difficile de contrefaire l'ensemble des détections d'un document d'identité.
- **La falsification** : Il s'agit d'un document original, sur lequel le fraudeur apporte une modification d'un ou plusieurs éléments. La falsification d'un document authentique, peut porter sur la date de validité, sur les mentions d'identité ou encore sur la photographie. Il est important d'examiner le document dans son détail, afin de constater un changement de photo ou de données personnelles.
- **L'usurpation d'identité** : L'usurpation d'identité se produit lorsqu'une personne utilise le document d'identité d'une autre personne. Dans ce cas, le document sera authentique ; en revanche la personne qui l'utilise n'est pas détentrice dudit document d'identité. C'est pour cela qu'il est très important d'examiner avec minutie la photo présente sur un document d'identité, et la comparer avec la personne qui est face à vous.
- **Les documents volés vierges** : Documents authentiques ayant été volés avant leur personnalisation, et qui auront été complétés par le voleur, le receleur ou le faussaire par la suite. Ils pourront alors être qualifiés de "documents falsifiés". Suite à une réorganisation complète du schéma de distribution des documents d'identité, ce type de fraude documentaire tend à se raréfier. En effet avec le passeport Delphine, les documents étaient issus de l'Imprimerie Nationale, puis étaient personnalisés dans les préfectures et mairies. Le vol de ces documents vierges survenait principalement lors de

leur convoyage, ou en mairie. Désormais, avec le passeport biométrique, la confection du document d'identité est finalisée à l'Imprimerie Nationale.

- **Les documents fantaisistes** : Les documents fantaisistes sont créés de toutes pièces. A la différence d'une contrefaçon, le document présenté n'est pas une copie : il s'agit d'un document créé de toutes pièces par le faussaire.

## 2.6.5 les approches utilisées pour la détection et traitée les fraudes

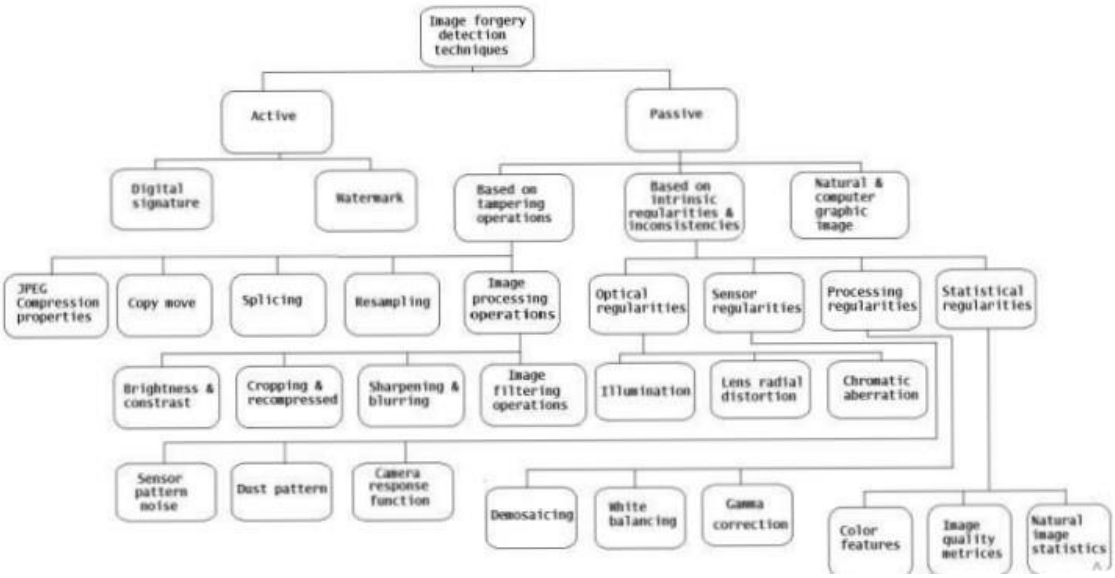



**Figure 2. 9** Classification de Birajdar & Mankar (2013) des recherches menées en Image Forensics *[1]*.

### 2.6.5.1 Approches active

Des recherches sont donc menées pour lutter contre la fraude avant même que celle-ci n'ait lieu : il s'agit de protéger le document en y insérant des éléments de sécurité qui permettront d'authentifier plus sûrement et simplement le document [1].

Ces approches se divisent en deux groupes : les approches des « données cachées » et celles des « signatures numériques » [1].

#### 2.6.5.1.1 données cachées

La première approche pour sécuriser un document est de cacher des informations en son

sein afin que le destinataire, averti, puisse les récupérer et vérifier l'authenticité du document [1].

Le filigrane est traditionnellement une technique utilisée dans la création de papier qui fait apparaitre un dessin ou un texte quand on regarde le papier d'une certaine façon, principalement par transparence. Ainsi de nombreux billets de banques comportent des filigranes, qui sont très difficilement falsifiables (Cox et al. 2008). Bas et al. (2016) appellent « filigrane numérique » (digital watermarking) une insertion d'éléments qui doivent être invisibles, robustes (insensibles aux perturbations que pourrait subir le document) et sécurisés (difficiles à extraire sans une clé ou un algorithme) dans un média [1].

#### 2.6.5.1.2 signature

Ces techniques de signature de document permettent l'authentification dans le cas d'un échange de ces documents entre des personnes averties et équipées des moyens de chiffrement et de déchiffrement de ces codes et/ou du code original, ce qui n'est pas le cas dans la plupart des échanges de documents [1].

### 2.6.6.2 Approches passive

Contrairement aux approches dites actives, les approches passives de détection de fausses images peuvent s'appliquer à n'importe quel type d'images, sécurisées ou non, de scènes naturelles ou de documents, photographies ou dessins... La détection de faux documents parmi les originaux est une tâche de classification pour laquelle il s'agit de trouver les meilleurs indices, ou caractéristiques, de l'image et de sélectionner et entrainer le meilleur classifieur pour différencier les documents authentiques et les faux documents (Birajdar & Mankar 2013) [1].

Cette approche contient deux sous-domaines de l'intelligence artificielle **:**

D'apprentissage automatique (Machine Learning) qui comprend plusieurs techniques (SVM, KNN, BinaryForest)

Apprentissage automatique profond (Deep Learning) qui comprend des techniques comme CNN, GNN**,** LSTM, **AutoEncodeur.**

## 2.7 Conclusion

Dans ce chapitre, nous avons essayé de faire un récapitulatif sur les notions élémentaires de vision par ordinateur et de traitement d'image, et nous avons présenté les notions de l'image numérique et ses différents caractéristiques (pixels, résolution, etc.) ainsi que le Codage des couleurs, en fin nous avons expliqué la stratégie de détection d'anomalie et la détection des fraudes en particulier.

Dans ce qui suit, nous allons parler sur les différentes approches pour traiter les fraudes documentaires.

.

.

# Chapitre 3 : conception et implémentation

## 3.1 Introduction

Ce chapitre est consacré à la conception et la mise en œuvre de notre système qui permettra d'identifier la détection des fraudes documentaires en utilisant une méthode appartenant à la famille de techniques de Deep Learning. Nous allons donc décrire les différentes parties de la conception de notre système, tout en expliquant les détails relatifs à chaque étape. Par la suite nous allons présenter les détails de l'implémentation. Les résultats obtenus sont illustrés à la fin du chapitre avec une discussion.

Puisque nous avons implémenté un auto-encodeur convolutif, nous avons vu qu'il est nécessaire d'expliquer son principe avant d'entamer la phase de conception

## 3.2 Autoencodeur

Un auto-encodeur est un réseau de neurones artificiels dans lequel la couche d'entrée a le même nombre de neurones que celle de la sortie. Il est utilisé pour l'apprentissage non supervisé et vise à reconstruire l'entrée avec le minimum d'erreur possible [31].

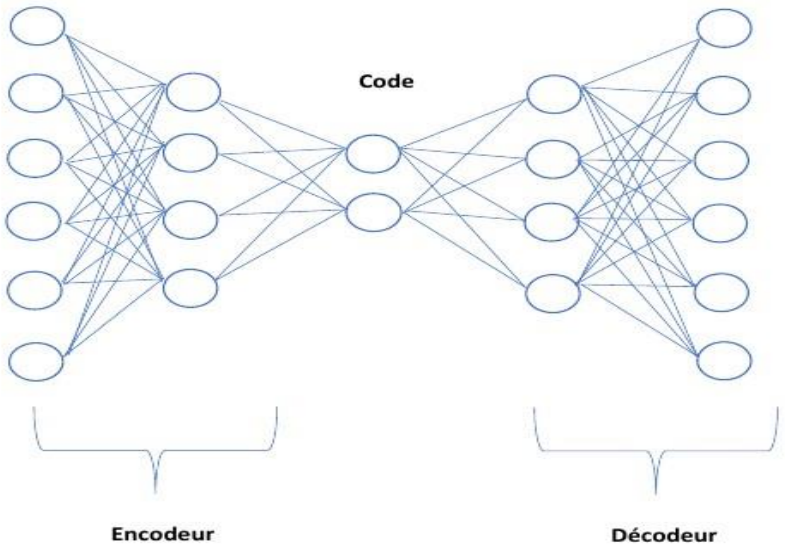



**Figure 2. 10** Architecture d'un auto-encodeur avec 2 couches d'encodeur et de décodeur et une couche cachée dite code *[31]*

L'encodeur représente la partie du réseau qui compresse l'entrée dans un espace latent représentant la couche code.

Le code est la couche cachée qui est généralement représentée sous forme compressée dans une dimension réduite. Il constitue aussi l'entrée alimentée du décodeur. Le décodeur est la partie du réseau qui tente de reconstruire l'entrée à partir de l'espace latent [31].

Il existe plusieurs modèles d'auto-encoder, nous allons choisir dans notre travail l'auto-

encodeur convolutif puisqu'il est classé parmi les meilleures modèles d'extractions des caractéristiques.

### 3.2.1 Les auto-encoder convolutif (CAE)

Un auto-encodeur convolutionnel (CAE) est un réseau neuronal (un cas particulier d'un modèle d'apprentissage non supervisé) qui est formé pour reproduire son image d'entrée dans la couche de sortie. Dans ce type d'auto encodeur, les couches d'encodeur sont appelées couches de convolution et les couches de décodeur sont également appelées couches de dé-convolution. Le côté dé-convolution est également connu sous le nom d'amplification ou de convolution de transposition.

Les CAE sont des extracteurs de caractéristiques à usage général, contrairement aux AE qui ignorent complètement la structure de l'image 2D. En effet, dans les AE, l'image doit être déroulée en un seul vecteur et le réseau doit être construit en respectant la contrainte sur le nombre d'entrées. En d'autres termes, les AE introduisent une redondance dans les paramètres, forçant chaque caractéristique à être globale (c'est-à-dire à couvrir tout le champ visuel), contrairement aux CAE [32].

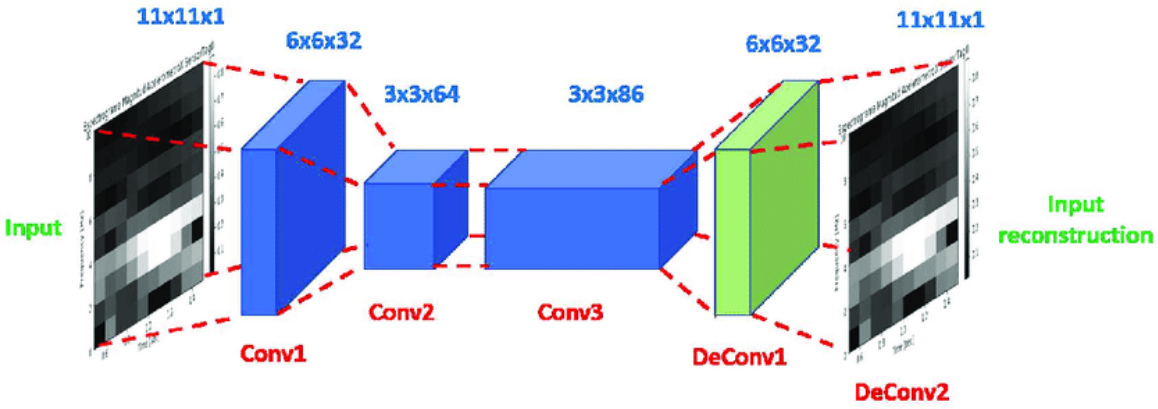



**Figure 2. 11** architecure d'un autoencoder convolutif *[33]*

#### 3.2.1.1 Les couches de réseaux de neurones convolutionnels

**1.Couche convolutionnelle (CONV)**

La couche convolutionnelle (en anglais convolution layer) (CONV) utilise des filtres qui scannent l'entrée I suivant ses dimensions en effectuant des opérations de convolution. Elle peut être réglée en ajustant la taille du filtre F et le stride S. La sortie O de cette opération est appelée feature map ou aussi activation map [34].

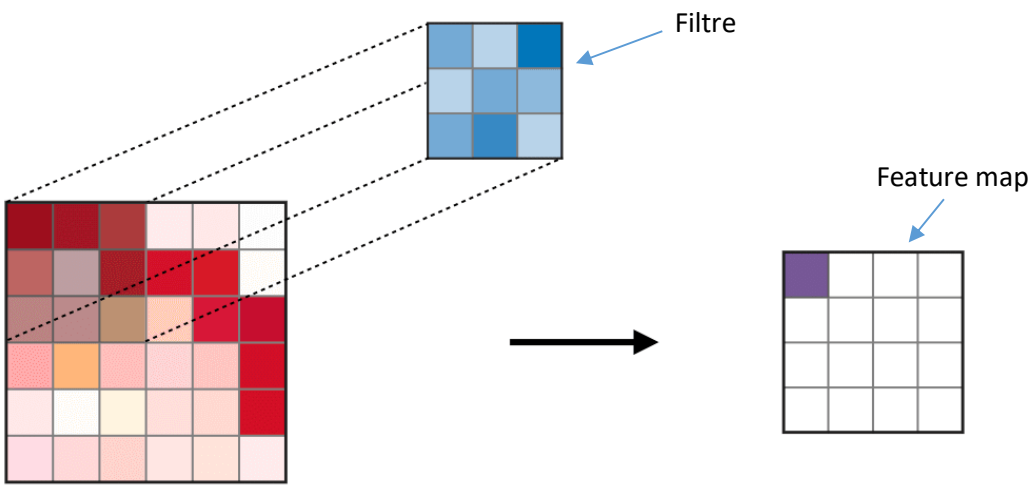



**Figure 2. 12** La couche convolutionnelle *[34]*.

## 2. Couche de mise en commun (Pooling)

La couche de pooling (en anglais pooling layer « POOL ») est une opération de sous-échantillonnage typiquement appliquée après une couche convolutionnelle [34]. Le pooling réduit la taille spatiale d'une image intermédiaire, réduisant ainsi la quantité de paramètres et de calcul dans le réseau. Il est donc fréquent d'insérer périodiquement une couche de pooling entre deux couches convolutives successives d'une architecture CNN pour contrôler l'overfitting (sur-apprentissage). L'opération de pooling créait aussi une forme d'invariance par translation.

Le pooling fournir un gros gain en puissance de calcul. Cependant, en raison de la réduction agressive de la taille de la représentation (et donc de la perte d'information associée), la tendance actuelle est d'utiliser de petits filtres (type 2x2). Il est aussi possible d'éviter la couche de pooling mais cela implique un risque sur-apprentissage plus important. Les types de pooling les plus populaires sont le max et l'average pooling, où les valeurs maximales et moyennes sont prises, respectivement [34]. Le tableau ci-dessous illustre les deux types de la couche mise en commun.

.

### 3.2.1.2 Paramètres du filtre

La couche convolutionnelle (CONV) contient des filtres pour lesquels il est important de savoir comment ajuster ses paramètres.

**1.Dimensions d'un filtre**

Un filtre de taille F × F appliqué à une entrée contenant C canaux est un volume de taille F × F × C qui effectue des convolutions sur une entrée de taille I × I × C et qui produit un feature map

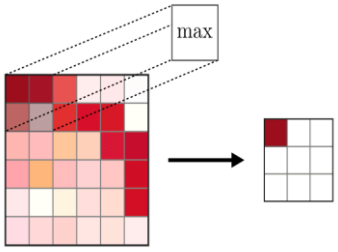
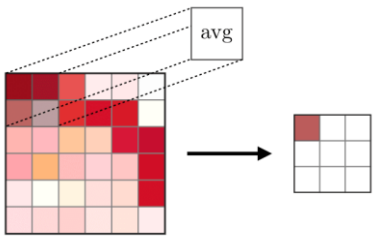
| **Type** | Max pooling | Average pooling |
|---|---|---|
| **But** | Chaque opération de pooling sélectionne la valeur maximale de la surface. | Chaque opération de pooling sélectionne la valeur moyenne de la surface. |
| **Illustration** |  |  |
| **Commentaires** | - Garde les caractéristiques détectées.<br>- Plus communément utilisé. | - Sous-échantillonne la feature map.<br>- Utilisé dans LeNet. |

**Tableau 2. 1** Les types de la couche mise en commun

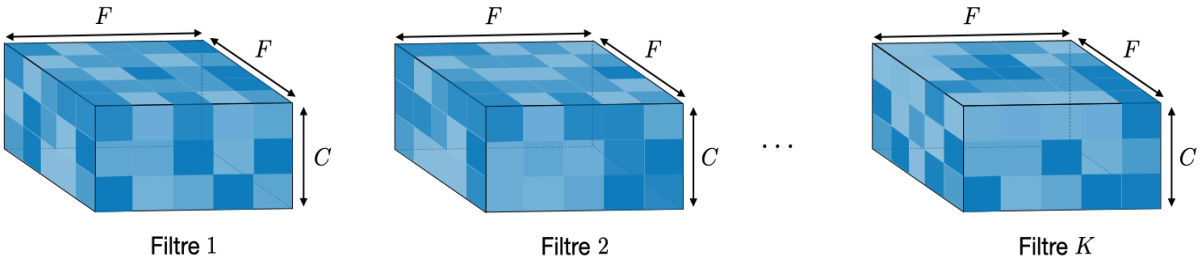



**Figure 2. 13** Dimensions d'un filtre *[34]*.

Remarque : appliquer K filtres de taille F × F engendre un feature map de sortie de taille O × O × K.

**2.Stride**

Dans le contexte d'une opération de convolution ou de pooling, la stride S est un paramètre qui dénote le nombre de pixels par lesquels la fenêtre se déplace après chaque opération [34].

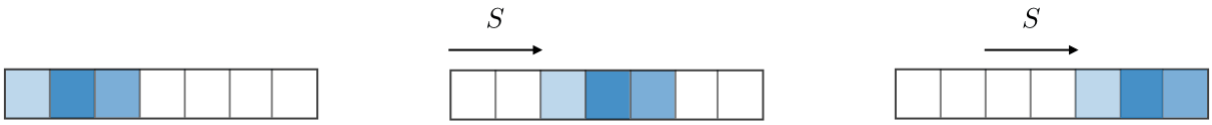



**Figure 2. 14** La fenêtre se déplace avec 3 pixels (S=3) après chaque opération *[34]*.

### 3.2.1.3 Fonctions d'activation

Les fonctions d'activation communément utilisées sont :

**1.Unité linéaire rectifiée**

La couche d'unité linéaire rectifiée (en anglais rectified linear unit layer « ReLU ») est une fonction d'activation **g** qui est utilisée sur tous les éléments du volume. Elle a pour but d'introduire des complexités non-linéaires au réseau [34].

## 3.3. Fonction de perte de l'indice de similarité structurelle (SSIM)

*SSIM* est utilisé comme métrique pour mesurer la similarité entre deux images données.

La métrique SSIM (Structural Similarity Index) extrait 3 caractéristiques clés d'une image : Luminance, Contraste et Structure [35] .

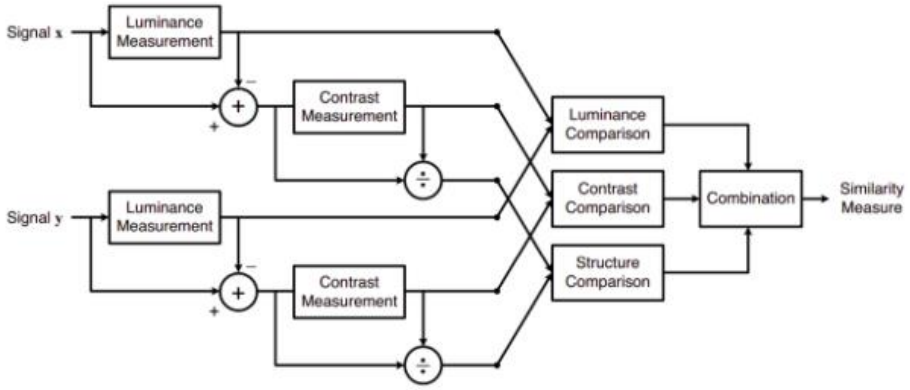



**Figure 2. 15** L'architecture de fonction de perte [35]

Indice de similarité structurelle entre 2 images données qui est une valeur comprise entre -1 et +1. Une valeur de +1 indique que les 2 images données sont très similaires ou identiques tandis qu'une valeur de -1 indique que les 2 images données sont très différentes.

Pour des images similaires, la fonction de perte SSIM sera plus petite et pour les images anormales, la fonction de perte SSIM sera plus grande.

## 3.4 Conception générale de notre système

Le processus de détection de fraude documentaires comporte quatre étapes :

- Encoder l'image
- Décoder cette image
- Calculer la fonction d'erreur
- Détecter la fraude par cette métrique (fonction d'erreur)

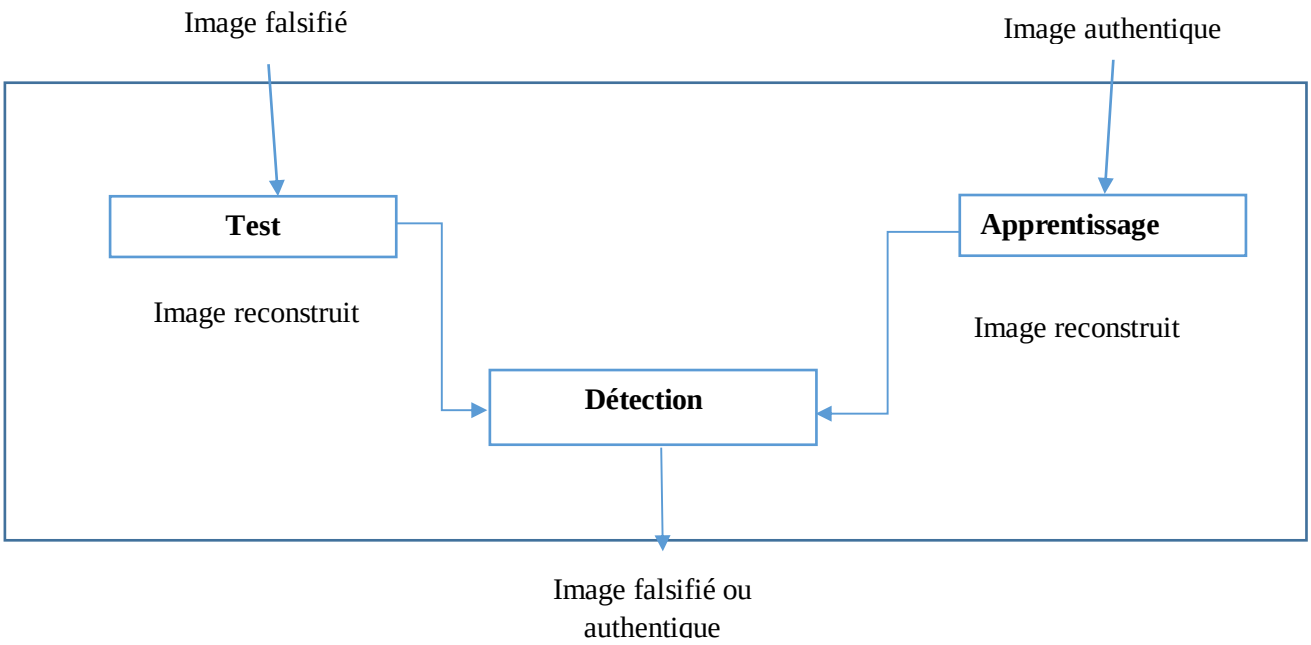



**Figure 3. 1** Schéma générale de notre système de détection de fraude documentaires

### 3.4.1 Conception détaillé

Notre système est décomposé de deux phases principales : la phase d'entrainement et la phase de test, il contient aussi une phase de détection, cette dernière va nous permettre de détecter l'image falsifié. Dans ce qui suit nous détaillons ces phases.

#### 3.4.1.1 Phase d'entrainement

##### 3.4.1.1.1 Prétraitement

Le prétraitement des données est la première étape vers l'élaboration d'un modèle d'apprentissage profond. Dans notre cas notre base de données était un mélange entre des images couleur (RGB) et en niveaux de gris, nous avons donc la transformé en niveaux de gris pour qu'elle soit homogène et le traitement sera plus facile.

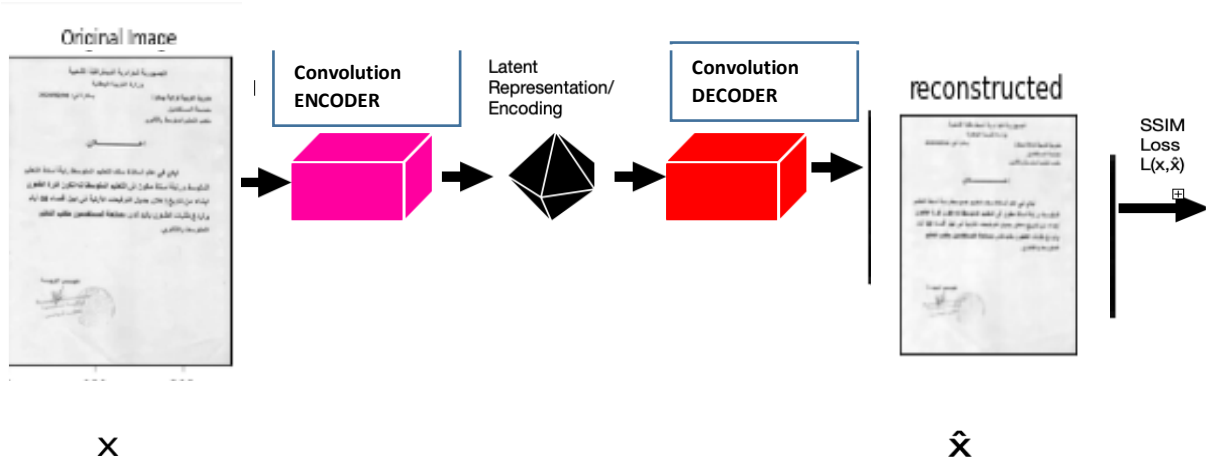



**Figure 3. 2** Modèle d'entrainement

Notre modèle d'entrainement effectué sur une base d'images authentique est un auto-encodeur convolutif qui est essentiellement un système d'encodeur-décodeur qui fait la reconstruction d'une image d'entrée en une image de sortie. Ceci est réalisé par deux sous-systèmes :

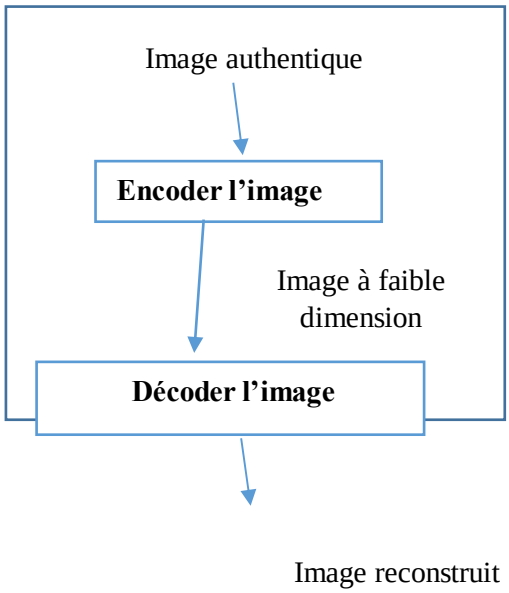



**Figure 3. 3** phase d'apprentissage

#### 3.4.1.1.2 Encodeur

L'Encodeur prend comme entrée une image et fait l'extraction de caractéristique. A cette étape il fait aussi la compression de cette dernière pour obtenir une image à faible dimension.

#### 3.4.1.1.3 Décodeur

Le décodeur fait la reconstruction de l'image à faible dimension afin d'obtenir une image de même taille que l'image d'origine.

### 3.4.1.2 Phase de test

Nous allons tester ce modèle sur la base d'images falsifié. Pour cela nous allons appliquer notre modèle à cette base pour obtenir

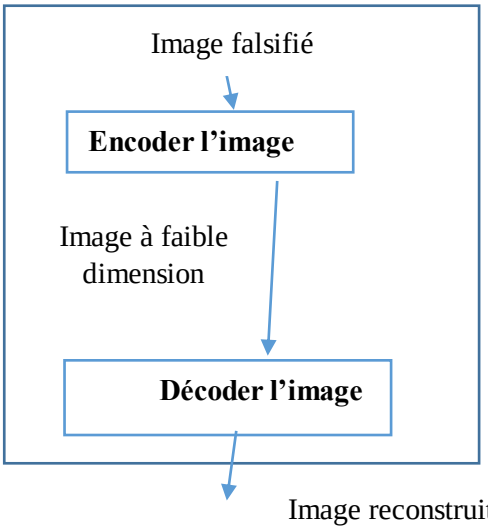



**Figure 3. 4** phase de test

### 3.4.1.3 Phase détection

Calculer l'erreur entre les deux images reconstruites des deux phases si l'erreur de reconstruction est supérieure à certain seuil l'image prédite est falsifié sinon elle est authentique.

.

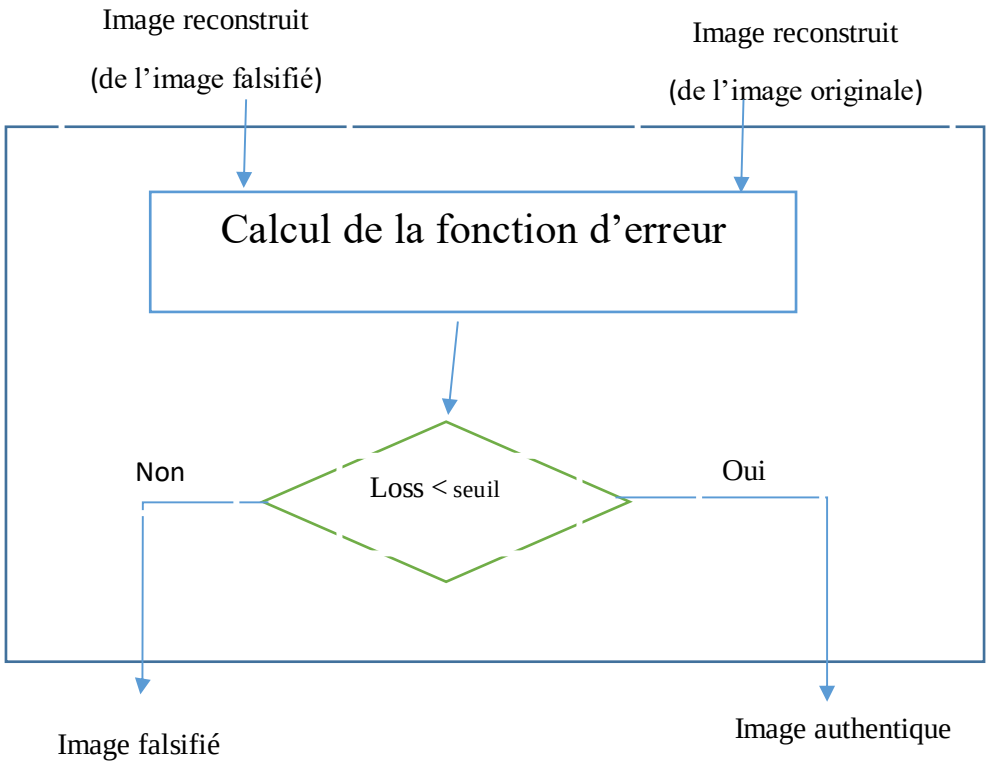



**Figure 3. 5** phase détection

## 3.5 Astuces d'apprentissage profond

### 3.5.1 Entraîner un réseau de neurones

#### 3.5.1.1 Itération

Mise à jour unique des pondérations d'un modèle pendant l'apprentissage. Une itération consiste à calculer les gradients des paramètres en termes de perte sur un seul lot de données.

#### 3.5.1.2 Lot (batch)

Ensemble d'exemples utilisés dans une itération (c'est-à-dire, une mise à jour du gradient) de l'entraînement du modèle.

#### 3.5.1.3 Taille de lot (batch size)

Nombre d'exemples d'un lot. La taille de lot est habituellement fixée pendant les processus d'apprentissage.

#### 3.5.1.4 Époque (Epoch)

Cycle d'apprentissage complet sur l'intégralité de l'ensemble de données de manière à ce que chaque exemple ait été vu une fois. Une itération représente ainsi *N/taille du lot* itérations d'apprentissage, où *N* est le nombre total d'exemples.

#### 3.5.1.5 Gradient descent sur mini-lots

Durant la phase d'entraînement, la mise à jour des coefficients n'est souvent basée ni sur tout le training set d'un coup à cause de temps de calculs coûteux, ni sur un seul point à cause de bruits potentiels. À la place de cela, l'étape de mise à jour est faite sur des mini-lots, où le nombre de points dans un lot est un paramètre que l'on peut régler [36].

#### 3.5.1.6 Fonction de loss

Pour pouvoir quantifier la performance d'un modèle donné, la fonction de loss (en anglais loss function) est utilisée pour évaluer la mesure dans laquelle les sorties vraies $y$ sont correctement prédites par les prédictions du modèle $\boldsymbol{z}$ [36].

## 3.6 Implémentation d'une architecture d'apprentissage profond

### 3.6.1 Langage, Logiciels et librairies

#### 1.Python

Python est un langage de programmation de haut niveau interprété (il n'y a pas d'étape de compilation) et orienté objet avec une sémantique dynamique. Il est très sollicité par une large communauté de développeurs et de programmeurs. Python est un langage simple, facile à apprendre et permet une bonne réduction du cout de la maintenance des codes. Les bibliothèques (packages) python encouragent la modularité et la réutilisabilité des codes.

Python et ses bibliothèques sont disponibles (en source ou en binaires) sans charges pour la majorité des plateformes et peuvent être redistribués gratuitement [37].

**Figure 3. 6** Python logo

**2.Google Colab**

Est un produit de Google Research. Colab permet à n'importe qui d'écrire et d'exécuter le code Python de son choix par le biais du navigateur. C'est un environnement particulièrement adapté au machine learning, à l'analyse de données et à l'éducation.

**Figure 3. 7** Google Colab logo

**3.Tensorflow**

TensorFlow est une bibliothèque de logiciels open source pour le calcul numérique haute performance. Son architecture flexible permet un déploiement facile du calcul sur une variété de plates-formes (CPU, GPU, TPU), et des ordinateurs de bureau aux clusters de serveurs en passant par les périphériques mobiles et périphériques.

Développé à l'origine par des chercheurs et des ingénieurs de l'équipe Google Brain au sein de l'organisation IA de Google, il est livré avec un support solide pour l'apprentissage automatique et l'apprentissage en profondeur et le noyau de calcul numérique flexible est utilisé dans de nombreux autres domaines scientifiques.


**Figure 3. 8** Tensorflow logo

**4.Keras**

Keras est une puissante bibliothèque libre Python pour le développement et l'évaluation de modèles d'apprentissage profond.


**Figure 3. 9** Keras logo

## 3.6.2 La base de données

Nous avons utilisé une base de donnes qui est construite dans le LABO LESIA, contenant 512 images documentaire. La bases de données est divisée en 2 classes. Comme suit :

Classe 1 : Contient des images authentiques. Dont 173 images sont utilisées pour l'apprentissage et 22 images sont utilisées pour la validation.

Classe 2 : Contient 317 images falsifiées, qui sont utilisées pour le test.

## 3.6.3 Configuration utilisée dans l'implémentation

- La configuration du matériel utilisé dans notre implémentation est :
- Un PC portable Lenovo i5 8 gén CPU 1.8 GHZ.
- Carte graphique Nvidia MX150 4GB.
- RAM de taille 6 GO.
- Disque dur de taille 2 T
- Système d'exploitation Windows 10.

## 3.6.4 Préparation de base de données

1. Télécharger la base de données dans Google Colab
2. Fractionnement et préparation d'une base de données

```
+ Code  + Texte

import numpy as np
from keras.preprocessing.image import img_to_array
from keras.layers import Dense, Conv2D, MaxPooling2D, UpSampling2D
from keras.models import Sequential
import cv2
import matplotlib.pyplot as plt
import tensorflow as tf
from tensorflow.keras import layers
from keras.preprocessing.image import ImageDataGenerator
from keras.preprocessing.image import ImageDataGenerator, array_to_img, img_to_array, load_img

# augmentation on the train set
data_gen = ImageDataGenerator(
        rescale=1./255)

x_train = data_gen.flow_from_directory(directory=train_loc , target_size=(256, 256), class_mode='input',batch_size=5
                                       ,color_mode='grayscale' )

data_test =  ImageDataGenerator(
        rescale=1./255)
x_test =data_test.flow_from_directory(directory=validation , target_size=(256, 256), class_mode='input',batch_size=5
                                      ,color_mode='grayscale' )

anomaly_generator = data_test.flow_from_directory(
    directory= test_loc,
    target_size=(256, 256),
    class_mode='input',
    color_mode='grayscale'
    )
```

**Figure 3. 10** Préparation de base de données

# 3.7 Implémentation de notre modèle autoencodeur convolutionnel

## 1. Importer des bibliothèques et des modules

Pour construire un auto-encoder convolutif (CAE) avec keras, nous devons d'abord importer des bibliothèques

```
import keras
from keras.models import Sequential
from keras.layers import Dense, Activation, Flatten, Input
from keras.layers import Conv2D, MaxPooling2D, UpSampling2D
import matplotlib.pyplot as plt
from keras import backend as K
import numpy as np
from keras.preprocessing.image import ImageDataGenerator, array_to_img, img_to_array, load_img
from PIL import Image, ImageChops
from sklearn.neighbors import KernelDensity
import random
import numpy as np
import tensorflow as tf
import matplotlib.pyplot as plt

from tensorflow.keras import layers
from tensorflow.keras.models import Model
```

**Figure 3. 11** les bibliothèque et modèles nécessaires pour l'implémentation de l'autoencoder convolutif

## 2. Le modèle CAE

Notre Auto-encodeur est implémenté par le code suivant :

```
input = layers.Input(shape=(256, 256, 1))

# Encoder
x = layers.Conv2D(32, (3, 3), activation="relu", padding="same")(input)
x = layers.MaxPooling2D((2, 2), padding="same")(x)
x = layers.Conv2D(16, (3, 3), activation="relu", padding="same")(x)
x = layers.MaxPooling2D((2, 2), padding="same")(x)
x = layers.Conv2D(1, (3, 3), activation="relu", padding="same")(x)
x = layers.MaxPooling2D((2, 2), padding="same")(x)
x = layers.Conv2D(1, (3, 3), activation="relu", padding="same")(x)
x = layers.MaxPooling2D((2, 2), padding="same")(x)

# Decoder
x = layers.Conv2DTranspose(1, (3, 3), strides=2, activation="relu", padding="same")(x)
x = layers.Conv2DTranspose(1, (3, 3), strides=2, activation="relu", padding="same")(x)
x = layers.Conv2DTranspose(16, (3, 3), strides=2, activation="relu", padding="same")(x)
x = layers.Conv2DTranspose(32, (3, 3), strides=2, activation="relu", padding="same")(x)
x = layers.Conv2D(1, (3, 3), activation="sigmoid", padding="same")(x)

# Autoencoder
autoencoder = Model(input, x)
autoencoder.compile(optimizer="adam", loss="binary_crossentropy")
autoencoder.summary()
```

**Figure 3. 12** l'implémentation de l'autoencoder convolutif

| Les paramètres | informations |
|---|---|
| Input Shape | (256,256,1) |
| Epoches | 2000 |
| Batch size | 5 |
| Layer activations | Relu ,sigmoid |
| optimizer | Adam |
| Loss | Binary Crossentropy |

**Tableau 3. 1** spécification des paramètres

## 3. Compilation du modèle CAE

```
autoencoder.compile(optimizer="adam", loss="binary_crossentropy")
```

**Figure 3. 13** compilation du modèle

## 4. Modèle summary

Une fois que notre modèle est prêt, nous pouvons appeler la méthode summary () pour afficher son contenu

```
autoencoder.summary()
```

**Figure 3. 14** l'instruction d'affichage

```
Model: "model_18"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 input_13 (InputLayer)       [(None, 256, 256, 1)]     0

 conv2d_48 (Conv2D)          (None, 256, 256, 32)      320

 max_pooling2d_36 (MaxPoolin  (None, 128, 128, 32)     0
 g2D)

 conv2d_49 (Conv2D)          (None, 128, 128, 16)      4624

 max_pooling2d_37 (MaxPoolin  (None, 64, 64, 16)       0
 g2D)


 conv2d_50 (Conv2D)          (None, 64, 64, 1)         145

 max_pooling2d_38 (MaxPoolin  (None, 32, 32, 1)        0
 g2D)

 conv2d_51 (Conv2D)          (None, 32, 32, 1)         10

 max_pooling2d_39 (MaxPoolin  (None, 16, 16, 1)        0
 g2D)

 conv2d_transpose_28 (Conv2D  (None, 32, 32, 1)        10
 Transpose)

 conv2d_transpose_29 (Conv2D  (None, 64, 64, 1)        10


  conv2d_transpose_30 (Conv2D  (None, 128, 128, 16)     160
  Transpose)

  conv2d_transpose_31 (Conv2D  (None, 256, 256, 32)     4640
  Transpose)

  conv2d_52 (Conv2D)          (None, 256, 256, 1)       289

=================================================================
Total params: 10,208
Trainable params: 10,208
Non-trainable params: 0
_________________________________________________________________
```

**Figure 3. 15** Modèle summary

## 5. Entrainement du modèle CAE

Pour entrainer le modèle, nous avons utilisé la méthode (fit)

```
hist=autoencoder.fit(x_train,
                epochs=2000  ,
                batch_size=None,
                shuffle=True,
                validation_data=x_test)

plt.plot(hist.history['loss'], label = 'train')
plt.plot(hist.history['val_loss'], label = 'test')
plt.title(' :  Loss  &  Validation Loss')
plt.ylabel('Loss')
plt.xlabel('Epoch')
plt.legend()
plt.show()
```

**Figure 3. 16** Entrainer le modèle

Après l'entrainement on peut obtenir la base d'images reconstruit

```
pred = autoencoder.predict(x_test)
```

**Figure 3. 17** instruction de prédiction pour les images authentiques

## 6. Teste notre Modele CAE :

Nous allons tester la prédiction du modèle sur les données falsifiées

```
pred_an = autoencoder.predict(anomaly_generator)
```

**Figure 3. 18** linstruction de prédiction pour les images falsifiées

## 3.8 Résultat et discussion

### 3.8.1 Résultat obtenu pour le modèle CAE

La première étape consiste à faire un entrainement du modèle, pour cela nous avons utilisé une base d'image authentique pour apprendre au modèle à reconnaitre les images normales.

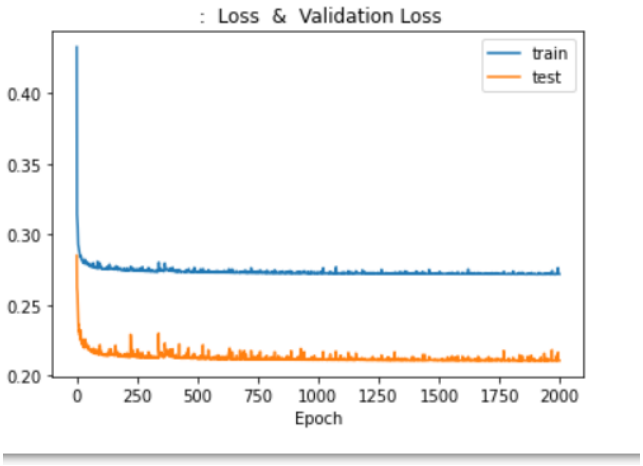



**Figure 3. 19** modèle de perte

Les tracés des courbes d'apprentissage 3.19 montrent un bon ajustement (fit) parce que :

- L'erreur d'apprentissage et de la validation diminue avec le nombre d'époque.

1. **Test de reconstruction d'image par le modèle**

La figure suivante montre un exemple de reconstruction d'une image à partir d'une image originale

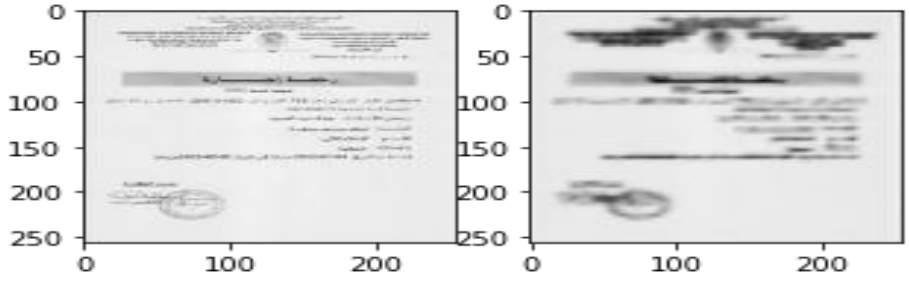



**Figure 3. 20** Exemple d'une image originale et sa reconstruction

## 2. Le premier test du modèle pour la détection

Pour voir l'efficacité de notre autoencodeur nous avons le testé dans trois cas :

Cas N01 : au début nous avons calculé la fonction de perte entre l'image reconstruit de l'image originale et celle de l'image falsifiée de la même image (courbe rouge dans la figure 3.21) nous avons remarqué que la valeur de la fonction de perte (loss) est dans l'intervalle [0.002 – 0.015] ce qui signifie que notre système peut détecter qu'une image est falsifié.

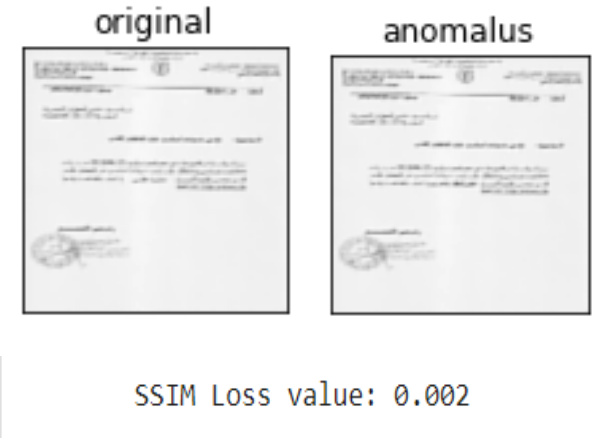



**Figure 3. 21** Exemple de fonction de perte d'image original et celle image falsifié

Cas N° 2 : la deuxième expérience était entre une image reconstruite à partir d'une image originale et celle d'une image falsifiée d'une autre image (courbe violet dans la figure 3. 21) nous avons remarqué que la valeur de la fonction de perte (loss) est dans l'intervalle

[0.416 – 0.832] ce qui signifie que notre système peut détecter qu'une image est différente de l'image.


**Figure 3. 22** Exemple de fonction de perte image originale et celle d'une image falsifiée d'une autre

Image

Cas N° 3 : afin de s'assurer du bon fonctionnement de autoencodeur, nous avons ajouté une autre expérience, dans cette dernière nous avons calculé la fonction de perte entre deux images reconstruite à partir du même image (courbe marron dans la figure 3. 21) nous avons remarqué que cette fonction est toujours égale à 0. Ce qui signifie que le système peut détecter qu'une image est authentique.


**Figure 3. 23** Exemple de fonction de perte entre deux images reconstruite à partir du même image

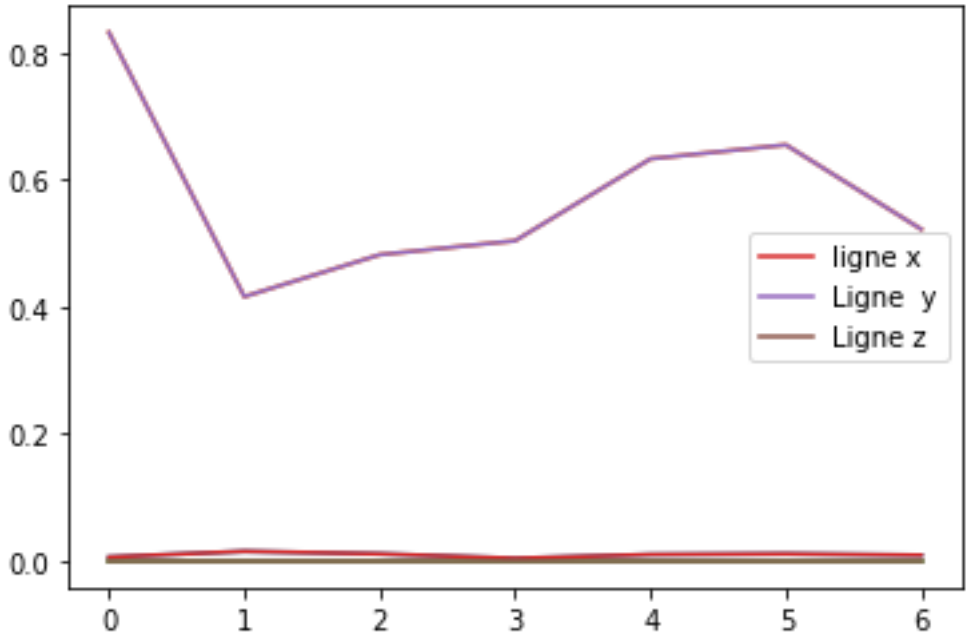



**Figure 3. 24** schéma illustre les valeurs de fonction de perte SSIM entre des images reconstruites dans les 3 cas précédents

### 3. Le deuxième test du modele pour la détection d'anomalie

Dans ce test de détection d'anomalie nous avons utilisé une technique appelée

estimation de la densité du noyau pour faire une comparaison avec la fonction de perte.

Au lieu de se fier uniquement à l'erreur de reconstruction on considère également la probabilité d'une image dans l'espace latent (la couche la plus compressée du réseau d'encodeur automatique). En visualisant notre ensemble de données d'image comme une collection de vecteurs, l'estimation de la densité du noyau mesure la densité de chaque partie de l'espace vectoriel occupée par les données d'entrainement.

En implémentant cette méthode avec l'ensemble des images d'apprentissage et l'ensemble des images falsifiées, nous avons obtenu les histogrammes de la figure 3.22. Dans cette figure l'histogramme bleu indique la distribution de la densité des images normales (authentique) et l'histogramme vert représente la distribution de la densité des images anormale (falsifier), nous remarquons dans ce résultat que les deux histogrammes sont très proche ceci est dû du fait que les images falsifiées sont construites à partir des images normales avec une légère modification.

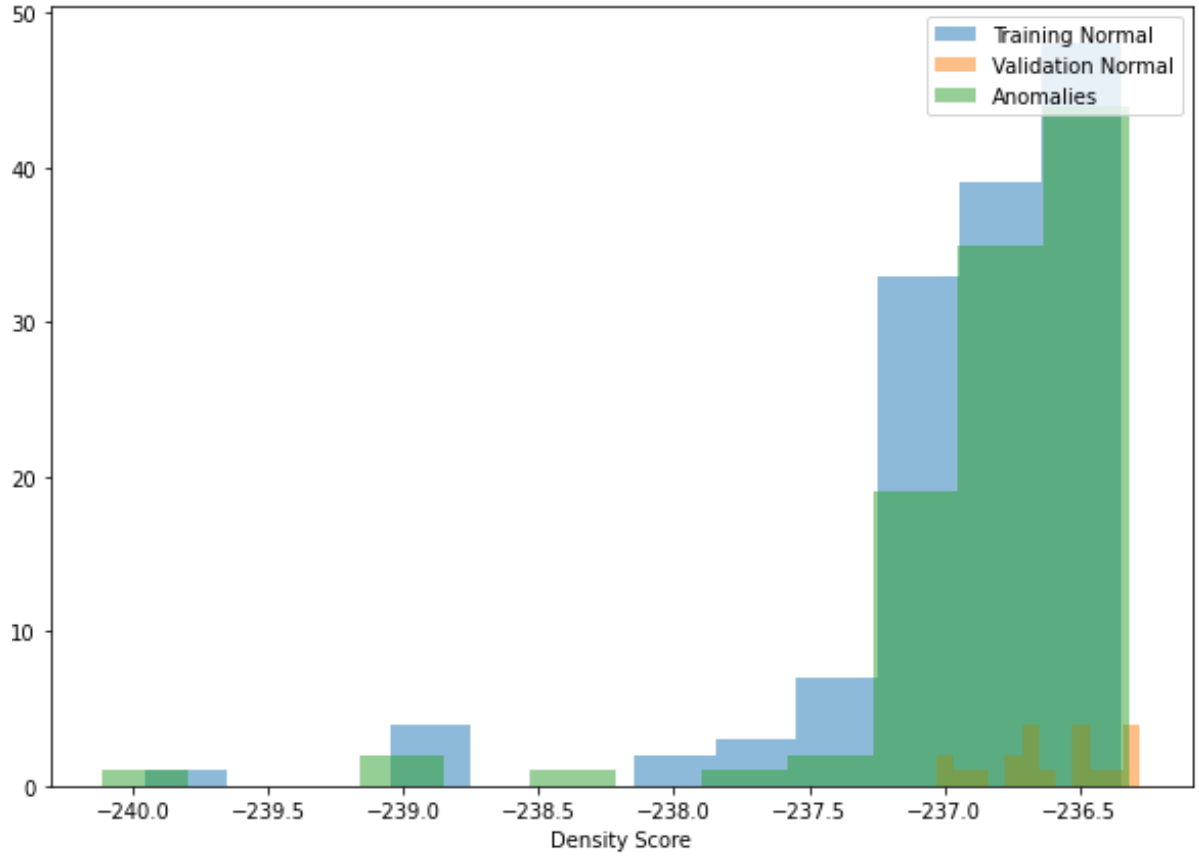



**Figure 3. 25** Estimation de la densité du noyau comme mesure de l'anomalie /authentique

## 4. Comparaison entre les deux méthodes de détection d'anomalies

En utilisant les deux méthodes de détection d'anomalie nous avons remarqué que la méthode de perte est plus convenable à notre modèle et à notre base d'image, puisqu'elle nous a permis de détecter les images normales des images anormales. Tandis que la méthode de distribution de densité peut être plus convenable si le taux de falsification des images est élevé.

### 5. L'évaluation du modèle

Dans notre système nous avons utilisé la fonction d'erreur (loss) « binary_crossentropy » comme fonction de perte pour le modèle de classification binaire. Cette fonction calcule la perte d'entropie croisée entre les étiquettes vraies et les étiquettes prédites.

Donc nous allons faire l'évaluation de notre modèle par cette fonction, ainsi la valeur du perte (loss) est 26%, ce résultat est acceptable car plus le taux d'érreur est bas alors nous ayons une bonne reconstruction.

Le but d'entrainement est de diminuer la valeur de perte pour obtenir une meilleure reconstruction.

```
[ ] test_eval = autoencoder.evaluate(anomaly_generator, verbose=0)

[ ] print('Test loss:', test_eval)

Test loss: 0.2682780921459198
```

**Figure 3. 26** résultat d'évaluation

## 3.9 Conclusion

Nous avons présenté dans ce chapitre notre approche de détection d'anomalie basée sur les réseaux de neurones convolutionnels, pour cela nous avons utilisé le modèle autoencodeur convolutif. Les résultats obtenus sont satisfaisant en termes de précision et d'erreur.

# Conclusion générale

L'échange de documents a toujours été sujet à suspicion sur l'authenticité du document reçu, sur l'identification de l'auteur ou de l'éditeur, ou sur son intégrité physique. De nombreuses recherches ont été effectuées pour sécuriser l'échange et authentifier les images envoyées ou reçues, qu'ils soient sous format papier ou format numérique. La lutte contre la fraude est un enjeu économique très important, que ce soit pour les entreprises, pour les administrations ou pour les particuliers. Plusieurs études tendent à montrer que les entreprises cherchent de plus en plus à se doter d'outils performants de détection des fraudes, quel que soit leur domaine d'activité [1].

Partant de ce principe, nous avons conçu un système de détection d'anomalies dans les fraudes documentaires. Durant notre étude, nous nous somme attacher à :

Faire une étude sur l'apprentissage automatique

Donner une vue générale sur la vision par ordinateur, le traitement d'image et l'extraction de caractéristique

Concevoir et implémenter un système permettant de faire la détection des anomalies dans les images falsifiées basé sur un modèle d'apprentissage profond qui est l'autoencodeur convolutif.

Le système que nous avons proposé est un noyau d'un système de détection de des anomalies dans les fraudes documentaires, il peut être amélioré par l'ajout de plusieurs composantes telles que l'intégration d'un LSTM basé sur les réseaux de neurones récurent pour localiser l'endroit falsifié.

# Références

[1] C. Artaud., «Détection des fraudes : de l'image à la sémantique du contenu : application à la vérification des informations extraites d'un corpus de tickets de caisse.,» pp. 25-28, 2019.

[2] «eurodecision,» [En ligne]. Available: https://www.eurodecision.com/algorithmes/machine-learning.

[3] «Apprentissage_automatique,» [En ligne]. Available: https://fr.wikipedia.org/wiki/Apprentissage_automatique.

[4] «Connaissez-vous l'overfitting ?,» [En ligne]. Available: https://datascientest.com/connaissez-vous-overfitting. [Accès le 8 juin 2022].

[5] «regularisation,» [En ligne]. Available: https://dataanalyticspost.com/Lexique/regularisation/.

[6] O. Delalleau, «''Apprentissage machine efficace : théorie et pratique'',» *Département d'informatique et de recherche opérationnelle Faculté des arts et des sciences..*

[7] E. Nadaraya, «"On estimating regression",» *Theory of Probability and its Applications,,* vol. 9, pp. 141-142, 1964.

[8] j. Weizenbaum, « "ELIZA-a computer program for the study of natural language communication between man and machine",» *Commun. ACM,,* vol. 9, pp. 36-45, 1966.

[9] «mylittleneuron,» [En ligne]. Available: https://mylittleneuron.com/2020/10/20/introduction-machine-de-boltzmann-restreinte/.

[10] K. M. S. e. H. W. Hornik, «Multilayer feedforward networks are universal approximators,» *Neural Networks,,* vol. 2, pp. 359-366, 1989.

[11] E. Mathieu-Dupas, «Algorithme des k plus proches voisins pondérés et application en diagnostic».

[12] M. Rosenblatt, «"Remarks on some nonparametric estimates of a density function",» *The Annals of Mathematical Statistics,* Vols. %1 sur %2 vol. 27, no 3, p. p. 832–837., 1956.

[13] E. Parzen, «"On the estimation of a probability density function and mode",» *Annals of Mathematical Statistics,* Vols. %1 sur %2vol. 33,, p. 1064–1076, 1962.

[14] «Modèle de mélange gaussien,» [En ligne]. Available: https://fr.wikipedia.org/wiki/Mod%C3%A8le_de_m%C3%A9lange_gaussien.

[15] «Astuce_du_noyau,» [En ligne]. Available: https://fr.wikipedia.org/wiki/Astuce_du_noyau.

[16] L. J. F. R. O. S. Breiman, « Classification and Regression Trees, Wadsworth International Group, Belmont, CA.,» 1984.

[17] F. Jedrzejewski, « "11 Méthodes bayésiennes",» *. Mathématiques pour l'imagerie médicale,Les Ulis: EDP Sciences,* pp. 167-180, 2021.

[18] «Vision_par_ordinateur,» [En ligne]. Available: https://fr.wikipedia.org/wiki/Vision_par_ordinateur. [Accès le 6 juin 2022].

[19] M. M. [. 12], ««Indexation et recherche d'images par contenu symbolique (approche statistique) »,» Université Mohamed Khider – BISKRA., 2012.

[20] DJABEUR DJEZZAR Mohammed Rafik, «''Mise au Point d'une Application de Reconnaissance de Formes'', Mémoire de fin d'études pour l'obtention du diplôme de Master en Informatique».

[21] 8 juin 2022. [En ligne]. Available: http ://di.univ-blida.dz :8080/jspui/bitstream/123456789/11363/1/pfe.ouachemi-ouchfoun.classification.pdf..

[22] L. B. e. A. BENYAGOUB., «"Utilisatition des Algorithmes de L'Int́eligence Artificielle appliquer `a l'Aide `a la Conduite".Th`ese de doct.,» 2021..

[23] «La détection des objets dans des images.,» [En ligne]. Available: http://e-biblio.univ-mosta.dz/bitstream/handle/123456789/13141.

[24] A. A. e. G. Hassina., «Réalisation d'un syst`eme de lecture des plaques d'immatriculation Algérienne.Thse de doct. Université Mouloud Mam-,» 2016..

[25] M. Nadia., *Thèse de doct. Universite mohamed boudiaf'des sciences et de la technologie d'oran,* 2014.

[26] «datascience,» [En ligne]. Available: https://datascience.eu/fr/mathematiques-et-statistiques/introduction-detection-des-anomalies/. [Accès le 8 juin 2022].

[27] «detectiondanomalie,» [En ligne]. Available: https://dataanalyticspost.com/Lexique/detection-danomalie/#:~:text=Le%20but%20de%20la%20d%C3%A9tection,la%20distribution%20de%20probabilit%C3%A9%20observ%C3%A9e.. [Accès le 8 juin 2022].

[28] «detection d'anomalie,» [En ligne]. Available: https://fr.wikipedia.org/wiki/D%C3%A9tection_d%27anomalies. [Accès le 8 juin 2022].

[29] [En ligne]. Available: https://definir-tech.com/app2/25589/comment-appelle-t-on-ce-type-d-usurpation-d-identite-par-telephone-comment-appelle-t-on-ce-type-d-hameconnage. [Accès le 8 juin 2022].

[30] «shop.ctms,» [En ligne]. Available: https://shop.ctms.fr/fraude-documentaire/content/238-quels-sont-les-differents-types-de-fraudes-documentaires-.

[31] A. M. SOW, «CLASSIFICATION, RÉDUCTION DE DIMENSIONNALITÉ ET RÉSEAUX DE».

[32] «pgaleone.eu convolutional-autoencoders,» [En ligne]. Available: https://pgaleone.eu/neural-networks/2016/11/24/convolutional-autoencoders/?fbclid=IwAR0w4xgOP8S0fW5FuVxieRXQwIsf8_7XcwQmZkRwb0aen8XC8W1YY8Njiaw.

[33] «researchgate.net,» [En ligne]. Available: https://www.researchgate.net/figure/Convolutional-Autoencoder-Architecture-used-in-this-paper_fig4_337326792?fbclid=IwAR3BUdwt7_mi1p7s0BPqUxAv1DLpe1z6uvSwesnisb1y61v--DKpBpn3t_4.

[34] «stanford.edu,» [En ligne]. Available: https://stanford.edu/~shervine/l/fr/teaching/cs-230/pense-bete-reseaux-neurones-convolutionnels.

[35] M. I. A. C. B. F. I. H. R. S. S. M. I. a. Zhou Wang, «Image Quality Assessment: From Error Visibility to,» p. 5.

[36] A. A. e. S. Amidi, « «CS 230 – Apprentissage profond Petites astuces,» Université de stanford».

[37] [En ligne]. Available: https://docs.python.org/fr/3/tutorial/..

[38] G. E. T. J. S. e. D. H. A. Hinton, «“Boltzmann machines : Constraint satisfaction networks thatlearn”, rapport technique TR-CMUCS-84-119,» *Carnegie-Mellon University, Dept. of Computer Science.,* 1984.

[39] G. E. e. T. J. S. «Hinton, «“Learning and relearning in Boltzmann machines”,,» *dans Parallel Distributed Processing : Explorations in the Microstructure of Cognition,» . ,* Vols. %1 sur %2Volume 1 : Foundations, édité par D. E. Rumelhart et J. L. McClelland, MIT, 1986.

[40] M. J. e. M. I. J. Wainwright, «“Graphical models, exponential families, and variational inference”, Foundations and Trends in Machine Learning, vol. 1, no 1-2, p. 1–305.,» *, Foundations and Trends in Machine Learning,* vol. 1, pp. 1-305, 2008.

[41] K. M. S. e. H. W. Hornik, «Multilayer feedforward networks are universal approximators,» *Neural Networks,* vol. 2, pp. 359-366, 1989.

[42] B. e. A. J. S. Scholkopf, Learning with Kernels : Support Vector Machines, Regularization, Optimization and Beyond, MIT Press, Cambridge, MA., 2002.

[43] N. F. e. G. E. H. «. R. B. C. (.-1.-2. S. Sabour, ««Dynamic Routing Between Capsules,»,» (2017-10-26)..

[44] G. S. Watson, «Smooth regression analysis,» *Sankhya - The Indian Journal of Statistics,* vol. vol.26, pp. 359-372, 1964.

[46] «Vision Par ordinateur,» [En ligne]. Available: / / fr . wikipedia . org / wiki /.

[47] J. Weber., «Segmentation morphologique interactive pour la fouille de séquences vidéo,» 2011.

[48] «cours-14-introduction-a-la-vision-par-ordinateur,» [En ligne]. Available: https://www.electronique-mixte.fr/formation-pdf/formation-pdf-traitement-dimage-vision-par-ordinateur/cours-14-introduction-a-la-vision-par-ordinateur/. [Accès le 6 juin 2022].