Chapitre II

1. Introduction

Les systèmes de reconnaissance automatique basés sur les techniques de *Machine Learning* « ML » ont enregistré des performances remarquables. Ces performances sont relatives aux modalités d'entrées utilisées telles que les textes, les images, les flux de parole et les vidéos, ainsi que la complexité de la tâche a exécuté.

Entre autres, l'évolution importante enregistrée dans les techniques et les algorithmes utilisés dans pour l'apprentissage automatique et celle de l'apprentissage profond couplé avec les puissances de calculs actuelles ont permet d'entamer de problématiques plus complexes comme la reconnaissance automatique des liens de parenté.

Dans ce contexte, nous présentons dans ce chapitre les techniques de ML utilisées pour le développement des systèmes de reconnaissance automatique des liens de parenté à base des images faciales. Ainsi, nous présentons une étude bibliographique sur les travaux réalisés dans ce contexte avec un tableau de synthèse récapitulatif.

2. Les types de systèmes de reconnaissance automatique

Le développement des systèmes de reconnaissance automatique en ML est basé sur la stratégie déployée pour la phase d'apprentissage automatique ainsi que les algorithmes de reconnaissance utilisés. Généralement, dans le domaine de reconnaissance faciale et les liens de parenté nous trouvons trois catégories : les systèmes classiques basés sur les algorithmes d'apprentissage classique de ML, les systèmes basés sur les algorithmes de l'apprentissage profond et les systèmes basés sur l'apprentissage des mesures « *Metric Learning* ». Dans ce mémoire et tenant compte de nos objectifs, nous intéressons sur les deux premiers types de systèmes.

2.1. Les systèmes de reconnaissance classiques

Généralement, un système de reconnaissance automatique qui utilise les techniques d'apprentissage automatique classique comme les de vérification de la parenté est constituée de deux phases principales :

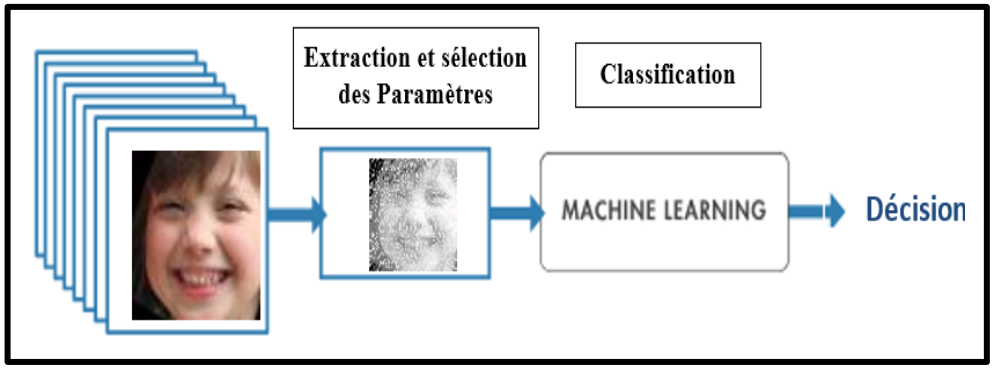


Figure 2.1 : Les étapes d'un système de reconnaissance automatique classique.

2.1.1. Extraction de caractéristiques

Cette étape représente le cœur du système de vérification après la localisation du visage dans l'image faciale. Elle permet l'extraction des informations discriminantes, utiles et non redondantes (vecteur de caractéristiques), à travers la description de points d'intérêts associés aux caractéristiques faciales, telles que le skin, les yeux, les sourcils, le nez et la bouche. Ces derniers sont utilisés pour la construction des modèles de représentation pour les algorithmes de classification et décision. L'efficacité de cette étape à une influence directe sur les performances des systèmes de vérification des liens de parenté. L'extraction des caractéristiques relatives aux visages utilise plusieurs méthodes qui sont classées généralement en trois approches (globales, locales et hybrides), parmi lesquelles, nous citons celles que nous avons utilisés lors de développement de notre contribution pour la détection automatique des liens de parenté, qui sont : LBP « *Local Binary Pattern* », TPLBP « *Three-Patch Local Binary Patterns* », FPLBP « *Four-Patch Local Binary Patterns* », HOG « *Histogram of oriented gradients* ».

a. Motif binaire local (LBP)

Cette technique était initialement proposée par *Ojala* [35]. Afin de caractériser les textures présentes dans des images en niveaux de gris. Elle était largement utilisée dans diverses applications notamment la reconnaissance faciale. Le principe général de cette technique est de comparer le niveau de gris de chaque pixel avec les valeurs des niveaux de ses huit voisins dans un voisinage 3 × 3 en soustrayant la valeur du pixel central. Afin de générer un motif binaire. Les valeurs strictement négatives obtenues sont codées avec un "zéro" (0) et les autres avec un "un" (1). Les pixels de ce motif binaire sont alors multipliés par des poids donnés par les puissances de deux (2 poids) et sommés afin d'obtenir un code LBP du pixel central ou courant. Ainsi, nous obtenons pour chaque image avec des pixels dont l'intensité est comprise entre 0 et 255 et elles utilisent les histogrammes pour le choix des descripteurs [36]. La **figure 2.2** illustre le principe de cette méthode.

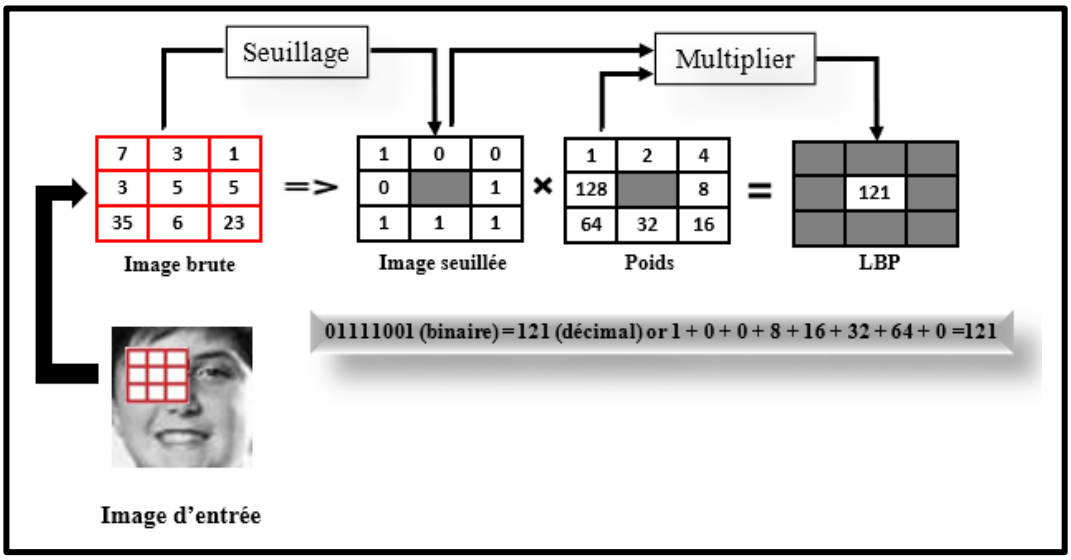


Figure 2.2 : Exemple d'extraction des caractéristiques en utilisant l'opérateur LBP basique.

Cependant, afin de traiter la texture à différentes échelles et de capturer les caractéristiques dominantes avec des structures à grande échelle. La technique LBP a été étendue en utilisant des voisinages de taille déférente (multi-échelle) [37]. Dans ce cas, un cercle de rayon R autour du pixel central et les valeurs des P points échantillonnés sur le bord de ce cercle sont prises et comparées avec la valeur du pixel central. La **figure 2.3**, illustre trois voisinages pour des valeurs de R et P différentes.

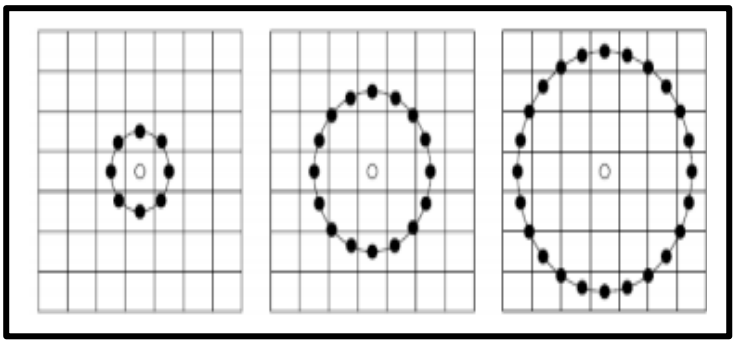


Figure 2.3 : Trois voisinages pour des R et P différents [36].

Formellement, en donnant un pixel central en coordonnées (x_c, y_c) avec le niveau de gris g_c dans l'image d'entrée, ses niveaux gris qui correspondent à P pixels voisins sur un cercle de rayon R sont $(g0, g1, \dots, gP-1)$. Le LBP résultant peut être exprimé sous forme décimale du pixel g_c comme suit :

$$\boldsymbol{LBP(x_c, y_c)_{P,R} = \sum_{i=0}^{p-1} u(g_i{}^{P,R} - g_c).2^i} \qquad (2.\ 1)$$

Où ***u(x)*** est la fonction de seuillage, donnée par :

$$\boldsymbol{u(x) = \begin{cases} 1 & si\ x \geq 0 \\ 0 & autrement \end{cases}} \qquad (2.\ 2)$$

La **figure** ci-dessous illustrée des exemples de quelques modalités biométriques avec leurs images LBP basique.

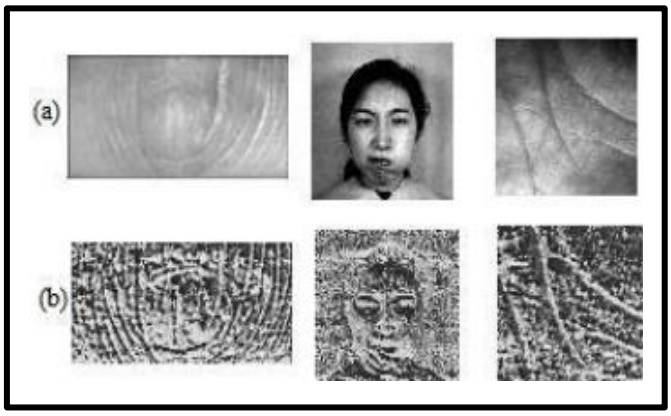


Figure 2.4 : (a) Images réelles des modalités biométriques **(b)** Images LBP basique correspondantes [38].

En effet, pour les résultats obtenus dans les applications du monde réel, la propriété la plus importante du descripteur LBP c'est qu'il est invariant contre les changements monotones du niveau gris causés comme les changements d'éclairage. Aussi sa facilité de calcul, qui permet l'analyse des images complexe.

b. LBP à trois patchs (TPLBP)

Le LBP à trois patches TPLBP « *Three-Patch Local Binary Patterns* » est une variante de LBP proposée par *Wolf, et al* [39]. Cette méthode est basée sur la comparaison des valeurs de trois patchs pour produire une valeur d'un seul bit dans le code attribué à chaque pixel. Pour chaque pixel de l'image, elle considère un patch (w × w) centré sur le pixel, et S patches supplémentaires distribuées uniformément dans un anneau de rayon r. Pour le paramètre α, elle utilise des paires de patchs α (le diamètre du cercle). La valeur d'un seul bit est définie la similitude des deux patches par rapport au patch central. Le code résultant a des bits S par pixel. Le TPLBP donne des résultats prometteurs en reconnaissance faciale. Cette méthode est donnée par l'équation suivante :

$$\boldsymbol{TPLBP}_{r,S,w,\alpha}(p) = \sum_i^S f\left(d(C_i, C_p) - d(C_{i+\alpha \ mod \ S}, C_p)\right).2^i \qquad (2.\ 3)$$

Où C_i et $C_{i+\alpha \ mod \ S}$ sont deux patchs le long de l'anneau et C_p est le patch central, d est toute fonction de distance entre deux patchs, et f est une fonction de seuillage qui est calculée comme suit :

$$f(x) = \begin{cases} 1 & if \ x \geq \tau \\ 0 & if \ x < \tau \end{cases} \qquad (2.4)$$

La valeur τ : est le seuil de comparaison, et généralement supérieure à zéro (p. ex., $\tau = 0{,}01$) pour assurer une certaine stabilité dans les régions uniformes [40]. Le principe de la méthode d'opérateur TPLBP est illustré à la **figure 2.5**.

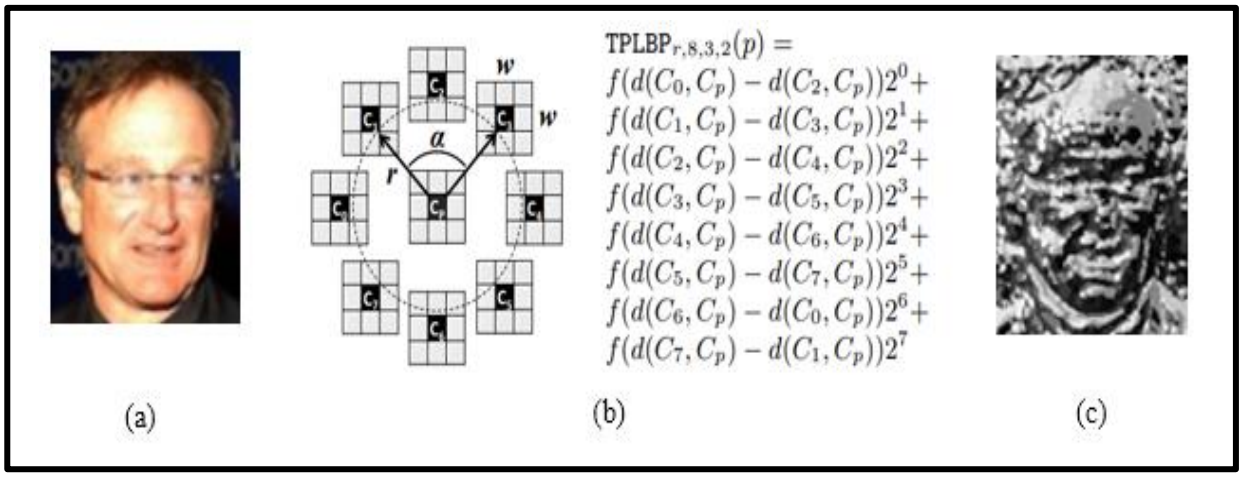


Figure 2.5 : (a) L'image réelle de visage. **(b)** Le code TPLBP calculé avec les paramètres S= 8, w= 3 et α= 2. **(c)** La représentation TPLBP d'une image réelle de visage [39].

Chaque image est divisée en une grille de régions qui ne se chevauchent pas et un histogramme de chaque code binaire est calculé pour chacune de ces régions. Chacun de ces histogrammes a été normalisées à la longueur de l'unité, leurs valeurs ont été tronquées à 0.2, puis normalisées de nouveau à la longueur unitaire. Finalement, l'image est représentée par ces histogrammes concaténés en un seul vecteur.

c. LBP à quatre patchs (FPLBP)

La technique LBP à quatre patchs FPLBP « *Four-Patch Local Binary Patterns* » est une autre variante de LBP, elle est presque la même que TPLBP avec une différence dans le nombre d'anneaux. Avec FPLBP, deux anneaux avec rayon r1 et r2 sont utilisés et centrés dans le pixel, et S patchs de taille (w × w) distribués autour de ces deux anneaux. Pour produire les codes (FPLBP) à quatre patchs, elle compare deux patchs symétriques centraux dans l'anneau intérieur avec deux patchs symétriques centraux dans l'anneau extérieur. Après la comparaison, un bit dans chaque pixel est pris en compte en fonction de laquelle des deux patchs sont les plus similaires. Ainsi, pour les S patchs le long de chaque cercle, S/2 est le nombre de paires symétriques centrales et qui est aussi la longueur des codes binaires produits [39]. La définition formelle d'opérateur FPLBP est la suivante :

$$\boldsymbol{FPLBP}_{r_1,r_2,S,w,\alpha}(p) = \sum_i^{S/2} f\left(d\left(C_{1i}, C_{2,i+\alpha \bmod S}\right) - d\left(C_{1,i+S/2}, C_{2,i+\frac{S}{2}+\alpha \bmod S}\right)\right).2^i \qquad (2.5)$$

Le principe de la méthode FPLBP est illustré à la **figure 2.6**.

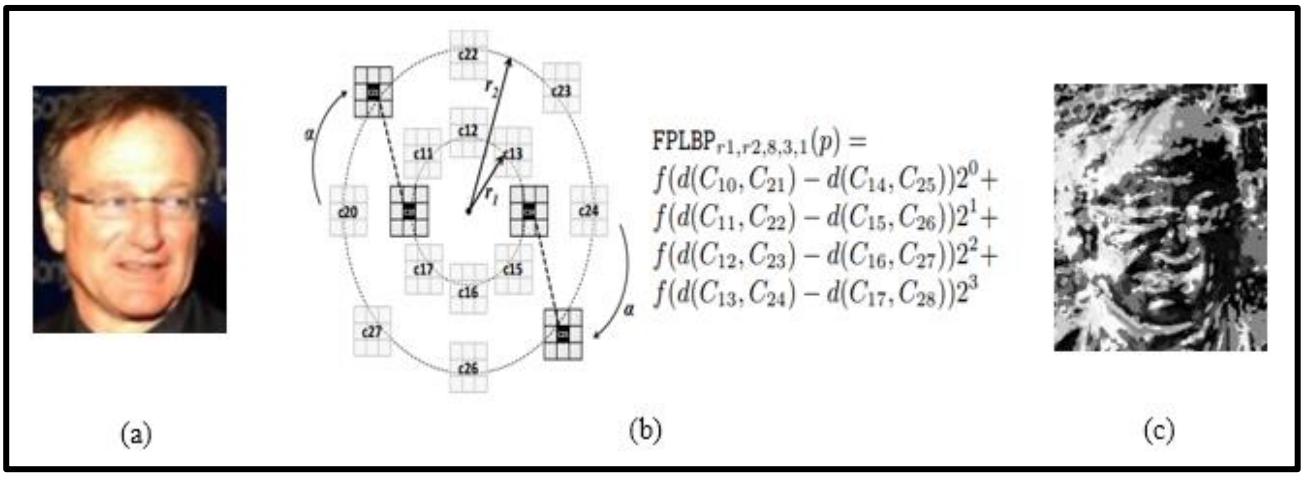


Figure 2.6 : (a) L'image réelle de visage. **(b)** Le code FPLBP calculé avec les paramètres S= 8, α= 1. **(c)** La représentation FPLBP d'une image réelle de visage [39].

d. Histogramme des gradients orientés (HOG)

L'histogramme des gradients orientés HOG « *Histogram of oriented gradients* » introduit par *Dalal et Triggs* [41], est actuellement l'une des méthodes les plus performantes et les plus utilisées pour l'apparence et la forme locale d'un objet dans une image. Il peut être décrit par la distribution de l'intensité du gradient ou la direction des contours. Ceci peut être fait par la division de l'image en petites régions adjacentes, appelées cellules, et le calcul pour chaque cellule l'histogramme des directions du gradient (orientations des contours) pour les pixels à l'intérieur de cette cellule. La concaténation de ces histogrammes représente alors le vecteur de caractéristiques du descripteur « HOG ». Pour de meilleurs résultats, les histogrammes locaux sont normalisés en contraste, par le calcul d'une mesure de l'intensité sur des groupes de cellules adjacentes, appelées des blocs. Cette dernière est utilisée pour normaliser toutes les cellules du bloc. L'algorithme de cette technique contient les étapes suivantes :

1- **Prétraitement :** dans cette étape les images d'entrées sont premièrement converties en niveaux de gris après une normalisation selon l'information de luminance. Ensuite, l'image I (x, y) est divisée en N cellules régulières (N petites régions) et des blocs. Les valeurs de

gradient sont calculées pour chaque pixel en avec un filtre dérivatif 1-D centré, dans les directions horizontales et verticales. Pour cela les masques suivants sont utilisés :

$$\boldsymbol{D_x = [-1 \quad 0 \quad 1]} \qquad (2.6)$$

$$\boldsymbol{D_Y = \begin{bmatrix} 1 \\ 0 \\ -1 \end{bmatrix}} \qquad (2.7)$$

$$\begin{cases} \boldsymbol{G_x(x,y) = I(x,y) * D_x} \\ \boldsymbol{G_y(x,y) = I(x,y) * D_y} \end{cases} \qquad (2.8)$$

2- **Calcul du gradient :** Le gradient d'un pixel (x, y) d'une image est un vecteur caractérisé par son amplitude et sa direction. Le calcul du gradient est une étape critique dans la formation de descripteurs.

L'amplitude d'une image se calcule comme suit :

$$\boldsymbol{|G(x,y)| = \sqrt{G_x(x,y)^2 + G_y(x,y)^2}} \qquad (2.9)$$

Et la direction du gradient est donnée par :

$$\boldsymbol{\alpha(x,y) = arctan\left(\frac{G_x(x,y)}{G_y(x,y)}\right)} \qquad (2.10)$$

Où G_x (x, y) et G_y (x, y) sont les gradients dans la direction horizontale et verticale, au pixel (x, y).

3- **Construction de l'histogramme :** La troisième étape est la création des histogrammes de l'orientation des gradients par des cellules carrées de petite taille de 4×4 à 12×12 pixels. Chaque pixel de la cellule vote alors pour une classe de l'histogramme, en fonction de l'orientation du gradient à ce point. Le vote du pixel est pondéré par l'intensité du gradient en ce point.

4- **Formation et normalisation des blocs :** afin d'éviter les disparités dues aux variations d'illumination, ainsi que l'introduction de redondance dans le descripteur. Les cellules sont regroupées par bloc (concaténation des histogrammes des cellules), le vecteur de valeur du bloc est ensuite normalisé. Le vecteur de HOG pour chaque cellule est normalisé en utilisant des vecteurs dans leurs blocs respectifs. En utilisant l'équation de normalisation (L2-norme) suivant :

$$\boldsymbol{V_n = \frac{V}{\sqrt{\|V\|_2^2 + e^2}}} \qquad (2.11)$$

Où : $\boldsymbol{V}$ est le vecteur contenant tous les histogrammes d'un bloc donné, $\boldsymbol{\|V\|_2}$est la norme $L2$ du vecteur $\boldsymbol{V}$ du descripteur et $\boldsymbol{e}$ est une petite constante.

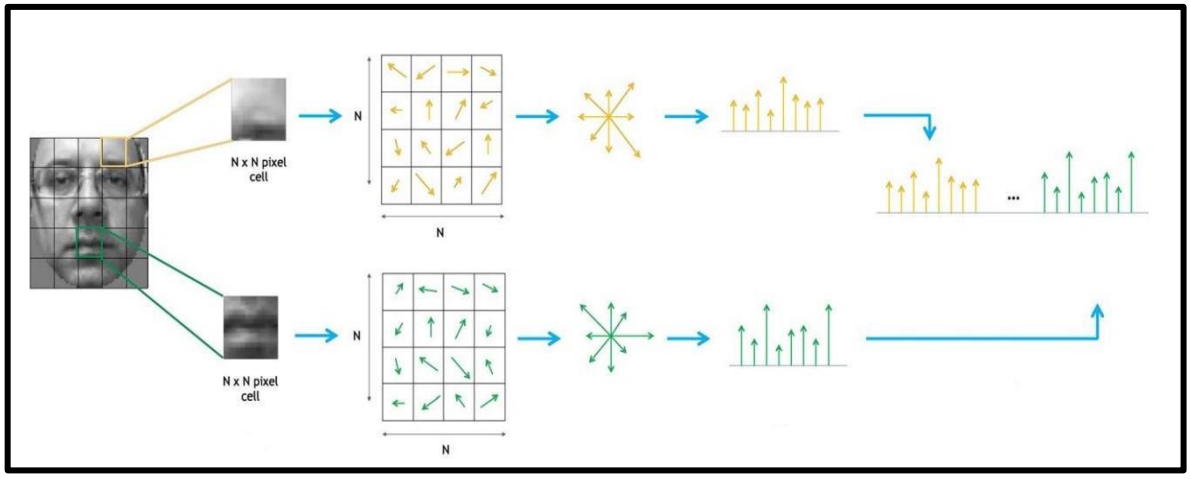


Figure 2.7 : Exemple d'extraction l'histogramme avec le descripteur **HOG** [42].

2.1.2. La classification

La classification automatique de l'image est le processus d'attribution automatique d'une classe à une image avec un système de classification. C'est ainsi que nous trouvons la classification des objets, des scènes, des textures, de la reconnaissance faciale.

La classification c'est l'étape finale dans le processus de vérification de lien de parenté, elle consiste à modéliser les paramètres extraits d'une modalité (le visage) des deux individus en se basant sur leurs caractéristiques communes. Un modèle est un ensemble d'informations utiles, discriminantes et non redondantes qui caractérise deux ou plusieurs individus ayant des similarités, ces derniers seront regroupés dans la même classe (classé les deux personnes en deux classes « classe parenté » ou « classe non-parenté »). Selon les caractéristiques extraites précédemment, les algorithmes de classification diffèrent, récemment beaucoup ont été proposées pour la classification dans la littérature, on trouve des techniques basées sur la mesure de la similarité ou du calcul de la distance tels que distance euclidienne, Minkowski et la métrique cosinus qui mesure la similarité de deux vecteurs de caractéristique.

Il y a d'autres méthodes se basent sur la classification des caractéristiques par un seul classificateur comme SVM « *Support Vector Machine* », KNN « *K-Nearest Neighbors* », toutes ces méthodes consistent à déterminer le modèle le plus proche des visages d'entrées parmi les images de la base de données.

a. Support Vector Machine (SVM)

La technique « *Support Vector Machine* (SVM) », également appelé Séparateur à Vaste Marge, développé par Vapnik en 1995 [43], est l'une des méthodes les plus populaires dans la famille des approches supervisé. La technique est destinée à résoudre des problèmes de classification ou de régression. Son but est de sépare linéairement avec un hyperplan optimal les données d'apprentissage (appartenant à des classes différentes) et maximiser la distance

entre les classes à partir d'un classificateur linéaire [44]. Cette technique est appliquée dans divers domaines, à savoir la reconnaissance des caractères, la reconnaissance d'images et le diagnostic médical, etc.

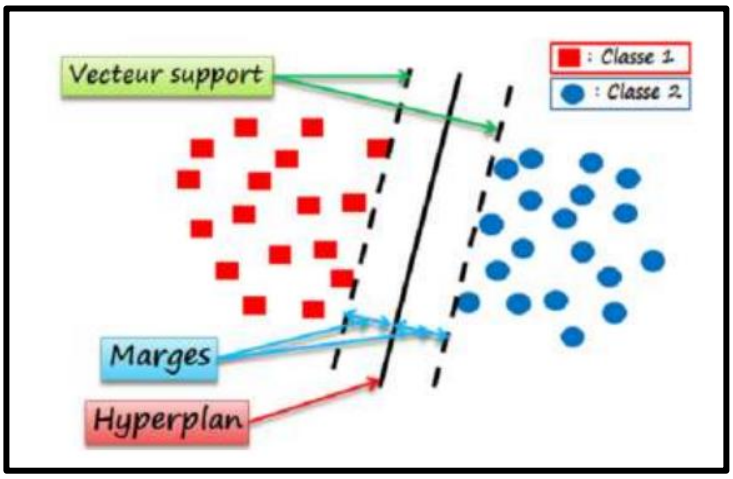


Figure 2.8 : Principe de la classification en utilisant la technique de SVM.

b. K-Nearest Neighbors (KNN)

L'algorithme « *K-Nearest Neighbors* (KNN) », également appelé K plus proche voisins (K-PPV) est un algorithme supervisé non paramétrique pour la classification [45]. Cet algorithme est largement utilisé pour plusieurs tâches d'intelligence en classification, segmentation d'images et vidéo. Le principe de l'algorithme est de regrouper les échantillons selon leur voisinage, c'est-à-dire que chaque point est affecté à la classe la plus représentée parmi ses k voisins les plus proches. Cet algorithme est basé sur les deux éléments principaux suivants :

- **K** : le nombre de cas les plus proches à utiliser.
- Une métrique pour mesurer le plus proche voisin.

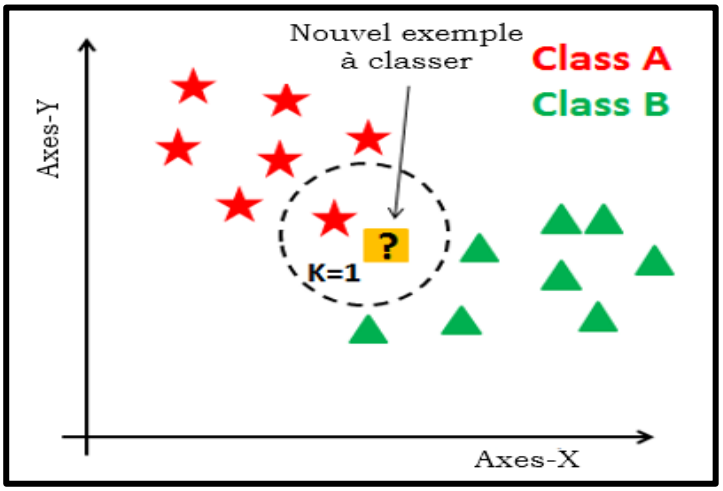


Figure 2.9 : La tâche de la classification en utilisant l'algorithme K-Nearest Neighbors (KNN). *Pour k = 1 la classe majoritaire du point central est la classe A* [46].

2.2. Les systèmes de reconnaissance basés sur l'apprentissage profond

L'apprentissage profond « Deep Learning » peut être considéré comme un sous-ensemble des algorithmes de Machine Learning (ML) et l'intelligence artificielle (IA). L'apprentissage profond est une technique d'apprentissage basée les réseaux de neurones artificiels ANN « *Artificiel Neural Networks* », ils sont conçus pour imiter la façon dont les humains pensent et apprennent. Ces réseaux de neurones artificiels, comprenant de nombreuses couches et stimulent l'apprentissage en profondeur pour aider les systèmes intelligents à comprendre les tâches de perception complexes avec une précision maximale. L'apprentissage profond définit ses propres paramètres de base, ce qui permet à la machine d'apprendre par elle-même sans le recours aux techniques classiques d'extraction des paramètres. Les techniques d'apprentissage profond sont utilisées dans la classification des images, la traduction des langues et la reconnaissance vocale...etc. Il peut être utilisé pour résoudre tout problème de reconnaissance de formes [47, 48].

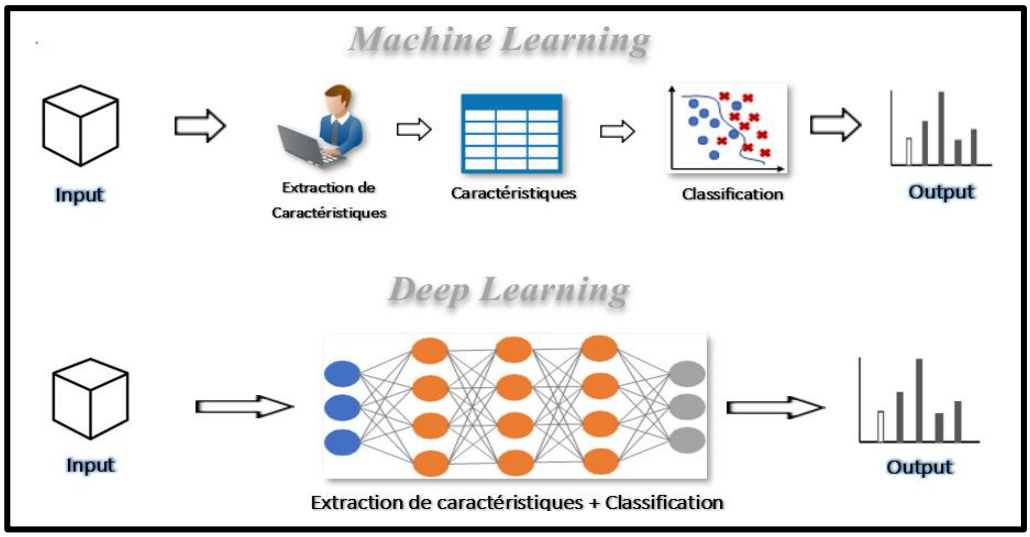


Figure 2.10 : Le procède du ML traditionnel comparé à celui du Deep Learning [49].

2.2.1. Fonctionnement de l'apprentissage profond

L'apprentissage profond se différencie de l'ensemble des algorithmes d'apprentissages de la Machine Learning par une architecture de réseaux de neurones artificiels multicouches complètement intégrés qui lui offre une meilleure autonomie dans son apprentissage. Car dans sa version « classique ». Les algorithmes de la Machine Learning nécessitent généralement des données préalablement traitées, pour son entraînement. Ces techniques peuvent donner des systèmes prédictifs, où leur exactitude dépend de la qualité des caractéristiques extraites. Les algorithmes d'apprentissage automatique « classique » ont besoin d'être guidés afin de pouvoir reconnaître les propriétés d'un élément donné. Au contraire, dans l'apprentissage profond, le

programme repère lui-même les caractéristiques nécessaires à son objectif. Ce qui le rend particulièrement pertinent pour l'apprentissage non supervisé et pour l'exploitation du grand volume de données [50].

Le principe d'apprentissage profond est sur l'entrainement sur un grand nombre d'exemples étiquetés afin de concevoir des modèles pour chaque classe dans l'espace d'étiquetage. Ainsi, après la mise à jour des différents paramètres de chaque couche neuronale, le système compare les résultats de prédiction obtenus pour un ensemble d'exemples de test par rapport aux étiquetages. Ainsi, avec un processus itératif, il ajuste les poids placés sur les différentes couches neuronales pour corriger son erreur en utilisant l'algorithme de rétropropagation « *back-propagation* ». Le processus est répété des milliers de fois jusqu'à l'atteint des performances optimales du système de reconnaissance automatique. Cette technique d'apprentissage est appelée « *supervised learning* » ou apprentissage supervisé. L'autre technique d'apprentissage non supervisé. Les réseaux de neurones doivent reconnaître des patterns au sein des ensembles de données pour apprendre par eux-mêmes quels éléments d'une photo peuvent être pertinents. La **figure 2.11** représente un exemple d'un système de reconnaissance automatique basée sur l'apprentissage profond avec un étiquetage trinaire : trois classes « Cat », « Dog » et « Raccoon ».

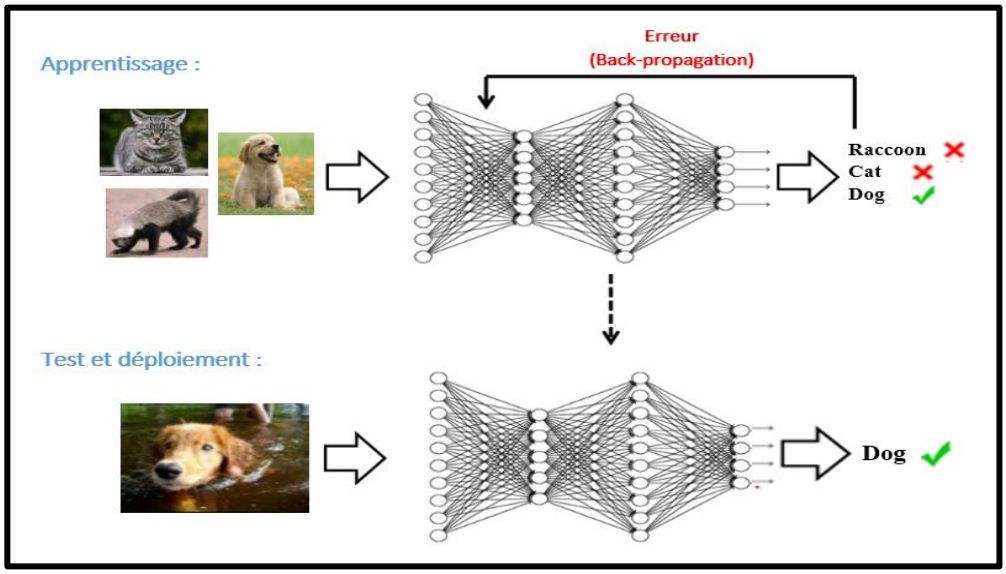


Figure 2.11 : Approche d'apprentissage en profondeur [51].

2.2.2. Les applications du l'apprentissage profond

L'apprentissage en profondeur ou plus communément appelé apprentissage hiérarchique est une discipline qui et évolue rapidement et prolifère et change notre façon de voir les technologies. Ainsi, il est encore plus difficile d'avoir une vision de ses futures applications. En particulier, Les principales avancées ont été dans l'application de l'apprentissage en profondeur à l'analyse

multimédia, y compris le traitement de l'audio et du langage naturel, traitement de l'image. Il a émergé dans de nombreux autres domaines, notamment les systèmes automatiques de diagnostics médicaux dans le domaine de la santé en général et dans celui de la bio-informatique pour l'étude de l'ADN, ainsi que dans le domaine de la sécurité informatique, etc. Dans la section suivante, nous citons certains domaines d'application des algorithmes d'apprentissage profond.

- Reconnaissance d'image.
- Traduction automatique.
- Traitement du langage naturel.
- La reconnaissance faciale.
- La reconnaissance vocale.
- La conduite autonome.
- Le diagnostic médical et les soins de santé (Health-care).
- La modération automatique des réseaux sociaux [47, 52, 53].

2.2.3. Les modèles de réseaux de neurones profonds (Architecteurs)

Il existe une variété d'architectures des réseaux de neurones basées apprentissage profond proposé dans les littératures. En effet, ces architectures dépendent étroitement par la complexité de la tâche et les caractéristiques des données d'entrées. Ces données d'entrées peuvent être des images isolées, des séquences d'images ou du vidéo, du texte et même des données dans un mode d'acquisition en temps réel. Pour cela, l'architecture de ces réseaux doit être adaptée selon les caractéristiques de données d'entrées. En général, nous trouvons trois grandes catégories d'architectures :

- ✓ Les réseaux neuronaux profonds **DNN** « Deep Neural Networks ».
- ✓ Les réseaux de neurones convolutifs **CNN** « *Convolutional Neural Network* ».
- ✓ Les réseaux de neurones récurrents **RNN** « *Recurrent Neural Network* ».
- ✓ Les réseaux de neurones récurrents à mémoire court-terme et long terme **LSTM** « *Long ShortTerm Memory* ».

Cependant, dans notre problématique les entrées d'un système de reconnaissance automatique des liens de parenté sont des paires d'images indépendantes (isolés). Ainsi, l'architecture d'apprentissage profond utilisée dans ces cas est les réseaux convolutif. Pour cela, nous intéressants dans les sections suivantes seulement sur l'architecture des réseaux de neurones convolutifs (CNN).

3. Les réseaux de neurones convolutifs (CNNs)

3.1. Principe

Les réseaux de neurones convolutifs (ou réseaux de neurones à convolutions, ou CNNs ou ConvNet) sont des types de réseaux de neurones artificiels acycliques « *feed-forward* ». Leur fonctionnement est inspiré par les processus biologiques, ils consistent en un empilage multicouche de perceptrons, dont le but est de prétraiter de petites quantités d'informations. Le nom « réseaux de neurones convolutifs » indique que le réseau emploie une opération linéaire mathématique entre matrices appelée convolution.

Les réseaux de neurones convolutifs sont actuellement parmi les modèles les plus performants pour la reconnaissance automatique des images. Ils comportent deux phases :

- La première est la phase de convolution, elle fonctionne comme un extracteur de caractéristiques des images à travers une succession de filtres, ou noyaux de convolution. Elle permet la création de nouvelles images intermédiaire appelée cartes de convolutions « *Feature maps* ». Au final, les cartes de convolutions sont alignées et concaténées dans des vecteurs de caractéristiques.
- La deuxième représente la phase de reconnaissance, elle est constituée de couches entièrement connectées « *Fully-connected* ». Le rôle de cette phase est la classification des entrées en se basant sur les résultats de la première phase. La sortie est généralement des vecteurs de probabilités selon l'étiquetage de données utilisées [54].

La **figure 2.12** présente une illustration du principe du réseau de neurones convolutifs. Les détails des éléments de chaque phase sont l'objectif des sections suivantes.

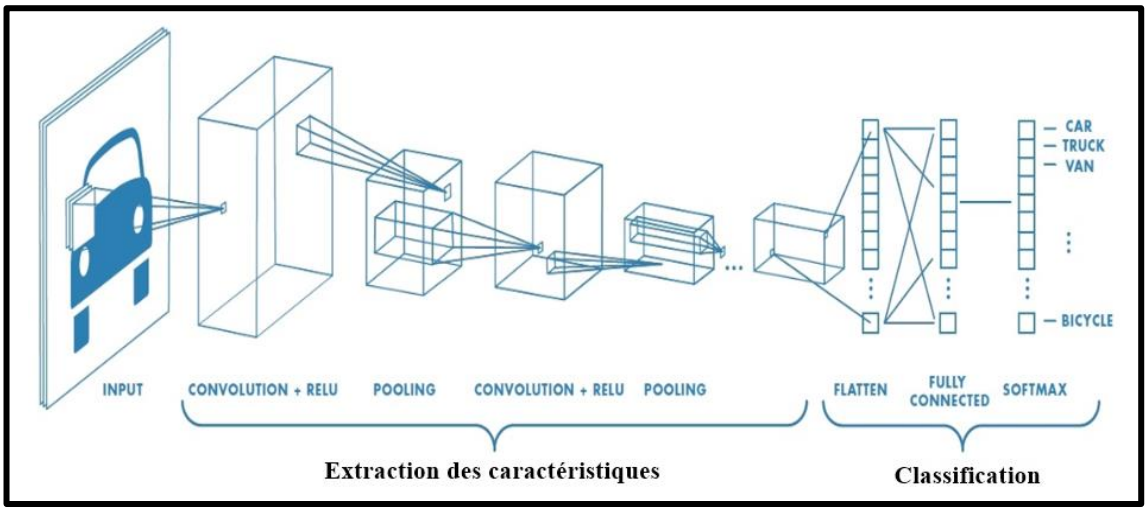


Figure 2.12 : Les deux parties qui composent l'architecture des réseaux neuronaux convolutifs [55].

3.2. La phase de convolution

La convolution est l'un des principaux éléments constitutifs d'un CNN. Elle fonctionne comme un processus de filtrage, c'est-à-dire appliquée un balayage sur les données d'entrée avec un filtre « *kernel* » de taille fixe. Pour chaque position du filtre, les valeurs des deux matrices en superposition (filtre et la partie de l'image à traiter) sont multipliées. Chaque valeur ainsi inférée est projetée dans une nouvelle matrice. Cette matrice représente une nouvelle image qui fait ressortir les caractéristiques recherchées au travers du filtre. La sortie de la convolution, que l'on peut appeler **«** carte de caractéristiques **»**, à des dimensions plus petites que celle de l'image en entrée [56]. La **figure 2.13** illustre le principe de convolution.

La phase de convolution pour l'architecture CNN est formée par un ensemble de couches de traitement en cascades. Dans la section suivante, nous présentons les éléments de cette phase :

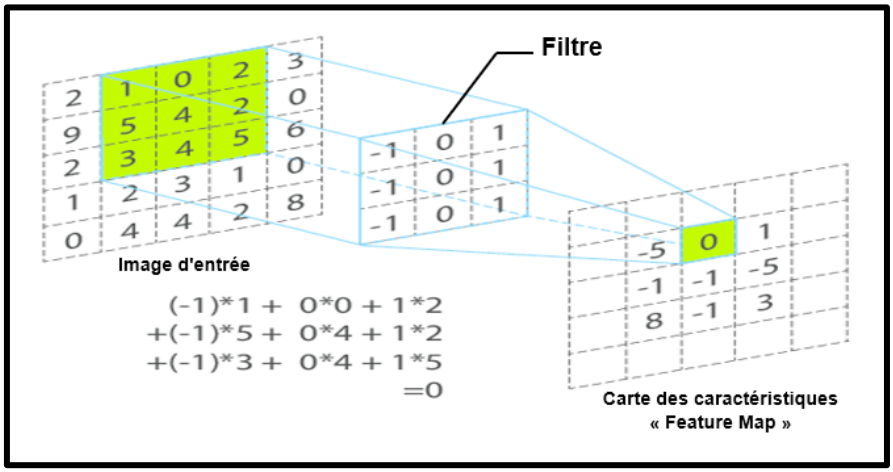


Figure 2.13 : Exemple de calcul de la convolution [57].

3.2.1. La couche de convolution (CONV)

La couche de convolution est la couche de construction de base d'un CNN, elle permet de créer des cartes de caractéristiques pour chaque paire (image, filtre). Elle permet de grader les informations discriminantes de chaque filtre appliqué à l'image. Trois paramètres essentiels pour la configuration de cette couche : la profondeur, le pas et la marge.

- **Profondeur de la couche (nombre de filtres) :** représente le nombre de filtres pour chaque couche de convolution. Elle représente la profondeur d'espace de recherche des caractéristiques discriminante pour les images d'entrées.
- **Le pas (stride) :** représente le pas de chevauchement des filtres dans l'image d'entrée. Il permet d'assurer la continuité des informations intra filtres pour une couverture globale des caractéristiques extraites.

- **La marge (zero-padding) :** permet de contrôler les dimensions de sortie des images filtrées pour chaque couche de convolution de sortie. En particulier, il est parfois souhaitable de garder la même dimension pour les images filtrées [58, 59].

3.2.2. La couche de pooling (POOL)

La couche de « Pooling » est souvent placée entre deux couches de convolution, elle reçoit en entrée plusieurs cartes de caractéristiques « Feature maps », et applique à chacune une opération de regroupement « pooling ». Il permet de réduire la taille de l'image filtrée tout en conservant les caractéristiques les plus importantes qu'elle contient, la **figure 2.14** illustre le principe de cette couche. Les deux fonctions les plus utilisées sont :

- **Max-pooling :** Elle garde la valeur la plus élevée de la zone définie.
- **Mean-pooling :** Elle garde la moyenne des valeurs de la zone définie.

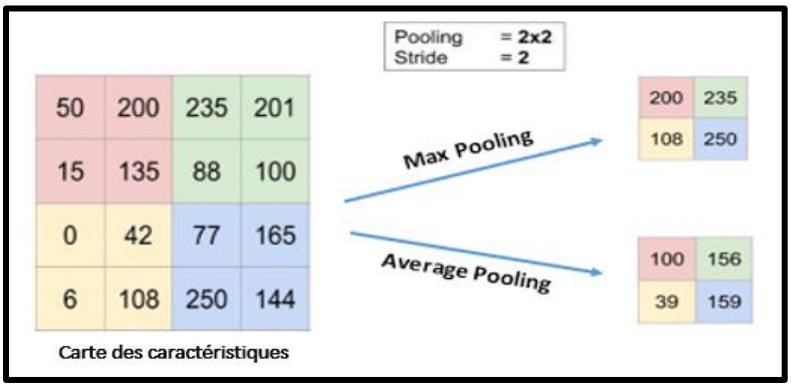


Figure 2.14 : Exemple avec types de pooling (avec un filtre "**2x2**" et un pas de "**2**") [60].

3.2.3. La couche d'activation

Comme toute architecture basée les réseaux de neurones artificiels, la fonction d'activation est le noyau de décision pour chaque couche de convolution et elle permet la correction des erreurs dans le processus de rétropropagation. En effet, il existe plusieurs fonctions d'activation, mais la plus utilisée dans la phase de convolution dans les architectures CNN est la fonction « Relu ».

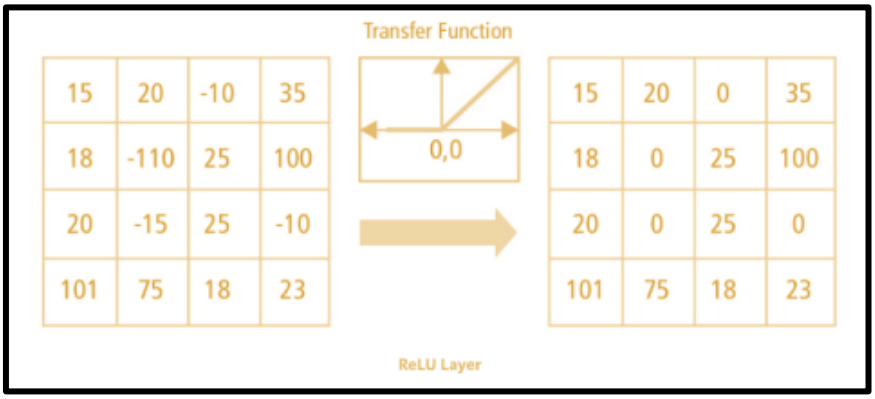


Figure 2.15 : La couche ReLU [61].

Cette fonction est définie par la fonction $f(x) = max\ (0, x)$. Appelée aussi « fonction d'activation non saturante », elle remplace toutes les valeurs négatives reçues en entrées par des zéros. Cependant, les autres fonctions utilisées sont :

- La fonction sigmoïde $f(x) = \ (1 + e\ - x)\ - 1$.
- La fonction tangente hyperbolique $f(x) = tanh(x)$.
- La fonction tangente hyperbolique saturante $f(x) = |tanh(x)|$ [58].

3.3. La phase d'évaluation ou reconnaissance

3.3.1. La couche de mise à plat « *flattening* »

Le « *flattening* » consiste à créer le vecteur de caractéristiques qui représente l'image d'entrée. C'est une étape de regroupement de toutes les images filtrées obtenues en deux dimensions à partir d'une image source « x » dans un espace d'une seule dimension. Le résultat de cette couche devient l'entrée du processus de reconnaissance ou de prédiction.

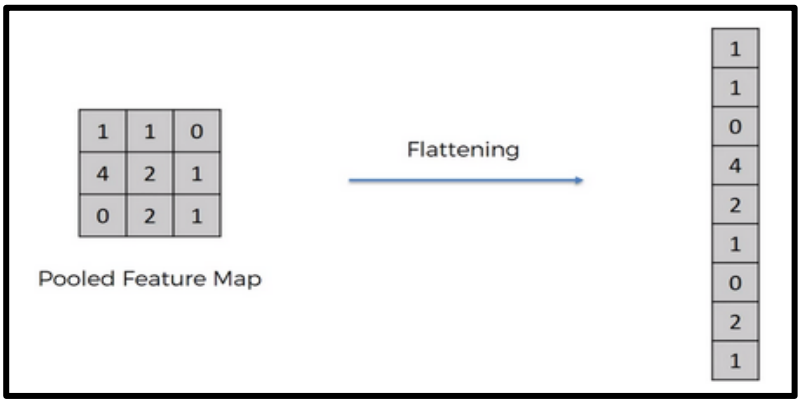


Figure 2.16 : Illustration du rôle de la couche flattening [61].

3.3.2. La couche dropout

La couche du dropout consiste à surmonter le problème de surapprentissage par la désactivation d'un ensemble aléatoire des connexions dans le réseau. Au moment de l'entraînement, l'opération consiste à l'abandon temporaire d'une unité choisie au hasard et de toutes ses connexions du réseau pendant l'entraînement. Ainsi, pour chaque nouvel élément d'entrée, le réseau sélectionne de manière aléatoire un sous-ensemble de neurones, formant une architecture de couche différente. Ces architectures utilisent des poids communs, mais étant donné que l'apprentissage ne dépend pas de neurones et de connexions spécifiques, la couche dropout peut aider à prévenir le surapprentissage « *Over fitting* » [62, 63].

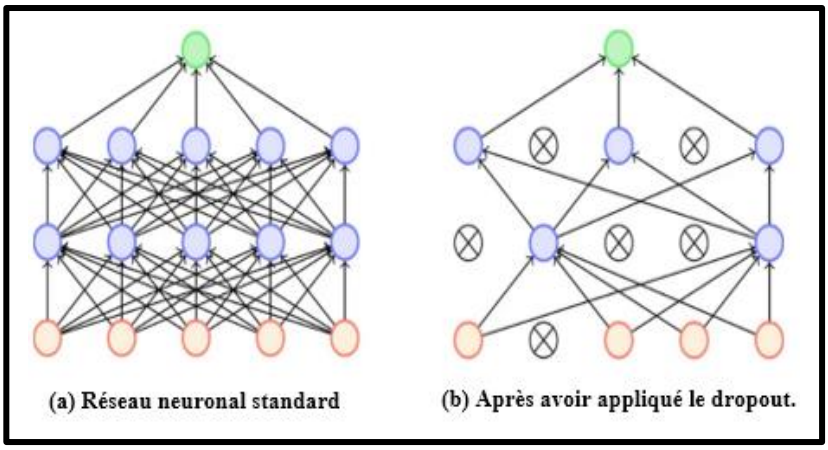


Figure 2.17 : Illustration de fonctionnement du dropout [64].

3.3.3. Couche entièrement connectée (Fully-Connected)

La couche entièrement connectée « dense » ou FC constitue toujours la dernière phase d'un CNN après plusieurs couches de convolution et de max-pooling. Les neurones d'une couche entièrement connectée ont des connexions à toutes les sorties de la couche précédente (comme on le voit régulièrement dans les réseaux de neurones réguliers).

Cette couche reçoit un vecteur plat en entrée résultant de l'étape de flattening et crée un nouveau vecteur en sortie. Pour ce faire, elle applique une combinaison linéaire puis éventuellement une fonction d'activation aux valeurs reçues en entrée. L'intérêt de la couche FC à classifier l'image lors de l'entrée du réseau, elle renvoie un vecteur de taille N, Où N est le nombre de classes. Chaque élément du vecteur indique la probabilité pour l'image en entrée d'appartenir à une classe [58].

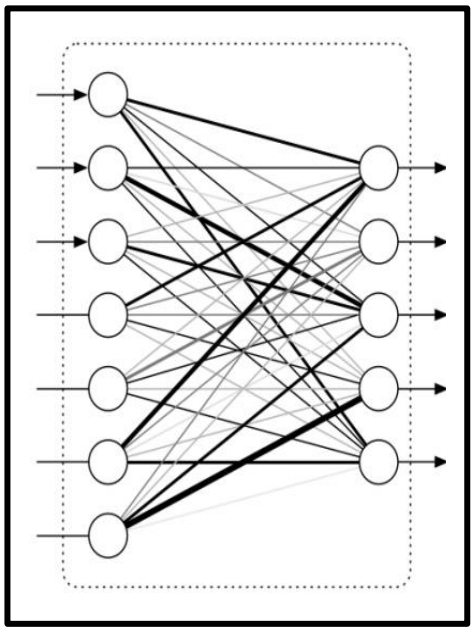


Figure 2.18 : Principe de la couche entièrement connectée (FC) [65].

3.4. Choix des hyperparamètres de CNNs

Malheureusement, tous les aspects définis dans l'architecture des CNN, il y a toujours une longue liste de paramètres qui doivent être définis manuellement pour permettre aux CNN d'avoir de meilleurs résultats. Les CNN utilisent plus d'hyperparamètres même si les règles habituelles pour les taux d'apprentissage et des constantes de régularisation s'appliquent toujours, il faut prendre en considération les notions suivantes :

- ✓ Pour chaque couche convolutive, combien de caractéristiques doit-on choisir ? (**Nombre de filtres**)
- ✓ Combien de pixels doit-on considérer dans chaque caractéristique ? (**Forme de filtres**)
- ✓ Pour chaque couche de pooling, quelle taille de fenêtre doit-on choisir ? Quel pas ?
- ✓ Pour chaque couche entièrement connectée supplémentaire, combien de neurones cachés doit-on définir ?

3.4.1. Nombre de filtres

Comme la taille des images intermédiaires diminue avec la profondeur du traitement, le nombre de filtres dans les couches proches de l'entrée est réduit, tandis que l'inverse se retrouve dans les couches plus proches de la sortie où elles sont plus grandes. Pour égaliser le calcul à chaque couche, le produit du nombre de caractéristiques et le nombre de pixels traités sont généralement choisis pour être à peu près constants à travers les couches. Pour préserver l'information en entrée, il est très important de conserver le nombre de sorties intermédiaires (c.-à-d. de multiplier le nombre d'images intermédiaires par le nombre de positions de pixels) afin d'augmenter (au sens large) d'une couche à l'autre. Le nombre d'images intermédiaires contrôle directement la puissance du système, dépend du nombre d'exemples disponibles et la complexité du traitement.

3.4.2. Forme de filtres

Les formes de filtre varient grandement dans la littérature. Ils sont souvent choisis en fonction de l'ensemble de données. Les meilleurs résultats sont ceux obtenus sur les images de MNIST (28x28) avec la taille de 5x5 sur la première couche, tandis que les ensembles de données d'images naturelles (souvent avec des centaines de pixels dans chaque dimension) ont tendance à utiliser de plus grands filtres de première couche de 12x12, et aussi 15x15 [66]. Donc, elle est relative à la tâche et la taille et le nombre des images en entrée et généralement elle est obtenue par les tests empiriques.

3.4.3. Forme de Max pooling

Les valeurs typiques sont 2 × 2. Des volumes d'entrée très importants peuvent justifier une mise en commun de 4 × 4 dans les couches inférieures (premières couches). Cependant, le choix de formes plus grandes réduira considérablement la dimension du signal et peut entraîner une perte excessive d'informations.

3.5. Quelques réseaux convolutifs célèbres (pré-entraînés)

Il est plus usuel d'utiliser des modèles de **CNNs** déjà entrainer et de les réadapter pour le problème de classification ou autre, c'est ce que l'on appelle l'apprentissage par transfert « *Transfer learning* ». Il s'agit de transférer la connaissance acquise et l'apprentissage d'un modèle traitant un problème de classification général vers un autre type de problème particulier. Il existe un grand nombre de réseaux **CNN** pré-entrainés, les plus connus sont :

3.5.1. LeNet

Les premières applications réussies des réseaux convolutifs ont été développées par *LeCun et al* dans les années 1990 [67]. Parmi ceux-ci, le plus connu est l'architecture « LeNet » utilisée pour lire les codes postaux, les chiffres, la reconnaissance de caractères manuscrits, etc.

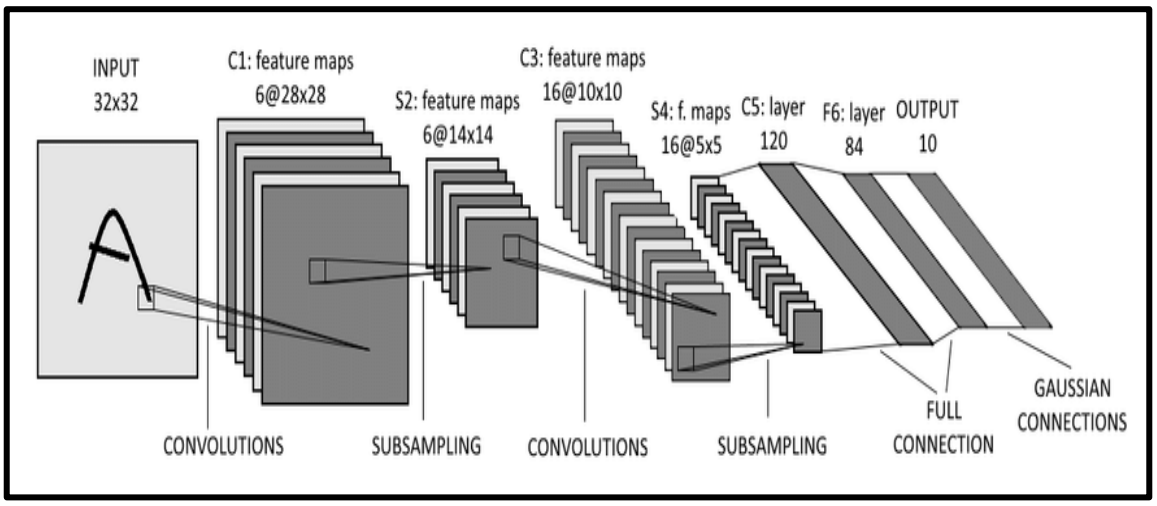


Figure 2.19 : Architecture de **LeNet** [68].

3.5.2. AlexNet

Le premier travail qui a popularisé les réseaux de neurones convolutifs dans la vision par ordinateur, développé par *Alex Krizhevsky et al* [69]. L'AlexNet a été soumis au challenge ImageNet ILSVRC en 2012 et a nettement surperformé tous ses concurrents précédents et a remporté le défi en réduisant l'erreur de top-5 de 26% à 15 ,3%. Le réseau avait une architecture très similaire à celle de LeNet, mais il était plus profond, plus grand et contenait huit couches ; les cinq premières étaient des couches de convolutions empilées les unes sur les autres et les trois dernières étaient des couches entièrement connectées et ils ont utilisé la fonction d'activation

non saturante de Relu, qui a montré une amélioration des performances d'entrainement. Il a été entraîné avec 60 millions de paramètres et 650 000 neurones pendant 5 à 6 jours sur deux GPUs GTX 580 de 3 Go.

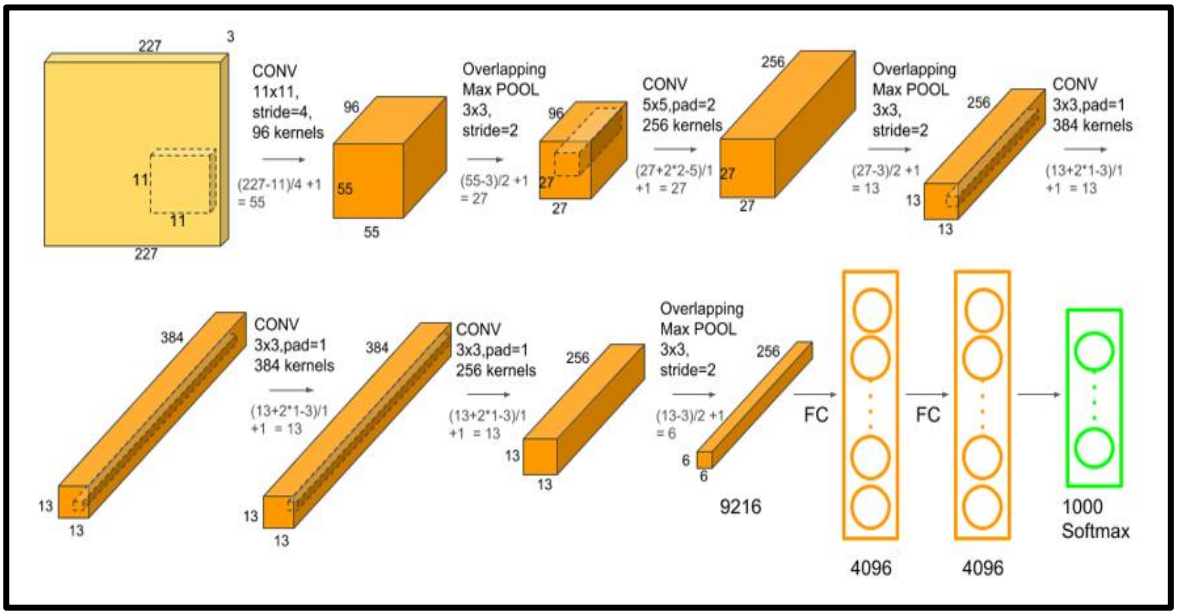


Figure 2.20 : Architecture de **AlexNet** [70].

3.5.3. VGGNet

C'est un réseau de neurones convolutif très performant dans le cadre du challenge de reconnaissance visuelle à grande échelle Image Net, où sa principale contribution a été de montrer que la profondeur du réseau est un composant essentiel pour de bonnes performances. VGGNet a été développé par le Visual Geometry Group de l'Université d'Oxford, d'où le nom VGG, en 2014 par *K. Simonyan et al* [71]. C'était l'un des modèles célèbres soumis à l'ILSVRC-2014 en atteignant le modèle à 92,7% de précision de test top-5 dans la base d'ImageNet. VGGNet, également connu sous le nom de VGG-16, se compose de 16 couches de poids ; 13 couches convolutives plus trois couches entièrement connectées et, de manière attrayante, présente une architecture extrêmement homogène, car il est très facile à comprendre et qui ne réalise que 3 x 3 convolutions et un regroupement 2x2 du début à la fin. VGG-16 a été formé pendant des semaines et utilisait des GPU NVIDIA Titan Black, et l'inconvénient de VGG est que ce réseau est généralement gros, plus coûteux à évaluer et utilise beaucoup plus de mémoire et de paramètres (environ 138 M).

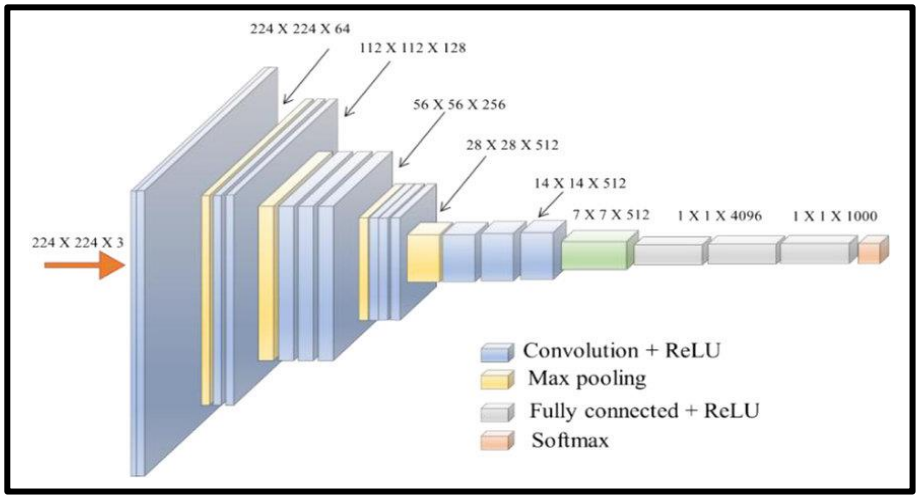


Figure 2.21 : Architecture de **VGGNet** [72].

3.5.4. ResNet

Le réseau *Residual Neural Network* (ResNet) développé par *Kaiming He et al* [73]. Il a été le gagnant de la 1ère place à la compétition de classification ILSVRC 2015. ResNet a introduit une architecture innovante avec des « sauts de connexions » et une lourde normalisation par lots. Ces connexions sont également appelées unités « *Gated* » ou unités « *Recurrentes gated* » et ont une forte similitude avec les éléments récents réussis appliqués dans les RNNs. L'intérêt de derrière la création ce type de réseaux de neurones convolutifs est de résoudre le problème de descente de gradients « *Vanishing* » dans les réseaux de neurones convolutifs très profond. Grâce à cette technique, ils ont pu former un réseau de neurones avec 152 couches et environ 60 millions paramètres, tout en ayant une complexité inférieure à celle de VGGNet. Il atteint un taux d'erreur de 3,57% parmi les cinq premiers (top-5), ce qui est supérieur à la performance au niveau humain sur le corpus utilisé.

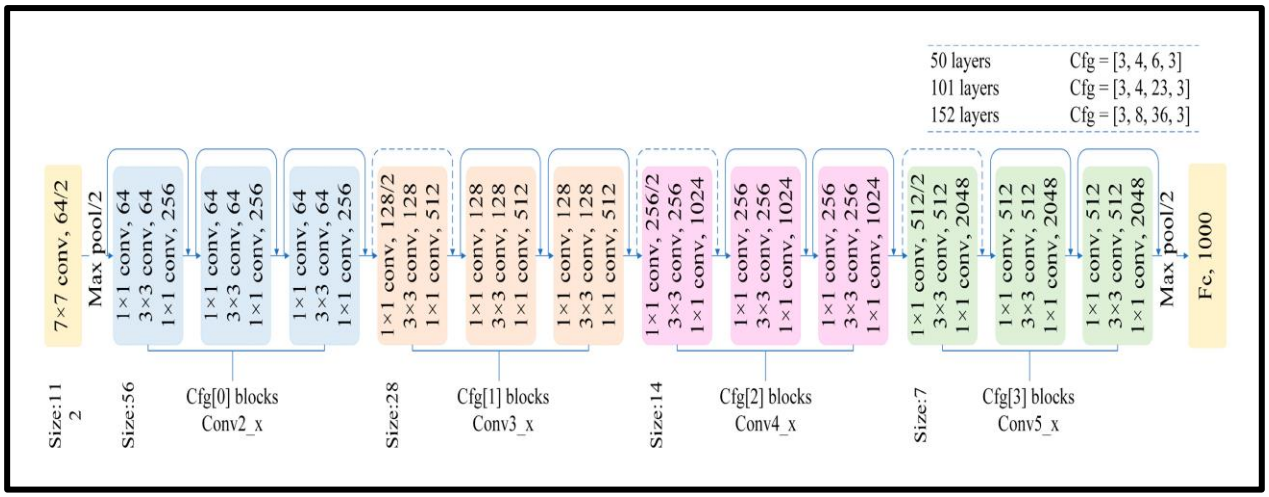


Figure 2.22 : Architecture de **ResNet** [74].

4. Mesures d'évaluation de performance des systèmes

Afin de mesurer les performances des systèmes de reconnaissance automatique, nous trouvons dans les littératures plusieurs métriques et courbes qui permettent l'évaluation de ces systèmes. Dans ce contexte, nous citons dans les sections suivantes les métriques utilisées dans les systèmes de vérification automatiques des liens de parenté.

4.1. Matrice de confusion

Cette matrice est définie dans le domaine de recherche d'information « *Information Retrieval* », elle permet d'évaluer des résultats de prédictions d'un classifieur par la comparaison des valeurs réelles d'étiquetage Y des données d'entrées avec les valeurs prédites Ŷ fournies par le modèle de classification. La matrice de confusion est une matrice dont les colonnes sont les classes obtenues et les lignes sont les classes désirées (Valeurs réelles) pour tout l'échantillon de test utilisé. Ainsi, la diagonale contient l'évaluation des échantillons bien classée (prédictions correctes), hors diagonale les éléments mal classés (prédictions incorrectes) comme le présente le Tableau 3.2 [75].

Classe réelle	**Classe prédite**	
	C1	C2
C1	***VP***	***FN***
C2	***FP***	***VN***

Tableau 2.1 : Matrice de confusion dans le cas d'une classification binaire.

Avec :

- **VP** sont les vrais positifs, c'est-à-dire les observations qui ont été classées positives et qui le sont réellement.
- **FP** sont les faux positifs, c'est-à-dire les individus classés positifs et qui sont en réalité des négatifs.
- De la même manière, les **FN** sont les faux négatifs et **VN** sont les vrais négatifs.

4.2. Courbe ROC (*Receiver Operating Characteristic*)

La courbe ROC est un graphe qui représente les performances d'un modèle de classification pour tous les seuils de classification. Cette courbe trace le taux des vrais positifs en fonction de celui des faux positifs. La courbe ROC montre ces taux pour différents seuils et plus cette courbe est éloignée de la première bissectrice, plus le modèle de classification à une forte capacité à discriminer. L'indicateur synthétique associé à la courbe ROC est la surface située sous la

courbe, c'est l'AUC « *Aera Under the Curve* ». Un modèle de classification est performant si l'AUC est proche de 1. À l'inverse, un modèle de classification n'est pas discriminant si l'AUC est proche de 0,5.

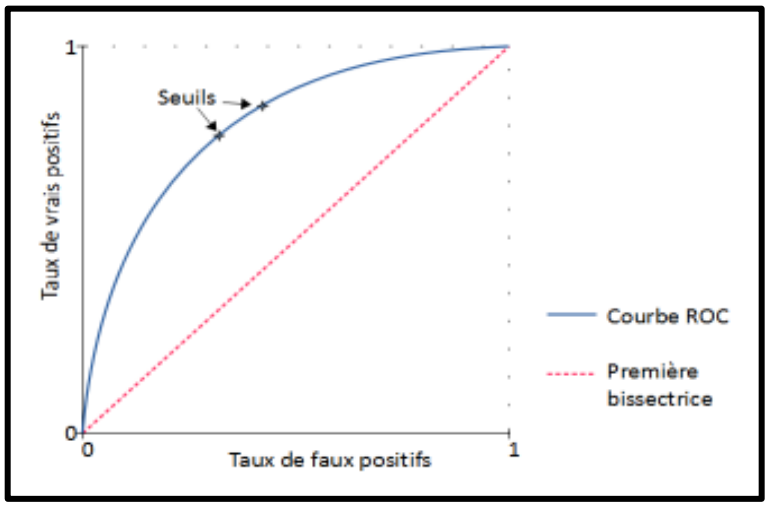


Figure 2.23 : Courbe ROC (Receiver Operating Characteristic) [76].

4.3. Les métriques d'évaluation

Il existe plusieurs métriques numériques d'évaluation utilisées dans les systèmes de reconnaissance automatiques, parmi eux nous citons :

- **Accuracy :** désigne la proportion des prédictions correctes effectuées par le modèle de classification.

$$\mathbf{Accuracy} = \frac{\mathbf{VP} + \mathbf{VN}}{\mathbf{VP} + \mathbf{VN} + \mathbf{FP} + \mathbf{FN}}$$

- **Précision :** de ceux que l'algorithme a pu classer, combien sont corrects.

$$\mathbf{Précision} = \frac{\mathbf{VP}}{\mathbf{VP} + \mathbf{FP}}$$

- **Rappel :** de ceux qui existent, combien l'algorithme a pu trouver.

$$\mathbf{Rappel} = \frac{\mathbf{VP}}{\mathbf{VP} + \mathbf{FN}}$$

- **F-mesure :** moyenne harmonique entre la précision et le rappel.

$$\mathbf{F-mesure} = \frac{2 * (\mathbf{Précision} * \mathbf{Rappel})}{(\mathbf{Précision} + \mathbf{Rappel})}$$

5. Travaux connexes pour la vérification automatique des liens de parenté

Les systèmes de reconnaissance des liens de parenté fonctionnent sur deux modes : la vérification et l'identification. Cependant, il existe également des systèmes basés uniquement sur la vérification de parenté, c'est-à-dire sans définir la classe de la relation de parenté entre les deux images qui sont traitées. En effet, la majorité des travaux proposés dans les littératures qui seront cités dans les sections suivantes traitent le problème de vérification de l'existence des liens de parenté.

Entre autres, les architectures d'apprentissage automatique et profond ont révélé leurs efficacités dans de nombreux problèmes de reconnaissance automatique complexes avec l'utilisation des algorithmes intelligents et de réseaux de neurones artificiels pour extraire et apprendre des informations discriminantes. Par conséquent, divers travaux et contributions anciennes et récentes ont été proposés comme une solution à ce sujet de la vérification de la parenté à partir d'images faciales. Ces travaux peuvent être classés en trois approches principales : basés sur les caractéristiques, basés sur l'apprentissage profond et basés sur l'apprentissage des métriques. Dans ce qui suit, nous en avons mentionné quelques travaux récents sur chaque approche.

5.1. Les systèmes basés sur les caractéristiques locales

Lu et al [77]**,** ont proposés une nouvelle méthode d'apprentissage de caractéristiques discriminantes (PDFL) basée sur un prototype pour la vérification de parenté, vise à apprendre des caractéristiques discriminantes de niveau intermédiaire pour mieux caractériser la relation des images de visage pour la vérification de parenté, contrairement à des méthodes qui utilisent des descripteurs de caractéristiques artisanaux de bas niveau. Dans ce travail, ils ont appliqué trois descripteurs de caractéristiques différents, pour extraire des informations différentes de chaque image faciale, avec les deux bases « KinFaceW-I » et « KinFaceW-II ». Ils ont enregistré les résultats suivants : les caractéristiques du descripteur SPLE ont donné de bien meilleurs résultats que l'autre descripteur (LBP, SIFT) avec une précision de 65,6% pour la base « KinFaceW-I ». Tandis que, pour la base « KinFaceW-II », ils ont obtenu une précision de 76.4% comme le résultat le plus élevé ainsi avec les caractéristiques SPLE et l'autres caractéristiques SIFT, LBP ils ont enregistré les taux de reconnaissance 64% et 69.9% respectivement. Ainsi, afin d'utiliser plusieurs caractéristiques de bas niveau pour l'apprentissage des caractéristiques de niveau intermédiaire, ils ont proposé en outre une méthode PDFL multi-vue « MPDFL » afin d'améliorer les performances de vérification. Ils ont enregistré meilleur score 70.1% pour la base « KinFaceW-I » et 77% avec la base « KinFaceW-II ».

Liu et al [78]**,** ont proposé une méthode innovante de vecteur de caractéristique de Fisher héréditaire « IFVF » pour la vérification de parenté. Cette approche proposée vise à découvrir les relations génétiques entre les enfants et les parents afin de conserver à la fois la capacité discriminante et la capacité héréditaire. Plus précisément, le vecteur de Fisher est dérivé pour chaque image en agrégeant les caractéristiques de descripteur SIFT échantillonnées de manière dense dans l'espace de couleur de l'adversaire, puis une transformation héréditaire est apprise, qui maximise la similitude entre les images de parenté tout en minimisant celle entre les images de non-parenté pour chaque paire d'images simultanément. Enfin, ils ont utilisé une mesure de similitude cosinus pour le calcul de la similarité entre les vecteurs. La méthode proposée est évaluée sur les deux bases « KinFaceW-I » et « KinFaceW-II » où les résultats expérimentaux ont montré les taux de vérification de 73.45% et 81.60% respectivement.

Duan et al [79]**,** ont proposé une méthode de soustraction de caractéristiques pour supprimer la partie sans lien de parenté de la caractéristique locale pour la vérification de la parenté. Ceci est fait en utilisant une méthode de descente de gradient pour réduire simultanément la distance caractéristique entre les paires d'images de visage avec la parenté et maximiser la distance entre les paires de non-parenté. Sur la base de la caractéristique soustraite, la vérification est réalisée au moyen de comparaison de distance « Gaussian ». Les tests expérimentaux ont montré parmi les trois descripteurs : LPQ, SIFT et LBP utilisées dans l'étude, que le descripteur LPQ a donné les meilleurs résultats avec des taux de vérification de 70.9% et 77,1% sur les bases de données « KinFaceW-I » et » KinFaceW-II » respectivement.

Xu et al [80]**,** ont suggéré un modèle de similarité robuste pour apprendre une mesure de similarité linéaire clairsemée selon lequel le score de similarité de chaque paire d'images de visage positive est supérieur au score de similarité moyen de certaines paires d'images de visage négatives. Ils ont aussi développé un algorithme d'apprentissage de la similitude en ligne « OSL-A » pour une application plus évolutive. Ils ont évalué empiriquement la méthode proposée sur les bases « KinFaceW-I » et « KinFaceW-II » où ils ont également appliqué deux descripteurs de visage HOG et SIFT, pour extraire les traits du visage, les résultats étaient comme suit : Avec la base « KinFaceW-I », avec les caractéristiques du descripteur HOG le score est 77.92% ainsi avec les caractéristiques SIFT le score est 76.35%. Pour la base « KinFaceW-II », ils ont enregistré les scores 77.10% pour les caractéristiques HOG et de 74.90% pour les caractéristiques SIFT.

Alirezazadeh et al [81], ont suggéré une méthode basée sur la détection des parties principales du visage telle que les yeux, le nez et la bouche avec l'image complète du visage, comme une première étape. Ces parties seront utilisées pour l'extraction des caractéristiques :

LBP, SPLE et LPQ. Ensuite, pour réduire la redondance des vecteurs de caractéristiques et améliorer le taux de vérification, la méthode de sélection de caractéristiques nommer KinFS « *Kinship Feature Selection* » est proposée pour la sélection des caractéristiques discriminantes. Cette méthode est basée sur l'algorithme RSFS « *Random Subset Feature Selection* ». Pour la classification, ils ont utilisé le classifieur SVM, pour déterminer la relation de parenté ou de non-parenté. Les résultats obtenus ont enregistré les taux de 78.9% et 84,5% sur les ensembles de données « KinFaceW-I » et « KinFaceW-II » respectivement.

5.2. Les systèmes à base d'apprentissage profond

Zhang et al [18], proposent d'extraire les caractéristiques de haut niveau pour la vérification de parenté basée sur des réseaux de neurones convolutifs profonds (**CNN**). Compte tenu de l'importance des points clés faciaux, ils ont également extrait des caractéristiques basées sur les points clés pour la vérification de la parenté. Avec les bases « KinFaceW-I » et « KinFaceW-II », les résultats sont 74.8% et 77.5% pour la première et 85.3%, 88.4% pour la deuxième base. Ainsi, ils ont montré que leur approche proposée a apporté des améliorations significatives par rapport aux méthodes précédentes.

Lu et al [82]**,** ont présenté une méthode discriminante d'apprentissage métrique profond « **DDML** » pour la vérification du visage et de la parenté, l'objectif de cette méthode est de capturer la variété non linéaire. Ils ont utilisé un réseau de neurones profond afin d'apprendre un ensemble de transformations non linéaires hiérarchiques. Pour chaque image de visage de la base de données « KinFaceW-I »et « KinFaceW-II », ils ont extrait quatre descripteurs de caractéristiques. Avec la base « KinFaceW-I », le résultat le plus élevé était obtenu avec les caractéristiques LPQ avec une précision de 81.4% et les autres caractéristiques : DSIFT, HOG et LBP ont donné moins avec 78.4%, 77.5% et 75.5% respectivement. Concernant la base « KinFaceW-II », qui a atteint presque les mêmes résultats en obtenant également la plus grande précision de 82.2% avec les caractéristiques LPQ, puis suivie par les caractéristiques DSIFT avec 79.1% et HOG, LBP avec 78.1% et 77.6% respectivement. Ils ont aussi développé une méthode d'apprentissage multidimensionnel profond discriminant (DDMML) pour apprendre conjointement plusieurs réseaux de neurones et qui a donné comme la meilleure précision de 83.5% avec « KinFaceW-I » et 84.3% avec « KinFaceW-II ».

5.3. Les systèmes basés sur l'apprentissage des métriques

Yan et al [83], ont proposé une méthode d'apprentissage métrique repoussé du voisinage (NRML) pour la vérification de parenté, la mesure proposée vise à apprendre une mesure de distance sous laquelle les échantillons intra-classes (avec une relation de parenté) sont

rapprochés le plus possible et pousser des échantillons inter-classes (sans relation de parenté) dans un voisinage, simultanément, de sorte que des informations plus discriminantes peuvent être exploitées pour la vérification. Les résultats des tests empiriques sur les deux bases « KinFaceW-I » et « KinFaceW-II » obtenus avec le classificateur SVM portaient à quatre caractéristiques : avec la première base, les caractéristiques SPLE ont donné une précision de 64.3%, SIFT, LBP et TPLBP ont donné de moins de résultats avec une précision de 63.8%, 63.7% et 62.9% respectivement. Cependant, avec la seconde base la plus grande précision est de 75.7% avec les caractéristiques SPLE, et les caractéristiques TPLBP, LBP et SIFT ont donné une précision de 70.1%, 69.5% et 62.8% respectivement. Leur objectif était aussi la meilleure utilisation de plusieurs descripteurs de caractéristiques afin d'extraire des informations complémentaires, et pour cela ils ont proposé une autre méthode NRML multi-vue « MNRML » pour rechercher une métrique de distance commune pour effectuer une fusion des caractéristiques multiples afin d'améliorer les performances de vérification de parenté, où il a donné de bons résultats avec une précision de 76.5% pour la base « KinFaceW-II » et de 69.9% pour la base « KinFaceW-I ».

Yan et al dans [84], ont proposé est une méthode d'apprentissage multi métrique discriminante « DMML » pour la vérification de parenté via l'analyse d'images faciales afin d'apprendre plusieurs métriques de distance avec des multiples caractéristiques extraites par des différents descripteurs de caractéristiques : LBP, SPLE et SIFT. Elle prouve son efficacité par rapport aux méthodes d'apprentissage mono-métriques et multi-métriques existantes. La méthode proposée avec le classifieur SVM était plus efficace qu'avec d'autres classifieur comme KNN. Pour les résultats obtenus, ils ont enregistré des taux de vérification de 72.25%, 70.62% et 69.5% respectivement pour la base « KinFaceW-I » ont donné. Tandis que pour la base « KinFaceW-II », ils ont enregistré les taux 78.25%, 76.88% et 74,75% respectivement.

Kou et al [85]**,** ont proposé une nouvelle approche de vérification de la parenté en apprenant une mesure de similitude clairsemée d'une manière en ligne, qui se concentre sur l'apprentissage d'une mesure de similitude génétique pour augmenter l'évolutivité pour les applications pratiques avec une quantité sans cesse croissante de données. Ils ont mené des expériences pour vérifier l'efficacité de leur approche les bases de données de parenté disponibles « KinFaceW-I » et « KinFaceW-II ». Pour les représentations caractéristiques de l'image du visage, ils ont utilisé trois descripteurs : SIFT, LBP, et SPLE et ils ont donné les résultats suivants : Avec la base « KinFaceW-I », le résultat le plus élevé était obtenu avec le descripteur SPLE avec une précision de 65.7% et les autres descripteurs : SIFT et LBP ont donné moins avec 65.5% et 63.5% respectivement. Concernant l'autre base « KinFaceW-II », qui a atteint presque les mêmes

résultats en obtenant également la plus grande précision de 74.8% avec le descripteur SPLE, puis suivie par les descripteurs LBP et SIFT avec 68.9% et 66.7% respectivement.

Zhou et al [86]**,** ont proposé une technique d'apprentissage de similarité d'ensemble « ESL », elle est fondée sur une fonction de similarité bilinéaire clairsemée pour modéliser les caractéristiques relatives aux images faciales avec des liens de parenté. La proposition exploite l'ensemble des paires d'images qui ont des liens de parenté en générant un ensemble de modèles de similitude afin d'accroitre la capacité de généralisation du système. Pour tester leur méthode, ils ont utilisé les bases « KinFaceW-I » et « KinFaceW-II ». Ils ont choisi deux descripteurs de caractéristiques : LBP et HOG pour extraire les caractéristiques discriminantes. Le taux de vérification atteint par les caractéristiques HOG a dépassé le taux de LBP avec 78.6% et 74.1% respectivement pour la base « KinFaceW-I » et les taux de 75.7% et 74.3% pour la base « KinFaceW-II ».

Hu et al, dans [87], ont présenté une méthode d'apprentissage multi-métrique à grande marge « LM^3L » pour la vérification du visage et de la parenté, qui apprend conjointement plusieurs métriques de distance globale en vertu desquelles les corrélations des différentes représentations de caractéristiques de chaque échantillon sont maximisées. Les expériences sur la base « KinFaceW-II » et avec les deux descripteurs : LBP, SIFT ont enregistré un taux de vérification de 78.7%. Pour mieux exploiter les structures locales des images de visages, ils ont proposé également une métrique locale d'apprentissage « LML » et une méthode d'apprentissage multi-métrique à grande marge locale « L^2M^3L » pour apprendre un ensemble de métriques locales, où il atteint un taux de vérification de 80.0%.

Jiwen Lu et al [88], ont proposé une méthode d'apprentissage métrique de parenté « KML » avec un modèle couplé de réseau neuronal profond « DNN ». KML apprend une métrique de similitude profonde couplée de sorte que les paires d'images avec la relation de parenté sont rapprochées, tandis que celles sans relation de parenté sont poussées aussi loin que possible, et les résultats montrent la supériorité de ce modèle. Ils ont extrait LBP, HOG, SIFT, VGG comme caractéristiques, des deux bases « KinFaceW-I » et « KinFaceW-II ». Ils ont obtenu les résultats suivants : Les caractéristiques VGG ont donné un résultat beaucoup mieux que d'autres caractéristiques : HOG, SIFT et LBP avec un taux de vérification 82.8% contrairement au 78.9%, 78.7% et 76.6% respectivement, et c'est ce qu'elle a donné à partir résultats de la base « KinFaceW-I ». Alors que la base « KinFaceW-II » a donné, comme résultats et comparaisons entre eux, avec les caractéristiques VGG, ont a également donné le meilleur résultat avec un taux de vérification de 85.7%, par rapport à d'autres caractéristiques : SIFT, LBP et HOG qui a donné moins avec un taux de 81%, 80.6% et 80.3%, respectivement.

Classe	Méthode	Caractéristiques	Classifieur	Base de données	Taux de verification (%)				
					F-S	F-D	M-S	M-D	Moyennes
Caractéristiques	PDFL & MPDFL [77]	LBP+SPLE+ SIFT	SVM	KinFaceW-I	68.2	63.5	61.3	**69.5**	65.6
					73.5	67.5	66.1	73.1	**70.1**
				KinFaceW-II	77.0	74.3	77.0	**77.2**	76.4
					77.3	74.7	77.8	**78.0**	**77.0**
	IFVF [78]	SIFT	Cos-sim	KinFaceW-I	73.3	71.7	71.1	**77.5**	73.4
				KinFaceW-II	**85.6**	75.4	82.8	82.6	**81.6**
	A feature subtraction method [79]	LBP+LPQ+ SIFT	Distance euclidienne (Gaussian)	KinFaceW-I	**75.4**	63.8	69.9	74.6	70.9
				KinFaceW-II	**82.4**	76.2	76.6	73.2	**77.1**
	OSL-A [80]	HOG+SIFT	SVM	KinFaceW-I	**82.7**	75.0	71.8	82.3	**77.9**
				KinFaceW-II	**82.0**	77.2	75.6	73.6	77.1
	Feature in High-dimensional (KinFS) [81]	LPQ+SPLE+ LBP	SVM	KinFaceW-I	77.8	73.5	79.6	**84.6**	**78.9**
				KinFaceW-II	**87.5**	79.8	83.9	86.7	**84.5**
Apprentissage profond	CNN [18]	CNN	CNN	KinFaceW-I	76.1	71.8	78.0	**84.1**	**77.5**
				KinFaceW-II	89.4	81.9	89.9	**92.4**	**88.4**
	DDML & DDMML [82]	LPQ+DSIFT+ LBP+HOG	SVM	KinFaceW-I	83.8	77.0	78.1	**86.6**	81.4
					86.4	79.1	81.4	**87.0**	**83.5**
				KinFaceW-II	**84.8**	82.6	79.4	81.8	82.2
					87.4	83.8	83.2	83.0	**84.3**
Apprentissage métrique	NRML & MNRML [83]	LBP+SPLE+SIFT +TPLBP	SVM	KinFaceW-I	64.1	59.1	63.9	**71.0**	64.3
					72.5	66.5	66.2	72.0	**69.9**
				KinFaceW-II	76.8	73.1	76.8	**77.0**	75.7
					76.9	74.3	77.4	**77.6**	**76.5**
	DMML [84]	LBP+SPLE+ SIFT	SVM	KinFaceW-I	74.5	69.5	69.5	**75.5**	**72.2**
				KinFaceW-II	78.5	76.5	76.5	**79.5**	**78.2**
	Learning a Genetic Measure [85]	SPLE+SIFT+ LBP	SVM	KinFaceW-I	66.1	62.2	64.3	**70.0**	65.7
				KinFaceW-II	74.9	71.0	**76.9**	76.4	**74.8**
	ESL [86]	HOG+LBP	KNN	KinFaceW-I	**83.9**	76.0	73.5	81.5	**78.6**
				KinFaceW-II	**81.2**	73.0	75.6	73.0	**75.7**
	LMMML & LLMMML [87]	LBP+SIFT	SVM	KinFaceW-II	**82.4**	74.2	79.6	78.7	78.7
					82.4	78.2	78.8	80.4	**80.0**
	KML [88]	LBP+HOG+ SIFT+VGG	KNN	KinFaceW-I	83.8	81.0	81.2	**85.0**	**82.8**
				KinFaceW-II	**87.4**	83.6	86.2	85.6	**85.7**

Tableau 2.2 : Etude comparatifs entre les performances des travaux susmentionnés avec les bases **KinFaceW-I** et **KinFaceW-II**.

6. Conclusion

Dans ce chapitre, nous avons présenté les techniques d'apprentissage utilisées dans les systèmes de reconnaissance automatique pour les images. En effet, la vérification automatique des liens de parenté est une application de ces systèmes. Nous avons présenté les deux grandes catégories de ces systèmes : les systèmes qui utilisent l'architecture dite classique en deux phases : extraction de caractéristiques et reconnaissance ou classification. Ainsi que les systèmes basés sur l'apprentissage profond. Ensuite, nous avons effectué une étude bibliographique sur les travaux effectués dans cette problématique avec un tableau de synthèse.

Cependant, les résultats obtenus dans ces travaux montrent que l'utilisation des architectures CNN est prometteuse tout en tenant compte des contraintes de calculs et espace mémoire. Car la particularité des systèmes de vérification automatique des liens de parenté est dans leurs données d'entrées avec une paire d'images ce qui rend la tâche plus complexe. En revanche, les plateformes de développement en cloud comme *Google Colab* ou *Kaggle* nous encouragent d'utiliser ces architectures dans notre problématique. Dans le chapitre suivant, nous présentons notre contribution pour un système de vérification automatique des liens de parenté.