

République Algérienne Démocratique et Populaire
Ministère de l'Enseignement Supérieur et de la Recherche Scientifique
Université DJILALI BOUNAAMA
KHEMIS MILIANA



Faculté des Sciences et de

la Technologie

Département de Mathématique et d'informatique

Mémoire Présenté

Pour l'obtention de diplôme

MASTER en Informatique

Option : Génie logiciel et système distribué

Thème :

Les réseaux de neurone convolutionnel (CNN) pour la classification des images associées aux places de stationnement d'un parc de véhicule

Réaliser par :

GHENNAM Bochra

SMARA Safia

Encadré par :

Mr. KHALFIA

Jury:

Mr.HARBOUCHE.....Président

Mr.BOUKADOUM.....Examineur1

Mme.KESSAISSIA.....Examineur2

Année Universitaire 2018/2019

Remerciements

*Nous remercions Allah qui nous a aidés à
réaliser
ce travail*

*Nous remercions également notre encadreur
KHALFI Ali pour
l'aide et les conseils concernant ce travail*

*Nous tenons aussi à remercier les membres du
jury pour avoir accepté
d'examiner et d'évaluer ce travail.*

dédicace

A mes parents ma mère et mon père pour

Leur

patience, leur amour, leur soutien et leur

encouragement

A mes chers frères qui sont ma fierté

A mes sœurs

A mon fiancé

A ma famille et ma belle famille

A mes amis avec qui j'ai vécu les plus belles expériences

De ma vie

A tous ceux qui ont toujours cru en moi et poussé à réussir,

je vous dédie

ce travail.

Bochra

dédicace

A mes parents ma mère et mon père pour

Leur

patience, leur amour, leur soutien et leur

encouragement

A mes chers frères qui sont ma fierté

A mes sœurs

*A ma famille et mes amis avec qui j'ai vécu les plus belles
expériences*

De ma vie

tous qui mon aide de près ou de loin à réaliser

je vous dédie

ce travail.

safia

Liste des tableaux.....	I
Liste des figures.....	II
Introduction générale.....	1
Résumé.....	III
Chapitre I : Notion de base sur les images	
I.1. Introduction.....	4
I.2. Notions de bases.....	4
I.2.1. Définition d'une image.....	4
I.2.2. Les types d'image.....	4
I.2.3. Caractéristiques de l'image.....	5
I.2.3.1 Pixel.....	5
I.2.3.2 Résolution.....	5
I.2.3.3. Dimention.....	5
I.2.3.4. Bruit.....	6
I.2.3.5 Voisinage.....	6
I.2.3.6. Contraste.....	6
I.2.3.7. Luminance.....	7
I.2.3.8 Contour.....	7
I.2.4. Les caractéristiques visuelles de l'image	7
I.2.4.1. Couleur.....	7
I.2.4.1.1 l'histogramme de couleur.....	7
I.2.4.1.2 les espaces de couleur.....	8
I.2.4.1.3. les moments statistiques.....	8
I.2.4.2. Texture.....	9
I.2.4.2.1. la matrice de cooccurrences.....	9
I.2.4.3. La forme.....	10
I.2.5. Vecteur descripteur.....	10
I.2.6. Mesure se similarité.....	10
I.2.7. Distance de similar.....	11
I.2.7.1. La distance euclidienne.....	11
I.2.7.2. La distance de manhattan.....	11
I.2.7.3. la distance de tchebychev.....	11
I.2.8. Recherche par similarité visuelle.....	12
I.2.9. Classification par similarité visulle.....	12
I.3. Conclusion.....	12

Chapitre II : Machine learning et classification des images

II.1 Introduction.....	14
II.2.Méthodes de classification.....	14
II.2.1 Méthodes supervisées.....	14
II.2.2. Méthodes non supervisées.....	14
II.3. Indicateurs de performance en classification.....	14
II.3.1. Matrice de confusion.....	14
II.3.2. Courbe ROC.....	16
II.4.Classification des images et l'apprentissage automatique.....	16
II.5. l'apprentissage automatique.....	17
II.6. performance et surapprentissage.....	18
II.7. types d'apprentissage.....	19
II.7.1. Apprentissage supervisé.....	19
II.7.2. Apprentissage non supervisé.....	20
II.8. les différents types d'algorithmes.....	21
II.8.1. la regressionlineare.....	21
II.8.2. Algorithme KNN.....	21
II.8.3. Le classifieurnaif de bayes.....	21
II.8.4. K-means.....	22
II.8.5. les arbres de decision.....	22
II.8.6. les forrets aléatoires.....	23
II.8.7. les machines a vecteur de support.....	24
II.8.8. Perceptron multicouches.....	24
II.9.les différents types des réseaux de neurone.....	25
II.9.1. Adaline.....	25
II.9.2. les réseaux RBF.....	26
II.9.3. les réseaux de hopfield.....	27
II.9.4. les réseaux kohonen.....	28
II.10.Deep learning.....	28
II.11. Quelques Algorithmes du deeplearning.....	28
II.11.1. Réseau de neurone réccurent.....	28
II.11.2.Réseau de neurone profond.....	29
II.11.3.les machines de Boltzmann restreintes(RBM).....	29
II.12. Domaines d'application.....	29
II.13.les réseaux de neurone convolutionels.....	30

II.13.1.la couche de convolution.....	30
II.13.2. la couche de pooling.....	31
II.13.3.fonction d'activation.....	32
II.13.4 couche entierement connectée.....	34
II.13.5. couche de perte.....	34
II.14.choix des parametres.....	35
II.14.1. Nombre des filtres.....	35
II.14.2. forme de filtre.....	35
II.15. Méthode régularisation.....	35
II.15.1. Dropout.....	35
II.15.2 Drop-connect.....	35
II.15.3 pooling stochastique.....	36
II.16. Etat de l'art.....	36
II.17. Conclusion.....	37
Chapitre III : Implémentation	
III.1. Introduction.....	39
III.2. Logiciels et librairies utilisés dans l'implémentation.....	39
III.2.1. Tensorflow.....	39
III.2.2. Keras.....	39
III.2.3. Python.....	40
III.3. Configuration utilisé dans l'implémentation.....	40
III.4. Dataset.....	41
III.4.1.PKLot.....	41
III.4.2.CNRPark.....	41
III.5.Architecture de notre modele.....	41
III.6. Résultats obtenus et discussions.....	44
III.7. coude de source.....	49
III.8. Conclusion.....	51
Conclusion générale.....	53
Bibliographie.....	54

Liste des tableaux

Tableau III.1: architecture du modèle.....	40
Tableau III.2 Configuration du modèle.....	41
Tableau III.3: résultat d'exécution de code avec PKLot.....	45
Tableau III.4: résultat d'exécution de code avec CNRPark.....	46
Tableau III.5: comparaison d'accuracy avec CarNet.....	46

Liste des figures

Figure I.1 voisinage d'ordre 4 (a), diagonal (b), d'ordre 8 (c).....	6
Figure II.1: Matrice de confusion dans le cas binaire	15
Figure II.2: Courbe Roc.....	16
Figure II.3: graphe représente le sur-apprentissage.....	19
Figure II.4: apprentissage supervisé.....	20
Figure II.5 Illustration de la différence entre régression linéaire et classification linéaire.....	20
Figure II.6: apprentissage non supervisé.....	21
Figure II.7: Partitionnement de données avec l'algorithme K-moyennes.....	23
Figure II.8: Exemple d'arbre de décision avec les classes qui sont les feuilles.....	21
Figure II.9 : perceptron multicouche.....	23
Figure II. 10 : Structure du réseau Adaline.....	23
Figure II.11: représentation d'un réseau RBF.....	24
Figure II.12 : la convolution.....	29
Figure II.13 : types de pooling.....	30
Figure II.14: Fonction segmoïde.....	31
Figure II.15: Fonction relu.....	31
Figure II.16: Fonction Softmax.....	32
Figure III.1: logo de Tensorflow.....	37
Figure III.2: logo de Keras.....	38
Figure III.3: logo de Python.....	38
Figure III.4: training Puc- test Puc.....	41
Figure III.5: training Puc- test UFPR04.....	41
Figure III.6: training Puc- test UFPR05.....	42
Figure III.7: training UFPR04- test UFPR04.....	42
Figure III.8: training UFPR04-test UFPR05.....	42
Figure III.9: training UFPR05- test UFPR05.....	43
Figure III.10: training UFPR05-test UFPR04.....	43
Figure III.11: training UFPR05-test Puc.....	43
Figure III.12: training UFPR04-test PUC.....	44
Figure III.13: Nombre des images mal et bien classé par la base de donnée.....	44

الملخص

الشبكات العصبية هي مجموعة من الخوارزميات التي تستلهم عملها الخلايا العصبية البيولوجية ، وقد تم تطوير هذه الشبكات لحل مشكلات: التحكم ، التعرف على الأشكال أو الكلمات ، القرار ، والحفظ ، في هذا العمل تم استخدام نوع جديد من الشبكة العصبية وهو الشبكة العصبية التلافيفية لتصنيف الصور المرتبطة بمواقف السيارات اتلذلك اقترحنا بنية وقمنا بإجراء مقارنات ، من أجل الحصول على نتائج جيدة.

Résumé

Les réseaux de neurones sont un ensemble d'algorithmes dont le fonctionnement est inspiré des neurones biologiques, Ces réseaux ont été développés pour résoudre des problèmes: de contrôle, de reconnaissance de formes ou de mots, de décision, et de mémorisation, Dans ce travail on a utilisé un nouveau type de réseaux de neurones qui est le réseau de neurone convolutionnel pour la classification des images associées aux places de stationnement dans un parking,

Alors on a proposé une architecture et on a fait des comparaisons afin d'obtenir de bons résultats.

Abstract

Neural networks are a set of algorithms whose operation is inspired by biological neurons, these networks have been developed to solve problems: control, recognition of shapes or words, decision, and memorization, in this work a new type of neural networks was used which is the convolutional neuron network for the classification of images for car parking occupancy detection

So we proposed an architecture and made comparisons, in order to obtain good results.

Ces dernières années, avec le développement des systèmes informatiques et logiciels complexes, l'intelligence artificielle s'est immiscée dans de nombreux domaines et elle a pénétré dans presque tous les secteurs, de la construction, des transports et de la fabrication à l'intelligence économique, à l'éducation et aux soins de santé

De plus en plus avide d'expérimentations technologiques, les villes deviennent le nouveau terrain de jeu des solutions d'intelligence artificielle (IA), dont l'objectif ultime et commun à toutes les villes intelligentes est de mettre l'innovation, sous toutes ses formes, au service du citoyen pour élaborer aujourd'hui, la ville de demain, ont-ils fait observer, relevant que l'IA s'annonce comme l'un des éléments clés du futur développement urbain

Une « Smart City » ou « ville intelligente » utilise les technologies numériques pour améliorer la performance de ses services et le bien-être de ses habitants, réduire les coûts et les sources de consommation et engager de manière plus efficace et active le dialogue avec ses citoyens.

L'un des grands chantiers de la ville intelligente est le smart parking, le smart parking a été développé pour diverses raisons. Depuis plusieurs années, les villes ont remarqué que leurs conducteurs avaient de réels problèmes pour trouver une place de parking facilement. Il y a peu de place et le délai de stationnement est souvent mal adapté, Cela entraîne un bouchonnement des villes et un problème de pollution se pose également,

Habituellement, Quand un automobiliste cherche une place de parking, il tourne en rond jusqu'à trouver une place libre, ce qui entraîne des encombrements des rues mais s'il connaît la localisation des places libres, il peut en réserver une et aller directement à cette dernière

Et pour cela, la solution trouvée est d'utiliser des capteurs pour déterminer l'état des différents espaces. Cela nécessite l'installation des capteurs dans chaque espace de stationnement, ce qui pourrait être coûteux, en particulier dans les parcs de stationnement avec un grand nombre de places disponibles.

Dans notre travail, nous proposons une approche pour la détection de l'occupation de stationnement basé sur le Deep Learning. Et spécialement les réseaux de neurones à convolution, Le Deep Learning est une branche de l'intelligence artificielle qui vise à développer des techniques permettant aux ordinateurs d'apprendre des tâches de perception complexes, comme la vue et l'ouïe, Il fournit une précision quasi humaine dans la classification des images.

Pour ce faire, nous avons structuré notre mémoire en trois chapitres :

- Dans le premier chapitre on va présenter les notions de base sur les images
- Le deuxième chapitre est consacré à machine learning et classification des images
- Dans la troisième chapitre, on va montrer la partie expérimentale de notre travail et on discute les différents résultats obtenus et à la fin on termine par une conclusion général

Chapitr 1

I.1 Introduction:

En vision par ordinateur, il y a différents types d'image et différents caractéristiques associés à une image. Dans ce chapitre nous allons aborder les notions de base sur les caractéristiques principales des images permettant à la recherche et à la classification des images basée sur le contenu visuel. Nous présenterons des mesures de similarité des images pour chercher les images similaires. Nous terminerons ce chapitre par une conclusion.

I.2 Notion de base

I.2.1 Définition d'une image

Une image est une représentation du monde, une représentation d'une personne ou d'un objet par la peinture, la sculpture, le dessin, la photographie, le film, etc. C'est aussi un ensemble structuré d'informations qui, après affichage sur l'écran, ont une signification pour l'œil humain [1].

On peut définir une image comme un tableau bidimensionnel dont chaque élément (pixel) représente une surface élémentaire de l'image. La disposition de ces pixels est généralement en ligne et colonne [2].

Mathématiquement, Une image est la projection sur un plan d'une scène 3D, Elle peut être définie comme une fonction à deux variables $f(x, y)$ où (x, y) est la position d'un point de l'espace sur le plan de projection et $f(x, y)$ est l'intensité (ou brillance) au point de coordonnées (x, y) [3].

I.2.2. Types d'images

▪ Image binaire

Une image binaire est une image pour laquelle chaque pixel ne peut avoir pour valeur que 0 ou 1, elle peut être représentée par une matrice booléenne dont chaque élément signifie Vrai (1 = blanc) ou Faux (0 = noir).

▪ Image couleur RGB

Chaque pixel possède une couleur décrite par la quantité de rouge (R), vert (G) et bleu (B). Chacune de ces trois composantes est codée sur l'intervalle [0, 255], ce qui donne $256^3 = 16\,777\,216$ couleurs possibles [4].

- **Image en niveau de gris**

Une image en niveaux de gris est un tableau de valeurs. Chaque case de ce tableau stocke une valeur, ces valeurs sont enregistrées dans l'ordinateur sous forme de nombres entiers entre 0 et 255, la valeur 0 correspond au noir, et la valeur 255 correspond au blanc. Les valeurs intermédiaires correspondent à des niveaux de gris allant du noir au blanc .

- **L'image matricielle (ou bitmap):**

Elle est composée de petits points appelés « pixels » que l'on ne voit pas à l'œil nu. Lors de l'agrandissement d'une image matricielle, cette dernière devient floue car les pixels ressortent, ce sont les carrés qui apparaissent sur l'écran.

- **L'image vectorielle :**

Elle est composée de lignes de segments qui sont liés entre eux par des formules mathématiques. Il s'agit d'un système de proportionnalité et de coordonnées. Grâce à la vectorisation, chaque élément a une place bien définie ce qui empêche la déformation de l'image. [5].

I.2.3. Caractéristiques d'une image

I.2.3.1. Pixel

Le mot pixel provient d'une abréviation de l'expression britannique (Picture élément)[6],il représente l'unité de base d'une image. L'ensemble de ces pixels est contenu dans un tableau de deux dimensions constituent l'image finalement obtenue[7]. Chaque pixel est associé à un niveau de gris n ou des niveaux de composantes couleur codé sur N bits et représentent respectivement le niveau de luminosité ou de couleur de la zone correspondante dans la scène observé. Chaque pixel est localisé par ses coordonnées x et y [8].

I.2.3.2 Résolution

La résolution d'une image est le nombre de pixels contenus dans l'image par unité de longueur, elle s'exprime le plus souvent en PPP (Point Par Pouce) ou en DPI (Dot Per Inch), parfois en point

par cm, La résolution définit la qualité d'une image, plus la résolution est grande plus l'image est précise dans les détails[9].

I.2.3.3 Dimension

C'est la taille de l'image, elle se présente sous forme de matrice dont les éléments sont des valeurs numériques, le nombre de ligne de cette matrice multiplié par le nombre de colonnes nous donne le nombre total des pixels dans l'image [10].

I.2.3.4. Bruit

Le bruit est un phénomène parasite aléatoire (suivant une distribution de probabilité connue ou non) [11], il correspond à des perturbations soit du dispositif d'acquisition, soit de la scène observée elle-même [12].

Les sources de bruit d'une image sont nombreuses et diverses :

- bruits liés aux conditions de prise de vue (bougé, éclairage de la scène)
- bruits liés aux capteurs (appareil numérique de bas de gamme.
- bruits liés à l'échantillonnage.
- bruits liés à la nature de la scène (poussières, rayures) [13].

I.2.3.5. Voisinage

Un pixel p de coordonnées (x,y) a quatre voisins horizontaux et verticaux dont les coordonnées sont $(x+1,y)$, $(x-1, y)$, $(x,y+1)$, $(x,y-1)$, cet ensemble représente le voisinage d'ordre 4 de p , les quatre voisins diagonaux de p ont pour coordonnées $(x+1,y+1)$, $(x+1,y-1)$,

$(x-1, y+1)$, $(x-1, y-1)$. Les réunions de ces deux ensembles constituent l'ensemble des voisins d'ordre 8[14]

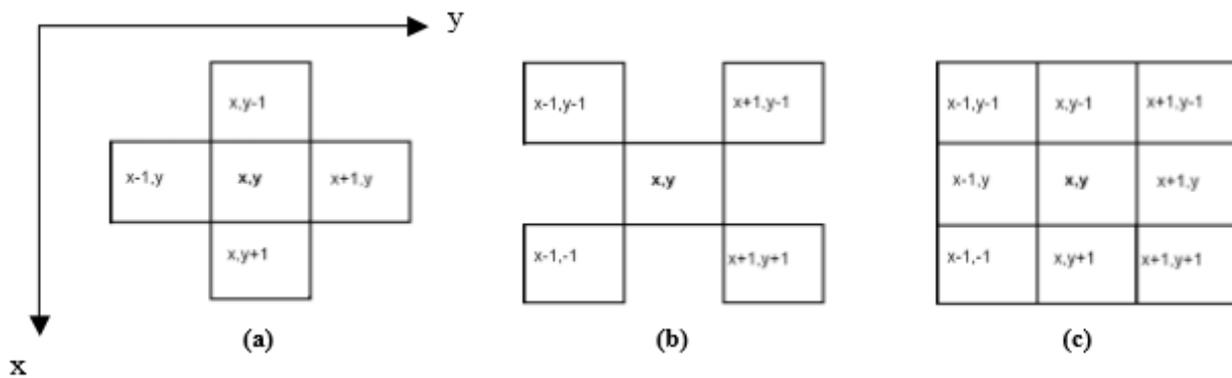


Figure I. 1 voisinage d'ordre 4 (a), diagonal (b), d'ordre 8 (c)

I.2.3.6 Contraste

Le contraste (de contraster, se tenir contre) est l'opposition de deux choses dont l'une fait ressortir l'autre[15], il est défini comme un coefficient exprimant le confort apprécié par l'œil qui sait différencier les puissances ou éclairages issues de deux zones proches sur une même image[16]. Il indique la différence de densité entre les tons les plus clairs et ceux les plus foncés d'une image. Pour simplifier, une image très contrastée comportera presque que du blanc et du noir, alors qu'une image peu contrastée contiendra énormément de valeurs intermédiaires mais pas nécessairement de véritables noirs et blancs [17].

I.2.3.7 Luminance

La luminance est l'intensité lumineuse d'une source lumineuse étendue dans une direction donnée, divisée par l'aire apparente de cette source dans cette même direction. L'unité de luminance lumineuse est la candela par mètre carré, symbole cd/m^2 [18].

I.2.3.8 Contour

Une limite entre deux (ou un groupe de) pixels dont la différence de niveau de gris (couleur) est significative [19]. Un contour correspond à une variation d'intensité ou à une discontinuité entre les propriétés de deux ensembles de points.

I.2.4 Les caractéristiques visuelles de l'image

I.2.4.1 Couleur

La couleur est l'un des premiers descripteurs visuels utilisé pour la recherche d'images auquel les travaux de recherche s'y sont intéressés. Cet intérêt provient de la robustesse que la couleur montre par rapport aux changements d'échelle et d'orientation, son indépendance au fond, à la taille et aux occlusions des images. L'attribut couleur est primordiale dans notre quotidien pour décrire l'environnement et pour identifier les objets d'intérêt dans une scène et, aussi dans divers domaines d'applications tels que les multimédias (indexation, classification, etc.)[20].

I.2.4.1.1 L'histogramme de couleur

Définition :

On définit l'histogramme des niveaux de gris d'une image comme étant la fonction $h : [0..L-1] \rightarrow N$ qui associe à chaque niveau de gris entre 0 et L-1 la quantité de pixels de l'image qui possèdent cette intensité lumineuse [21].

L'histogramme d'une image peut être représenté par un vecteur dont chaque

Composante est un nombre de pixels de niveau de gris correspondant à son indice.

Les histogrammes sont faciles et rapides à calculer, et robustes à la rotation et à la translation.

Cependant il y a quatre problèmes en utilisant d'histogrammes pour l'indexation et la recherche d'images [22].

- a- les histogrammes sont de grandes tailles, donc par conséquent il est difficile de créer une indexation rapide et efficace.
- b- les histogrammes ne possèdent pas d'informations spatiales sur les positions des couleurs. Dans certains cas, il y a des images différentes, mais ces images ont les mêmes histogrammes.
- c- les histogrammes sont sensibles à de petits changements de luminosité. c'est-à-dire, c'est difficile pour comparer des images similaires dans des conditions différentes.
- d- on ne peut pas faire la comparaison partielle des images (objet particulier dans une image), puisqu'on doit calculer globalement l'histogramme sur toute l'image.

I.2.4.1.2 Les espaces de couleur

Le mode colorimétrique de fondement pour les applications numériques est basé sur le fonctionnement des écrans vidéo (ordinateur, télévision...) qui restituent les teintes par l'addition en lumière (faisceaux d'électrons) de trois couleurs primaires : le rouge (R), le vert (G) et le bleu (B). Avec ce sous-système une teinte Θ est obtenue par une combinaison linéaire :

$$\Theta = \alpha * R + \beta * G + \gamma * B \text{ avec } \alpha, \beta, \gamma \in [0,1] \dots\dots\dots(1)$$

Dans le cas d'une image RGB codé sur 8 bits (cas le plus courant), chaque couleur primaire est codé sur un octet et peut donc prendre n'importe quelle valeur dans l'intervalle [0, 255]. Le noir et le blanc correspondent alors respectivement au triplet (0, 0, 0) et (255, 255, 255).

L'espace TSV (Teinte Saturation Valeur - en anglais HSV) est le plus utile pour la segmentation et la reconnaissance et il a été prouvé un espace très fort dans le système de recherche des images. Dans cet espace, on peut séparer pour un pixel l'intensité du pixel (valeur) et la couleur du pixel (teinte et saturation).[23]

I.2.4.1.3. Les Moments statistiques

On calcule seulement des dominantes caractéristiques de couleur tel que l'espérance, la variance et d'autres moments. On peut calculer l'espérance, la variance, les moments sur chaque composante par la formule suivante :

$$E = \frac{1}{N} \sum_{j=1}^N p_j \dots\dots\dots(2)$$

$$\delta = \left(\frac{1}{N} \sum_{j=1}^N (p_j - E)^2 \right)^{\frac{1}{2}} \dots\dots\dots(3)$$

$$s = \left(\frac{1}{N} \sum_{j=1}^N (p_j - E)^3 \right)^{\frac{1}{3}} \dots\dots\dots(4)$$

Où E est l'espérance, δ est la variance, s est le troisième moment [24].

I.2.4.2. Texture:

La texture est le second attribut visuel largement utilisé dans la recherche d'images par le contenu après l'attribut couleur, c'est une caractéristique fondamentale des images, en général il n'existe pas une définition précise de la texture en vision par ordinateur on parle fréquemment de répétition de motifs similaire, elle peut être caractérisé par l'attribut de contraste, de directionnalité, de régularité et de la périodicité du motif. La texture permet de différencier des régions de couleurs similaires, mais de sémantiques différentes.

Plusieurs méthodes utilisées pour analyser la texture [25]. On peut citer :

I.2.4.2.1. La matrice de cooccurrences

Cette méthode permet de déterminer la fréquence d'apparition d'un motif formé de deux pixels séparés par une certaine distance d dans une direction particulière par rapport à l'horizontale, plusieurs statistiques peuvent alors être calculé à partir de la matrice cooccurrences [26], citons par exemple :

$$\text{Energie} = \sum_{i=0}^{N-1} \sum_{j=0}^{N-1} (P_{ij}(d, \theta))^2 \dots \dots \dots (5)$$

$$\text{Inertie} = \sum_{i=0}^{N-1} \sum_{j=0}^{N-1} ((i-j)^2 P_{ij}(d, \theta)) \dots \dots \dots (6)$$

$$\text{Entropie} = - \sum_{i=0}^{N-1} \sum_{j=0}^{N-1} (\log P_{ij}(d, \theta) P_{ij}(d, \theta)) \dots \dots \dots (7)$$

$$\text{Contraste} = \sum_{i=0}^{N-1} \sum_{j=0}^{N-1} ((i-j)^2 P_{ij}(d, \theta)) \dots \dots \dots (8)$$

I.2.4.3. La forme

L'attribut forme est l'un des attributs bas niveau également le plus utilisé pour décrire le contenu visuel des images. Ces attributs sont utilisés pour décrire la structure géométrique générique du contenu visuel. Zhang et al [27] ont proposé de classifier les descripteurs de forme en deux familles :

- ✓ Descripteurs orientés région : ils décrivent les objets selon la distribution spatiale des pixels qui les constituent.
- ✓ Descripteurs orientés contour : ils décrivent les objets selon leur contour externe.

I.2.5. Vecteur descripteur

Le vecteur descripteur est un vecteur caractéristique de l'image construite à partir des attributs extraits de l'image. Il se présente généralement sous forme d'un vecteur nN composantes réelles.

I.2.6. Mesure de similarité

La recherche d'images similaires par le contenu est basée sur la similarité des caractéristiques visuelles telles que la couleur, la texture ou la forme. La fonction distance utilisée pour évaluer la similarité dépend des critères de la recherche, mais également de la représentation des caractéristiques. L'idée principale est généralement d'associer à chaque image un vecteur

multidimensionnel représentant les caractéristiques de l'image, et de mesurer la similarité des images en utilisant une fonction de distance entre les vecteurs.[28] Une fonction de distance $d(x, y)$ entre les points x, y est une métrique si :

- $d(x,y) \geq 0$
- $d(x,y) = 0 \Leftrightarrow x=y$
- $d(x,y) = d(y,x)$.
- $d(x,y) \leq d(x,z)+d(z,y)$.

I.2.7. Distance de similarité

Lorsque les données sont assimilées à des vecteurs, ce qui est souvent le cas, la distance de Minkowski est fréquemment employée. Elle est donnée par :

$$d_m(V_1, V_2) = \left(\sum_{i=1}^N W_i X |V_{1,i} - V_{2,i}|^m \right)^{\frac{1}{m}} \dots\dots\dots(9)$$

m représente l'ordre est W représente la matrice de pondération

En faisant varier m on obtient différentes de type de fonctions, la distance de Minkowski du premier ordre ($m=1$) est une distance de Manhattan et la distance de Minkowski du deuxième ordre ($m=2$) est une distance Euclidienne.

Le choix d'une valeur appropriée pour m dépend de l'importance que nous voulons accorder aux plus grandes différences. Ainsi les grandes valeurs de m donnent progressivement plus d'importance aux différences les plus grandes et quand m tend à l'infini la distance de Minkowski tend vers la distance de Tchebychev [29].

I.2.7.1. La distance euclidienne

Lorsque $m=2$ on obtient la distance euclidienne :

$$d(V_1, V_2) = \sqrt{\sum_{i=1}^N W_i X (V_{1,i} - V_{2,i})^2} \dots\dots\dots(10)$$

Elle est invariable aux translations et aux rotations des données dans les espaces des attributs.

I.2.7.2. La distance de manhattan

Pour $m=1$ on obtient la distance de Manhattan :

$$d(V_1, V_2) = \sum_{i=1}^N (|V_{1,i} - V_{2,i}|) \dots \dots \dots (11)$$

Cette distance est aussi connue sous le nom de city-block. [30]

I.2.7.3. Distance de tchebychev

Pour $m=\infty$ on obtient la distance de Tchebychev :

$$d(V_1, V_2) = \max_i (|V_{1,i} - V_{2,i}|) \dots \dots \dots (12)$$

Elle est adaptée aux données de grande dimension. Cette distance examine la différence absolue entre les différents pairs des vecteurs.[31]

I.2.8. Recherche par similarité visuelle:

Il s'agit de retrouver les vecteurs les plus similaires à un vecteur requête au sens d'une mesure de similarité donnée. La recherche par similarité peut être exécutée soit par une recherche des K plus proches voisins (k-ppv), soit par une recherche à ϵ près appelée également recherche par intervalle [32]

I.2.9. Classification par similarité visuelle :

Consiste à regrouper les vecteurs descripteurs les plus similaires en plusieurs ensembles. Cette classification est basée sur le calcul de la distance de similarité entre l'ensemble des vecteurs descripteurs de la base d'images, tout en basant sur les mesures de similarité.

I.3. Conclusion:

Dans ce chapitre nous avons présenté les caractéristiques de base de l'image. Le contenu visuel de l'image a été présenté à travers d'un vecteurs descripteur qui sera exploiter dans le contexte d'évaluation de la similarité des images. Le présent chapitre nous a permet de comprendre que l'image est un objet complexe à analyser correctement et qui porte plusieurs information visuelles issus de la couleur, texture ou la forme.

Chapitre 2

II.1. Introduction :

Ce chapitre est divisé en trois sections. Pour la première section nous allons présenter les méthodes de classification. Dans la deuxième section nous aborderons des algorithmes de la machine learning et les notions fondamentales de l'apprentissage en profondeur (deep Learning). enfin pour la troisième section nous allons présenter les travaux associés à la classification des images du stationnement dans un parc.

II.2. Méthodes de classifications :

Les méthodes de classification les plus communes peuvent être séparées en deux grandes catégories : les méthodes de **classification supervisée** et les méthodes de **classification non supervisée**.

II.2.1. Méthode supervisé :

Les classes sont définies à partir des besoins de l'utilisateur et correspondent à des unités sémantiques de l'image, ce qui nécessite une étape d'apprentissage préalable à la classification [37]. Donc la classification supervisée est un processus inductif où l'utilisateur considère un ensemble d'exemples étiquetés préalablement : la cible à apprendre est connue (classe d'appartenance) [33].

II.2.2. Méthode non supervisé :

La classification non supervisée désigne un corpus de méthodes ayant pour objectif de dresser ou de retrouver une typologie existante caractérisant un ensemble de n observations, à partir de p caractéristiques mesurées sur chacune des observations [34]. Dans le cas non supervisé, l'opérateur n'a besoin d'aucune information a priori sur la scène à classer. Soit il existe une collection universelle de classes déjà définies indépendamment de l'image, soit le nombre de classes et leurs caractéristiques sont définies automatiquement durant le processus de classification [35].

II.3. Indicateur de performance en classification :

II.3.1 Matrice de confusion :

La première manière d'évaluer un classifieur consiste à confronter les valeurs observées de la variable dépendante Y avec les valeurs prédites \hat{Y} fournies par le modèle. L'outil privilégié est la matrice de confusion[36].

La matrice de confusion C'est un tableau de contingence confrontant les classes obtenues (colonnes) et les classes désirées (lignes) pour l'échantillon. Sur la diagonale principale on trouve donc les valeurs bien classées, hors diagonale les éléments mal classés [37].

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

Figure II.1:Matrice de confusion dans le cas binaire

- 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[38]

Cette matrice permet de déduire les paramètres suivants [39]:

Précision : De ceux que l'algorithme a pu classer, combien sont corrects

$$\text{Précision} : \frac{VP}{VP+FP}$$

Rappel: De ceux qui existent, combien l'algorithme a pu trouver

$$\text{Rappel} : \frac{VP}{VP+FN}$$

F-mesure: Moyenne harmonique entre la précision et le rappel

$$F\text{-mesure} : \frac{2 * (\text{précision} * \text{rappel})}{\text{précision} + \text{rappel}}$$

II.3.2. Courbe Roc:

Est un graphe qui montre la performance d'un modèle pour tous les seuils, Il s'agit du 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 à une forte capacité à discriminer[40].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[41].

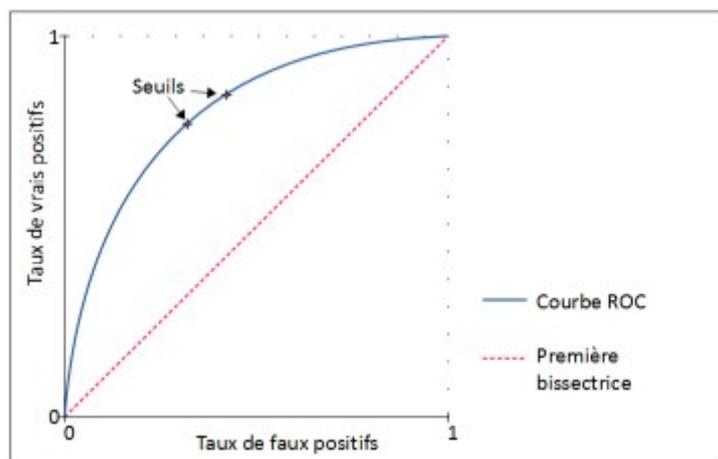


Figure II.2: Courbe Roc

II.4. La classification des images et le machine learning :

La classification d'images est un problème fondamental en machine learning, qui a de nombreuses applications concrètes [42]. Lorsque l'apprentissage automatique et la classification des images sont intégrés, les ordinateurs deviennent capables d'exécuter des tâches visuelles qui, jusqu'à récemment, ne pouvaient être effectuées que par des humains. L'ensemble, ces technologies offrent le potentiel

de percées dans l'automatisation, offrant de nouvelles opportunités numériques pour les entreprises dans divers domaines [43]

Pour une machine, une image est un tableau de nombres indiquant la luminosité (ou la couleur) de chaque pixel, et un signal sonore une suite de nombres indiquant la pression de l'air à chaque instant [44].

On utilise le machine learning lorsqu'il est difficile ou impossible de définir les instructions explicites à donner à un ordinateur pour résoudre un problème, mais que l'on dispose de nombreux exemples illustratifs [45]. La machine learning repose sur deux piliers fondamentaux, d'une part, les données, qui sont les exemples à partir duquel l'algorithme va apprendre ; et d'autre part, l'algorithme d'apprentissage, qui est la procédure que l'on fait tourner sur ces données pour produire un modèle. On appelle entraînement le fait de faire tourner un algorithme d'apprentissage sur un jeu de données [46].

Dans sa forme la plus utilisée, l'apprentissage machine est supervisé, une technique courante en IA, permettant aux machines d'apprendre. Concrètement, pour qu'un programme apprenne à reconnaître une voiture, par exemple, on le « nourrit » de dizaines de milliers d'images de voitures, étiquetées comme telles. Un « entraînement », qui peut nécessiter des heures, voire des jours. Une fois entraîné, il peut reconnaître des voitures sur de nouvelles images [47].

Jusqu'à récemment les systèmes « classiques » d'apprentissage en reconnaissance de forme étaient composés de deux blocs : Un extracteur de caractéristique (Feature extractor), programmé à la main et Un classifieur entraînable, comme une Machine à Vecteur de Support, pour classer l'image. L'extracteur de caractéristiques, transforme le tableau de nombres représentant l'image en une série de nombres, un vecteur de caractéristiques, dont chacun indique la présence ou l'absence d'un motif simple dans l'image [44].

II.5. L'apprentissage automatique :

La machine learning est un domaine captivant. Issu de nombreuses disciplines comme les statistiques, l'optimisation, l'algorithmique ou le traitement du signal, c'est un champ d'études en mutation constante qui s'est maintenant imposé dans notre société [48]. L'apprentissage automatique est un type d'intelligence artificielle qui confère aux ordinateurs la capacité d'apprendre sans être explicitement programmés [49]. Il consiste en la mise en place d'algorithmes ayant pour objectif d'obtenir une analyse prédictive à partir de données, dans un but précis [50].

L'apprentissage automatique consiste alors à programmer des algorithmes permettant d'apprendre automatiquement de données et d'expériences passées, un algorithme cherchant à résoudre au mieux un problème considéré.

La machine learning est utilisée depuis des décennies dans la reconnaissance automatique des caractères ou les filtres anti-spam, il sert maintenant à protéger contre la fraude bancaire, recommander des livres, films, ou autres produits adaptés à nos goûts, identifier les visages dans le viseur de notre appareil photo, ou traduire automatiquement des textes d'une langue vers une autre [45]

II.6. Performance et surapprentissage :

La performance est correspond à la véracité des résultats [51]. L'optimisation de la performance se fait en améliorant le modèle. Pour cela on modifie l'algorithme utilisé, on change les paramètres et on acquiert plus d'expérience [52].

Le phénomène de surapprentissage dégrade la performance des [algorithmes](#) de [machine learning](#), le sur-apprentissage est quand vous trouvez un bon modèle des données d'entraînement, mais ce modèle ne se généralise pas. Par exemple: un élève qui a mémorisé les réponses à des tests d'entraînement obtiendra de bons résultats à un test d'entraînement, mais pourrait avoir de mauvais résultats au test final. Il faut mettre de côté les «données de test» qui ne sont jamais entraînées. Une fois la formation terminée, nous exécutons le modèle sur les données de test finales [53].

Dans quels cas se produit l'overfitting ? Un algorithme d'apprentissage, par exemple [supervisé](#), cherche le modèle qui exprime le mieux la relation entre des [données](#). L'overfitting intervient lorsque l'algorithme sur-apprend (overfit), autrement dit, lorsqu'il apprend à partir des données mais aussi à partir de patterns (schémas, structures) qui ne sont pas liés au problème, comme du [bruit](#). Ainsi, l'overfitting est caractérisé par une erreur de type [variance](#) très élevée, Concrètement, on observe généralement de l'overfitting lorsqu'on utilise des modèles très complexes sur des problèmes simples mais bruités [54]

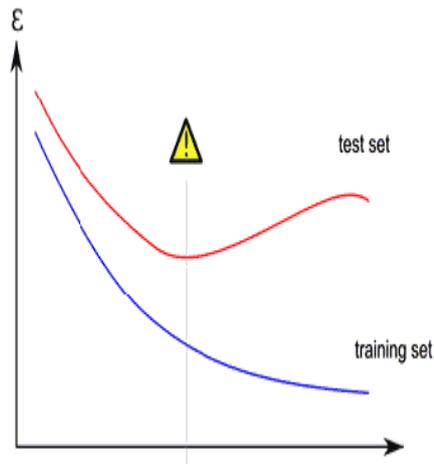


Figure II.3: graphe représente le sur-apprentissage

Le graphe montre l'évolution de l'erreur commise sur l'ensemble de test par rapport à celle commise sur l'ensemble d'apprentissage, les deux erreurs diminuent mais dès que l'on rentre dans une phase de sur-apprentissage, l'erreur d'apprentissage continue de diminuer alors que celle du test augmente.

II.7. Types d'apprentissage :

II.7.1. Apprentissage supervisé :

L'apprentissage supervisé a pour but d'établir des règles de comportement à partir d'une base de données contenant des exemples de cas déjà étiquetés. La base de données est en principe un ensemble de couples entrées / sorties $\{(X, Y)\}$. Le but est d'apprendre à prédire pour toute nouvelle entrée X , la sortie Y [55].

Les méthodes d'apprentissage supervisé se donnent alors comme objectif général de construire à partir de la base d'apprentissage, des classifieurs, ou fonctions de classement. Une telle fonction permet, à partir de la description d'un objet, de reconnaître un attribut particulier [56].

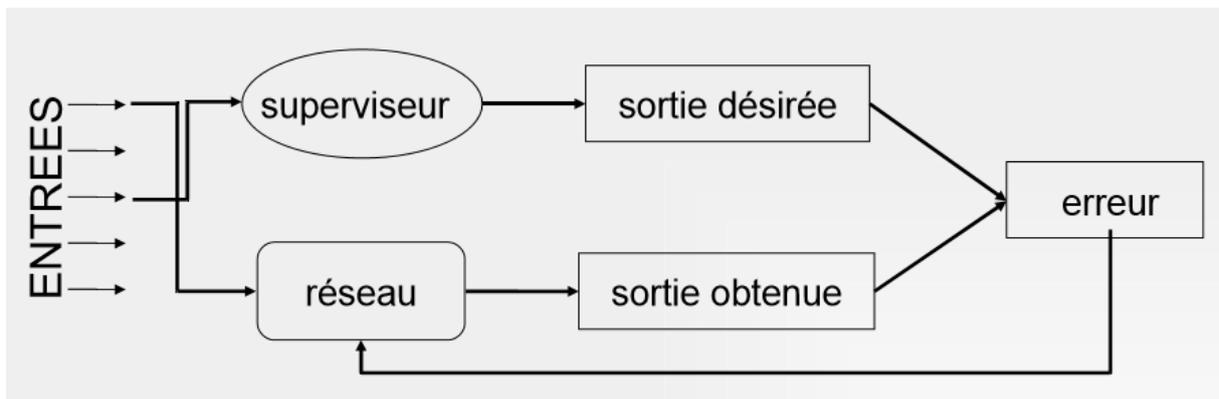


Figure II.4: apprentissage supervisé

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

- ❖ **Régression** : lorsque la valeur cible à prédire est continue
- ❖ **classification** : ou catégorisation, lorsque la valeur cible à prédire est discrète.

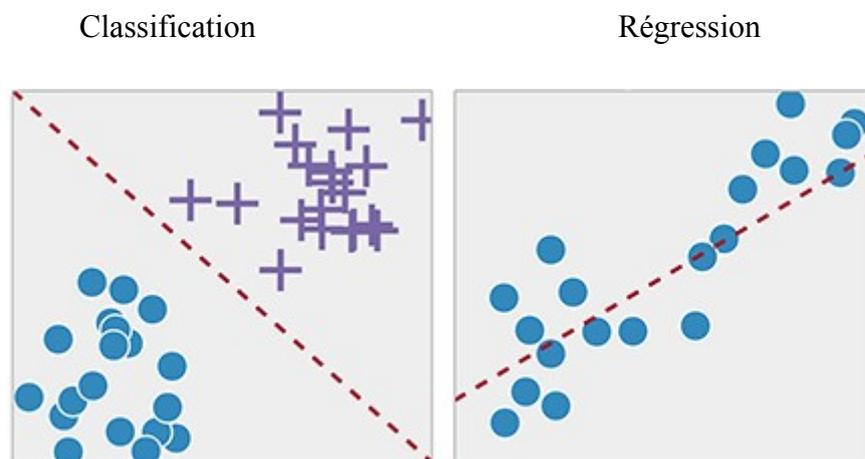


Figure II.5.1 Illustration de la différence entre régression linéaire et classification linéaire

II.7.2. Apprentissage non supervisé :

L'apprentissage non-supervisé vise à caractériser la distribution des données, et les relations entre les variables, sans discriminer entre les variables observées et les variables à prédire [57].

Différentes tâches sont associées à l'apprentissage non supervisé :

- Clustering (segmentation, regroupement) : construire des classes automatiquement en fonction des exemples disponible
 - L'apprentissage non supervisé est très souvent synonyme de clustering.
- Règles d'association : analyser les relations entre les variables ou détecter des associations.
- Réduction de dimensions.

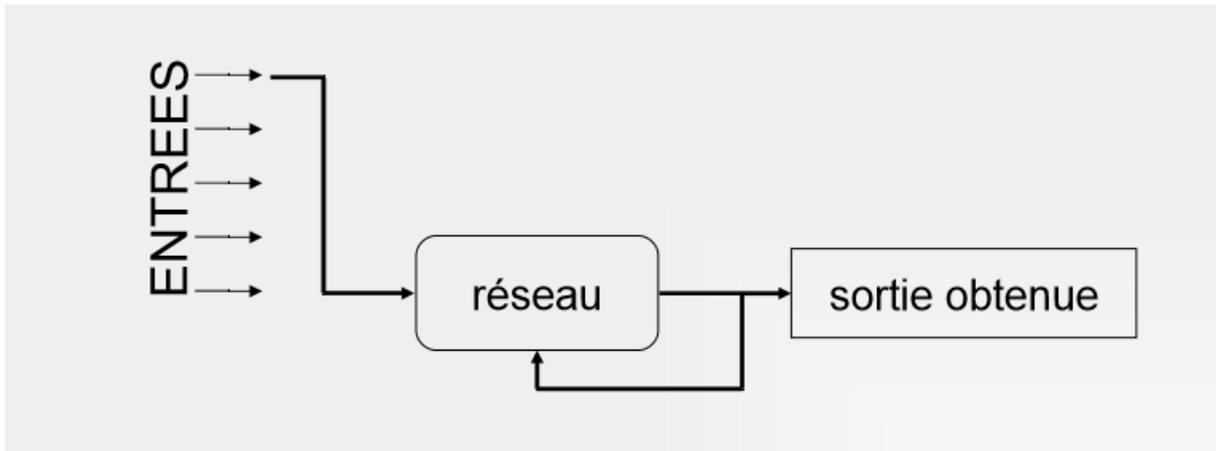


Figure II.6: apprentissage non supervisé

II.8. Les différents types d'algorithmes:

II.8.1. La régression linéaire:

C'est une modélisation linéaire qui permet d'établir des estimations dans le futur à partir d'information provenant du passé [61], le but de la régression linéaire est d'expliquer une variable Y à l'aide d'une variable X (reps plusieurs variables $X_1 X_2 \dots X_q$), la variable Y est appelée variable dépendante ou variable explicative, elle est souvent calculée avec la méthode des moindres carrés qui permet de réduire les erreurs en ajoutant de l'information[58].

II.8.2. Algorithme KNN:

L'algorithme des k-voisins les plus proches est une méthode d'apprentissage à base d'instance, la méthode ne nécessite pas de phase d'apprentissage, c'est l'échantillon d'apprentissage associé à une fonction de distance et à une fonction de choix de la classe en fonction des classes des voisins les plus proches, qui constitue le modèle[59].

- ◆ principe:

Nous supposons que une base de données d'apprentissage constitue de (N) couples <entrée-sortie> pour estimer la valeur de sortie d'une nouvelle entrée (X), la méthode des K plus proche voisins consiste à prendre en compte(de façon identique) les K échantillons d'apprentissage dont l'entrée est la plus proche de la nouvelles entrée (X), selon une distance à définir[60].

II.8.3. Le classifieur naïf de bayes:

Un classifieur naïf de Bayes est un classifieur probabiliste basé sur l'application du théorème de Bayes avec l'hypothèse naïve[61], l'algorithme détermine la classe d'une instance avec un modèle probabiliste, la probabilité que l'instance appartienne à une classe est calculée en multipliant la probabilité d'obtenir la classe avec la probabilité d'obtenir chaque caractéristique pour cette classe, en fonction des valeurs moyennes et des variances qui sont calculées en premier lien, l'algorithme choisi ensuite d'attribue à l'instance la classe ayant la plus grande probabilité[62]

La classification naïve bayésienne est basée sur le théorème de Bayes avec une forte indépendance des hypothèses.

$$P(A|B) = \frac{P(B|A) \cdot P(A)}{P(B)} \dots\dots\dots(13)$$

Où P(A/B) désigne la [probabilité conditionnelle](#) de A sachant B.

II.8.4. K-means:

La classification K-Means a pour but de partitionner n objets en k grappes dans lesquelles chaque objet appartient au groupe avec la moyenne proche. Cette méthode produit exactement k différents groupes de la plus grande distinction possible. Le meilleur nombre de grappes k conduisant à la plus grande séparation (distance) n'est pas connu a priori et doit être calculé à partir des données.

Algorithme :

- Regroupe les données en k groupes où k est prédéfini.
- Sélectionnez k points au hasard en tant que centres de cluster.
- Attribuez des objets au centre de cluster le plus proche en fonction de la fonction de distance euclidienne.

- Calculez le centroïde ou la moyenne de tous les objets de chaque groupe.
- Répétez les étapes 2, 3 et 4 jusqu'à ce que les mêmes points soient attribués à chaque groupe lors de tours consécutifs[63].

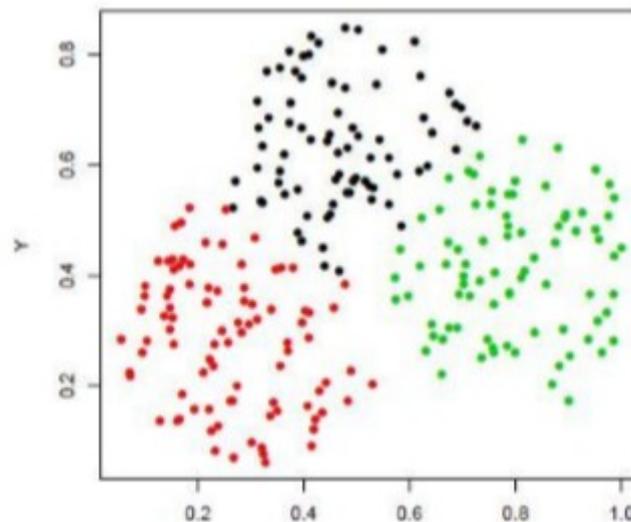


Figure II.7:Partitionnement de données avec l'algorithme K-moyennes.

II.8.5. Les arbres de décision:

Un arbre de décision est une représentation visuelle d'un algorithme de classification de données suivant différents critères qu'on appellera décisions (ou nœuds).

Principe :

Prédire la valeur d'un attribut (variable cible ou variable exogène) à partir d'un ensemble de valeurs d'attributs (variables prédictives ou variables endogènes)

Une méthode simple, supervisé, et très connue de classification et de prédiction.

- Un arbre est équivalent à un ensemble de règles de décision : un modèle facile à comprendre.
- Un arbre est composé de :
 - nœuds: classes d'individus de plus en plus fines depuis la racine
 - arcs: prédicats de partitionnement de la classe source.[64]

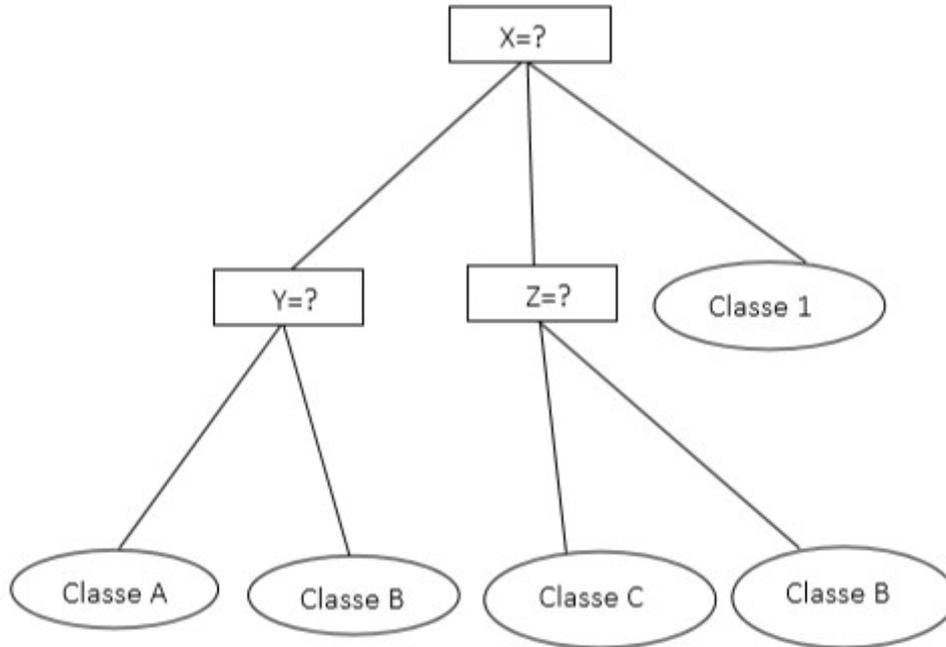


Figure II.8: Exemple d'arbre de décision avec les classes qui sont les feuilles

II.8.6. Les forêts aléatoires:

Les forêts aléatoires sont donc une amélioration du bagging pour les arbres de décision CART dans le but de rendre les arbres utilisés plus indépendants (moins corrélés).

Caractéristiques :

- Elles donnent de bons résultats surtout en grande dimension,
- Elles sont très simples à mettre en œuvre,
- Elles ont peu de paramètres.

Algorithme « forêts aléatoires » [Bre01] :

- On tire au hasard dans la base d'apprentissage B B échantillons avec remise $z_i, i=1, \dots, B$ (chaque échantillon ayant n points).
- Pour chaque échantillon i on construit un arbre CART $G_i(x)$ selon un algorithme légèrement modifié : à chaque fois qu'un nœud doit être coupé (étape 'split') on tire au hasard une partie des attributs (q parmi les p attributs) et on choisit le meilleur découpage dans ce sous-ensemble[65]

II.8.7. Les machine à vecteur de support:

Le SVM est un modèle discriminant qui tente de minimiser les erreurs d'apprentissage tout en maximisant la marge séparant les données des classes. La maximisation de la marge est une méthode de régularisation qui réduit la complexité du classifieur.

Les réseaux de neurones éprouvent de grandes difficultés à traiter les données de très haute dimension, de plus, ils possèdent un grand nombre de paramètres d'apprentissage à fixer par l'utilisateur. Pour surmonter ce problème, on procède souvent par la sélection d'une partie des attributs des données pour réduire la dimension de l'espace d'entrée. Mais dans ce cas, on aura besoin d'utiliser des hypothèses simplificatrices qui ne se vérifient pas toujours en pratique. Par ailleurs, une méthode issue récemment d'une formulation de la théorie de l'apprentissage statistique due en grande partie à l'ouvrage de Vapnik en 1995 intitulé *the nature of learning theory* surmonte ce problème en imposant que le nombre de paramètres soit linéairement lié au nombre des données d'apprentissage. Cette technique est appelée Support Vector Machines (SVM) ou machines à vecteurs de support. L'avantage principale provient du fait qu'il ya peu de paramètres à régler comparativement aux réseaux de neurones. Elle laisse très peu de place aux paramètres utilisateurs, bien que récemment proposée elle a fait l'objet d'un nombre important de publications[66]

II.8.8. Le perceptron multicouche

Les Perceptrons Multicouches (PMC) sont des réseaux de neurones bouclés comportant au moins trois couches de neurones, en particulier, une couche d'entrée, au moins une couche cachée et une seule couche de sortie. Les fonctions d'activation de la couche d'entrée suivent généralement la fonction identité, tandis que les fonctions d'activation des neurones dans les couches cachées et dans la couche de sortie suivent d'une manière générale une fonction mathématique avec des dérivées continues premières et secondes. Citons comme exemple de fonctions d'activation les fonctions identité, logistique sigmoïde, tangente hyperbolique, exponentielle et sinus. Voir aussi le module STATISTICA Réseaux de Neurones Automatisés.[67]

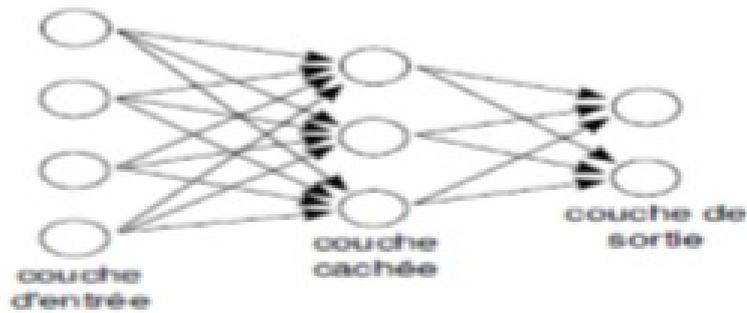


Figure II.9 : perceptron multicouche

II.9. Les différents types de réseaux de neurones :

II.9.1. Adaline :

L'Adaline (ADAPtive Linear NEURon) désigne un neurone formel adaptatif. Le réseau Adaline a été développé par Widrow. Il est constitué d'un unique neurone effectuant la combinaison linéaire de ses entrées. Il s'agit en fait d'un Perceptron sans saturation des sorties.

La règle d'apprentissage de ce réseau consiste à minimiser l'erreur quadratique en sortie du réseau de neurone. La règle d'apprentissage est identique à celle du Perceptron, à la différence près que ce sont les entrées non-saturées qui sont prises en compte [68]

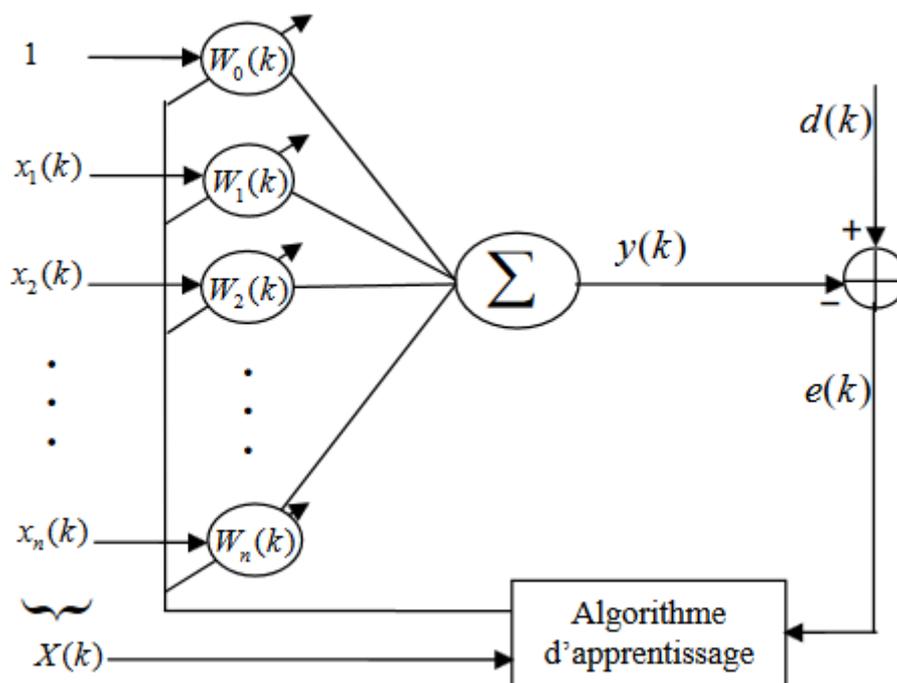


Figure II. 10 : Structure du réseau Adaline.

La sortie estimée $y(k)$ du signal de référenced (k) sera composée par la relation linéaire suivante :

$$Y(k) = W^T(k) X(k)..$$

$$\text{Avec } W^T(k) = [W_0(k) \ W_1(k) \ W_2(k) \ \dots \ W_n(k)]..$$

$$\text{Et } X^T(k) = [1 \ x_1(k) \ x_2(k) \ \dots \ x_n(k)]..$$

II.9.2. Les réseaux RBF :

L'idée générale des réseaux RBF dérive de la théorie d'approximation des fonctions, ces réseaux sont une architecture Feedforward puissante. Ce type de réseaux a été introduit pour la première fois par Hardy, et la théorie correspondante a été développée par Powell, ensuite, ces réseaux ont pris le terme de réseaux de neurones grâce à Broomhead et Lowe. Un réseau RBF est un réseau à trois couches: une couche d'entrée, une couche cachée composée defonctions noyaux et une couche de sortie, dont les neurones sont généralement animés par unefonction d'activation linéaire[69]

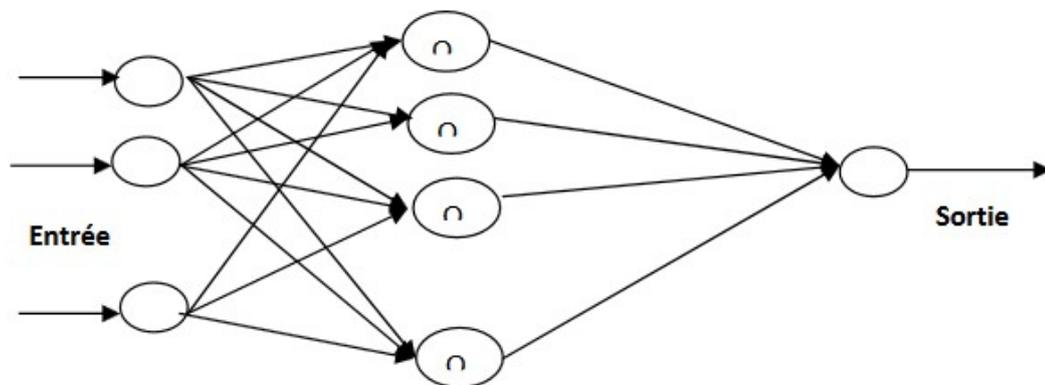


Figure II.11:représentation d'un réseau RBF

La sortie du réseau est simplement une combinaison linéaire des sorties des neurones RBF multipliéspar le poids de leur connexion respective

II.9.3. Les réseaux de hopfield :

Les réseaux deHopfieldont des réseaux à minimisation d'énergie : ils sont composés deneurones entièrement interconnectés ; ils n'ont pas une "entrée" et une "sortie", mais ilsévoluent à partir d'un

état initial. D'une façon générale, un réseau de Hopfield est donc défini par n neurones, et par une matrice de transfert W , symétrique [70].

- Chaque neurone a 2 états possibles : -1 ou +1
- L'activation du neurone i est :

$a_i = 1$ si $w_{i1} * y_1 + \dots + w_{ij} * y_j + \dots + w_{in} * y_n > 0$.

$a_i = -1$ sinon .

Avec :

y_i état du neurone i

w_{ij} coefficient synaptique entre les neurones i et j

$w_{ij} = w_{ji}$ (matrice W symétrique avec diagonale nulle)

II.9.4. Les réseaux de kohonen :

Il s'agit d'un réseau non supervisé avec un apprentissage compétitif. On apprend non seulement à modéliser l'espace des entrées avec des prototypes, mais également à construire une carte à une ou deux dimensions permettant de structurer cet espace. Les neurones de ce réseau sont constitués d'un vecteur de poids dans l'espace des entrées. La carte des neurones définit quant à elle des relations de voisinage entre les neurones. L'algorithme d'apprentissage du réseau de Kohonen est de type compétitif. Les cartes de Kohonen peuvent être utilisées dans le cadre de la projection de données multi-variées, d'approximation de densité ou de classification. Elles ont été utilisées avec succès en reconnaissance de la parole, classification et traitement d'images [71].

II.10. Deep learning :

Le deep learning ou apprentissage profond est un domaine du machine learning. Le deep learning se rapprocherait du cerveau humain, il va plus loin que la connexion entre les données et les algorithmes puisqu'il permet à la machine d'apprendre et de progresser grâce à son expérience . Le DL, qui fait appel à la fois aux connaissances en neurosciences, aux mathématiques et aux progrès technologiques, est aujourd'hui plébiscité comme une véritable révolution dans le domaine de l'intelligence artificielle [72].

L'apprentissage profond est un apprentissage réalisé sur un réseau de neurones avec plusieurs couches cachées. Le principe du Deep Learning repose sur un apprentissage hiérarchique couche

par couche. Entre chaque couche interviennent des transformations non linéaires et chaque couche reçoit en entrée la sortie de la couche précédente[73]

Ces réseaux sont capables de catégoriser les informations des plus simples aux plus complexes. Pour un objet par exemple, la première couche détecte des petits contours élémentaires, la seconde assemble ces contours en motifs puis les motifs en parties d'objets puis ces parties en objets[74].

II.11. Quelques algorithmes du deep learning :

II.11.1. Réseau de neurone récurrent : les RNNs – ont été introduit en machine learning afin de pouvoir traiter des données séquentielles, La caractéristique principale des Récurrent Neural Networks est que les neurones internes à une couche peuvent être reliés entre eux. Ils peuvent également être reliés à eux même, et a des neurones de couches précédentes en sens inverse [75].

Au cours des dernières années, un type de RNN est devenu la norme grâce à ses excellentes performances sur des tâches aussi nombreuses que variées : les réseaux de neurones à base de cellules Long Short-Term Memory (LSTM).

II.11.2. Réseaux de neurone profond :

Les réseaux de neurones profonds, ou Deep Neural Networks, ne sont autres que des réseaux multicouches, d'architecture classique, mais ils comportent plusieurs couches cachées et c'est la manière de gérer leur apprentissage qui a donné, depuis 2006, un regain d'intérêt à leur étude[76].

II.11.3. Les machines de Boltzmann restreintes (RBM)

La machine restreinte de Boltzmann notamment est le premier programme de deep learning à fonctionner sur une base de données non supervisée. Ce réseau de neurones fonctionne avec une «couche visible» et une couche invisible. Dans la première couche, toutes les données ne sont pas forcément étiquetées. Les informations sont ensuite traitées par une couche «invisible», qui va permettre d'avoir un étiquetage correct [77]

II.12. Domaine d'application :

L'apprentissage profond est une technique d'apprentissage automatique qui a considérablement amélioré les résultats dans de nombreux domaines tels que la vision par ordinateur, la reconnaissance de la parole et la traduction automatique. Les techniques d'apprentissage profond permettent, à l'aide de données, de résoudre de nombreux problèmes dans de nombreux domaines de l'économie tels que la santé, le transport, le commerce, la finance ainsi que l'énergie.

Il a d'autres usages dans les outils d'aide à la décision, dans les jeux tels que le Go avec AlphaGo et même dans l'exploitation de données structurées, dans la cybersécurité et d'une manière générale dans la recherche scientifique comme en génomique.

Autres domaine :

- voiture autonome
- diagnostic médical
- modération automatique des réseaux sociaux,
- recommandations personnalisées,
- exploration spatiale
- chatbots (agents conversationnels),

II.13. Réseau de neurone convolutionnel :

Un réseau de neurone convolutionnel est un type particulier de réseau de neurone qui se base sur l'opération de convolution. Les réseaux à convolution sont dérivés des architectures de type perceptron multi-couches (Multi Layer Perceptron : MLP), cependant ils utilisent des poids partagés, liés à la fenêtre de convolution, qui leur permettent une extraction implicite de caractéristiques locales[78]. Les CNN sont particulièrement adaptés à la reconnaissance d'images.

Un réseau est dit convolutif quand chaque neurone reçoit ses informations non pas de toute la couche précédente, mais seulement des neurones situés dans son champ réceptif.

Chaque couche dite de convolution - balaye l'ensemble de la couche précédente en appliquant à chaque petite région un même traitement local. Dans le cas d'une image, les régions sont classiquement des carrés de pixels ; la convolution permet alors par exemple d'y détecter des contours ou de flouter l'image. Ce filtre balaye ainsi l'ensemble de la sortie de la couche précédente et a pour rôle de détecter des propriétés locales. Entre chaque couche de convolution, on rajoute une couche - dite de pooling - qui ne conserve que les k plus grandes valeurs, où k est un hyper paramètre à sélectionner [79].

Une architecture CNN est formée par un empilement de couches de traitement Indépendantes.

II.13.1. La couche de convolution :

Quand on lui présente une nouvelle image, le CNN ne sait pas exactement si les caractéristiques seront présentes dans l'image ou où elles pourraient être, il cherche donc à les trouver dans toute l'image et dans n'importe quelle position. En calculant dans toute l'image si une caractéristique est présente, nous faisons un filtrage. Les mathématiques que nous utilisons pour réaliser cette opération sont appelés une convolution, de laquelle les réseaux de neurones à convolution tiennent leur nom. La couche de convolution est la composante clé des réseaux de neurones convolutifs, et constitue toujours au moins leur première couche Son but est de repérer la présence d'un ensemble de features dans les images reçues en entrée [80]

Elle fonctionne comme un extracteur de caractéristiques des images. Une image est passée à travers une succession de filtres, ou noyaux de convolution, On obtient pour chaque paire (image, filtre) une carte d'activation, ou feature map, qui nous indique où se situent les features dans l'image : plus la valeur est élevée, plus l'endroit correspondant dans l'image ressemble à la feature [81]. Finalement, les valeurs des dernières feature maps sont concaténées dans un vecteur. Ce vecteur définit la sortie du premier bloc, et l'entrée du second.

Trois hyper paramètres permettent de dimensionner le volume de la couche de convolution :

1. **Profondeur de la couche K** : nombre de noyaux de convolution
2. **Le pas S**: contrôle le chevauchement des champs récepteurs
3. **La marge (θ) ou zero padding P** : Cette marge permet de contrôler la dimension spatiale du volume de sortie.

Pour chaque image de taille $W \times H \times D$ en entrée, où W est sa largeur en pixels, H sa hauteur en pixels et D le nombre de canaux (1 pour une image en noir et blanc, 3 pour une image en couleurs) la

couche de convolution renvoie une matrice de dimensions $WC \times HC \times DC$, où $WC = \frac{w - F + 2P}{S} + 1$ et

$$HC = \frac{H - F + 2P}{S} \text{ et } DC = K$$

Où F est la taille des filtres et le S c'est le pas avec lequel on fait glisser la fenaitre, le P correspond au zéro padding, et K c'est le nombre des filtres.

Choisir $P = \frac{F-1}{2}$ et $S=1$ permet ainsi d'obtenir des feature maps de même largeur et hauteur que celles reçues en entrée

Où F est la taille du filtre, chaque filtre est de dimensions $F \times F \times D$ pixels [82].

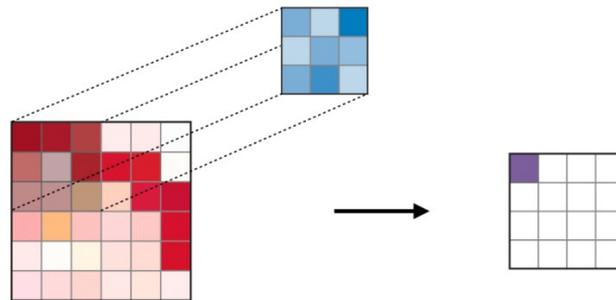


Figure II.12 : la convolution

II.13.2. La couche de pooling :

Un autre outil très puissant utilisé par les CNNs s'appelle le Pooling. Qui est une méthode permettant de prendre une large image et d'en réduire la taille tout en préservant les informations les plus importantes qu'elle contient.

Le pooling est une fonction de sous-échantillonnage spatial. Elle prend toujours en entrée des feature maps de taille $n_x \times n_y \times D$, et réduit les deux premières dimensions spatiales [83]. Les deux méthodes les plus utilisées pour appliquer cette opération sont les suivantes, soit on fait la moyenne des valeurs de la zone (pooling average), soit on extrait uniquement la valeur la plus élevée (pooling max). Et nous utilisons celle qui applique l'opération max au résultat de chaque filtre.

Pour cela, on découpe l'image en cellules régulières, puis on garde au sein de chaque cellule la valeur maximale. En pratique, on utilise souvent des cellules carrées de petite taille pour ne pas perdre trop d'informations.

Après cette étape de sous-échantillonnage nous obtenons une feature map qui est définie comme :

$$I_k^{(l)} = \text{pool}(S_k^{(l)}).$$

Avec $I_k^{(l)}$ une feature map de la couche l , $\text{pool}()$ l'opération de pooling et $S_k^{(l)}$ la valeur de sortie du neurone k de la couche l .

La couche de pooling permet de réduire le nombre de paramètres et de calculs dans le réseau. On améliore ainsi l'efficacité du réseau et on évite le sur-apprentissage.

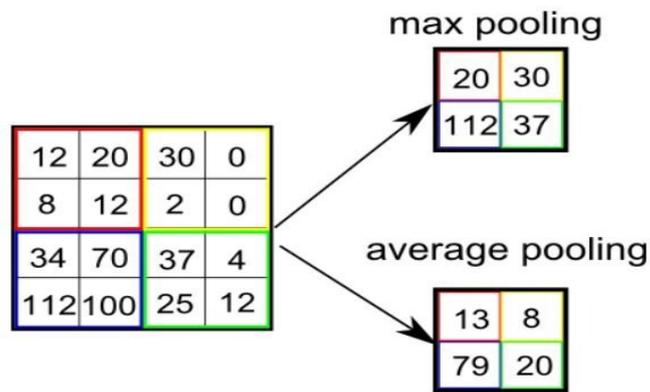


Figure II.13 : types de pooling

II.13.3. Fonction d'activation :

Une fonction d'activation est une fonction mathématique utilisée sur un signal. Elle va reproduire le potentiel d'activation que l'on retrouve dans le domaine de la biologie du cerveau humain. Elle va permettre le passage d'information ou non de l'information si le seuil de stimulation est atteint. Concrètement, elle va avoir pour rôle de décider si on active ou non une réponse du neurone[84]. Pour les réseaux de neurone à convolution, une fois la convolution effectuée, une fonction d'activation est appliquée sur toutes les valeurs de l'image filtrée. Il existe beaucoup de fonctions d'activations :

◆ Sigmoidé :

Historiquement, la fonction sigmoidé est la fonction d'activation la plus ancienne et la plus populaire [85], elle est définie comme :

$$S(x) = \frac{1}{1 + e^{-z}}$$

e est la constante exponentielle, à peu près égale à 2,71828. Un neurone qui utilise un sigmoïde comme fonction d'activation est appelé un neurone sigmoïde. Nous fixons d'abord la variable z à la somme pondérée des entrées, puis la transmettons à la fonction sigmoïde.

$$Z = b + \sum_i w_i x_i$$

$$S(x) = \frac{1}{1 + e^{-z}} \dots\dots\dots(14)$$

La sigmoïde a des propriétés non-linéaires telles que :

- bornée inférieurement par 0
- saturée quand les entrées deviennent grandes
- bornée supérieurement par 1

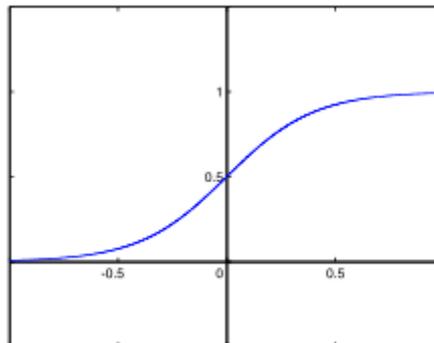


Figure II.14:Fonction sigmoïde

◆ **ReLU :**

ReLU est une fonction d'activation très couramment utilisée. Acronyme de Rectified Linear Unit (unité linéaire rectifiée), elle permet tout simplement de remplacer les résultats négatifs par zéro. La fonction ReLU est interprétée par la formule: $f(x) = \max(0, x)$ [86]

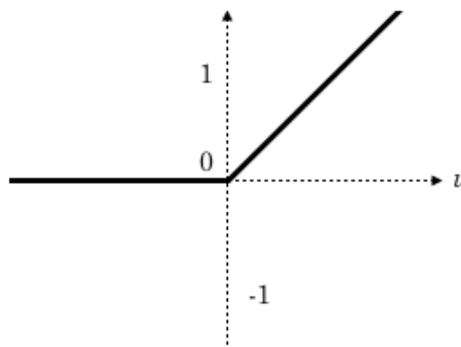


Figure II.15:Fonction relu

◆ **SoftMax :**

Elle réalise une exponentielle normalisée (c'est-à-dire que la somme des sorties totalise 1). En combinaison avec la fonction d'erreur d'entropie croisée, elle permet de modifier les réseaux perceptrons multicouches pour l'estimation des probabilités de classes[87].

Elle prend en entrée un vecteur $z=(z_1, \dots, z_k)$ de K nombres réels et qui en sort un vecteur $\sigma(z)$ de K nombres réels strictement positifs et de somme 1.

La fonction est définie par :

$$\sigma(\mathbf{z})_j = \frac{e^{z_j}}{\sum_{k=1}^K e^{z_k}} \dots\dots\dots(15)$$

Pour tout $j \in \{1, \dots, k\}$

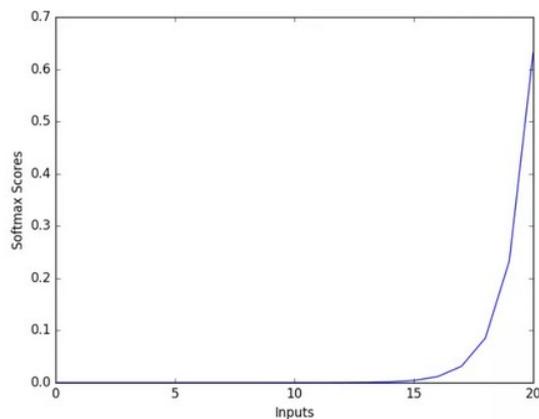


Figure II.16:Fonction SoftMax

II.13.4. Couche entièrement connectées :

Les neurones d’une couche entièrement connectée sont connectés avec chacun des neurones du volume correspondant à son entrant, c'est-à-dire avec chacun des neurones de la couche précédente. Cette couche sort un vecteur de K dimensions, où K représente le nombre de classes que le réseau sera capable de prédire. Ce vecteur contient les probabilités de classement pour chaque classe d’une image [88],

La couche FC consiste à multiplier le vecteur en entrée par la matrice contenant les poids. Le fait que chaque valeur en entrée soit connectée avec toutes les valeurs en sortie explique le terme fully-connected [81]

II.13.5. Couche de perte (Loss) :

La couche de perte suit la couche entièrement connectée et gère les ajustements de poids sur le réseau. Avant que l'entraînement du réseau ne commence, les poids des couches convolution et entièrement connectées reçoivent des valeurs aléatoires. Ensuite, pendant l'entraînement, la couche de perte vérifie en permanence les suppositions de la couche entièrement connectée par rapport aux valeurs réelles dans le but de minimiser autant que possible la différence entre l'estimation et la valeur réelle. La couche de perte effectue cela en ajustant les poids à la fois dans la couche de convolution et dans les couches entièrement connectées.

II.14. Choix des paramètres :

Malheureusement, tous les aspects des CNNs ne sont pas aussi intuitifs à apprendre et comprendre que ce que nous avons vu jusqu'à présent. Ainsi, il y a toujours une longue liste de paramètres qui doivent être définis manuellement pour permettre au CNN d'avoir de meilleurs résultats.

II.14.1. Nombre des filtres : En pratique, un CNN apprend seul les valeurs de ces filtres pendant le processus de l'entraînement. Les paramètres tels que le nombre de filtres, sont spécifiés par le scientifique avant le lancement du processus de l'entraînement. Plus nous avons des filtres, plus les caractéristiques d'image sont extraites et plus le réseau devient performant au niveau de l'extraction des caractéristiques et de la classification des images.

II.14.2. Forme du filtre : Les formes de filtre varient grandement dans la littérature. Ils sont généralement choisis en fonction de l'ensemble de données.[89]

II.15. Méthodes de régularisation :

Cette technique clé de [machine learning](#) vise à limiter le surapprentissage et à contrôler l'erreur de type *variance* pour aboutir à de meilleures performances.

Il existe plusieurs méthodes de régularisation :

II.15.1. Dropout :

Dropout est une nouvelle méthode de régularisation qui a permis d'améliorer considérablement les résultats et la performance des réseaux de neurones. Cette méthode consiste à éteindre et rallumer

les neurones aléatoirement, dans le cadre d'entraînements successifs. Une fois les séries d'entraînements terminées, on rallume tous les neurones et on utilise le réseau comme d'habitude. Cette technique a montré non seulement un gain dans la vitesse d'apprentissage, mais en déconnectant les neurones, on a aussi limité des effets marginaux, rendant le réseau plus robuste et capable de mieux généraliser les concepts appris [90]

II.15.2. Drop-connect : Un peu la même chose que Dropout, mais sur les connexions, nécessite de ré-implémenter un nouveau module `dropconnect` linéaire, au lieu de désactiver des neurones, on va simplement désactiver les connexions entrantes (toujours de façon aléatoire) sur une couche depuis la précédente. D'un point de vue du réseau, cela revient à instancier les valeurs des poids des connexions à 0.

II.15.3. Pooling Stochastique :

Le pooling stochastique est une méthode de régularisation inspirée par le dropout. Au lieu de toujours capturer la plus forte activité au sein de chaque région de pooling comme le fait le max-pooling, le pooling stochastique sélectionne les activations de manière aléatoire en fonction d'une distribution multinomiale [91].

II.16. Etat de l'art :

Face à l'augmentation continue du nombre de véhicules dans le monde, il est nécessaire d'apporter de nouvelles solutions de gestion du trafic. Il devient parfois extrêmement difficile de trouver une place de stationnement disponible, notamment en zones urbaines denses.

Les techniques de détection de l'occupation d'un parking revêtent une grande importance pour une gestion efficace des parkings. Connaître en temps réel la disponibilité de places de parking gratuites et communiquer avec les utilisateurs peut être une grande aide pour réduire les files d'attente, améliorer l'évolutivité et le temps nécessaire pour trouver une place vide dans un parking. Des solutions sont proposées pour la détection des occupations des stationnements qui sont basées sur les différents algorithmes du machine learning

Giuseppe Amato*, Fabio Carrara*, Fabrizio Falchi*, Claudio Gennaro* and Claudio Vairo*[92] deux architectures CNN nommées mAlexNet et mLeNet ont été proposées. Les modèles proposés comportent moins de cinq couches et prennent en entrée une image RVB 224x224. Pour la partie expérimentale ils sont utilisés PKlot et CNRpark

Afin d'évaluer les performances des deux architectures, ils ont effectué des expériences en utilisant CNRpark (A et B), considérant deux scénarios d'application possibles: un scénario à une seule caméra, dans lequel les données de train et de test proviennent de la base de données. Même point

de vue, et un scénario à plusieurs caméras, dans lequel les données d'apprentissage et les données de test proviennent de points de vue différents.

Les deux CNN, lorsqu'ils sont testés dans le scénario avec une seule caméra, fonctionnent mieux sur le sous-ensemble CNRPark A. le modèle mAlexNet atteint une précision de 99.60% dans le CNRPark A. Cependant, de très bons résultats sont également obtenus pour le sous-ensemble du CNRPark B,

Une architecture performante a été proposée par Giuseppe Amato, Fabio Carrara, Fabrizio Falchi, Claudio Gennaro, Carlo Meghini, Claudio Vairo en 2016 [93], pour la détection visuelle d'occupation des stationnements, l'architecture est suffisamment petite pour fonctionner sur une caméra intelligente, Ils ont utilisées trois couches convolutions et 2 couches entièrement connectés, l'architectures contient environ 1 1340 paramètres

Ils ont comparés ses performances en ce qui concerne l'utilisation de l'architecture AlexNet, pour valider leurs approches, ils ont construit une base de données appelée CNRPark-EXT, de plus ils ont testés leurs méthode avec PKLot, , leurs architecture atteint des valeurs de précision de 98,27% dans UFPR04 / PUC et 90,13% dans la configuration CNRParkOdd / CNRParkEven et 90,71% dans la configuration CNR-375ParkEven / CNRPark.

Ching-Chun et Sheng-Jyh (Huang et Sheng-Jyh (2010)) [94] ont également proposé une approche établie sur l'apprentissage pour l'identification de l'état des places de parking rangée par rangée. Leur modèle d'apprentissage est fondé sur un framework bayésien hiérarchique. Il inclut trois étapes à savoir le modèle de classification, le modèle d'adjacence et le modèle sémantique. Les auteurs ont montré que le modèle qu'il ont proposé permet de gérer efficacement le problème d'occlusion inter-objet. En effet, sur 1500 images de test, les taux de fausses acceptations et de faux rejets sont respectivement 3.2% et 2%.

Paulo R.L. de Alencar, S. Oliveira, Alceu S. Britto, Eunelson J. Silva Jr, Alessandro L. [95] Ils ont présentés l'ensemble de données PKLot qui est composé de 695 899 images capturées à partir de deux parking avec trois vues différentes de la caméra, Les images disponibles montrent une grande variation de luminance car elles ont été capturées sous différentes conditions (ensoleillé, pluvieux et couvert nuageux) sans aucun contrôle sur l'illumination, ils ont évalué deux descripteurs texturaux, les modèles binaires locaux et la quantification de la phase locale avec un classificateur support vector machine (SVM) pour détecter les places de stationnement vacantes, et ils ont démontrés que les descripteurs à base de texture sont une bonne alternative pour distinguer les espaces de stationnement vides et occupés. Dans les expériences où le même point de vue a été utilisé pour le training et le test, ils ont atteint des taux de reconnaissance exceptionnels, plus de 99%, Le principal défi, réside dans la construction d'un classificateur général qui est en mesure de détecter les places de stationnement des parcs de stationnement qui n'ont pas été utilisés pour le training. En ce sens, le

meilleur résultat obtenu par le classificateur à base de texture était d'environ 89 %, d'après cet article ils ont dit que la baisse observée en termes de performances montre que des investigations supplémentaires sont nécessaires pour créer des schémas de classification moins dépendants de l'ensemble du Training,

II.17. Conclusion:

Dans ce chapitre nous avons exploré le domaine de classification des images en utilisant les approches de la machine learning issu les notions de base de la machine learning et leur algorithmes. L'approche deep learning en particulier les CNN pour la classification des images ont été abordées ce qui nous a permis par la suite de proposer dans le chapitre suivant notre modèle CNN.

Chpitr 3

III.1. Introduction :

Ce chapitre est consacré pour la présentation de l'architecture de notre modèle et son implémentation pour entraîner le modèle nous allons exposer les caractéristiques de la base de données PKlot et CNRPark puis nous allons discuter sur les résultats obtenus.

III.2. Les outils et les librairies utilisés :

III.2.1. TensorFlow :

TensorFlow est une bibliothèque open source de Machine Learning, Créé par l'équipe Google Brain en 2011, sous la forme d'un système propriétaire dédié aux réseaux de neurones de Deep Learning, est une bibliothèque de logiciels open source pour le calcul numérique utilisant des graphes de flux de données. Les nœuds des graphes représentent des opérations mathématiques, tandis que les arêtes des graphes représentent les tableaux de données multidimensionnels (tenseurs) qui circulent entre eux. Cette architecture flexible vous permet de déployer des calculs sur un ou plusieurs processeurs ou GPU sur un ordinateur de bureau, un serveur ou un périphérique mobile sans réécrire le code [96].

Cette bibliothèque permet notamment d'entraîner et d'exécuter des réseaux de neurones pour la classification des chiffres écrits à la main, la reconnaissance d'image, les plongements de mots, les réseaux de neurones récurrents, les modèles sequence-to-sequence pour la traduction automatique, ou encore le traitement naturel du langage [97].



Figure III.1. logo de tensorflow

III.2.2. Keras :

Keras est une API de réseaux de neurones de haut niveau, écrite en Python et capable de s'exécuter sur TensorFlow, CNTK ou Theano.

Il a été développé pour permettre une expérimentation rapide [98].

- Permet un prototypage simple et rapide (convivialité, modularité et extensibilité)

- Prend en charge les réseaux convolutionnels et les réseaux récurrents, ainsi que les combinaisons des deux
- Fonctionne de manière transparente sur CPU et GPU.



FigureIII.2.logo de keras

III.2.3. Python :

Python est un langage objet interprété de haut niveau, créé au début des années quatre-vingt-dix par Guido Van Rossum au Centrum voor Wiskunde à Informatica, Amsterdam

Python est un langage de programmation facile à utiliser et puissant. Il offre des structures de données puissantes de haut niveau et une approche simple mais réelle de la programmation orientée-objet. La syntaxe élégante de python et le typage dynamique, ajoutés à sa nature interprétée, en font un langage idéal pour écrire des scripts et pour le développement rapide d'applications dans de nombreux domaines et sur la plupart des plates-forme[99]



FigureIII.3.logo de python

III.3. Configuration utilisé dans l'implémentation :

La configuration du matériel utilisé dans notre implémentation est :

- Un pc portable Dell i7 CPU 3.00 GHZ
- RAM de taille 8 GO
- Système d'exploitation Windows 8.1 64 bit

III.4. DataSets :

Pour le datasets nous avons utilisé PKLot , et CNRPark :

III.4.1. PKLot :

La base de donnée PKLot contient 12 417 images de parcs de stationnement et 695 899 images de places de stationnement segmentées à partir de ceux-ci, qui ont été vérifiées et étiquetées manuellement. Toutes les images ont été acquises sur les parkings de l'Université fédérale de Parana (UFPR) et de l'Université catholique pontificale de Parana (PUCPR), tous deux situés à Curitiba[100].

Les images sont organisées en trois répertoires, Chaque répertoire contient trois sous-répertoires pour différentes conditions météorologiques (nuageux, pluvieux et ensoleillé).À l'intérieur de chaque sous-répertoire, les images sont organisées par une date d'acquisition.

III.4.2. CNRPark :

L'ensemble de données CNRPark contient des images du stationnement ramassé en jours déférent de juillet 2015, à partir de 2 caméras distinctes A et B, qui ont été placés pour avoir des perspectives et angles de vue différent[101].

III.5. Architecture de notre modèle :

Notre travail a pour objectif de classifier des images associées aux places de stationnement dans un parking, et nous ne devons distinguer que deux classes: si un véhicule existe ou non dans une image d'espace de stationnement donnée.

Notre architecture est nommée m-ResNet, elle est constituée de trois couches de convolution dont chaque couche utilise un filtre de taille 3*3, et nous proposons d'utiliser 32 filtre dans chaque couche, toutes les couches convolutives conv1-3 avec une rectification linéaire Relu ,cette fonction force les neurones à retourner des valeurs positive, Une couche de Maxpooling est appliquée après chaque couche de convolution, afin de réduire la taille de l'image et la quantité des paramètres, une dilatation a été appliqué aussi pour chaque couche de convolution, Le raison pour laquelle nous utilisons la convolution dilatée est que nous devons éviter d'apprendre trop en profondeur. La convolution dilatée permet à un modèle d'apprendre de grandes caractéristiques tout en ignorant les petites. Deux couches entièrement connectées ont été ajoutées, Le nombre des unités pour les deux

couches égale à 1048, la couche de sortie avec une fonction sigmoïde car notre travail consiste d'une classification binaire.

Pour éviter le sur apprentissage, on a utilisé la régularisation dropout, Le terme «Dropout» se réfère à des unités ignorées, dans un réseau d'neurones pendant la phase d'entraînement, les unités ignorées ne sont pas considérées pendant une passe avant ou arrière particulière

Rappelons qu'une image est un ensemble de pixels. Chaque pixel étant un triplet de valeurs (R , G , B), notre modèle prend en entrée une image de taille ($54*32*3$)

Net	Conv1	Conv2	Conv3	Fc4	Fc5	Fc6
Load model	54*32*32	54*32*32	54*32*32	4096	4096	2
CarNet	Pool 2*2	Pool 2*2	Pool 2*2	Relu	Relu	seg
Notre	54*32*32	27*16*32	13*8*32	1048	1048	1
modèle	Pool 2*2	Pool 2*2	Pool 2*2	Relu	Relu	seg
m-ResNet						

Tableau III.1: architecture du modèle

Layer (type) Output Shape Param #

conv2d_1 (Conv2D) (None, 54, 32, 32) 896

batch_normalization_1 (Batch Normalization) (None, 54, 32, 32) 128

max_pooling2d_1 (MaxPooling2D) (None, 27, 16, 32) 0

conv2d_2 (Conv2D) (None, 27, 16, 32) 9248

batch_normalization_2 (Batch Normalization) (None, 27, 16, 32) 128

max_pooling2d_2 (MaxPooling2D) (None, 13, 8, 32) 0

conv2d_3 (Conv2D) (None, 13, 8, 32) 9248

batch_normalization_3 (Batch Normalization) (None, 13, 8, 32) 128

max_pooling2d_3 (MaxPooling2 (None, 6, 4, 32) 0

flatten_1 (Flatten) (None, 768) 0

dense_1 (Dense) (None, 1048) 805912

dropout_1 (Dropout) (None, 1048) 0

dense_2 (Dense) (None, 1048) 1099352

dropout_2 (Dropout) (None, 1048) 0

dense_3 (Dense) (None, 1) 1049

Total params: 1,926,089

Trainable params: 1,925,897

Non-trainable params: 192

Tableau III.2 Configuration du modèle

III.6. Résultat obtenue et discussion:

a. Résultat obtenue pour puc puc :

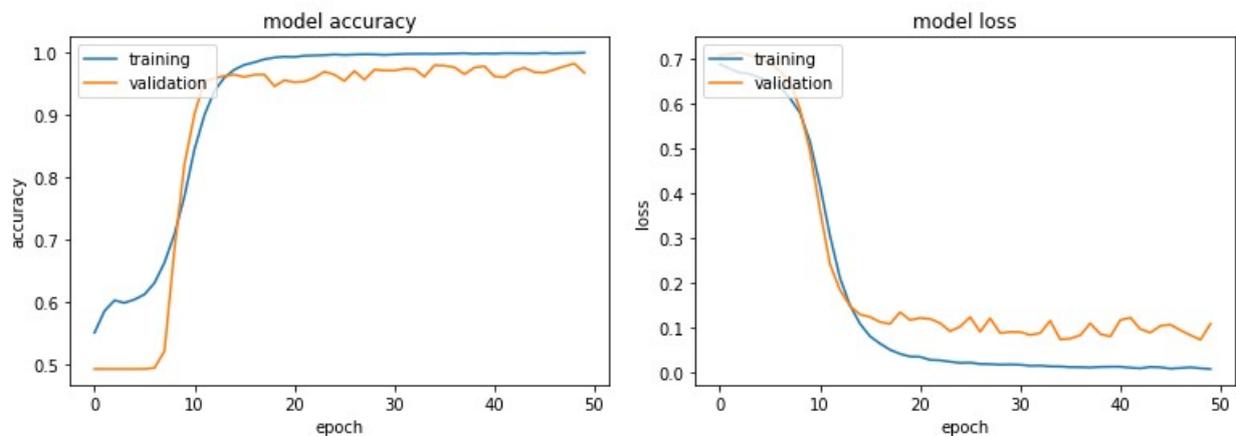


Figure III.4: training Puc- test Puc

b. Résultat obtenue pour puc-UFPR04 :

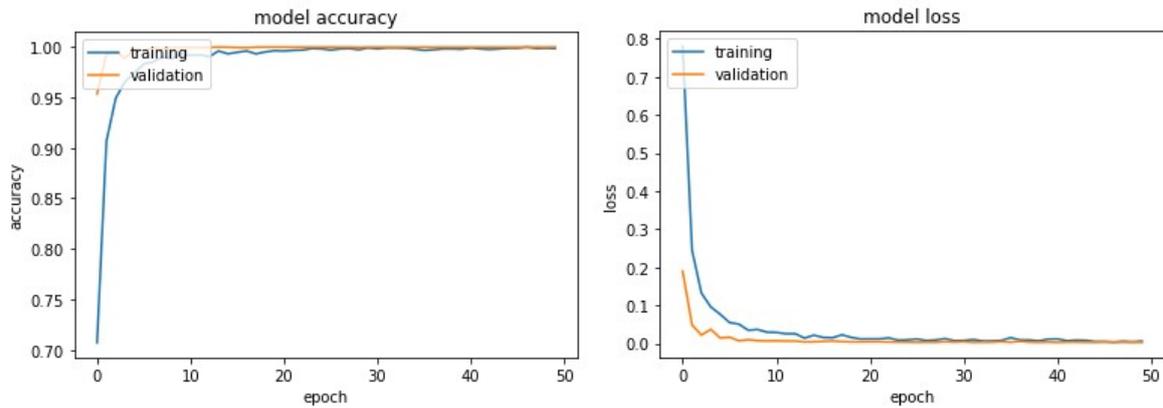


Figure III.5: training Puc- test UFPR04

c. Résultat obtenue pour PUC-UFPR05:

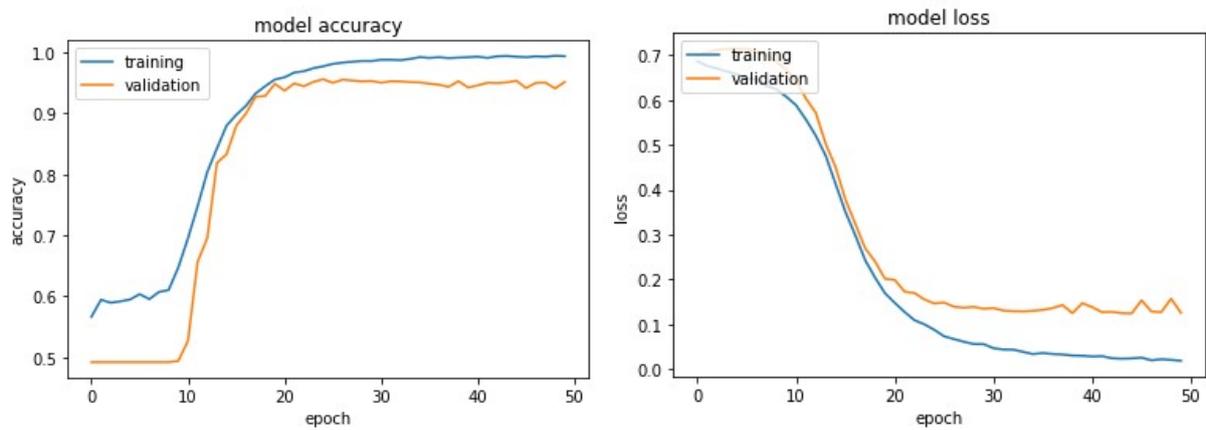


Figure III.6: training Puc- test UFPR05

d. Résultat obtenue pour UFPR04- UFPR04 :

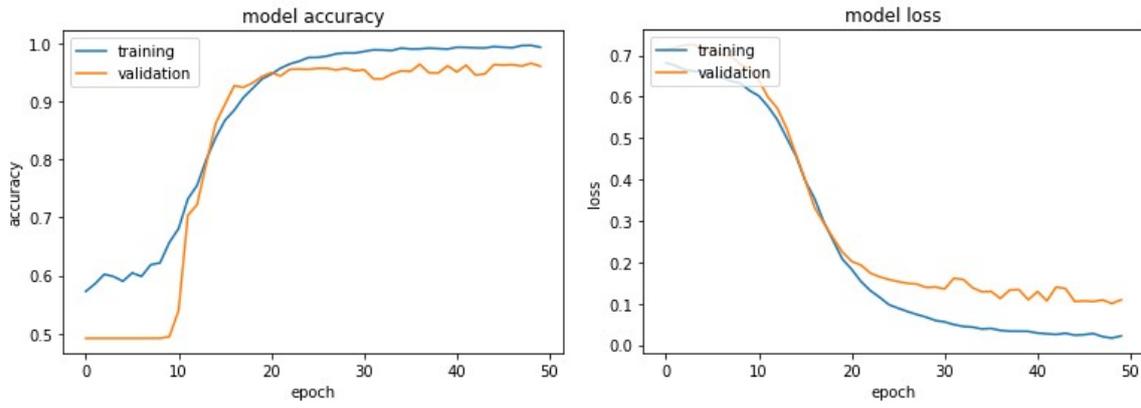


Figure III.7: training UFPR04- test UFPR04

E.Résultat obtenue pour UFPR04- UFPR045 :

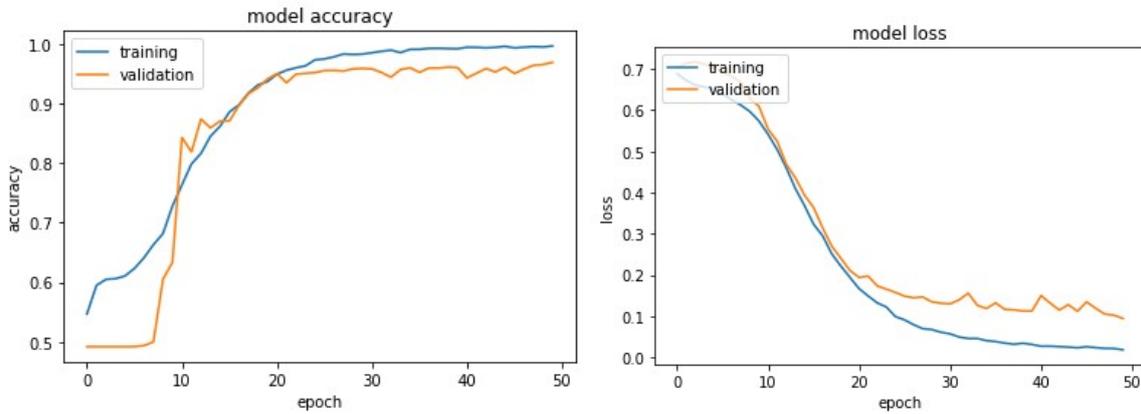


Figure III.8:training UFPR04-test UFPR05

f.Résultat obtenue pour UFPR05- UFPR05 :

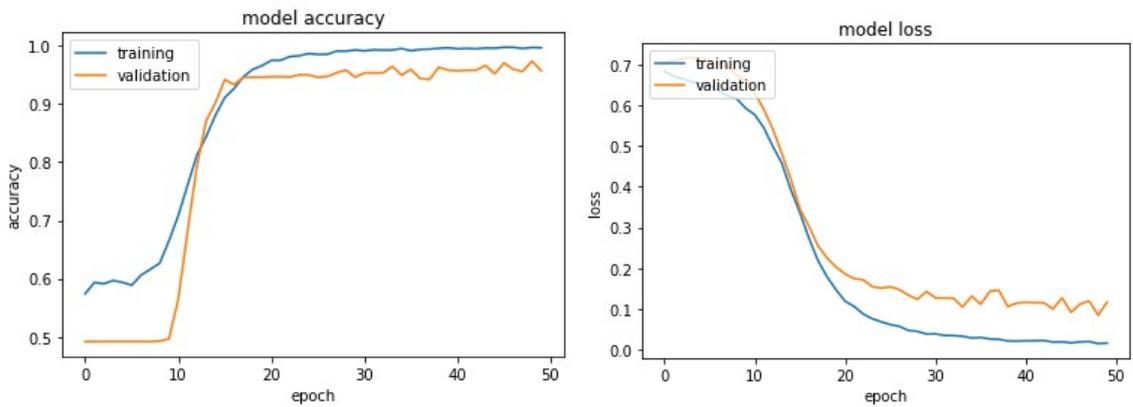


Figure III.9: training UFPR05- test UFPR05

g.Résultat obtenue pour UFPR05- UFPR04

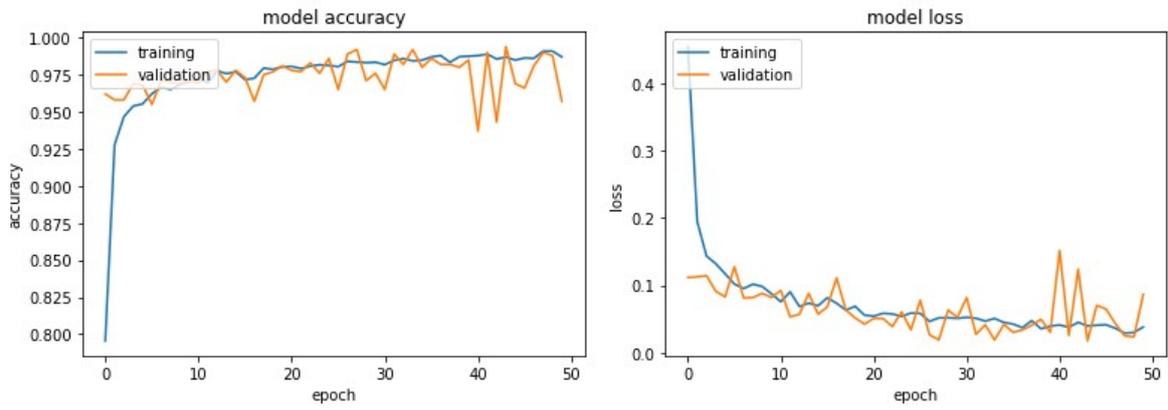


Figure III.10: training UFPR05-test UFPR04

h. Résultat obtenue pour UFPR05-test Puc :

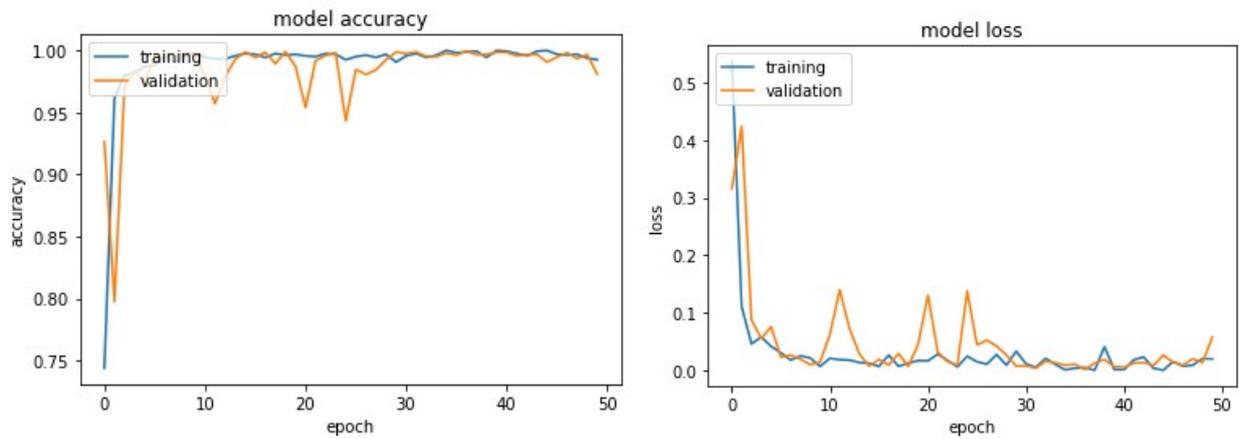


Figure III.11: training UFPR05-test Puc

I. Résultat obtenue pour UFPR04-test Puc :

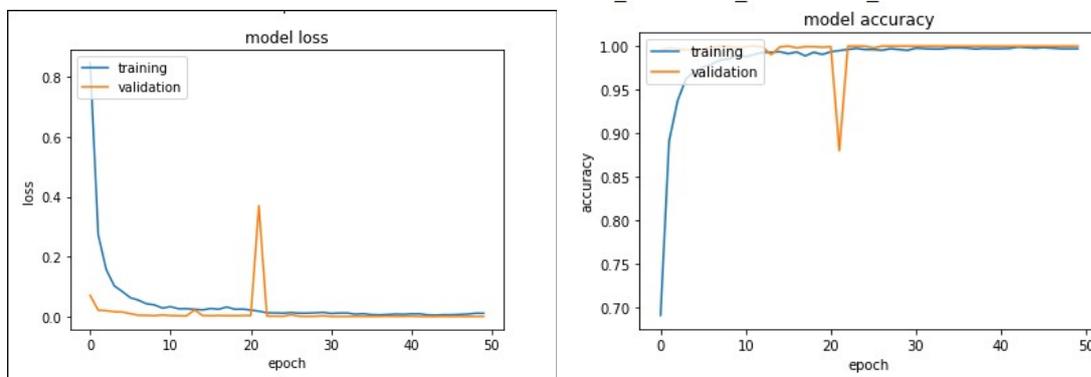


Figure III.12: training UFPR04-test PUC

I.Résultat obtenue pour CNRPARK-test CNRPARK :

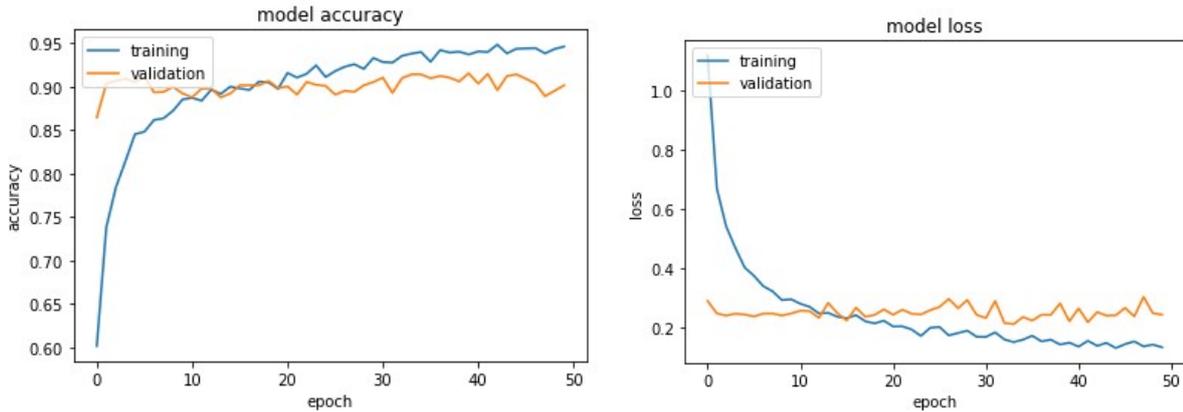


Figure III.13:training UFPR04-test PUC

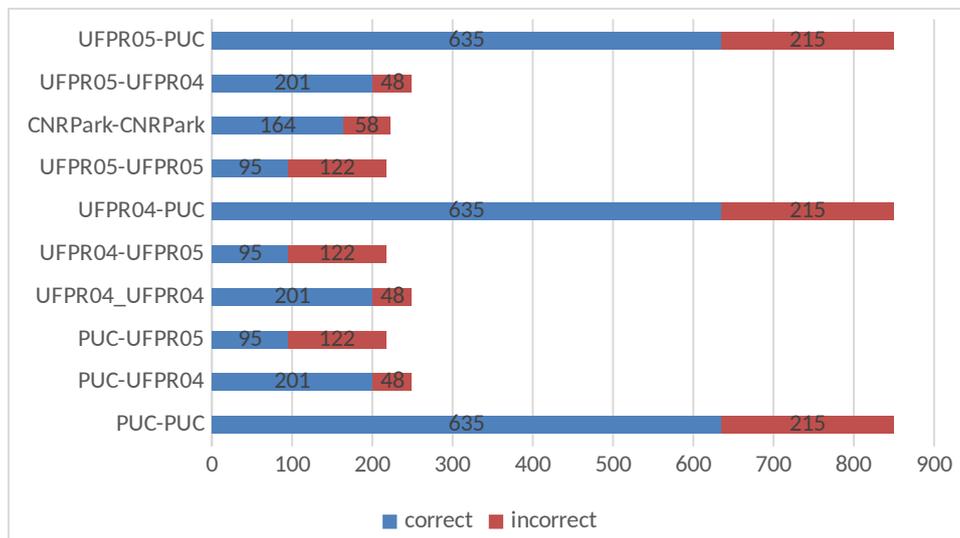


Figure III.14:Nombre des images mal et bien classé par la base de donnée

Après l’analyse des résultats obtenus, on constate les remarques suivantes :

D'après les figures : **Figure III.4, Figure III.5 et Figure III.7**, la précision dans l'apprentissage et la validation est presque stable au début puis elle augmente avec le nombre d'époque jusqu'à elle est stable. De même l'erreur d'apprentissage et de validation est stable au début puis il diminue.

D'après les figures : **Figure III.3, Figure III.6 et Figure III.9**, la précision de l'apprentissage augmente avec le nombre d'époque, jusqu'à elle est stable, ceci reflète qu'à chaque époque le modèle apprend plus d'information et la même chose pour la validation.

L’erreur d'apprentissage et validation est diminué avec le nombre d'époque.

D'après la **Figure III.8** la précision de l'apprentissage augmente avec le nombre d'époque jusqu'à elle est stable, par contre la validation elle est presque stable entre [0,92 0,97].

L'erreur d'apprentissage est diminuée, et la validation est presque stable entre [0 et 0,1]

D'après la **Figure III.3** la précision de l'apprentissage augmente avec le nombre d'époque jusqu'à elle est stable, par contre la validation elle est presque stable entre [0,95 0,99].

L'erreur d'apprentissage est diminuée, et la validation est presque stable entre [0 et 0,2]

D'après la **Figure III.10** la précision de l'apprentissage augmente avec le nombre d'époque jusqu'à elle est stable, par contre la validation elle est stable jusqu'à l'époque 20 elle diminue puis elle augmente

L'erreur d'apprentissage est diminuée.

J. Tableau de comparaison des résultats

Le tableau ci-dessous montre les différents résultats obtenus sur les différents datasets :

Non du modele	Sous ensemble D'apprentissage	Sous ensemble De test	Précision (%)	La moyenne (%)	Le nombre de paramètres	
					Mise a jours lors de apprentissage	Non mise a jours lors de l'apprentissage
CarNet	UPF04	UPF04	95.6	97.04	65,725,121	0
	UPF04	UPF05	97.6			
	UPF04	PUC	98.3			
	UPF05	UPF04	95.2			
	UPF05	UPF05	97.5			
	UPF05	PUC	98.4			
	PUC	UPF04	94.4			
	PUC	UPF05	97.6			
	PUC	PUC	98.8			
RosNet	UPF04	UPF04	98.27	96.64	1,925,897	192
	UPF04	UPF05	94.92			
	UPF04	PUC	94.75			
	UPF05	UPF04	96.92			
	UPF05	UPF05	92.89			
	UPF05	PUC	95.98			
	PUC	UPF04	96.59			
	PUC	UPF05	98.99			
	PUC	PUC	98.94			

Tableau III.3: résultat d'exécution de code avec PKLot

Data set	Training	Testing	Val-accuracy	Valid-accuracy	Accuracy
CNRPARK	CNRPARK	CNRPARK	0.9460	0.9014	0.9826

Tableau III.4: résultat d'exécution de code avec CNRPARK

Les tableaux : **Tableau III.2 et Tableau III.3** montre l'architecture utilisée avec plusieurs datasets . Les résultats obtenus sont exprimés en termes de précision d'apprentissage, de validation, de test. Le temps d'exécution est trop couteux. Le Data set PUC a présenté les meilleurs résultats trouvés. Le nombre d'époque et de couches de convolution reflètent ces bons résultats, cependant le temps d'exécution était très couteux (à cause du nombre des images).

Les résultats obtenus se sont améliorés à mesure que nous avons approfondie notre réseau et augmenté le nombre d'époque. La base d'apprentissage est également un élément déterminant dans les réseaux de neurones convolutionnels, il faut avoir une base d'apprentissage de grande taille pour aboutir à des meilleurs résultats.

Data set	Means accuracy
CarNET	0.9704
Load modele	
m-RosNET	0.9646
notre modele	

Tableau III.5 : comparaison d'accuracy avec CarNet

D'après le **tableau III.5** , nous avons comparé les résultats de notre architecture m-ResNet avec l'architecture CarNet, nous avons obtenu une accuracy moins que CarNet mais avec un moins nombre de paramètres.

III.7. Coude sources :

- **Input_shape (54, 32,3)** : la taille de l'image en entrée.

- **pool_size=(2,2)** : taille de pool.
- **Batch_size** : définit le nombre d'échantillons qui seront propagés sur le réseau.
- **ImageDataGenerator** : classe qui permet aux utilisateurs d'effectuer une augmentation d'image d'une manière très facile.
- **rescale=1. /255** : est un paramètre qui normalise les valeurs des pixels de l'image de manière à avoir une moyenne de zéro et un écart type de 1.
- **Epoche** : c'est le nombre d'itérations d'entraînement.

1. Classifier=séquentiel ()

- La fonction séquentielle initialise une pile linéaire de couches.

2. classifier.add(Conv2D(32,(3,3),padding='same',dilation_rate=1,input_shape=(54,32,3), activation='relu')):

- Cette commande permet de créer 32 features maps en utilisant un filtre de taille 3*3 et un padding égal à la taille de l'image précédente, avec une dilatation standard=1, et une fonction d'activation ReLu.

3. Classifier.add (BatchNormalization())

- Cette commande permet d'accélérer l'apprentissage du réseau en réduisant le décalage de covariable interne.

4. classifier.add (MaxPooling2D (pool_size= (2,2)))

- Cette commande permet de réduire la taille de l'image tout en conservant les caractéristiques importantes, en utilisant un filtre de taille 2*2.

5. classifier.add (Flatten())

- Un flatten qui va créer le vecteur final à envoyer au réseau de neurones artificiels (dense).

6. classifier.add (Dense (units=1048, activation='relu'))

- Cette commande permet de créer une couche entièrement connectée avec un nombre d'unité égal à 1048, avec une fonction d'activation relu.

7. classifier.add (Dropout (0.5))

- Cette commande exprime une régularisation dropout qui permet d'éviter le surapprentissage, elle consiste à désactiver les sorties de neurones avec une probabilité 0.5, 0.5 signifie de jeter au hasard la moitié des couches.

8. **classifier.add (Dense (units=1, activation='sigmoid'))**

- Cette commande permet de créer une couche de sortie avec une fonction d'activation sigmoid car le travail consiste d'une classification binaire.

9. **classifier.compile(loss='binary_crossentropy',optimizer=optimizers.SGD(lr=1e-4, momentum=0.9),metrics=['accuracy'])**

- Pour compiler le modèle on utilise cette commande, elle prend trois paramètres qui sont : la fonction Loss, l'Optimizer et Metrics, pour l'optimizer on a choisi le sgd (Stochastic Gradient Descent), Nous spécifions une fonction de perte que notre optimizer minimisera. Dans ce cas, puisque nous travaillons avec un problème à deux classes, nous utilisons la fonction `binary_crossentropy` et Puisque nous faisons un problème de classification, la métrique de précision (accuracy) est un bon choix.

10. **Classifier.summary() :**

- Permet de prévisualiser la disposition et la taille des paramètres de notre réseau.

11. **.Score = classifier.evaluate_generator (test_set, 5000, workers=12)**

- Cette commande permet d'évaluer notre modèle sur la base de test.

III.8. Conclusion :

Nous avons présenté dans ce chapitre une approche de classification basée sur les réseaux de neurones convolutionnels, pour cela on a testé notre modèle avec différents base de données et on a montré les différents résultats, et on a fait une comparaison avec le modèle CarNet.

Conclus

Notre société devient de plus en plus intelligente à l'heure actuelle, et comme tous les autres domaines de l'intelligence artificielle, le domaine de la classification d'image a connu une évolution majeure depuis l'apparition du Deep Learning

Dans ce projet nous avons proposé une approche pour la détection de l'état du stationnement, qui est basé sur les réseaux de neurones convolutionnel pour classer l'occupation des places de stationnement, et pour cela nous avons parlé des différents types des algorithmes du machine Learning et nous avons discuté des notions fondamentales des deep learning en général et des réseaux de neurones convolutionnels en particulier, en présentant les différents types de couches utilisées dans la classification. et aussi les méthodes de régularisation.

Afin de valider notre approche, nous avons utilisé deux bases de données qui sont PKLot et CNRPark et on a testé notre modèle avec les différents bases de données afin d'obtenir des meilleurs résultats en terme de précision et d'erreur, et nous avons comparé notre modèle m-ResNet avec le modèle CarNet et nous avons obtenue des bonnes résultats moins que CarNet mais avec un moins nombre de paramètres, pour la configuration nous avons utilisé spider

Bibliographie:

- [1] (<http://thesis.univ-biskra.dz/2271/6/Chapitre%2003.pdf>)
- [2](http://www-lagis.univ-lille1.fr/~bonnet/Outils_Simul/complements_image.pdf)
- [3](https://www-master.ufr-info-p6.jussieu.fr/2006/Ajouts/Master_esj_2006_2007/IMG/pdf/IntroDefinitions.pdf)
- [4](<https://bu.univ-ouargla.dz/master/pdf/GOUARAH-BOUAFFANE.pdf?idmemoire=32>)
- [5](http://www.numerical-tours.com/matlab/introduction_6_elementary_fr/)
- [6](<https://www.google.com/url?sa=t&rct=j&q=&esrc=s&source=web&cd=3&cad=rja&uact=8&ved=2ahUKEwiJisrk7ILjAhVEAmMBHbD6CjEQfjACegQIAxAB&url=http%3A%2F%2Fwww.clashinfo.com%2Fdico%2Fdefinition-p%2Fart7-pixel.html&usg=AOvVaw2P4jfXcNVDQdYBZ1vG4J53>)
- [7]:(https://www.google.com/url?sa=t&rct=j&q=&esrc=s&source=web&cd=3&cad=rja&uact=8&ved=2ahUKEwj05IKQ74LjAhXjz4UKHZASCAgQFjACegQIBBAC&url=http%3A%2F%2Fbib.univ-oeb.dz%3A8080%2Fjspui%2Fbitstream%2F123456789%2F8078%2F1%2FMIL-00050.pdf&usg=AOvVaw1uLxD_fnM_oZqrTtsvSNWD)
- [8]:(<https://www.google.com/url?sa=t&rct=j&q=&esrc=s&source=web&cd=1&cad=rja&uact=8&ved=2ahUKEwjbnbfD8ILjAhVDSxoKHd6uBB8QFjAAegQIBBAC&url=http%3A%2F%2Fwww-lisic.univ-littoral.fr>)

[%2F~vandenbroucke%2Fenseignement%2FCM_TI_01-Image-numerique.pdf&usg=AOvVaw3TSvUq6O6sOkhZz_gmtEWO\)](#)

[9]:([\[10\]: \(\[https://www.memoireonline.com/12/09/3040/m_La-liaison-automatique-des-plusieurs-images-percues-sur-un-scanner4.html\]\(https://www.memoireonline.com/12/09/3040/m_La-liaison-automatique-des-plusieurs-images-percues-sur-un-scanner4.html\)\)](https://www.google.com/url?sa=t&rct=j&q=&esrc=s&source=web&cd=4&cad=rja&uact=8&ved=2ahUKEwjjw_cz-8YLjAhUjAGMBHskzBuMQFjADegQIBhAC&url=https%3A%2F%2Fedu1d.ac-toulouse.fr%2Fpolitique-educative-31%2Fsite-ressources31%2Ffiles%2F04-photofiltre7-resolution-definition-simplifie.pdf&usg=AOvVaw2rO_ITBWglVb3_whs3kSXY))</p></div><div data-bbox=)

[11]:(https://www-master.ufr-info-p6.jussieu.fr/2010/spip.php?action=accéder_document&arg=103&cle=540a02c1737edc60917affb273a0b9dcb90dd421&file=pdf%2Ffiltrage-optim.pdf)

[12]: (https://www.memoireonline.com/12/09/3040/m_La-liaison-automatique-des-plusieurs-images-percues-sur-un-scanner4.html)

[13]:(https://www-master.ufr-info-p6.jussieu.fr/2010/spip.php?action=accéder_document&arg=103&cle=540a02c1737edc60917affb273a0b9dcb90dd421&file=pdf%2Ffiltrage-optim.pdf)

[14]:(https://www-master.ufr-info-p6.jussieu.fr/2010/spip.php?action=accéder_document&arg=103&cle=540a02c1737edc60917affb273a0b9dcb90dd421&file=pdf%2Ffiltrage-optim.pdf)

[15]:(<https://www-lisic.univ-littoral.fr/~porebski/Vision%20et%20Traitement%20d'images.pdf>)

[16]:(<http://deptinfo.unice.fr/~ol/L1MI-UA/memo6.pdf>)

[17]:(https://www.google.com/url?sa=t&rct=j&q=&esrc=s&source=web&cd=2&cad=rja&uact=8&ved=2ahUKEwiI1ezb-ILjAhUOHxoKHeCiCfUQFjABegQIABAC&url=http%3A%2F%2Fflaris.univ-angers.fr%2F_resources%2Flogo%2FDEATouhamiSami.pdf&usg=AOvVaw3yGpojBOUkWhKvjMwUohuy)

[18]:(<http://www.formules-physique.com/categorie/1050>)

[19]:(<http://www.infotographiste.com/lecons/lecons-de-photographie/contraste/>)

[20]:(<https://www.memoireonline.com/05/14/8871/La-recherche-d-images-par-la-semantique.html>)

[21]:(https://www-master.ufr-info-p6.jussieu.fr/2006/Ajouts/Master_esj_2006_2007/IMG/pdf/IntroDefinitions.pdf)

[22]:(https://www-master.ufr-info-p6.jussieu.fr/2006/Ajouts/Master_esj_2006_2007/IMG/pdf/IntroDefinitions.pdf)

[23]:(https://www-master.ufr-info-p6.jussieu.fr/2006/Ajouts/Master_esj_2006_2007/IMG/pdf/IntroDefinitions.pdf)

[24]:(https://www.google.com/url?sa=t&rct=j&q=&esrc=s&source=web&cd=3&cad=rja&uact=8&ved=2ahUKEwjmnLPitpTjAhUeBGMBHZaxA_EQFjACegQIAhAC&url=https%3A%2F%2Fpdfs.semanticscholar.org%2F8479%2Fbac9c0a819e04f92588d81b91dee4ecbbb11.pdf&usg=AOvVaw1ZpAlkbtWlzM-sv5rJPF5X)

[25]:(https://www.google.com/url?sa=t&rct=j&q=&esrc=s&source=web&cd=3&cad=rja&uact=8&ved=2ahUKEwjmnLPitpTjAhUeBGMBHZaxA_EQFjACegQIAhAC&url=https%3A%2F%2Fpdfs.semanticscholar.org%2F8479%2Fbac9c0a819e04f92588d81b91dee4ecbbb11.pdf&usg=AOvVaw1ZpAlkbtWlzM-sv5rJPF5X)

[26]:(https://www.google.com/url?sa=t&rct=j&q=&esrc=s&source=web&cd=3&cad=rja&uact=8&ved=2ahUKEwjmnLPitpTjAhUeBGMBHZaxA_EQFjACegQIAhAC&url=https%3A%2F%2Fpdfs.semanticscholar.org%2F8479%2Fbac9c0a819e04f92588d81b91dee4ecbbb11.pdf&usg=AOvVaw1ZpAlkbtWlzM-sv5rJPF5X)

[27]:(Gilson Braviano. ‘ ‘ Logique floue en segmentation d’images: seuillage par Entropie et structures pyramidales irrégulières ‘ ‘, thèse de doctorat à L’Université Joseph Fourier-

Grenoble 1. Octobre 1995.)

[21] (M. H. Yang, et N. Ahuja. ‘‘Detecting Human Faces in Color Images’’. Proc. IEEE Int'lConf. Image Processing, vol 1, pp 127-130, 1998.)

[22]:Gong Y., Proietti G., Faloutsos C.(1998), ‘‘Image Indexing and Retrieval Based on HumanPerceptual Color Clustering’’. Proc. Of International Conference on Computer Vision andPattern Recognition(CVPR), June 1998)

[23](<https://www.google.com/url?sa=t&rct=j&q=&esrc=s&source=web&cd=25&ved=2ahUKEwigmfCR0pTjAhWx4YUKHevkB5wQFjAYegQIBBAC&url=http%3A%2F%2Fdepot-e.uqtr.ca%2F5167%2F1%2F030350177.pdf&usg=AOvVaw15AbvIxyDSzpwSiaz4jzLz>)

[24] (R.M. Haralick. ‘‘ Statistical and structural approaches to texture ‘’. In proceeding of the IEEE. Pages 786-804, may 1979, number 5, vol.67.

[30](H. Tamura, S. Mori et T. Yamawaki. ‘‘ Texture features corresponding to visual perception ‘’. IEEE Transactions on Systems, Manand Cybernetics, SMC-8(6):460–473,1978.)

[25] (Zhang D. and G. Lu. ‘‘ Review of shape representation and description techniques. Pattern Recognition ‘’, 37:1-19, 2004.à)

[26](https://www.google.com/url?sa=t&rct=j&q=&esrc=s&source=web&cd=1&ved=2ahUKEwif_eG265TjAhVRx4UKHW0GDF4QFjAAegQIABAC&url=http%3A%2F%2Fceur-ws.org%2FVol-547%2F27.pdf&usg=AOvVaw243MStoym3A304fiJIIDbH)

[27] (Daoudi Imane ‘‘ Recherche par similarité dans les bases de données multimédia : application a la recherche par le contenu ’’ thèse de doctorat p 28 ,2008.)

[28] (Jeannim, S. ‘‘ MPEG-7 visual pary of experimentation model, Version 9.0 ‘’ dans ISO/IEC JTC1/SC29/WG11/N3914,55th Mpeg Meeting. Pisa, Italy.2001.pages 27, 39, 41.)

[29] (<https://www.google.com/url?sa=t&rct=j&q=&esrc=s&source=web&cd=1&cad=rja&uact=8&ved=2ahUKEwj7rfne8JXjAhVIAWMBHXFxA5wQFjAAegQIAhAC&url=https%3A%2F%2Fdl.ummt0.dz%2Fbitstream%2Fhandle%2Fummt0%2F647%2FBEDOUHENE%2520SA%25C3%258FDA.AUTO.2011.pdf%3Fsequence%3D1%26isAllowed%3Dy&usg=AOvVaw0HIEAto8TrRYbMtRwP3Ee7>)

[30] (https://www.google.com/url?sa=t&rct=j&q=&esrc=s&source=web&cd=5&cad=rja&uact=8&ved=2ahUKEwiI64zY7ZTjAhVSTBoKHbKaDd0QFjAEegQIABAC&url=http%3A%2F%2Ffliris.cnrs.fr%2FDocuments%2FLiris-2807.pdf&usg=AOvVaw12_bXfOgwcg0w4Zl9doGRl)

[31] (http://www.cesbio.ups-tlse.fr/data_all/theses/HDR_Ducrot_2005.pdf)

[32] ([https://www.google.com/url?](https://www.google.com/url?sa=t&rct=j&q=&esrc=s&source=web&cd=1&cad=rja&uact=8&ved=2ahUKEwial6H__4LjAhVKqxoKHQt8CnsQFjAAegQIABAC&url=http%3A%2F%2Fficube-bfo.unistra.fr%2Ffr%2Fimg_auth.php%2F1%2F12%2FPresentation_Teledetection_ICube_I.pdf&usg=AOvVaw0BVCdl0F2Nak-ykPgUfCJh)

[sa=t&rct=j&q=&esrc=s&source=web&cd=1&cad=rja&uact=8&ved=2ahUKEwial6H__4LjAhVKqxoKHQt8CnsQFjAAegQIABAC&url=http%3A%2F%2Fficube-bfo.unistra.fr%2Ffr%2Fimg_auth.php%2F1%2F12%2FPresentation_Teledetection_ICube_I.pdf&usg=AOvVaw0BVCdl0F2Nak-ykPgUfCJh](https://www.google.com/url?sa=t&rct=j&q=&esrc=s&source=web&cd=1&cad=rja&uact=8&ved=2ahUKEwial6H__4LjAhVKqxoKHQt8CnsQFjAAegQIABAC&url=http%3A%2F%2Fficube-bfo.unistra.fr%2Ffr%2Fimg_auth.php%2F1%2F12%2FPresentation_Teledetection_ICube_I.pdf&usg=AOvVaw0BVCdl0F2Nak-ykPgUfCJh))

[33] :([https://www.google.com/url?](https://www.google.com/url?sa=t&rct=j&q=&esrc=s&source=web&cd=1&cad=rja&uact=8&ved=2ahUKEwjY84zDgIPjAhUiyoUKHRGGDZkQFjAAegQIAhAC&url=http%3A%2F%2Fwww.agroparistech.fr%2FIMG%2Fpdf%2FClassificationNonSupervisee-AgroParisTech.pdf&usg=AOvVaw2xrEVwmmk42ZfQAwTOX_cs)

[sa=t&rct=j&q=&esrc=s&source=web&cd=1&cad=rja&uact=8&ved=2ahUKEwjY84zDgIPjAhUiyoUKHRGGDZkQFjAAegQIAhAC&url=http%3A%2F%2Fwww.agroparistech.fr%2FIMG%2Fpdf%2FClassificationNonSupervisee-AgroParisTech.pdf&usg=AOvVaw2xrEVwmmk42ZfQAwTOX_cs](https://www.google.com/url?sa=t&rct=j&q=&esrc=s&source=web&cd=1&cad=rja&uact=8&ved=2ahUKEwjY84zDgIPjAhUiyoUKHRGGDZkQFjAAegQIAhAC&url=http%3A%2F%2Fwww.agroparistech.fr%2FIMG%2Fpdf%2FClassificationNonSupervisee-AgroParisTech.pdf&usg=AOvVaw2xrEVwmmk42ZfQAwTOX_cs))

[34] ([https://www.google.com/url?](https://www.google.com/url?sa=t&rct=j&q=&esrc=s&source=web&cd=2&cad=rja&uact=8&ved=2ahUKEwj-yovggoPjAhUVD2MBHUpeBYEQFjABegQIBBAC&url=http%3A%2F%2Frepository.usthb.dz%2Fbitstream%2Fhandle%2F123456789%2F1907%2FRESUME.doc%3Fsequence%3D1%26isAllowed%3Dy&usg=AOvVaw3CWRRPbDa73e0CnyVNA4HI)

[sa=t&rct=j&q=&esrc=s&source=web&cd=2&cad=rja&uact=8&ved=2ahUKEwj-yovggoPjAhUVD2MBHUpeBYEQFjABegQIBBAC&url=http%3A%2F%2Frepository.usthb.dz%2Fbitstream%2Fhandle%2F123456789%2F1907%2FRESUME.doc%3Fsequence%3D1%26isAllowed%3Dy&usg=AOvVaw3CWRRPbDa73e0CnyVNA4HI](https://www.google.com/url?sa=t&rct=j&q=&esrc=s&source=web&cd=2&cad=rja&uact=8&ved=2ahUKEwj-yovggoPjAhUVD2MBHUpeBYEQFjABegQIBBAC&url=http%3A%2F%2Frepository.usthb.dz%2Fbitstream%2Fhandle%2F123456789%2F1907%2FRESUME.doc%3Fsequence%3D1%26isAllowed%3Dy&usg=AOvVaw3CWRRPbDa73e0CnyVNA4HI))

[35] (http://eric.unilyon2.fr/~ricco/tanagra/fichiers/fr_Tanagra_Spv_Learning_Curves.pdf)

[36] (https://www.memoireonline.com/03/08/978/m_analyse-acces-education-kinshasa8.html)

[37] ([http://www.ressources-actuarielles.net/EXT/ISFA/1226-02.nsf/0/ff787771527bede6c12580f4006f7e2d/\\$FILE/p1315227%20EL%20AHMER%20OUMAIMA.002.pdf/p1315227%20EL%20AHMER%20OUMAIMA.pdf](http://www.ressources-actuarielles.net/EXT/ISFA/1226-02.nsf/0/ff787771527bede6c12580f4006f7e2d/$FILE/p1315227%20EL%20AHMER%20OUMAIMA.002.pdf/p1315227%20EL%20AHMER%20OUMAIMA.pdf))

[38] ([https://www.google.com/url?](https://www.google.com/url?sa=t&rct=j&q=&esrc=s&source=web&cd=1&cad=rja&uact=8&ved=2ahUKEwiP7cCJiYPjAhVQXBoKHQyFAvsQFjAAegQIAxAC&url=https%3A%2F%2Fhal.archives-ouvertes.fr%2Ftel-01420603v2%2Fdocument&usg=AOvVaw2xAmYDOs-E0q4qcbGoPeb6)

[sa=t&rct=j&q=&esrc=s&source=web&cd=1&cad=rja&uact=8&ved=2ahUKEwiP7cCJiYPjAhVQXBoKHQyFAvsQFjAAegQIAxAC&url=https%3A%2F%2Fhal.archives-ouvertes.fr%2Ftel-01420603v2%2Fdocument&usg=AOvVaw2xAmYDOs-E0q4qcbGoPeb6](https://www.google.com/url?sa=t&rct=j&q=&esrc=s&source=web&cd=1&cad=rja&uact=8&ved=2ahUKEwiP7cCJiYPjAhVQXBoKHQyFAvsQFjAAegQIAxAC&url=https%3A%2F%2Fhal.archives-ouvertes.fr%2Ftel-01420603v2%2Fdocument&usg=AOvVaw2xAmYDOs-E0q4qcbGoPeb6))

[39] (https://www.memoireonline.com/01/12/5185/m_Mesures-et-determinants-de-la-confiance-des-menages-sur-la-situation-economique-au-Cameroun-cas28.html)

[40] (http://www.jms-insee.fr/2018/S08_3_ACTE_TARAYOUN_JMS2018.pdf)

[41] ([https://www.google.com/url?](https://www.google.com/url?sa=t&rct=j&q=&esrc=s&source=web&cd=1&cad=rja&uact=8&ved=2ahUKEwjtr-iXjIPjAhVwD2MBHaIMDiAQFjAAegQIABAB&url=https%3A%2F%2Fopenclassrooms.com%2Ffr%2Fcourses%2F4470531-classez-et-segmentez-des-)

[sa=t&rct=j&q=&esrc=s&source=web&cd=1&cad=rja&uact=8&ved=2ahUKEwjtr-iXjIPjAhVwD2MBHaIMDiAQFjAAegQIABAB&url=https%3A%2F%2Fopenclassrooms.com%2Ffr%2Fcourses%2F4470531-classez-et-segmentez-des-](https://www.google.com/url?sa=t&rct=j&q=&esrc=s&source=web&cd=1&cad=rja&uact=8&ved=2ahUKEwjtr-iXjIPjAhVwD2MBHaIMDiAQFjAAegQIABAB&url=https%3A%2F%2Fopenclassrooms.com%2Ffr%2Fcourses%2F4470531-classez-et-segmentez-des-)

[donnees-visuelles%2F5072281-utilisez-ces-features-pour-classifier-des-images&usg=AOvVaw3ofidMifp1cojMiUpOKiau\)](#)

[42] (<https://www.iflexion.com/blog/machine-learning-image-classification>)

[43] (<https://www.google.com/url?>

[sa=t&rct=j&q=&esrc=s&source=web&cd=1&cad=rja&uact=8&ved=2ahUKEwjUro7Ij4PjAhX76OAKHRjwC3gQFjAAegQIBhAC&url=https%3A%2F%2Fwww.college-de-france.fr%2Fmedia%2Fyann-lecun%2FUPL4485925235409209505_Intelligence_Artificielle_____Y._LeCun.pdf&usg=AOvVaw3E3PikGI1R9OfdB1gDzEv\)](#)

[44] (https://www.dunod.com/sites/default/files/atoms/files/Feuilletage_520.pdf)

[45] (https://complements.lavoisier.net/9782100780808_introduction-au-machine-learning_Chapitre1.pdf)

[46] (https://www.lemonde.fr/pixels/article/2015/07/24/comment-le-deep-learning-revolutionne-l-intelligence-artificielle_4695929_4408996.html)

[47] (https://www.dunod.com/sites/default/files/atoms/files/Feuilletage_520.pdf)

[48] (<https://www.google.com/url?>

[sa=t&rct=j&q=&esrc=s&source=web&cd=2&cad=rja&uact=8&ved=2ahUKEwj-qLeqmIPjAhUU4OAKHV8dA4UQFjABegQIAxAB&url=https%3A%2F%2Fwhatis.techtarget.com%2Ffr%2Fdefinition%2FMachine-Learning&usg=AOvVaw2adz0vzt0pqu5zos00q1rW\)](#)

[49] (https://sites.lesia.obspm.fr/emmanuel-grolleau/files/2017/08/SIS_Presentation_MachineLearning_Grolleau-1.pdf)

[50] (<https://www.nexworld.fr/machine-learning-retour-aux-sources/>)

[51] (<https://www.google.com/url?>

[sa=t&rct=j&q=&esrc=s&source=web&cd=3&cad=rja&uact=8&ved=2ahUKEwiR-vP_oYPjAhV75OAKHTLqBOUQFjACegQIAxAB&url=https%3A%2F%2Fwww.nexworld.fr%2Fmachine-learning-retour-aux-sources%2F&usg=AOvVaw2sdMsjUTxyz6DAqUA7WtCi\)](#)

[52] (<https://www.google.com/url?>

[sa=t&rct=j&q=&esrc=s&source=web&cd=3&ved=2ahUKEwjryqLG4sjiAhWIBGMBHQssB5oQFjACegQIAxAC&url=http%3A%2F%2Fdspace.univbiskra.dz%3A8080%2Fjspui%2Fbitstream%2F123456789%2F11124%2F1%2FDeep_earning_pour_Reconnaissance_du_Visage.pdf&usg=AOvVaw3Sx1vLmFrGC6DWiFzPicXq\)\)](#)

[53] (<https://dataanalyticspost.com/Lexique/overfitting/>)

- [54] (<https://docplayer.fr/amp/3792030-Universite-de-nice-sophia-antipolis-clustering-des-news-clustering-sous-le-projet-zone-el-fouzi-ilhame-06-03-2013.html>)
- [55] (<https://slideplayer.fr/slide/3156840/>)
- [56] <https://www.google.com/url?sa=t&rct=j&q=&esrc=s&source=web&cd=1&cad=rja&uact=8&ved=2ahUKEwjSlcOF4oTjAhV68eAKHdfECj8QFjAAegQIABAB&url=https%3A%2F%2Fdocplayer.fr%2F17260054-Cours-ift6266-apprentissage-non-supervise-de-varietes.html&usg=AOvVaw1vXPPKIdb5E6S0ylr2PkMk>)
- [57] (https://www.google.com/url?sa=t&rct=j&q=&esrc=s&source=web&cd=4&ved=2ahUKEwjMsqj36ITjAhW8D2MBHc7cDzQQFjADegQIABAC&url=http%3A%2F%2Ffermin.perso.math.cnrs.fr%2FFiles%2FChap3.pdf&usg=AOvVaw1QWn3l2oJmzKtGT8hE_Ut)
- [58] (https://www.memoireonline.com/12/09/2917/m_Algorithmes-dapprentissage-pour-la-classification-de-documents1.html)
- [59] https://www.google.com/url?sa=t&rct=j&q=&esrc=s&source=web&cd=16&ved=2ahUKEwj4rKQ7YTjAhW0AWMBHdu3CJk4ChAWMAV6BAgIEAI&url=http%3A%2F%2Fdspace.univ-setif.dz%3A8888%2Fjspui%2Fbitstream%2F123456789%2F3017%2F1%2Fthese_slimani_08122018f.pdf&usg=AOvVaw1g3j5z4FXGZnRB-jk9xZSX)
- [60] <https://www.google.com/url?sa=t&rct=j&q=&esrc=s&source=web&cd=2&cad=rja&uact=8&ved=2ahUKEwil8KGo8ITjAhXhDWMBHQ-vDU8QFjABegQIARAC&url=https%3A%2F%2Fpdfs.semanticscholar.org%2Fff19%2Fc9d1a4b74be95016401c91703958580b145a.pdf&usg=AOvVaw2TjWMFk78uQxvMhIVmEYQe>)
- [61] (https://www.google.com/url?sa=t&rct=j&q=&esrc=s&source=web&cd=4&ved=2ahUKEwj60PLR8ITjAhWSAGMBHTP7DGQOFjADegQIAxAC&url=http%3A%2F%2Fprofs.etsmtl.ca%2Fjalandry%2FRecherche%2FBouchard%2Fpdf%2FGPA792_Rapport_technique.pdf&usg=AOvVaw03uP9Soael3SJ-HO1ckl_)
- [62] (https://www.saedsayad.com/clustering_kmeans.htm)
- [63] (<https://www.google.com/url?sa=t&rct=j&q=&esrc=s&source=web&cd=1&cad=rja&uact=8&ved=2ahUKEwivi7SV84TjAhX0DGMBHcGvD4IQFjAAegQIBBAC&url=http%3A%2F>)

- http://www.lamsade.dauphine.fr/~atif%2Flib%2Fexe%2Ffetch.php%3Fmedia%3Dteaching%3Adecisiontrees.pdf&usg=AOvVaw3IPgy_nL1QD8hqqHOIRFts)
- [64] (<http://cedric.cnam.fr/vertigo/Cours/ml2/coursForetsAleatoires.html>)
- [65] (https://www.memoireonline.com/07/07/518/m_utilisation-machines-vecteurs-support-chiffres-manuscrits0.html)
- [66] (<https://www.statsoft.fr/concepts-statistiques/glossaire/p/perceptrons-multicouches.html>)
- [67] (http://nicolascormier.com/documentation/ia/cours_IA/node89.html)
- [68] (<https://rfia2012.files.wordpress.com/2012/02/les-rc3a9seaux-de-neurone-rbf.pdf>)
- [69] (<https://reussirlem1info.files.wordpress.com/2012/05/mc-chapitre-3-rc3a9seaux-hopfields.pdf>)
- [70] (<https://rfia2012.files.wordpress.com/2012/02/rc3a9seau-de-kohonenc2ab-les-carte-auto-organisatrices-c2bb.pdf>)
- [71] https://www.google.com/url?sa=t&rct=j&q=&esrc=s&source=web&cd=1&cad=rja&uact=8&ved=2ahUKewjJzrbosoTjAhUcBGMBHR1BDPgQFjAAegQIARAB&url=https%3A%2F%2Fwww.inria.fr%2Factualite%2Fmediacenter%2Fle-deep-learning-une-revolution-en-intelligence-artificielle&usg=AOvVaw3nx7u8H8wMiqeg7_mXZEYm)
- [72] (<https://www.google.com/url?sa=t&rct=j&q=&esrc=s&source=web&cd=11&ved=2ahUKewjgzve2tYTjAhXC5eAKHbDzARo4ChAWMAB6BAgDEAE&url=https%3A%2F%2Fwww.theses.fr%2F2016PA066223.pdf&usg=AOvVaw2L9rRjRuAHfd4twieY13A4>)
- [73] <https://www.google.com/url?sa=t&rct=j&q=&esrc=s&source=web&cd=3&cad=rja&uact=8&ved=2ahUKEwinuJmQuITjAhVPWxoKHVktA84QFjACegQIARAC&url=https%3A%2F%2Finria.fr%2Fcontent%2Fdownload%2F101809%2F1454700%2Fversion%2F1%2Ffile%2FDP%2BYLcunVF.pdf&usg=AOvVaw2MDSsxn-O15E0uOqkz9B0m>)
- [74] (https://www.google.com/url?sa=t&rct=j&q=&esrc=s&source=web&cd=9&cad=rja&uact=8&ved=2ahUKEwjLh-7JuYTjAhVoxYUKHd2HBg8QFjAIegQICBAC&url=http%3A%2F%2Fwww.lix.polytechnique.fr%2Fdascim%2Fdascis%2Ffiles%2FProjet_3A__Rapport_final.pdf&usg=AOvVaw20UI-RQYxyfOr_5LB_p-yb)
- [75] (<https://pdfs.semanticscholar.org/0b88/a059510a01889715eabf5fe737b326dae71b.pdf>)
- [76] (<https://foris.fr/wp-content/uploads/2018/07/Intelligence-Artificielle.pdf>)

- [77] (https://www.google.com/url?sa=t&rct=j&q=&esrc=s&source=web&cd=3&ved=2ahUKEwjX4I6wwYTjAhVKTB0KHXBVCT4QFjACegQIBBAC&url=http%3A%2F%2Femilie.caillault.free.fr%2Fdoc%2FEpoisson2002a.pdf&usg=AOvVaw32pukI_oxFUStMVjNzY9KU)
- [78] (<https://www.google.com/url?sa=t&rct=j&q=&esrc=s&source=web&cd=2&ved=2ahUKEwjqua6ZxoTjAhWrDmMBHd7VBnkQFjABegQIAhAC&url=http%3A%2F%2Ftlnarchives.atala.org%2Fateliers%2F2015%2FDEFT%2Fdeft-2015-long-008.pdf&usg=AOvVaw3oNRFvXlrk8OukZ4W6YTS9>)
- [79] (<https://www.datasciencetoday.net/index.php/en-us/deep-learning/173-les-reseaux-de-neurones-convolutifs>)
- [80] (<https://openclassrooms.com/fr/courses/4470531-classez-et-segmentez-des-donnees-visuelles/5083336-decouvrez-les-differentes-couches-dun-cnn>)
- [81] (<https://openclassrooms.com/fr/courses/4470531-classez-et-segmentez-des-donnees-visuelles/5088816-apprenez-a-construire-un-cnn-et-gagnez-du-temps-avec-le-transfer-learning>)
- [82] (https://www.google.com/url?sa=t&rct=j&q=&esrc=s&source=web&cd=1&cad=rja&uact=8&ved=2ahUKEwiFw4i50ITjAhXZDmMBHSJEA1UQFjAAegQIAxAB&url=http%3A%2F%2Fwww.connex.lip6.fr%2F~schwander%2Fenseignement%2F2017-2018%2Fm2bigdata_apprentissage%2Ftp6.pdf&usg=AOvVaw1VxFtSdIv2owixE7objEVr)
- [83] (<https://deeplylearning.fr/cours-theoriques-deep-learning/fonction-dactivation/>)
- [84] (https://ml4a.github.io/ml4a/fr/neural_networks/)
- [85] <https://www.actuia.com/faq/que-signifie-relu-en-matiere-dintelligence-artificielle/>)
- [86] <https://www.statsoft.fr/concepts-statistiques/glossaire/s/softmax.html>)
- [87] (https://www.google.com/url?sa=t&rct=j&q=&esrc=s&source=web&cd=1&cad=rja&uact=8&ved=2ahUKEwjAz_y82ITjAhUSGhQKHd2KDd8QFjAAegQIABAB&url=https%3A%2F%2Ffr.mathworks.com%2Fsolutions%2Fdeep-learning%2Fconvolutional-neural-network.html&usg=AOvVaw36MJDQLcm-KUKk-57C0AZ1)
- [88] (<https://www.google.com/url?sa=t&rct=j&q=&esrc=s&source=web&cd=10&ved=2ahUKEwifh9ea5LPhAhXNBGMBHZ3yANUQFjAJegQIBRAC&url=http%3A%2F%2Fwww.memoirepfe.fst-usmba.ac.ma%2Fget%2Fpdf%2F4266&usg=AOvVaw2IhTLoi4pJnz7Nokr8Zspx>)

- [89] (<https://www.google.com/url?sa=t&rct=j&q=&esrc=s&source=web&cd=1&cad=rja&uact=8&ved=2ahUKEwjik7Kn87TjAhWHGBQKH9DD1IQFjAAegQIABAC&url=http%3A%2F%2Fdspace.univ-tlemcen.dz%2Fbitstream%2F112%2F12235%2F1%2FClassification-des-images-avec-les-reseaux-de-neurones.pdf&usg=AOvVaw27Fg92XiAvV26P1ZgeEFCh>)
- [90] (<https://arxiv.org/ftp/arxiv/papers/1512/1512.01400.pdf>)
- [91] (<https://www.google.com/url?sa=t&rct=j&q=&esrc=s&source=web&cd=3&ved=2ahUKEwiy0ef62ZnjAhUQAWMBHeImC0YQFjACegQIABAC&url=http%3A%2F%2Fwww.nmis.isti.cnr.it%2Ffalchi%2FDraft%2F2016-ISCC-Draft.pdf&usg=AOvVaw0s9j3lGvcOl3ZGI9qI8uCI>)
- [92] (https://www.researchgate.net/publication/309540140_Deep_Learning_for_Decentralized_Parking_Lot_Occupancy_Detection)
- [93] (https://www.researchgate.net/publication/224312570_A_Bayesian_hierarchical_detection_framework_for_parking_space_detection)
- [94] (https://www.google.com/url?sa=t&rct=j&q=&esrc=s&source=web&cd=3&ved=2ahUKEwjQino2pnjAhUk6uAKHYM_CScQFjACegQIARAC&url=http%3A%2F%2Fwww.inf.ufpr.br%2Fflesoliveira%2Fdownload%2Fpklot-readme.pdf&usg=AOvVaw0BgYay1CrXl-rAlhsz18cz)
- [95] (<https://www.google.com/url?sa=t&rct=j&q=&esrc=s&source=web&cd=1&cad=rja&uact=8&ved=2ahUKEwiR4-3cxLnhAhUPRBoKH9Yb8BGYQFjAAegQIAxAB&url=https%3A%2F%2Fgithub.com%2Fkendryte%2Fkendryte-tensorflow&usg=AOvVaw3he3xPhmiE7H7rsjuRt08g>)
- [96] (<https://www.lebigdata.fr/tensorflow-definition-tout-savoir>)
- [97] (<https://github.com/keras-team/keras>)
- [98] (<https://www.google.com/url?sa=t&rct=j&q=&esrc=s&source=web&cd=24&ved=2ahUKEwjIysTvjmBhAhWcAGMBH ZJIATg4FBAWMAN6BAgFEAI&url=http%3A%2F%2Fpages.upf.pf%2FSebastien.Chabrier%2FMI%2FTutorielPython.pdf&usg=AOvVaw14-Yh4gkLPcVRgGKRkPwlv>)
- [99] (<https://www.google.com/url?sa=t&rct=j&q=&esrc=s&source=web&cd=2&ved=2ahUKEwjfxpKyxxtziAhVL8uAKHb6NAioQFjABegQIAxAC&url=http%3A%2F%2Fwww.inf.ufpr.br%2Fflesoliveira%2Fdownload%2Fpklot-readme.pdf&usg=AOvVaw0BgYay1CrXl-rAlhsz18cz>)
- [100] (<https://web.inf.ufpr.br/vri/databases/parking-lot-database/>)

[101] https://www.researchgate.net/publication/309540140_Deep_Learning_for_Decentralized_Parking_Lot_Occupancy_Detection)