

UNIVERSITÉ NAZI BONI

BURKINA-FASO

Unité-Progrès-Justice



École Supérieure d'Informatique (ESI)



Année académique : 2020-2021

MÉMOIRE DE MASTER

présenté par :

Fatoumata OUATTARA

Filière : **Informatique**

Spécialité : **Science de Données**

Comment implémenter un réseau de neurones profond pour la classification des images radars ?

Directeur de Mémoire : **Pr HATEM BEN STA**

Soutenu publiquement le : **15/02/2023**

Devant le jury composé de :

Président :

M. Théodore TAPSOBA, Professeur titulaire, Université Nazi BONI

Membres du jury :

M. Pasteur PODA, Maître de conférences, Université Nazi BONI

M. Hatem BEN STA, Professeur titulaire, Université de Tunis El Manar

Dédicace

Au Dieu tout puissant, maître des temps et des circonstances, pour tous ses bienfaits.

A la mémoire de :
ma grande sœur
Mariam OUATTARA

En l'honneur de :
Mes parents
Pio OUATTARA et Djèneba OUATTARA

En l'honneur de :
Mon mari
Dofyniwassouani Alain HOUEDE

A mon fils
Tohnsoué Panmhan Evan HOUEDE

Remerciements

Mes premiers remerciements vont à l'endroit de mon directeur de mémoire, Monsieur Hatem Ben Sta, Professeur à l'Université de Tunis El Manar et à l'Université Laval, pour avoir encadré et dirigé mes premiers pas dans la recherche et pour m'avoir proposé un sujet stimulant. Je crois sincèrement avoir eu une chance extraordinaire de devenir son élève, j'en serai toujours fière et honorée. Je suis également très reconnaissante de la disponibilité sans faille dont il a fait preuve durant tout ce temps, en dépit de son emploi du temps chargé. Il va sans dire que j'ai appris à ses côtés, aussi bien à travers ses qualités scientifiques irréprochables que humaines, il restera pour moi un exemple.

Je voudrais aussi présenter ma profonde reconnaissance à tous les enseignants de l'Ecole Supérieure d'Informatique (ESI) qui ont contribué à ma formation.

À tous mes parents, amis et proches, je voudrais vous dire merci parce qu'il y a toujours eu quelqu'un parmi vous pour m'encourager et me remonter le moral lorsque je faiblissais. Encore merci.

Résumé

Ce mémoire est consacré à la classification des images radars à l'aide des réseaux de neurones profonds. Il a été démontré que les CNNs connaissent une bonne performance quant à la reconnaissance des images radars. Cependant pour pouvoir appliquer le CNN sur de telles images, il faut pendre certaines réalités en compte notamment l'inaccessibilité aux données radars.

Le problème principal auquel les réseaux de neurones profonds font face lorsqu'il s'agit des images radars est ce manque d'ensemble conséquents de données. En réalité, les images radars sont des données en générale protégées et dédiées à la sécurité ou à la surveillance. Elles sont acquies grâce à des satellites radars à synthèse d'ouverture (SAR), des dispositifs routiers, ou domestiques etc.

Dans ce document nous cherchons des solutions pour contrer le manque d'images radars afin de pouvoir engranger de très bons résultats lors de la catégorisation de ces images à travers les deep convolutional neural networks. Des méthodes d'augmentations de données sont évoquées et au final nous implémentons un modèle de réseau de neurones convolutif auquel nous soumettons un ensemble de données ayant été augmenté grâce à un certain nombre de transformations réalisées sur les images dont nous disposons.

Mots clés : Apprentissage profond, réseaux de neurones profonds, CNNs, augmentation des données, images radars.

Abstract

This paper is devoted to the classification of radar images using deep neural networks. It has been shown that CNNs perform well in the recognition of radar images. But to apply CNN on these images, certain realities must be taken into account, such as radar data inaccessibility. The main problem that deep neural networks encounters around radar images is the lack of data. In fact, radar images are protected data and dedicated to security or supervision. They are acquired thanks to synthetic aperture radar satellites, road or domestic devices, etc. In this document we seek solutions to counter the lack of radar images in order to be able to reap very good results during the categorization of these images through deep convolutional neural networks. Methods of data augmentation are discussed and finally we implement a convolutional neural network model to which we submit a dataset that has been augmented by a number of transformations performed on the images we have.

Key-words : Deep learning, deep neural networks, CNNs, data augmentation, radar images.

Table des matières

Dédicace	i
Remerciements	ii
Résumé	iii
Abstract	iv
Liste des sigles et abréviations	x
Avant propos	xi
Introduction générale	1
1 Imagerie Radar	2
1.1 Introduction	2
1.2 Histoire du Radar	3
1.3 Principe de fonctionnement du radar	4
1.4 Problèmes liés aux images radars	6
1.4.1 L'acquisition de données	6
1.4.2 Le Speckle ou chatoiement	6
1.5 Approches de filtrage des images radars	7
1.5.1 Filtres passe-bas	8
1.5.2 Filtres passe-hauts	9
1.6 Conclusion	10
2 Deep learning	11
2.1 Introduction	11
2.2 Le machine learning	12
2.2.1 Définition	12
2.2.2 Une discipline mère : l'intelligence artificielle	12

TABLE DES MATIÈRES

2.2.3	Types d'apprentissages automatiques	13
2.3	Du machine learning au deep learning	14
2.3.1	Deep learning vs machine learning	14
2.4	Les réseaux de neurones artificiels	15
2.4.1	Origine des réseaux de neurones artificiels et naissance du deep learning	16
2.4.2	Quelques algorithmes de Deep learning	20
2.5	Les réseaux de neurones convolutifs	20
2.6	Conclusion	23
3	Classification des images radars à l'aide du deep learning	24
3.1	Introduction	24
3.2	Les motivations de la classification des images	25
3.3	État de l'art des processus de classification des images radars	26
3.3.1	Bref aperçu des méthodes de classification des images radars avant les CNNs	26
3.3.2	Revue littéraire des travaux sur la classification des images radars grâce aux CNNs	27
3.3.3	Analyse personnelle	28
3.4	Le manque de données radars disponibles : enjeux	29
3.5	Perspectives au manque de données radars pour la classification	29
3.5.1	Transformation des images	29
3.5.2	La simulation de données : les réseaux de neurones génératifs antagonistes à l'honneur	30
3.5.3	L'arrêt anticipé de l'apprentissage	32
3.5.4	Le dropout	32
3.6	Conclusion	33
4	Implémentation	34
4.1	Introduction	34
4.2	Outils et bibliothèques nécessaires	35
4.2.1	TensorFlow	35
4.2.2	Keras	35
4.2.3	Langage Python	35
4.3	Ensemble de données d'analyse	36
4.4	Préparation des données	38
4.4.1	Fractionnement de notre ensemble de données	38
4.4.2	Redimensionnement des données	38
4.4.3	Augmentation de données	38

TABLE DES MATIÈRES

4.5	Architecture du premier modèle sans augmentation de données	39
4.5.1	Résumé du modèle	40
4.5.2	Entraînement du modèle	40
4.5.3	Visualisation des résultats de l'entraînement	41
4.6	Formation du modèle après une augmentation par transformation des images	42
4.6.1	Augmentation des données	42
4.6.2	Entraînement	43
4.6.3	Évaluation du modèle	43
4.7	Conclusion	44
	Conclusion générale	45
	Bibliographie	46

Table des figures

1.1	Image radar d'un port	3
1.2	Fonctionnement du système radar	4
1.3	Imagerie radar : principe	5
1.4	Exemple d'image satellitaire radar	7
1.5	Filtre moyenneur appliqué a une image radar (char MSTAR)	8
1.6	calcul de la nouvelle valeur du pixel rouge d'une image, a travers la valeur moyenne des pixels voisin (3x3) de ce pixel. Le pixel vert represente la nouvelle valeur dans l'image transformée.	9
1.7	Principe du filtre médian	10
2.1	Machine learning vs Deep learning	15
2.2	Structure du neurone biologique	16
2.3	Le neurone formel de MacCulloch et Pitts	17
2.4	Architecture d'un réseau de neurones convolutif	20
2.5	Bloc de convolution	21
2.6	Architecture générale d'un CNN	22
2.7	Allure de la fonction ReLU	23
3.1	Processus de classification d'images en deep learning	26
3.2	l'augmentation des données par transformation	30
3.3	Simulation par le GAN	32
4.1	Quelques librairies open source de deep learning	35
4.2	Les cibles militaires : images optiques à gauche et images SAR à droite	37
4.3	Exemples d'images d'entrainement	37
4.4	Resumé du modèle	40
4.5	Entrainement du modèle 1	41
4.6	Taux de précision(accuracy) et de perte(loss) du model 1 sur les données d'entrainement(train) et de validation (val)	41
4.7	Evaluation de notre modele sur un jeux de test	42

TABLE DES FIGURES

4.8	Image du vehicule type 2S1 sous des transformations	42
4.9	courbes de precisions et de pertes sur les données d'entrainement et de validation	43
4.10	Prédiction sur de nouvelles données	43

Liste des sigles et abréviations

CNN : Convolutional Neural Network
DCNN : Deep Convolutional Neural Network
EM : ElectroMagnétiques
ESI : École supérieure d'informatique
FC : Fully Connected
GAN : Generative adversarial network
IA : Intelligence Artificielle
IRS : Ingénierie des Réseaux et Systèmes
ISI : Ingénierie des Systèmes d'Information
ML : Machine Learning
MSTAR : Moving and Stationary Target Acquisition and Recognition
NAZA : National Aeronautics and Space Administration
NLP : Natural Language Processing
RADAR : Radio Detection And Ranging
RAG : Réseau Antagoniste Génératif
ReLU : Rectified Linear Units
RNN : Recurrent Neural Network
RS/CAR : Réseaux et Systèmes/ Conception et Architecture de Réseaux
RS/CS : Réseaux et Systèmes/Cybersécurité
SAR : Synthetic Aperture Radar
SD : Sciences de Données
SI/SAD : Systèmes d'Information/ Système d'aide à la Décision
SVM : machines à vecteurs de support
TAL : Traitement Automatique des Langues
TLN : Traitement Automatique du Langage Naturel
UNB : Université Nazi BONI
VAE : Variational Autoencoders

Avant propos

L' Université Nazi Boni (UNB) est une université publique du Burkina Faso à caractère scientifique, culturelle et technique. Elle a été créée le 16 mai 1997 par le decret no97-254/PRES/PM/ MESSRS avec pour mission l'élaboration et la transmission de connaissances aux Hommes afin de répondre aux besoins de la nation. L'université Nazi BONI compte huit(8) établissements, deux(2) centres universitaires et quatre(4) laboratoires de recherche. Toutes ces structures œuvrent à donner aux étudiants des compétences dans leurs domaines respectifs à travers une formation pointue. L'École Supérieure d'Informatique (ESI), dont nous relevons, est un établissement d'enseignement supérieur et de recherche. Elle délivre des diplômes de niveau bac+3(Licence) et bac+5 (Master). Au niveau bac+3, elle prépare de nos jours, d'une part une licence en Ingénierie des Réseaux et Systèmes (IRS) et d'autre part une licence en Ingénierie des Systèmes d'Information (ISI). Au niveau bac+5, elle propose quatre formations : un master en Réseaux et Systèmes option Conception et Architecture de Réseaux (RS/CAR), un master en Réseaux et Systèmes option cybersécurité (RS/CS), un master en Systèmes d'Information option Système d'aide à la décision (SI/SAD) et un master en Sciences de Données (SD). L'obtention du diplôme de Master nécessite la validation de quatre (4) semestres dont le quatrième consiste à effectuer un stage ou des recherches d'au moins six (6) mois suivi d'une soutenance publique devant un jury. Cette évaluation a pour objectif de familiariser l'étudiant avec les exigences d'un milieu de travail ou de recherche et de donner une dimension professionnelle à sa formation estudiantine. C'est dans ce contexte que nous avons été encadré accompagné et initié à la compréhension, la maîtrise du domaine de la recherche.

Introduction générale

Crucial arme militaire pendant la guerre de 1939-1945, le radar est depuis 1935 un outil extraordinaire qui n'a cessé de conquérir le monde scientifique et industriel. Ses premières applications ont été orientées dans le sens de la détection de cibles et plus tard en imagerie.

Ces dernières années, le système radar a suscité un intérêt marqué auprès des chercheurs. Il utilise des longueurs d'ondes qui peuvent traverser les nuages et offre par conséquent des possibilités intéressantes dans les domaines du renseignement et de la surveillance.

Cependant, les images produites par les radars sont affectées par un bruit de nature multiplicative, c'est-à-dire qui augmente avec l'intensité du signal reçu au niveau du capteur. Ainsi malgré leur bonne résolution, le message qu'elles ramènent à la vision humaine reste flou. L'humain perçoit l'image comme un ensemble d'informations sémantiques dont il faut interpréter le contenu ; ce qui lui échappe concernant les images radars. Certes avec le développement de l'intelligence artificielle ce problème de non transparence des images radars peut être facilement contrôlé. Reconnaître, classer, sans avoir analysé visuellement devient notre quotidien grâce aux différentes branches de l'intelligence artificielle, plus particulièrement grâce au Deep Learning (l'apprentissage en profondeur) qui ne cesse de nous impressionner.

L'utilisation du Deep Learning dans la reconnaissance de cibles à partir d'images radars, a montré un potentiel intéressant. Mais, pour atteindre des résultats optimaux sur ces types de cibles, il est nécessaire de constituer des bases de données les plus exhaustives possible. Or l'acquisition et l'étiquetage d'une telle base est une tâche très fastidieuse, couteuse en temps, en ressources humaines et matérielles. Le but de ce travail, est donc de tirer partie des méthodes d'augmentations de données existantes pour pouvoir classer les cibles radars grâce aux réseaux de neurones profonds.

Ce mémoire est organisé comme suit : un premier chapitre, où nous étudions le système radar en retournant à ses origines tout en faisant ressortir les difficultés dont il fait face. Dans un deuxième chapitre, nous attaquons le domaine du Deep learning, où nous mettons en lumière un réseau de neurone très en vogue de nos jours, le réseau de neurones convolutifs CNN. Ensuite, un troisième chapitre qui nous fait rentrer dans le vif du sujet, en nous ramenant aux différents travaux réalisés sur la combinaison deep learning et imagerie radar. Nous terminons par un quatrième chapitre consacré à l'implémentation d'un modèle CNN pour classer nos données radars. Nous y détaillons le processus suivie.

Imagerie Radar

Sommaire

1.1	Introduction	2
1.2	Histoire du Radar	3
1.3	Principe de fonctionnement du radar	4
1.4	Problèmes liés aux images radars	6
1.4.1	L'acquisition de données	6
1.4.2	Le Speckle ou chatoiement	6
1.5	Approches de filtrage des images radars	7
1.5.1	Filtres passe-bas	8
1.5.2	Filtres passe-hauts	9
1.6	Conclusion	10

1.1 Introduction

Les capteurs radars sont venus avec un système d'imagerie très avantageux par rapport à l'imagerie optique : ils offrent une meilleure résolution d'images, sont adaptables aux changements du temps et de l'espace, exigent une imagerie très cohérente (détecte les plus fins signaux possibles), et demeurent très sensibles face aux propriétés géométriques des cibles microscopiques comme macroscopiques. Cependant le premier contact avec une image radar est toujours très déroutant, il n'est pas évident aux premiers abords de définir le sens de l'image. Pourquoi ces couleurs, pourquoi de si grandes surfaces sombres ? pourquoi toutes ces lignes, que représentent-elles ? sont-elles des défauts de restitution ? Autant de questions spontanées quant à la perception première d'une image issue du radar. [1.1.](#)

Dans ce chapitre nous revisitons l'histoire de ce système et cherchons à comprendre explicitement son fonctionnement général qui paraît si complexe.

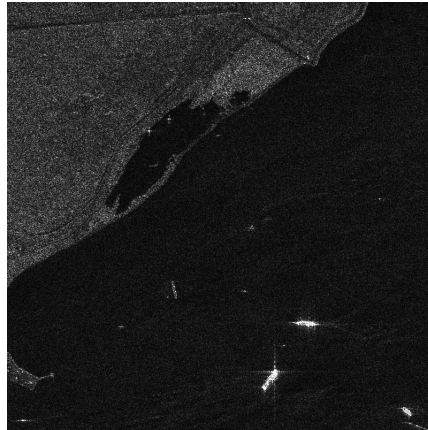


FIGURE 1.1 – Image radar d'un port

1.2 Histoire du Radar

Le besoin militaire de "détecter" des objets pour la défense aérienne et maritime, fut le principal moteur d'expansion du système radar.

Dès 1900, Nikola Tesla, un inventeur et ingénieur serbe, exprima la possibilité de détecter et mesurer le mouvement d'objets lointains à l'aide d'ondes électromagnétiques. Cette idée est reprise et développée par Hugo Gernsback le « Jules Verne du radar », dans son roman d'anticipation RALF 124 C41 +, paru en 1912. Cependant la première expérimentation concrète est faite par Christian Hülsmeyer qui breveta en 1904 son télémobiloscope, émetteur/récepteur radio fréquence, utilisé pour la détection à distance d'objets métalliques, qu'il avait testé avec succès à Cologne puis à Rotterdam avec des portées de l'ordre de 3 kilomètres sur des bateaux.[28]

Malgré cette réalisation intéressante, on ne comprit pas à l'époque l'intérêt réel de cette nouvelle application. Ainsi il fallut attendre plus de 20 ans après pour que les chercheurs reprennent les expériences. La véritable éclosion de cette technique se situe en effet entre 1930 et 1940 (pendant les 02 guerres mondiales), où des recherches et développements ont été menés simultanément dans de nombreux pays, notamment les USA, la Grande-Bretagne, l'Allemagne et la France.[28]

Le radar connue à l'origine pour la détection de mouvement, de vitesse, etc , va faire ses premiers pas en imagerie avec l'invention du radar à synthèse d'ouverture SAR. Il est depuis 1978 un outil extraordinaire pour le géographe. Pendant la décennie 90 toute une série de radars va être lancée sur satellite . Les images radars sont utilisées jusqu'ici par quelques spécialistes tels que les océanographes, glaciologues, météorologues...et inondent les grandes villes d'Europe et d'Asie.

1.3 Principe de fonctionnement du radar

Il existe de multiples types de radars, avec chacun leur spécificité malgré un fonctionnement général présentant des similitudes. Ainsi le radar d'une manière générale est constitué d'émetteur produisant des ondes électromagnétiques (ondes émises) de fréquence de plusieurs GHz. Ces ondes électromagnétiques sont alors envoyées par une antenne vers un cible à observer ou à interroger. Il peut s'agir par exemple d'un arbre, d'une voiture, d'une masse d'air ou d'un nuage. L'onde est alors réfléchi (onde rétro-diffusée) par la cible et retourne vers l'antenne, sur le récepteur. Les informations reçues peuvent alors être traitées et interprétées par l'utilisateur. La figure 1.2 nous résume ce enchaînement entre les différentes composantes du Système.

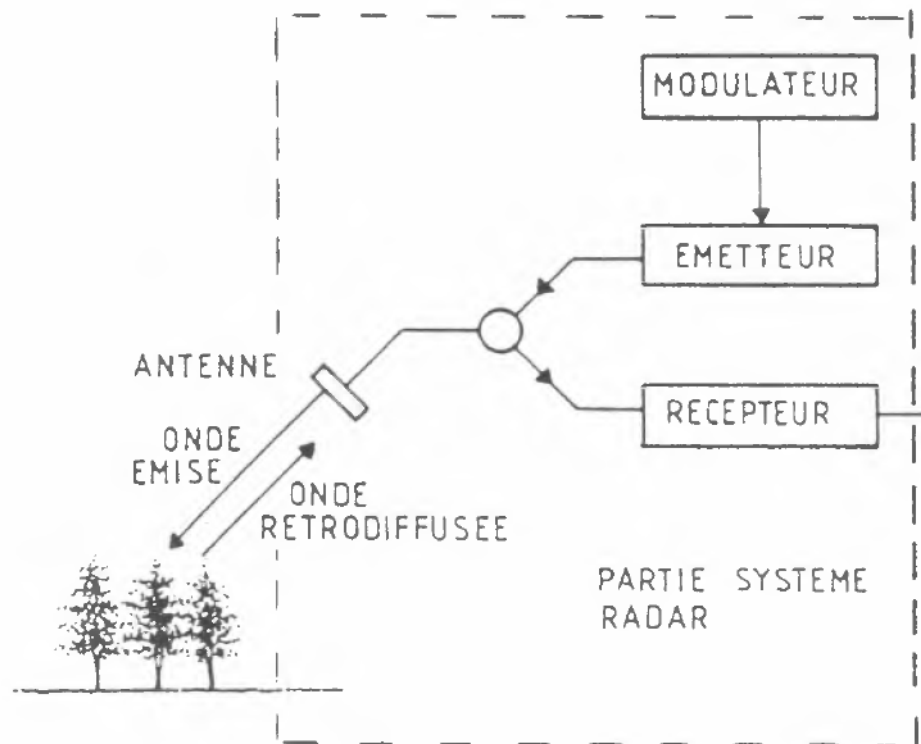


FIGURE 1.2 – Fonctionnement du système radar

En imagerie, le radar "imageur" fournit sa lumière pour éclairer une zone au sol (flèche 1 de la *figure 1.3*) et construit une image à des longueurs d'onde radio. Il utilise une antenne et un stockage informatique numérique pour enregistrer ses images. Dans une image radar, on ne peut voir que l'énergie qui a été réfléchiée vers l'antenne radar.

Les images radars contiennent plusieurs points de nature sel et poivre. Chaque pixel de l'image représente la rétro-diffusion radar pour cette zone du sol (flèche 2 de la *figure 1.3*) les zones plus claires représentent une rétro-diffusion élevée et les zones plus sombres représentent une faible rétro-diffusion. La rétro-diffusion du signal fournit des renseignements très utiles sur les propriétés géométriques et sur la nature des objets à la surface de la terre : rugosité de la surface, type de matériau (fer, bois,...), teneur en humidité Ainsi, une mer agitée (donc « rugueuse ») apparaîtra en blanc sur une image radar, tandis qu'une trace d'hydrocarbure très lisse et huileuse, sera en noir, tout comme une route au milieu d'une plaine. Les bateaux forment des points clairs lumineux ainsi que les constructions et villes, car ils comportent de nombreux angles et réflecteurs qui sont constitués de matériaux « réfléchissants » (éléments métalliques par exemple). En milieu forestier, les clairières apparaissent en sombre (milieu plus lisse). Les images radars sont facilement « brouillées » et donc difficiles à interpréter : on parle de « phénomène de speckle ». Les éléments en mouvement tels que les feuillages ou fortes pluies sont à l'origine de cela.

Les capteurs radars sont dit "actifs" et ne font pas appel à la lumière du soleil.

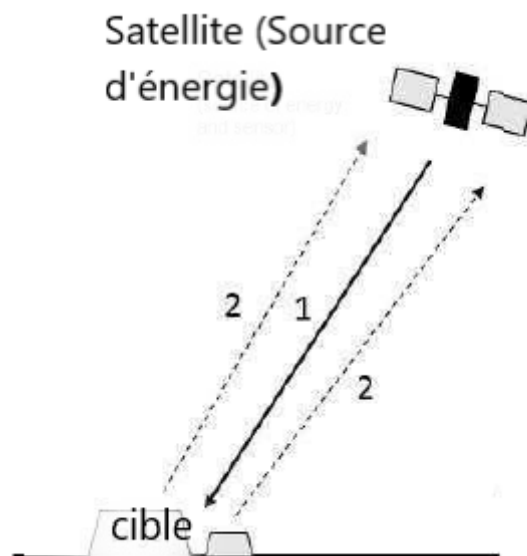


FIGURE 1.3 – Imagerie radar : principe

1.4 Problèmes liés aux images radars

1.4.1 L'acquisition de données

Les capteurs radars peuvent être montés sur des satellites ou sur des avions, mais peuvent également fonctionner au sol avec les antennes d'observation de la galaxie, sur des véhicules, ou même peuvent être installés à la maison. Vues les objectifs visés, les images radars sont difficilement accessible au citoyen lambda. L'État ou les organisations détenteurs d'importants budgets avec des objectifs bien précis, notamment la surveillance, le renseignement sont ainsi les propriétaires de ces données et sont donc libres de les commercialiser ou les laisser libres aux utilisateurs. Obtenir des images radars est un vrai parcours de combattant. Il existe plusieurs plateformes qui en offrent mais en nombre très limité. On peut citer TileDB dont les données sont utilisables directement en ligne, sentinel Hub Eo Browser qui offre des images gratuites et payantes. Certaines structures vont plutôt vendre ces données à l'image d'airbus qui est le premier fournisseur de données radars à la communauté de chercheurs et de scientifiques de la NASA. Capella Space est de même une entreprise de la silicon Valley lancé dans ce domaine. On est à ce jour à 41 satellites africains dans l'espace, mais combien sont dotés de capteurs radars ou disposent des données pour les citoyens ? jusque là aucun n'a été trouvé lors de nos recherches.

Avoir des données de satellites à capteurs actifs des régions africaines reste bien possibles mais a quel prix et avec quelles qualités ?

Vu la distance à laquelle se trouvent les satellites européens ou asiatiques, etc , les rétro diffusions dépendent de l'angle à laquelle les images sont prises, généralement latéralement. Certaines régions africaines peuvent être en moitié ou même hors du champs récepteur.

1.4.2 Le Speckle ou chatoiemment

Toutes les images radars contiennent une certaine quantité de ce que l'on appelle « speckle » ou chatoiemment. Cet effet se manifeste comme une texture « poivre et sel » ; nous pouvons le constater dans la figure 1.4. Le chatoiemment est produit par l'interférence aléatoire provenant de la diffusion multiple d'ondes qui se produit dans chaque cellule de résolution. Ce chatoiemment dégrade la qualité de l'image et peut rendre l'interprétation (visuelle ou numérique) complexe. Cet effet réduit également l'efficacité de la segmentation, de la classification et d'autres techniques d'extraction de l'information. Il est généralement préférable de réduire le bruit avant l'interprétation et l'analyse des images [26]. Toutefois, le chatoiemment peut être réduit considérablement si l'on prend plus d'images de la même région à des moments différents ou sous des angles d'incidence légèrement différentes. En moyennant plusieurs images, on tend à annuler la variabilité d'amplitude aléatoire et laisser le niveau d'amplitude uniforme inchangée[26].

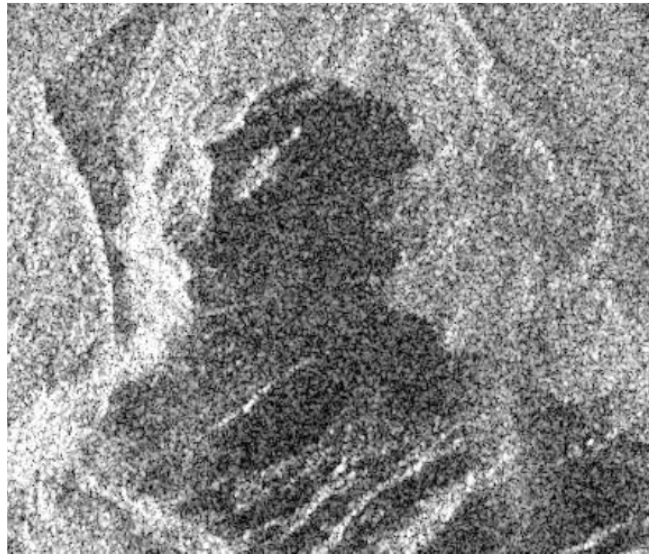


FIGURE 1.4 – Exemple d'image satellitaire radar

1.5 Approches de filtrage des images radars

Le filtrage d'une image, c'est cette opération qui consiste à éliminer le bruit pour faciliter le traitement et l'interprétation de cette image. Le bruit se produit généralement lors de la phase d'acquisition et de numérisation de l'image [19]. Le filtrage [22] [19] d'une image correspond à la convolution de la fonction image, $I(x, y)$, avec une fonction, $f(x, y)$, qui s'appelle réponse impulsionnelle du filtre. Il consiste simplement à remplacer chaque niveau de gris par une combinaison linéaire des niveaux de gris des points voisins. Les coefficients de cette combinaison sont donnés par la fonction réponse impulsionnelle. L'image filtrée est donnée par :

$$I_f(x, y) = (f * I)(x, y) \quad (1.5.1)$$

Dans la littérature, nombreux sont les auteurs qui se sont intéressés à ce problème de chatoiement sur les images classiques [17] [27] comme radars [25] [24] [17] [6] ... Les techniques utilisées pour la restauration de l'image classique dégradée peut de même s'appliquer sur l'image radar dégradée, mais pas aussi efficacement.

Parmi les approches proposées, en ce qui concerne nos recherches sur les retours radars, nous avons notamment l'approche multi-vues qui permet de minimiser le bruit via l'acquisition de différentes images, ou vues d'une même scène. De la moyenne de ces vues, résulte une image moins bruitée cf.[10].

Nous avons aussi l'approche numérique qui met en relief les différents filtres passe-bas comme le filtre moyen et passe-haut tel le filtre laplacien.

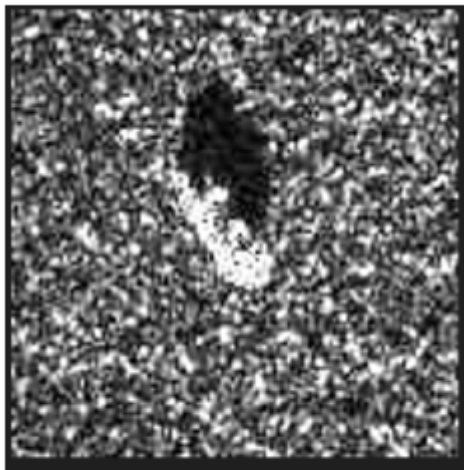
Les filtres peuvent être évalués principalement sur la base de leur capacité à :

- Lisser le plus possible les zones homogènes,
- Préserver la texture de l'image,
- Conserver la structure de l'image (lignes, contours, etc).

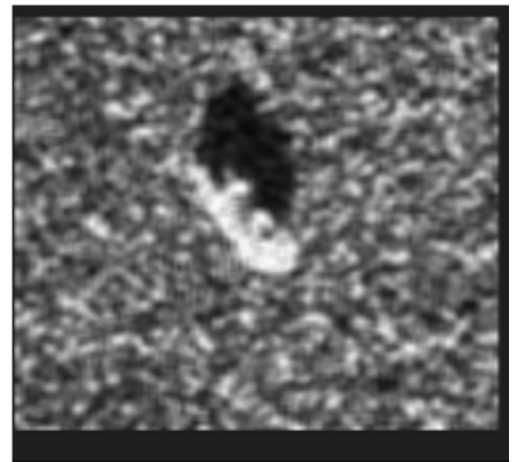
1.5.1 Filtres passe-bas

Les filtres passe-bas permettent de réduire le bruit, en éliminant les points "sels et poivre" pour lisser les images. On a par exemple :

Le filtre moyennneur : Le filtre moyennneur est une opération de traitement d'images utilisée pour réduire le bruit dans une image et/ou flouter une image. Par exemple, l'application d'un filtre moyennneur sur l'image de gauche donne l'image de droite.



- Image originale



- Image filtrée

FIGURE 1.5 – Filtre moyennneur appliqué a une image radar (char MSTAR)

Le filtre moyennneur fait parti de la catégorie des filtres d'images locaux car pour calculer la nouvelle valeur d'un pixel, il regarde la valeur des pixels proches. Concrètement, la valeur filtrée d'un pixel p est égale à la moyenne des valeurs des pixels proches de p . En général, on définit les « pixels proches de p » comme l'ensemble de pixels contenus dans un carré de largeur k centré sur p .

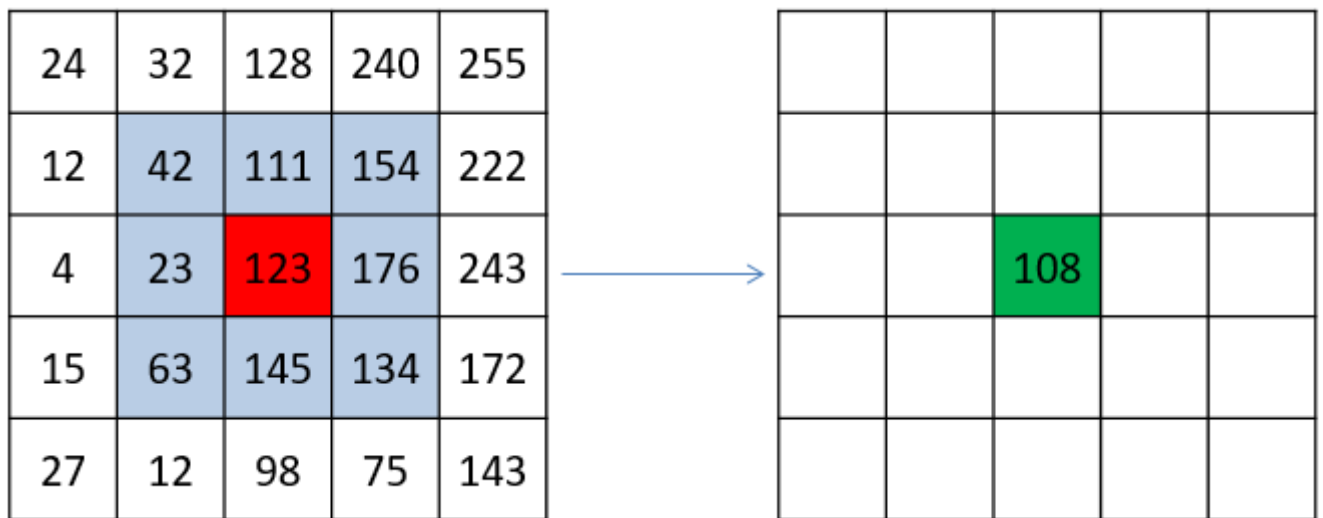


FIGURE 1.6 – calcul de la nouvelle valeur du pixel rouge d’une image, à travers la valeur moyenne des pixels voisin (3×3) de ce pixel. Le pixel vert représente la nouvelle valeur dans l’image transformée.

Filtre Gaussien :

Le filtre gaussien est un type de filtre qui utilise une fonction gaussienne pour calculer la transformation à appliquer à chaque pixel de l’image. La formule d’une fonction gaussienne en deux dimension est :

$$G_{\sigma}(x, y) = (1/2\pi\sigma^2) \exp -(x^2 + y^2/2\sigma^2) \quad (1.5.2)$$

Où x est la distance à l’origine sur l’axe horizontal, y est la distance à l’origine sur l’axe vertical et σ est l’écart type de la distribution. Ce filtre a une logique de lissage (élimination de détails et bruits) analogues au filtre moyennneur.

1.5.2 Filtres passe-hauts

Filtre médian

Le filtre médian a pour principe de remplacer la valeur du pixel central par la valeur médiane de la répartition (luminances triées par ordre croissant) des niveaux de gris des pixels situés à l’intérieur de cette fenêtre. Son but est de préserver les contours.

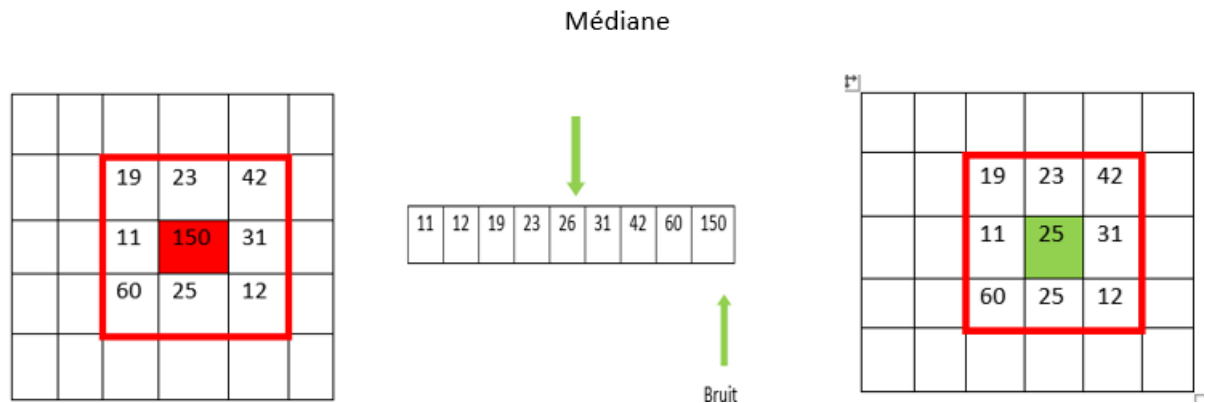


FIGURE 1.7 – Principe du filtre médian

L'intérêt du filtre médian est qu'un pixel non représentatif dans le voisinage n'affectera presque pas la valeur médiane. La valeur médiane qui s'impose, étant le niveau de gris d'un des pixels considérés, on ne crée pas alors de nouveaux niveaux de gris dans l'image. Ainsi lorsque le filtre passe sur un contour très marqué il le préservera mieux.

Filtre bilatéral

Un filtre bilatéral est utilisé pour lisser les images et réduire le bruit, tout en préservant les bords. Contrairement à ce filtre, le filtre médian entraîne souvent une perte d'information sur le contour.

1.6 Conclusion

Dans ce chapitre nous avons vu que les images radars sont des images assez bruitées et difficilement interprétables. Pour acquérir assez d'images radars il est aussi très difficile. Certes pour le facteur bruit, il existe bien des méthodes pour filtrer ces images. Avec l'évolution des technologies, dire qu'il est impossible de reconnaître une image de véhicule, de bateau ou autre n'existe plus. Dans le prochain chapitre nous verrons une technologie qui a révolutionner le monde dans la reconnaissance d'objets.

Deep learning

Sommaire

2.1 Introduction	11
2.2 Le machine learning	12
2.2.1 Définition	12
2.2.2 Une discipline mère : l'intelligence artificielle	12
2.2.3 Types d'apprentissages automatiques	13
2.3 Du machine learning au deep learning	14
2.3.1 Deep learning vs machine learning	14
2.4 Les réseaux de neurones artificiels	15
2.4.1 Origine des réseaux de neurones artificiels et naissance du deep learning	16
2.4.2 Quelques algorithmes de Deep learning	20
2.5 Les réseaux de neurones convolutifs	20
2.6 Conclusion	23

2.1 Introduction

Intelligence artificielle, machine learning, deep learning sont des mots que nous entendons quotidiennement de nos jours. C'est bien normal : c'est l'ère d'une révolution technologique où l'intelligence de la machine "s'apparente" à l'intelligence humaine.

Dans ce chapitre nous essayons de parcourir ces trois domaines, mais en nous focalisant sur le deep learning qui est une résultante des deux premières, en expliquant ses origines et apprendre d'avantage son utilisation.

2.2 Le machine learning

2.2.1 Définition

Qu'est-ce qu'apprendre ? comment apprend-on ? et que cela signifie-t-il pour une machine ?

La question de l'apprentissage fascine les spécialistes de l'informatique et des mathématiques tout autant que neurologues, pédagogues, philosophes ou artistes.

Fabien Benureau (2015) définit l'apprentissage en ces termes : « L'apprentissage est une modification d'un comportement sur la base d'une expérience ». En informatique, on parle d'apprentissage automatique, ou machine learning, quand un programme a la capacité d'apprendre sans que cette modification (de F. Benureau) ne soit explicitement programmée. Cette définition est celle donnée par Arthur Samuel (1959). On peut ainsi opposer un programme classique, qui utilise une procédure et les données qu'il reçoit en entrée pour produire en sortie des réponses, à un programme d'apprentissage automatique, qui utilise les données et les réponses afin de produire la procédure qui permet d'obtenir les secondes à partir des premières.

Les ingrédients secrets du machine learning sont essentiellement : d'une part, les données, qui sont les exemples à partir desquels l'algorithme va apprendre ; 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. Aucun algorithme d'apprentissage ne pourra créer un bon modèle à partir de données qui ne sont pas pertinentes. Ainsi un algorithme d'apprentissage auquel on fournit des données de mauvaise qualité ne pourra rien en faire d'autre que des prédictions de mauvaise qualité. De même, un modèle appris avec un algorithme inadapté sur des données pertinentes ne pourra pas être de bonne qualité.

2.2.2 Une discipline mère : l'intelligence artificielle

Dès 1950 les scientifiques imaginent les machines pensantes. Le mathématicien Alan Turing publie un article intitulé « l'ordinateur et l'intelligence », dans lequel il décrit comment savoir si une machine se rapproche de l'intelligence humaine. Cette méthode est nommée « le jeu de l'imitation » plus connu aujourd'hui sous le nom de Test de Turing. D'où les prémices de l'intelligence artificielle.

Pendant les années 1956, les scientifiques américains : John McCarthy, Marvin Minsky, Nathaniel Rochester et Claude Shannon invitent leurs confrères à une conférence sur les machines pensantes et utilisent pour la première fois le terme "Intelligence artificielle ou l'IA". A partir de cette date, c'est l'essor de l'intelligence artificielle ; les projets fleurissent. En 1957 le psychologue Frank Rosenblatt invente le premier programme d'apprentissage grâce à un réseau de neurones simple appelé le perceptron. La fin de l'année 1960 marque

une période de crise de l'intelligence artificielle. Les investissements s'écroulent et plusieurs chercheurs critiquent ce domaine qui tend à imiter l'intelligence humaine, d'absurdité. Ils évoquent entre autre l'impossibilité pour la machine, à travers les mathématiques, de penser, de mémoriser ou d'exprimer des sentiments. De là, l'on cherchera à redéfinir les objectifs même de l'IA. Dans les années 1980, les recherches reprennent, c'est l'époque des systèmes experts qui font les mêmes analyses qu'un expert humain dans un domaine bien précis. Mais ce n'est que dans les années 1990 que l'IA réintègre les laboratoires scientifiques avec l'évolution des outils de technologie et l'avènement du Big data. Il ne s'agira plus de coder des règles mais laisser les ordinateurs les découvrir seuls par corrélation et classification, sur la base d'une quantité massive de données.

2.2.3 Types d'apprentissages automatiques

Il existe plusieurs types d'apprentissage automatique. Les plus connus et utilisés sont l'apprentissage supervisé et non supervisé.

2.2.3.1 Apprentissage supervisé

L'apprentissage supervisé ou apprendre par l'exemple propose un mécanisme dans lequel, le système est guidé dans son apprentissage. On lui indique le type de résultat à atteindre en le nourrissant d'exemples. Pour cela, on lui fournit des données d'entrée(x) pour lesquelles le résultat(y) est connu et communiqué au système. Le but est qu'il puisse ensuite généraliser ce qu'il a appris pour des données non connues. Par exemple, si le système doit apprendre à reconnaître des bateaux dans une image, on lui fournit des images où le bateau est signalé et où le label « bateau » est associé. On parle ainsi de données étiquetées ou annotées. Le jeu de données d'entraînement annotées permet au système de calculer ses erreurs en comparant ses résultats avec les résultats connus et ainsi d'ajuster le modèle pour progresser. Une partie des données annotées (non utilisées pendant l'entraînement) pourra également servir à vérifier l'efficacité du modèle, une fois l'apprentissage terminé. L'apprentissage supervisé est effectué notamment dans un contexte de classification et de régression.

- La classification : c'est une tâche qui consiste à attribuer une classe à un objet.
- La régression : ici on n'attribue pas une classe mais une valeur mathématiques pouvant être un pourcentage ou une valeur absolue. Par exemple, une probabilité pour qu'une machine tombe en panne, ou le prix de vente idéal d'un objet en fonction de ses critères.

2.2.3.2 Apprentissage non supervisé

Dans le cas de l'apprentissage non supervisé, on ne donne pas d'exemples de résultats attendus au système. Seules les données d'entrée sont fournies et le système doit apprendre,

de manière autonome, la meilleure façon d'explorer les données. Il doit chercher à identifier dans le jeu de données une façon de les structurer (trouver des modèles ou « patterns ») ou encore à extraire des caractéristiques particulières. La performance permettant l'ajustement du modèle est alors appréciée grâce à des indicateurs objectifs, comme, par exemple, des calculs de variabilité interclasses. On parlera d'apprentissage par exploration.

Les tâches réalisables grâce à cette méthode sont notamment les tâches de clustering. Ici la machine essaie de grouper des objets dans des ensembles de données les plus homogènes possible. Cette technique peut sembler proche de la classification, mais à la différence de cette dernière, les classes ne sont pas pré-remplies par un humain.

Bon à savoir : Notons que si l'apprentissage non supervisé est une sorte de « Graal » [12] pour les chercheurs en intelligence artificielle [7] et si d'autres types d'apprentissage ouvrent de belles perspectives, l'apprentissage supervisé reste de loin le plus exploré et pratiqué à l'heure actuelle, notamment pour des applications de régression ou de classification pour lesquelles il donne de meilleurs résultats.

2.3 Du machine learning au deep learning

Le deep learning est une manière particulière de faire du machine learning. C'est une technique de machine learning reposant sur les réseaux de neurones. Il vise à mimer la "complexité" des neurones du cerveau humain. Les algorithmes du deep learning sont composés d'unités reliées entre elles et qui vont s'entre-activer. A l'instar du cerveau humain, chaque neurone peut produire un effet sur ceux auxquels il est connecté. Toutefois au lieu d'un signal électrique comme dans le cerveau humain, le réseau de neurones attribue un certain poids à différents neurones. Ces derniers sont organisés en couches et plus le nombre de couches est important, plus l'apprentissage est profond, d'où le terme "Deep". Le point suivant nous permettra de comparer amplement le deep learning à sa discipline mère qu'est le machine learning et de définir le pourquoi de son adoption.

2.3.1 Deep learning vs machine learning

L'apprentissage profond et l'apprentissage automatique sont tous les deux capables de gérer les grosses bases de données. Toutefois, les réseaux de neurones profonds évoluent beaucoup mieux avec plus de données que les outils traditionnels de Machine Learning qui ont des problèmes lorsqu'ils sont confrontés au Big Data. Souvent, le meilleur conseil pour améliorer la précision avec un réseau de neurones profond est simplement d'utiliser plus de données ! Avec les algorithmes de ML classiques, cette solution rapide et facile ne fonctionne pas aussi bien et des méthodes plus complexes sont souvent nécessaires pour améliorer le résultat de classification ou de régression. Par exemple, si nous ne disposons

que de 100 images, les arbres de décision, les K-plus proches voisins et d'autres modèles de ML nous seront beaucoup plus efficaces que l'utilisation d'un réseau de neurones profond. En effet, les techniques traditionnelles d'apprentissage automatique requièrent une expertise humaine en matière d'extraction de caractéristiques. Habituellement, une analyse exploratoire approfondie des données est d'abord effectuée. Une réduction de la dimension de la base pourrait alors être effectuée pour faciliter le traitement et rendre les motifs plus visibles, pour que les algorithmes d'apprentissage fonctionnent bien. Cela n'est pas nécessaire lors de l'utilisation d'un réseau profond vu son succès dans l'extraction automatique de caractéristiques via des transformations non-linéaires multi-couches.

le principal avantage des algorithmes d'apprentissage profond est qu'ils essaient d'apprendre les caractéristiques de haut niveau à partir de données de manière incrémentielle. La figure 2.1 explique d'avantage cette différence. Nous n'avons qu'à donner une entrée et la machine se charge d'extraire les caractéristiques.

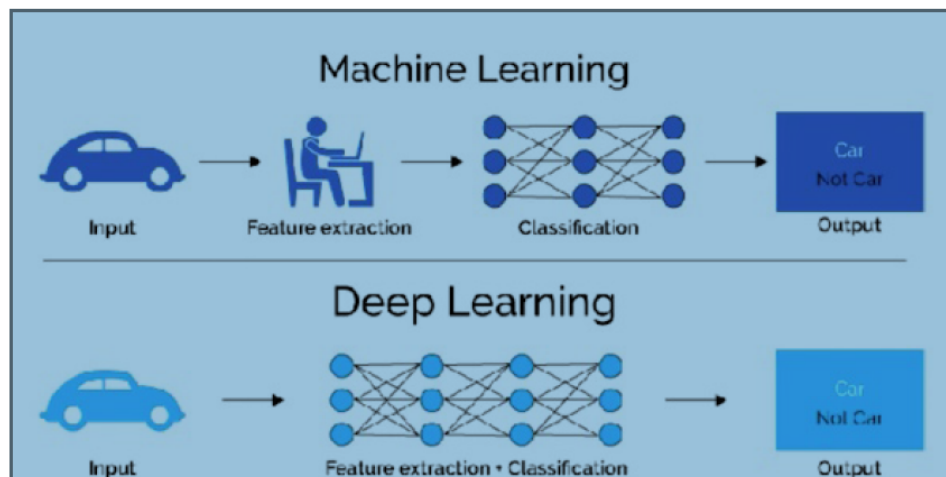


FIGURE 2.1 – Machine learning vs Deep learning

2.4 Les réseaux de neurones artificiels

On ne peut parler du Deep learning sans évoquer les réseaux de neurones artificiels. En effet, le deep learning est un ensemble d'outils et de méthodes d'apprentissage automatique basés sur l'utilisation « avancée » de réseaux de neurones. Les réseaux de neurones artificiels s'inspirent du fonctionnement des réseaux de neurones du cerveau humain. Le cerveau est constitué d'un très grand nombre de cellules spécialisées, les neurones, étroitement liées entre elles par des milliers d'interconnexions. Chaque neurone est capable de traiter de multiples informations en provenance de son environnement et de réagir en créant des messages électrochimiques.

Un neurone seul ne réalise que des opérations élémentaires, mais associés en réseau, les neurones permettent aux Hommes d'accomplir des tâches cognitives extrêmement com-

plexes. Les réseaux de neurones artificiels reprennent ce principe en inter-connectant de multiples unités élémentaires, les neurones formels, dans le but de résoudre des problèmes. Ils sont organisés en couches où chaque couche répond à un niveau d'abstraction d'un problème. Des calculs sont effectués grâce à des nombres, appelés paramètres ou poids du réseau, de telle sorte que l'imitation et la prévision soient aussi fidèles que possible. Ils résolvent des problèmes d'ordre statistique : analyse de données, prévision, classification. Ils s'appliquent de même à la robotique notamment dans le contrôle et le guidage de robots ou de véhicules, et répondent aux questions liées à la reconnaissance de formes, des signaux, des événements (lecture automatique de code postaux ou de chèque, reconnaissance d'images, reconnaissance de parole, etc) [14].

2.4.1 Origine des réseaux de neurones artificiels et naissance du deep learning

Le neurone artificiel de MacCulloch et Pitts :

On entend parler de neurone artificiel dès 1943 lorsque Warren MacCulloch et Walter Pitts publient leur premier modèle mathématique et informatique des neurones biologiques : le neurone formel. Rappelons le, en biologie les neurones sont des cellules excitables, connectées les unes aux autres et ayant pour rôle de transmettre des informations dans notre système nerveux. Chaque neurone est composé de plusieurs dendrites, d'un corps cellulaire et d'un axone. Les dendrites sont en quelque sorte les portes d'entrée du neurone. C'est à cet endroit, la synapse, que le neurone reçoit des signaux lui provenant des neurones qui le précèdent. Ces signaux peuvent être de type excitateurs ou inhibiteur.

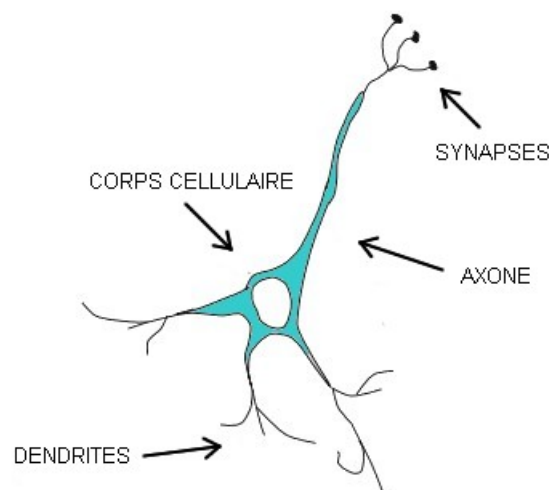


FIGURE 2.2 – Structure du neurone biologique

Ce que MacCulloch et Pitts ont essayé de faire, c'est de modéliser ce fonctionnement, en supposant que ce neurone pouvait être représenté par une fonction f de transfert, qui

prend en entrée des signaux $X(x_1, x_2, x_3, \dots)$ et qui retourne une sortie y . Dans cette fonction, on trouve deux grandes étapes :

Une agrégation qui fait la somme de toutes les entrées du neurone, en multipliant au passage chaque entrée par un coefficient $W(w_1, w_2, w_3, \dots)$. Ce coefficient représente en fait l'activité synaptique c'est à dire, le fait que l'activité soit excitateur auquel cas w vaut $+1$, ou-bien inhibiteur et w vaut -1 .

L'expression qu'on aura finalement à cette phase est :

$$f = w_1x_1 + w_2x_2 + w_3x_3 + \dots \quad (2.4.1)$$

Une fois cette étape réalisée, on passe à **l'activation** : on regarde le résultat du calcul effectué précédemment et si celui ci dépasse un certain seuil, en général si $f \geq 0$, alors le neurone s'active et retourne en sortie $y = 1$ sinon $y = 0$.

A l'origine, ce modèle n'était utilisé que pour traiter des problèmes logiques qui valent soit 0 ou 1.

En 1949, Donald Hebb met en évidence l'importance du couplage synaptique dans l'apprentissage par renforcement ou dégénérescence des liaisons inter-neurales lors de l'interaction du cerveau avec le milieu extérieur. Le principe est que lorsque deux neurones sont activés en même temps, alors la force de connexion augmente. Cependant, le premier modèle de réseau de neurone opérationnel est le perceptron inspiré du modèle rétinien et capable d'apprentissage.

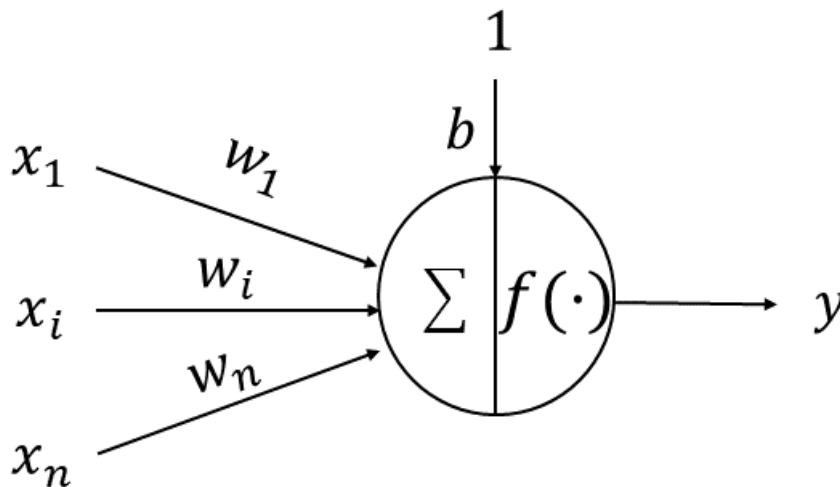


FIGURE 2.3 – Le neurone formel de MacCulloch et Pitts

Le perceptron de Franck Rosenblatt :

Face aux limites que présentaient le neurone de MacCulloch et Pitts, notamment le fait qu'il ne disposait pas d'algorithme d'apprentissage et donc qu'il fallait choisir relativement les paramètres w pour d'éventuelles applications au monde réel, Franck Rosenblatt proposa en 1957 le premier algorithme d'apprentissage, le perceptron.

Ce modèle ressemble beaucoup au précédent, il s'agit d'un neurone artificiel qui s'active lorsque la somme pondérée de ses entrées dépasse un certain seuil. Mais avec ça, le perceptron dispose d'un algorithme d'apprentissage lui permettant de trouver les valeurs des paramètres w afin d'obtenir les sorties y qui nous conviennent. Pour développer cet algorithme Franck Rosenblatt s'est inspiré de la théorie de Hebb. Donc à partir de cela, il développe un algorithme d'apprentissage qui consiste à entraîner un neurone artificiel, sur des données de référence (X, y) pour que celui-ci renforce ses paramètres W à chaque fois qu'une entrée X est activée en même temps que la sortie y présente dans ces données. Pour cela, il a imaginé une formule dans laquelle les paramètres W sont mis à jour, en calculant la différence entre la sortie de référence et la sortie produite par le neurone et en multipliant cette différence par la valeur de chaque entrée X ainsi que par un pas α d'apprentissage positive :

$$W = W + (\alpha * (y_{true} - y) * X) \quad \text{soit :} \quad (2.4.2)$$

y_{true} : sortie de référence

y : sortie produite par le neurone

X : entrée du neurone

α : vitesse d'apprentissage.

Si notre neurone sort une sortie différente de celle attendu, par exemple $y = 0$ alors que on voudrait avoir $y_{true} = 1$ alors la formule (2.4.2) donnera :

$$\begin{aligned} W &= W + \alpha(1 - 0)X \\ &= W + \alpha * (X) \end{aligned}$$

Donc pour les entrées ($x_1 = 1$) leur poids w_1 sera renforcé d'un petit pas α : $w_1 = w_1 + \alpha$. Cela rapprochera donc notre neurone de son seuil d'activation. Aussi longtemps qu'on sera en dessous de ce seuil, le coefficient w continuera d'augmenter grâce à notre formule, jusqu'à ce que $y_{true} = y$ et en ce moment (2.4.2) donnera $W = W + 0$ entraînant l'arrêt de l'évolution des paramètres. Les limites du perceptron mono-couche du point de vue performance ont été montrées en 1969 par les mathématiciens Minsky et Papert notamment parce que le perceptron est un modèle linéaire. Les travaux d'Hopfield en 1982 ont montrés que des réseaux de neurones artificiels étaient capables de résoudre

des problèmes d'optimisation et ceux de kohonen ont montrés qu'ils étaient capables de résoudre des tâches de classification et de reconnaissance.[21]

Le perceptron multicouche :

Le perceptron est un modèle linéaire, et avec ce dernier, on peut par exemple séparer deux classes de points. Cependant le seul ennui c'est qu'une grande partie des phénomènes universels ne sont pas linéaires et dans ces conditions le perceptron à lui seul n'est pas très utile. Certes en connectant plusieurs neurones il est possible de résoudre des problèmes plus complexes. Avec au moins deux couches(une couche d'entrée et une couche de sortie), on a là le premier véritable réseau de neurone artificiel : le perceptron multi-couche inventé par Geoffrey Hinton en 1986. Dans les années 1990 on commença à développer les premières variantes du perceptron multi-couche.

Le célèbre Yann LeCun inventa le premier réseau de neurone convolutif réseau capable de reconnaître et de traiter des images en introduisant au début de ces réseaux des filtres mathématiques qu'on appelle convolution et pooling. En 1957 apparaît les réseaux de neurones récurrents qui permettent de traiter des séries temporelles comme la lecture du texte ou encore la reconnaissance vocale.

L'envol du deep learning :

De leur invention jusque dans les années 2000, les réseaux de neurones utilisés étaient des réseaux peu profonds. Ils étaient utilisés parmi la gamme des outils de machine learning mais pour des usages limités du fait de la nécessité de disposer de grandes puissances de calculs . Aussi, d'autres algorithmes, tels les SVM ou les Random Forest, jugés plus performants, leur étaient souvent préférés. Mais tout cela c'était avant 2012. En fait, deux facteurs clés peuvent expliquer le saut de performance soudain qu'a réalisé le réseau de neurone artificiel : les progrès matériels, notamment ceux des processeurs graphiques, ou GPU, qui permettent de disposer d'une puissance de calcul bien supérieure à ce qu'elle était jusque dans les années 90. Internet et la constitution de gigantesques bases de données annotées (telle ImageNet) permettant de bénéficier de données d'apprentissage en quantité suffisante pour obtenir des résultats concluants. On avait la possibilité maintenant de rajouter de nouvelles couches cachées aux réseaux existants et vue cela, de nouvelles techniques furent mises au point : nouvelles architectures, nouveaux types de couches, nouvelles méthodes de calculs... C'est cet ensemble de nouvelles méthodes qui, aujourd'hui, constituent le Deep Learning.[7].

2.4.2 Quelques algorithmes de Deep learning

Le nombre d'architectures et d'algorithmes utilisés dans l'apprentissage en profondeur est large et varié. Nous avons notamment :

- Le réseau de neurones convolutif destiné au traitement des images. Une section sera totalement dédiée à ce réseau dans ce document pour une explication plus détaillée.
- Un réseau de neurones récurrent ou RNN est un type de réseau de neurones artificiels principalement utilisé dans la reconnaissance vocale et le traitement automatique du langage naturel (TAL, NLP, TALN). La récurrence de ce réseau vient du fait que les informations qu'elles calculent vont être stockées en mémoire pour être réutilisées sur un prochain calcul. Ainsi, lorsqu'on veut prédire une valeur à un temps $t+1$, le modèle RNN prend en compte les précédentes valeurs dans sa prédiction.
- Les réseaux antagonistes génératifs RAG ou GAN sont une classe d'algorithme d'apprentissage non supervisé. Ils permettent de générer des données avec un fort degré de réalisme.

En plus de ces trois algorithmes nous avons les réseaux de fonction de base radiale, les réseaux de mémoires à long et court termes et bien d'autres.

2.5 Les réseaux de neurones convolutifs

Les réseaux de neurones convolutifs désignent une sous-catégorie de réseaux de neurones. Cependant, les CNNs sont spécialement conçus pour traiter des images en entrée. L'idée derrière est de filtrer les images avant d'entraîner le réseau neuronal profond. Après avoir filtré l'image, les caractéristiques de celle-ci peuvent être plus proéminente et peuvent être récupérées pour identifier un objet. Un filtre est tout simplement un ensemble de multiplicateurs. L'architecture des CNNs est alors spécifique : elle est composée de deux blocs principaux comme indiqué dans la figure 2.4 ci dessous. .

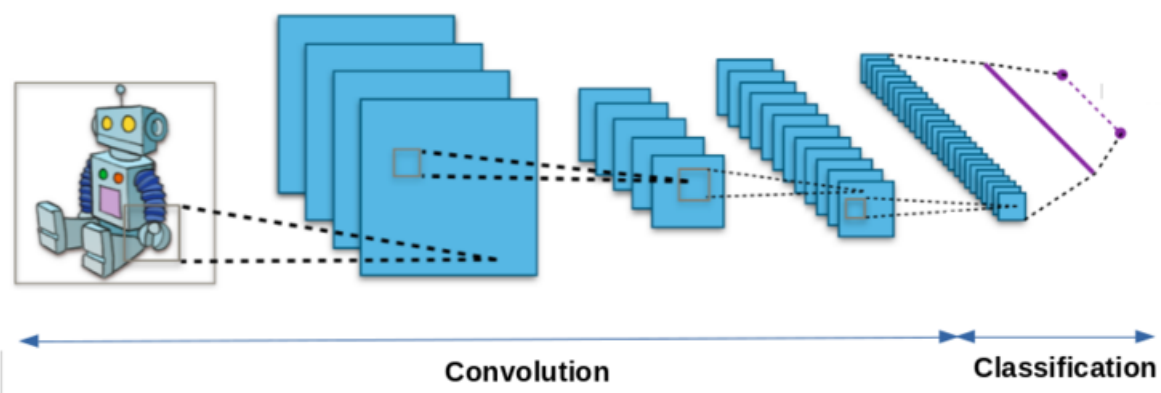


FIGURE 2.4 – Architecture d'un réseau de neurones convolutif

1— Le premier bloc (bloc de convolution) fait la particularité de ce type de réseau de neurones, puisqu'il fonctionne comme un extracteur de features (caractéristiques). Pour extraire les caractéristiques, le CNN applique des opérations de filtrage par convolution. La première couche filtre l'image avec plusieurs noyaux de convolution, et renvoie des "feature maps" ou cartes d'attributs, qui sont ensuite normalisées (avec une fonction d'activation) et redimensionnées. Ce procédé peut être réitéré plusieurs fois : on filtre les features maps obtenues avec de nouveaux noyaux, ce qui nous donne de nouvelles features maps à normaliser et redimensionner, et qu'on peut filtrer à nouveau, et ainsi de suite. Finalement, les valeurs des dernières features maps sont concaténées dans un vecteur. Ce vecteur définit la sortie du premier bloc, et l'entrée du second.

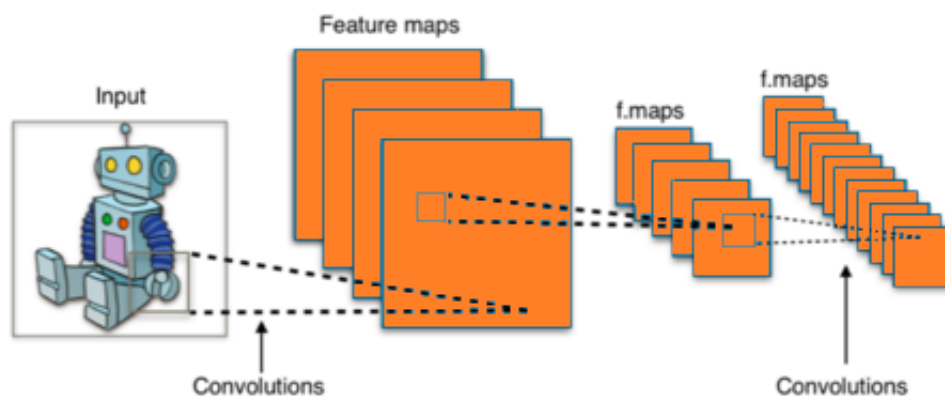


FIGURE 2.5 – Bloc de convolution

2— Le second bloc (classification) : les valeurs du vecteur en entrée sont transformées (avec plusieurs combinaisons linéaires et fonctions d'activation) pour renvoyer un nouveau vecteur en sortie. Ce dernier vecteur contient autant d'éléments qu'il y a de classes.

Il existe quatre types de couches pour un réseau de neurones convolutif : la couche de convolution, la couche de pooling ou sous-échantillonnage, la couche de correction ReLU et la couche fully-connected ou couche entièrement connectée (cf.figure 2.6).

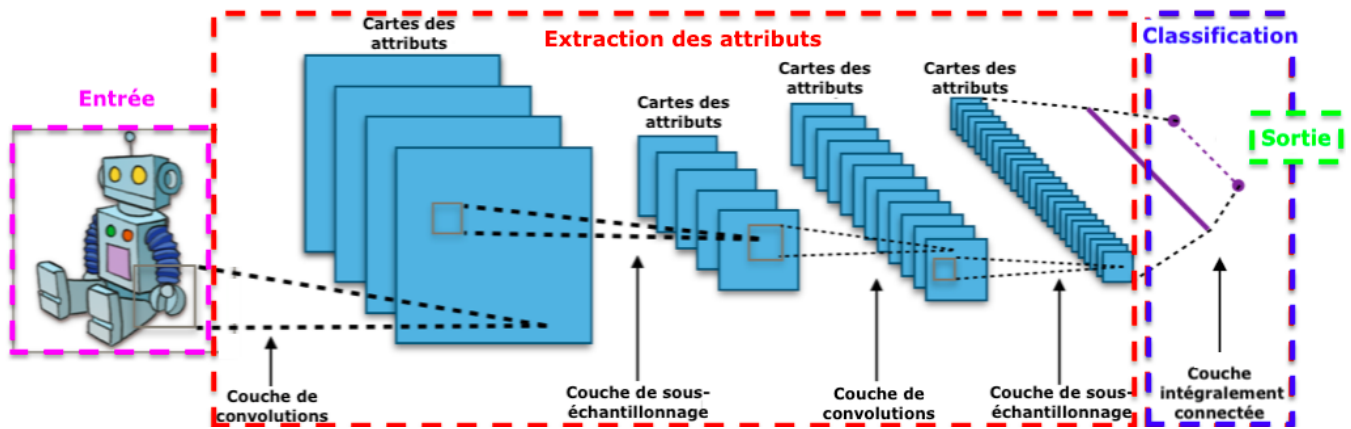


FIGURE 2.6 – Architecture générale d'un CNN

La couche de convolution :

Cette couche est la composante clé des réseaux de neurones convolutifs. son rôle est de repérer la présence d'un ensemble de features dans les images reçus en entrée. On parle de filtrage par convolution. Il s'agit en effet de faire glisser une fenêtre représentant la feature sur l'image et ainsi calculer le produit de convolution entre le feature et chaque portion de l'image balayée.

On obtient pour chaque paire (image, filtre) une carte d'activation ou feature maps qui indique où se situent les caractéristiques (features) dans l'image.

La couche de pooling :

Placée entre deux couches de convolutions, la couche de pooling reçoit en entrée plusieurs feature maps et applique à chacune d'elles l'opération de pooling.

Cette opération consiste à réduire la taille des images, tout en préservant leurs caractéristiques importantes. Le nombre de paramètres et le nombre de calculs dans le réseau se voient alors réduits. Ainsi on améliore l'efficacité du réseau tout en évitant le sur-apprentissage.

La couche de correction ReLU :

ReLU(Rectified Linear Units) désigne la fonction réelle non linéaire définie par :

$$ReLU(x) = \max(0, x) \quad (2.5.1)$$

Elle remplace donc toutes les valeurs négatives reçues en entrée par des zéros. Elle joue le rôle de fonction d'activation.

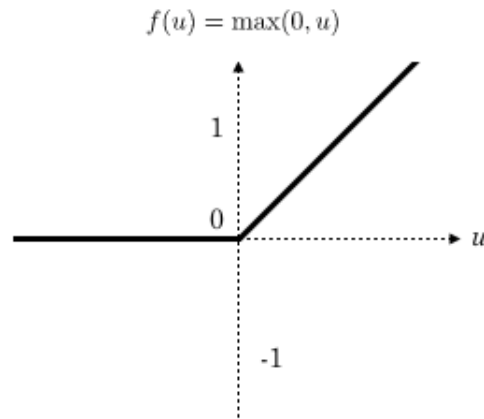


FIGURE 2.7 – Allure de la fonction ReLU

La couche fully-connected :

La couche fully-connected est toujours placée en fin d'architecture du CNN et est entièrement connectées à tous les neurones de sorties. Après avoir reçu un vecteur en entrée, cette couche se charge d'attribuer une classe à chaque entrée.

Ainsi se définit le fonctionnement et l'architecture des réseaux de neurones convolutifs dont la spécificité est sa partie convolutive.

2.6 Conclusion

Dans cette partie, nous avons parcouru le domaine du deep learning en apprenant plus sur ses origines. Nous avons visité ses algorithmes les plus utilisés notamment les réseaux de neurones profonds ou CNNs : ce sont des modèles de programmation puissants permettant notamment la reconnaissance d'images. A propos des images, quand on en parle, ce qui vient à l'esprit ce sont des images classiques, en couleurs, etc et concernant ces dernières on peut retrouver partout de multiples travaux qui ont été réalisés. On a vu dans le précédent chapitre le système radar et les images qu'il produit. Dans le chapitre suivant, nous verrons comment les CNNs classifient ces types d'images, la méthode diffère-t-elle du processus classique ?

Classification des images radars à l'aide du deep learning

Sommaire

3.1	Introduction	24
3.2	Les motivations de la classification des images	25
3.3	État de l'art des processus de classification des images radars	26
3.3.1	Bref aperçu des méthodes de classification des images radars avant les CNNs	26
3.3.2	Revue littéraire des travaux sur la classification des images radars grâce aux CNNs	27
3.3.3	Analyse personnelle	28
3.4	Le manque de données radars disponibles : enjeux	29
3.5	Perspectives au manque de données radars pour la classification	29
3.5.1	Transformation des images	29
3.5.2	La simulation de données : les réseaux de neurones génératifs antagonistes à l'honneur	30
3.5.3	L'arrêt anticipé de l'apprentissage	32
3.5.4	Le dropout	32
3.6	Conclusion	33

3.1 Introduction

Le radar est le plus souvent utilisé pour la surveillance de cibles en raison de sa capacité à pénétrer dans divers conditions météorologiques et de luminosité. Cependant force est de reconnaître que ces images sont très difficiles à cerner : elles sont bruitées, en quantité non suffisante et très difficile d'accès.

Par le passé, le système traditionnel de reconnaissance des images radars se divisait en trois points essentiels : le pré-traitement des images, l'extraction des caractéristiques la classification et l'identification.

De nos jours, l'apprentissage en profondeur a obtenu un succès de pointe dans de nombreux domaines notamment la classification, la détection, etc. Mais l'application réussie de cette technique nécessite une certaine quantité de données. Dans ce chapitre nous voyons comment appliquer les réseaux de neurones profonds aux images radars malgré des limites que nous posent ces images.

3.2 Les motivations de la classification des images

En rappel la classification des images consiste à répartir des images selon des classes établies au préalable. Classer une image lui fait correspondre une classe, marquant ainsi sa parenté avec d'autres images.

L'objectif de la classification d'images est d'élaborer un système capable d'affecter une classe automatiquement à une image. Ainsi, ce système permet d'effectuer une tâche d'expertise qui peut s'avérer coûteuse à acquérir pour un être humain en raison notamment de contraintes physiques comme la concentration, la fatigue ou le temps nécessité par un volume important de données images.

Les applications de la classification automatique d'images sont nombreuses et vont de l'analyse d'images médicales, aux données images industrielles, en passant par le domaine militaire. Ainsi on retrouve des applications dans le domaine médical comme la reconnaissance de cellules. Il existe des applications pour la reconnaissance d'écriture manuscrite pour les chèques, les codes postaux. Nous avons des applications dans le domaine urbain comme la reconnaissance de panneaux de signalisation, la reconnaissance de piétons, la détection de véhicules, la reconnaissance de bâtiments pour aider à la localisation. Dans le domaine de la biométrie, on en retrouve de même, comme la reconnaissance de visage, d'empreintes etc.

Le point commun à toutes ces applications est qu'elles nécessitent la mise en place d'une chaîne de traitement à partir des images disponibles, composée de plusieurs étapes afin de fournir en sortie une décision. Chaque étape de la mise en place d'un tel système de classification nécessite la recherche de méthodes appropriées pour une performance globale optimale.

Dans un contexte de deep learning il faut d'abord préparer les données, c'est à dire nettoyer la base de données, ajuster les tailles des images, diviser le jeu de données en sous groupes pour l'entraînement et le test etc, ensuite construire le modèle CNN par exemple, effectuer l'entraînement et enfin évaluer le modèle.

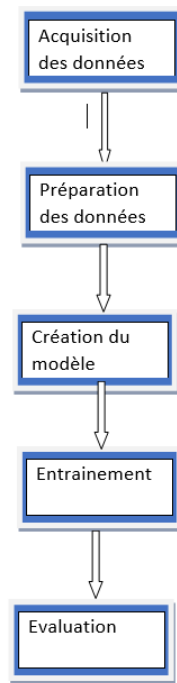


FIGURE 3.1 – Processus de classification d'images en deep learning

3.3 État de l'art des processus de classification des images radars

3.3.1 Bref aperçu des méthodes de classification des images radars avant les CNNs

Avant les grands succès de l'apprentissage profond pour la vision par ordinateur, une grande partie des recherches sur les algorithmes traditionnels a consisté à construire des pipelines modulaires élaborés, conçus à la main, de composants comprenant des sous-étapes de pré-traitement d'images conçues de même à la main, des étapes d'extraction de caractéristiques conçues à la main, un traitement potentiel supplémentaire des caractéristiques comme la normalisation et la discrimination, et enfin la classification. Pour différencier les approches, il faut notamment caractériser de manière générale les méthodes de représentation et de raisonnement.

De nombreuses approches de l'extraction et du pré-traitement des caractéristiques ont été étudiées. Les approches d'extraction de caractéristiques comprennent des techniques de vision par ordinateur telles que la détection des coins et des bords, le lissage, le filtrage et le masquage, des méthodes bayésiennes telles que l'ajustement de la distribution de probabilité et les modèles graphiques, la modélisation du radar et de ses propriétés, ainsi que des techniques de traitement du signal telles que les transformées en ondelettes.

Les techniques de pré-traitement comprennent le filtrage et le masquage dans le but de séparer la cible des fouillis et de l'ombre, les transformations pour estimer la longueur et la largeur de la cible, la translation de l'image et de la cible, la mise à l'échelle de l'image et de la cible, ainsi que les algorithmes d'estimation de la pose et le centrage utilisés pour normaliser la pose des cibles d'entrée. Des classificateurs tels que les machines à vecteurs de support (SVM), les algorithmes de boosting adaptatifs, les algorithmes génétiques, les classificateurs statistiques, les k-voisins les plus proches et la correspondance de modèles ont tous été utilisés avec diverses propriétés et compromis associés à chacun cf. [23] [4].

3.3.2 Revue littéraire des travaux sur la classification des images radars grâce aux CNNs

Les CNNs au cœur du processus, remplacent les pipelines de pré-traitement, les caractéristiques conçues à la main et les classificateurs par des modèles appris de bout en bout. Ces classificateurs appris de bout en bout offrent des avantages substantiels par rapport aux méthodes traditionnelles.

En 2021 L.st. Angel montre que nous pouvons tirer parti des architectures de modèles de DCNN et de GAN pour classer les images radars. Son expérience se passe dans un environnement domestique où il prend en compte trois classes de cibles (Hommes, chiens et chats). Compte tenu du manque de données radars, et n'étant pas prêt de collecter des ensembles de données radars qui peuvent être gourmands en ressources et sujets aux erreurs, il propose d'utiliser des techniques de multiplication des quelques images étiquetées grâce au réseaux antagonistes génératifs GAN. De cette manière, il est plus facile de développer un classificateur d'images radars très performant. De ce procédé il obtient à l'issue de ses tests une performance de 80%.

Cris. Coman et al. cf.[8] s'intéressent aux images radars à synthèse d'ouverture. Ils soulignent dans l'article l'importance de l'amplitude dans l'interprétation des images radars, mais proposent d'associer cette information à la notion de phase qui selon eux n'est pas à négliger. Ainsi, le défi pour eux, était d'exploiter la phase qui fournit des informations discriminatoires supplémentaires, pour augmenter la précision de la classification et résoudre certains problèmes liés à l'ajustement excessif et à la formation sur des ensembles de données réduits. Ils ont de même fait appel aux réseaux de neurones convolutifs mais à la différence de l'auteur précédent, leurs travail porte plutôt sur un jeux données SAR. Ils mènent leur expérience en comparant deux réseaux : l'un basé sur l'amplitude et l'autre sur l'amplitude et la phase. Les deux réseaux n'avaient que des entrées différentes ($88 * 88 * 1$ et $88 * 88 * 3$ respectivement). Le résultat fut positif dans le sens où la précision avait augmentée (97% contre 98%), certes restait toujours proche de la précision précédente.

H. Wang et al. reviennent sur le manque de données radars disponibles dans leur article intitulé "APPLICATION OF DEEP-LEARNING ALGORITHMS TO MSTAR DATA" cf. [15] tout en montrant l'importance de ce facteur quantité de données pour les réseaux de neurones profonds. Dans ce article, l'utilisation d'un modèle pré-entraîné est la solution préconisée.

R. Melzer, W.M. Sever et al.[23] remarquent tout d'abord une utilisation non justifiée des architectures de réseaux simples et artisanales pour la classification des images radars. Ensuite, jugent les méthodes systématiques de choix des configurations d'architecture et des hyperparamètres les plus performants, non plus exploitées. Enfin, selon ces auteurs les travaux antérieurs comparent souvent des détails tels que le choix de l'optimiseur ou les choix d'initialisation des poids plutôt que le choix de la topologie de l'architecture, le coût et l'efficacité du calcul, ou la capacité à généraliser à partir de quelques exemples à des données mesurées. Étant à recherche des meilleurs modèles pour les systèmes radars, spécifiquement à synthèse d'ouverture SAR, ils préconisent de s'inspirer des travaux de pointe existants en matière de classification d'images et de reconnaissance d'objets pour choisir les architectures et optimiser leurs topologies, leurs paramètres et leurs hyperparamètres de manière plus systématique, puis à comparer ces architectures sur les axes de performance. Mais malgré cette stratégie, un défi s'est opposé à eux : l'insuffisance de données. Pour pallier à cela, un appel fut fait à des outils de calcul électromagnétiques (EM) qui ont été développés pour simuler des retours radars.

Di Zhang et al. dans leur article "Transfer Learning with Convolutional Neural Networks for SAR Ship Recognition" dépeignent la capacité des CNNs à détourner le problème de manque d'images grâce au transfer learning. Ainsi, pour la classification des images radars de bateaux, ils utilisent un réseau pré-entraîné sur des données MSTAR de chars et voitures blindées de type VGG-X. En conclusion ils assurent qu'une simple procédure d'apprentissage de transfert pourrait donner de meilleurs résultats de reconnaissances de véhicules SAR. suffisantes.

3.3.3 Analyse personnelle

De la littérature, nous observons tout d'abord la diversité de données radars utilisées par les chercheurs : des systèmes radars domestiques aux capteurs radars satellites, etc. Nous retenons ce manque de données important qu'il faut réellement prendre en compte lors de l'implémentation d'un modèle d'apprentissage profond sur les images radars. Soulignons de même le fait que les réseaux de neurones convolutifs restent au cœur des méthodes proposées dans la littérature.

La remarque sur le facteur chatoiement ou bruit ne fait pas en réalité couler l'ancre des chercheurs. Les CNNs sont bien aguerris et peuvent recevoir en entrée des données bruitées ou pré-traitées. Ils détiennent cette force qu'est la couche de convolution, une couche

comportant un ensemble de filtres qui décèlent les moindres caractéristiques liées à l'image (courbes, lignes, formes, etc). De ce fait l'inaccessibilité aux ensembles de données pause le véritable problème à l'application du deep learning aux images radars.

Par ailleurs comme tout bon modèle de classification, il faut savoir choisir les bons paramètres qui optimisent l'apprentissage.

3.4 Le manque de données radars disponibles : enjeux

Un réseau de neurones apprend grâce à des exemples (jeux d'entraînements) qui lui sont soumis. Le but de cet apprentissage est de permettre au réseau de tirer de ces exemples des généralités et de pouvoir les appliquer à de nouvelles données par la suite. En intelligence artificielle, on parle de sur-apprentissage ou overfitting en anglais, quand un modèle a trop appris les particularités de chacun des exemples fournis en exemple. Il présente alors un taux de succès très important sur les données d'entraînement au détriment de ses performances générales réelles.

3.5 Perspectives au manque de données radars pour la classification

La data augmentation consiste à faire croître le nombre de données dont on dispose pour notre apprentissage. la réalisation de cette augmentation nécessite au préalable d'une base de données solide et donc assez représentative. Avec cette condition remplie, une augmentation de données nous permet de déboucher sur des modèles plus performants et fiables car robustes aux variations.

Pour procéder à la data augmentation, il existe plusieurs approches.

3.5.1 Transformation des images

Cette approche consiste à transformer les données existantes. Le but n'est donc pas de créer de toutes nouvelles images mais bien d'apporter une certaine variabilité dans les images sources. On va pouvoir augmenter la diversité et donc le champ d'apprentissage de notre modèle, qui va pouvoir mieux s'adapter pour prédire de nouvelles données. Le principe de cette méthode est relativement simple. L'image suivante (figure 3.2) met la lumière sur quelques transformations possibles.



FIGURE 3.2 – l'augmentation des données par transformation

En effet, en partant d'une simple image, nous pouvons la dupliquer autant de fois que nous avons des types de transformation différentes à lui appliquer. Et nous pouvons en augmenter davantage en croisant ces effets (cf. figure 3.2) sur une même image, et en y appliquant différentes valeurs de l'effet dans une fourchette donnée, pour avoir un résultat plus ou moins poussé.

3.5.2 La simulation de données : les réseaux de neurones génératifs antagonistes à l'honneur

Contrairement à la transformation de données qui permet de générer des variantes de données existantes, la simulation de données vient générer des données totalement nouvelles. Seul point commun : avoir un jeu de données initial suffisamment représentatif. Dans cette approche nous connaissons un réseau de neurone qui ne cesse d'impressionner le monde : les GANs.

Les réseaux antagonistes génératifs sont des architectures algorithmiques qui utilisent deux réseaux de neurones, s'opposant l'un à l'autre afin de générer de nouvelles instances synthétiques de données pouvant passer pour des données réelles. Les GANs ont été introduits dans un article d'Ian Goodfellow et d'autres chercheurs de l'Université de Montréal, dont Yoshua Bengio, en 2014.

3.5.2.1 Éléments constitutifs d'un GAN

Le GAN est principalement composé des éléments suivants :

- Une collection de données d'entraînement qui servira de référence pour le modèle.

- Un réseau de neurones génératif dont la morphologie est appropriée à la génération des données, contrairement aux réseaux de neurones adaptés à la classification, la régression, etc. Ces réseaux de neurones auront une couche de sortie permettant de produire une donnée résultante, et non une classe spécifique ou une valeur spécifique. Dans un modèle de réseaux de neurones antagonistes, la partie générative sera alimentée initialement par du bruit : des données complètement aléatoires, et non des images ou quoi que ce soit d'autre. L'objectif de ce réseau de neurones est de pouvoir générer, à partir du bruit, des images qui s'apparentent à nos cibles, en n'ayant aucun accès direct aux données d'entraînement. Il devra donc avoir de l'aide pour apprendre à générer des formes adéquates à partir de données complètement aléatoires.
- Un réseau de neurones discriminatif a pour objectif d'aider le réseau de neurones génératif à créer des données vraisemblables. Ce réseau étant entraîné à reconnaître nos cibles, il sera en mesure de distinguer si un nouvel enregistrement est « réel » ou non. Donc, chaque fois qu'il recevra une image générée par le réseau de neurones génératif, il évaluera si elle représente un de nos cible ou pas.
- Le mécanisme d'adversité oppose un réseau de neurones qui tente de générer des images réalistes pouvant se faire passer pour des images réelles, et un réseau discriminatif qui cherche à identifier les fausses images. Rappelons que les premières images créées par le réseau de neurones génératif sont loin d'être réalistes, car elles sont générées à partir d'un bruit de pixels totalement aléatoire ; il n'existe aucun indice sur les formes pouvant constituer un objet donné. Chaque image générée sera jugée comme étant réelle ou pas par le réseau de neurones discriminatif. S'il juge que l'image n'est pas réelle, il propagera un signal d'erreur qui aidera le réseau de neurones génératif à s'ajuster et à générer des formes différentes. Donc, ce processus itératif aidera le réseau de neurones génératif à former du contenu si réaliste que le réseau de neurones discriminatif sera incapable de voir la différence entre les images générées et les images réelles. Ce processus est référencé comme étant une adversité. Un GAN serait donc capable de produire des images si justes qu'un modèle d'apprentissage machine serait incapable de les distinguer.

3.5.2.2 Cadre d'utilisation des GANs

Dans ce travail, nous nous intéressons à la capacité des GANs à créer des images, ressemblant aux images que nous cherchons à classifier ; certes il est important de survoler un peu ce que ces réseaux peuvent réaliser avec d'autres types de données ou ces mêmes

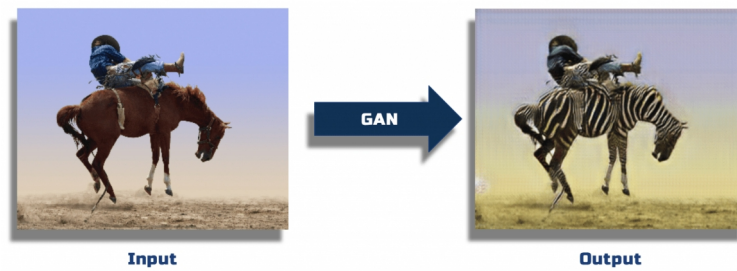


FIGURE 3.3 – Simulation par le GAN

images. Les GANs peuvent utiliser des extraits vidéo pour prédire la suite d'une vidéo et ainsi prolonger indépendamment la vidéo à la fin d'un enregistrement. Pour cela, tous les éléments d'une vidéo, les mouvements, les traitements mais aussi les modifications de l'arrière-plan comme la pluie et les nuages sont pris en compte. Les GAN peuvent également générer des images en se basant sur des descriptions. Ils peuvent par exemple, en se basant sur un scénario, élaborer tout seuls un storyboard, etc.

D'autres approches qui permettent aussi d'entraîner des modèles génératifs sont les auto-encodeurs variationnels (VAEs ou Variational Autoencoders) ou des approches hybrides entre les GANs et les VAEs.

3.5.3 L'arrêt anticipé de l'apprentissage

L'arrêt anticipé interrompt la phase d'entraînement avant que le modèle n'apprenne tous les moindres détails des images. Cependant, il est important de choisir le bon moment, sinon le modèle ne donnera toujours pas de résultat précis.

3.5.4 Le dropout

Les couches "FC" (Fully Connected) occupent la majeure partie de la mémoire du CNN. Pour prévenir le sur-apprentissage, la méthode du dropout est utilisée pour "éteindre" les neurones aléatoirement (avec une probabilité prédéfinie, souvent un neurone sur deux) ainsi que les neurones périphériques. Ainsi, avec moins de neurones, le réseau est plus réactif et peut donc apprendre plus rapidement. À la fin de la séance d'apprentissage, les neurones "éteints" sont "rallumés" (avec leurs poids originaux). Plus la couche FC est proche de l'image source, moins on éteindra de neurones.

L'objectif est d'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.

3.6 Conclusion

En récapitulatif, nous avons parcourue la revue littéraire qui nous a permis de saisir le vrai soucis des images radars face au deep learning notamment l'insuffisance de données sources. Ainsi il est impératif de remédier à ce problème par exemple en augmentant les données afin d'y pallier avant de prétendre à une quelconque classification. Il faut dire que la meilleure manière pour nous d'évaluer concrètement cette situation serait d'arriver à une implémentation pour saisir vraiment l'importance du manque de données en apprentissage en profondeur.

Chapitre 4

Implémentation

Sommaire

4.1	Introduction	34
4.2	Outils et bibliothèques nécessaires	35
4.2.1	TensorFlow	35
4.2.2	Keras	35
4.2.3	Langage Python	35
4.3	Ensemble de données d'analyse	36
4.4	Préparation des données	38
4.4.1	Fractionnement de notre ensemble de données	38
4.4.2	Redimensionnement des données	38
4.4.3	Augmentation de données	38
4.5	Architecture du premier modèle sans augmentation de données	39
4.5.1	Résumé du modèle	40
4.5.2	Entraînement du modèle	40
4.5.3	Visualisation des résultats de l'entraînement	41
4.6	Formation du modèle après une augmentation par transformation des images	42
4.6.1	Augmentation des données	42
4.6.2	Entraînement	43
4.6.3	Évaluation du modèle	43
4.7	Conclusion	44

4.1 Introduction

Dans ce chapitre, nous allons présenter l'architecture d'un modèle CNN mise au point grâce au langage de programmation python, pour la reconnaissance des images radars

MSTAR. Pour une bonne compréhension, nous organisons ce chapitre comme suit : une première section dans laquelle nous présentons les outils et librairies nécessaires dans l'implémentation ; suivie d'une deuxième section, où nous présentons nos données d'analyse ; dans une troisième section nous faisons la préparation des données ; une quatrième section est consacrée à l'architecture du modèle et nous terminons par une cinquième section dédiée à la formation du modèle après augmentation des images.

4.2 Outils et librairies nécessaires

L'open source contribue activement au développement de la recherche scientifique. Elle met gratuitement à la disposition de la communauté scientifique plusieurs librairies entre autres : Tensorflow, Pytorch, Theano, Mxnet, Scikit-learn (cf. figure 4.1).



FIGURE 4.1 – Quelques librairies open source de deep learning

4.2.1 TensorFlow

Plus populaire que Pytorch et les autres, Tensorflow supplante pour l'heure ses concurrents en terme de compétence, d'outillages associés et le tout, allié à une facilité d'utilisation. Parmi ses briques les plus prisées figure son fameux tensorboard qui est une console de visualisation graphique permettant de monitorer l'entraînement des modèles dans l'optique d'optimiser les paramètres d'ajustement.

4.2.2 Keras

Keras est une API de réseaux de neurones de haut niveau, écrite en Python et capable de fonctionner sur TensorFlow ou Theano. Il a été développé en mettant l'accent sur l'expérimentation rapide. Il a été développé à la suite de l'effort de recherche du projet ONEIROS (Open-ended Neuro-Electronic Intelligent Robot Operating System), et son principal auteur et mainteneur est François Chollet, un ingénieur Google.

4.2.3 Langage Python

Python est un langage de programmation de haut niveau interprété (il n'y a pas d'étape de compilation) et orienté objet avec une sémantique dynamique. Il est très sol-

licité par une large communauté de développeurs. Python est un langage simple, facile à apprendre et permet une bonne réduction du cout de la maintenance des codes. Les bibliothèques python encouragent la modularité et la réutilisabilité des codes. Python et ses bibliothèques sont disponibles sans charges pour la majorité des plateformes et peuvent être redistribués gratuitement.

4.3 Ensemble de données d'analyse

Les données alimentent l'apprentissage automatique. La mobilisation de ces données est un processus fastidieux mais reste la condition préalable à la conception de modèles et d'analyses exacts. Dans notre étude, nous nous sommes intéressé particulièrement aux images radars générées grâce aux activités de surveillance et de reconnaissance de cibles aériennes et terrestres. Cette mobilisation est un réel parcours du combattant. En effet, il existe assez de sources, mais une grande difficulté d'accessibilité aux données pour des raisons de sécurité. En s'orientant vers des sites ouverts, nous avons pu collecter quelques images satellites prises grâce aux capteurs radars. Comme satellites doté de capteur actif, nous avons par exemple ENVISAT lancé en 2002, TerraSAR-X une réalisation allemande opérationnel depuis 2008, PALSAR un satellite de télédétection japonaise de 4 tonnes lancé en 2006 et la famille SENTINEL notamment ; Sentinel 1A venue en 2014 pour remplacer ENVISAT par l'union européenne et 1B lancé en 2016 mais hors service en 2022. Parmi ces données satellitaires notre choix s'est porté sur le jeu de données MSTAR (Moving and Stationary Target Acquisition and Recognition) (cf. [11]) recueillies par le Sandia national Laboratory dans le cadre d'un projet parrainé conjointement par la Defense Advanced Research Projects Agency et Air Force Research Laboratory. Ces données sont disponibles sur internet et contiennent des images radars à synthèse d'ouverture (SAR) de 10 cibles dont des chars, des véhicules blindés, des systèmes d'armes et des véhicules du génie militaire (véhicules blindés de transport de troupes : Slicy, BRDM-2, BTR-60 ; char : T-62 ; systèmes d'armes : 2S1 ; unité de défense aérienne : ZSU-234 ; camion : ZIL-131 ; bulldozer : D7).

Auparavant, c'était des analystes formés qui étaient chargés d'exploiter les données SAR brutes pour répondre aux besoins d'identification de cibles d'intérêts ou d'activités significatives. Ainsi, les cibles étaient manuellement recherchées et classées. Avec les réseaux de neurones profonds nous allons apporter cette facilité de reconnaissance et d'exploitation automatique de ces images. Ci-dessous les types de cibles regroupées dans notre jeux de données.

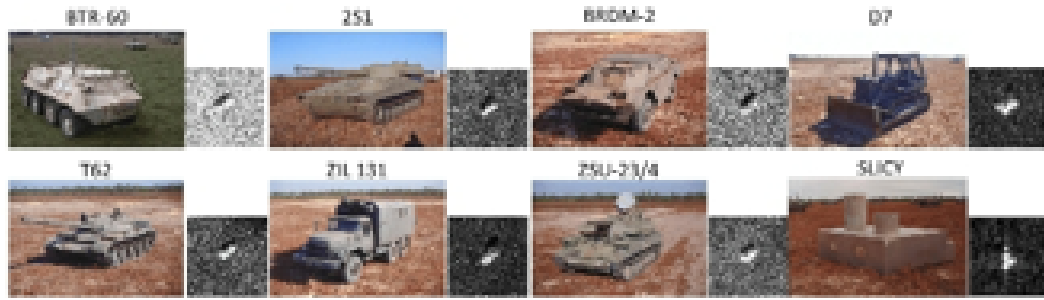


FIGURE 4.2 – Les cibles militaires : images optiques à gauche et images SAR à droite

Explorons le jeu de données en affichant de manière aléatoire quelques images des cibles.

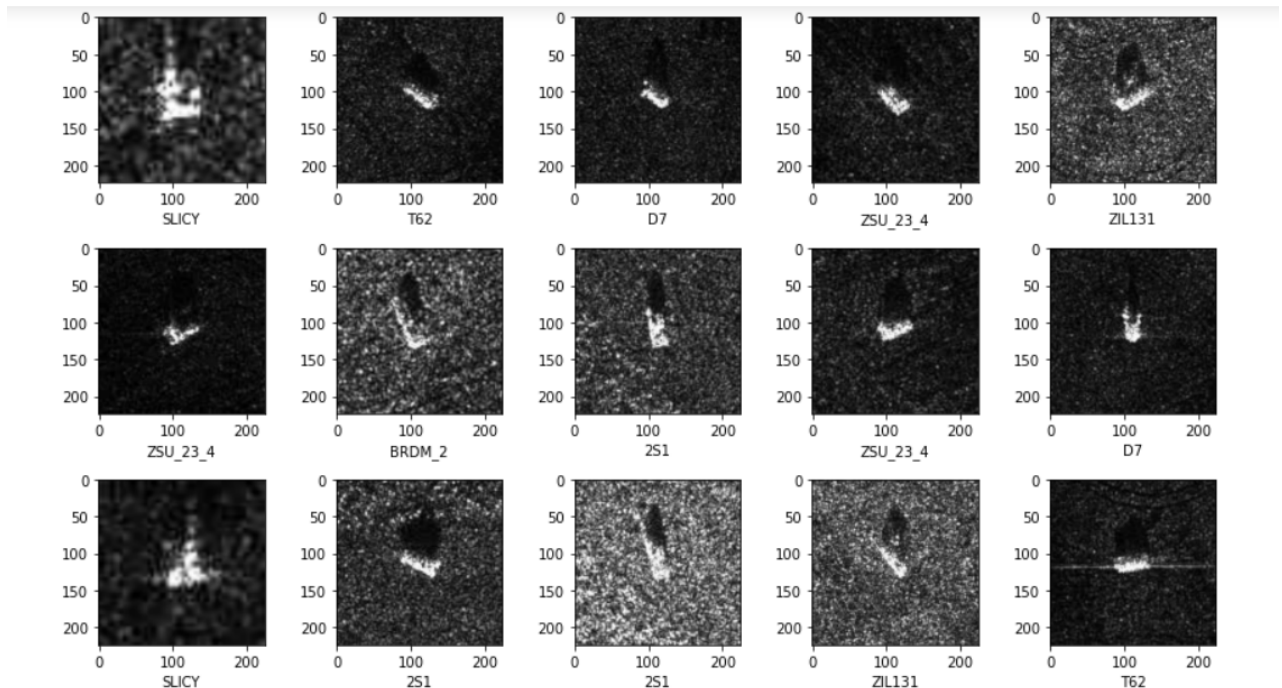


FIGURE 4.3 – Exemples d'images d'entraînement

Il faut dire que chaque type de cible appartient à un dossier et le nom de ce dossier est automatiquement attribué comme étiquette à la cible. Nous avons au total 2121 images que nous considérons bien peu pour un bon modèle.

A partir de 1000 images il est possible de faire une classification et espérer avoir un résultat acceptable ($\simeq 70\%$) sur le jeu d'entraînement, ce qui ne sera peut-être pas évident pour le jeu de test. Le but dans la classification, c'est d'avoir un classificateur d'une grande précision et généralisable. En fait, plus le nombre d'images augmente, plus la machine agrandit son champ d'apprentissage et donc arrive à mémoriser certains détails qui feront que sa précision en terme de classification sera très satisfaisante.

4.4 Préparation des données

Afin de mieux comprendre le côté pratique de notre projet de recherche, nous allons présenter dans ce qui suit les différentes étapes de préparation des données.

4.4.1 Fractionnement de notre ensemble de données

Cette étape consiste à créer au moins deux ensembles différents de données à savoir : les données d'entraînement et les données de validation.

Les données de formation ont pour rôle de permettre un bon apprentissage du modèle lors de l'entraînement. Quant aux données de validation, elles permettent de juger l'efficacité du modèle. Nous utiliserons 80% de notre ensemble de données pour l'entraînement et 20% pour la validation du modèle. Pour se faire, on maintient le même nombre de classes dans le dossier d'entraînement que dans le dossier de validation. Le nombre d'images varie mais la proportion des classes restent telle. Une petite portion de nos données sera réservée comme des données de test. Cette portion ne fait partie ni de l'entraînement, ni de la validation.

4.4.2 Redimensionnement des données

L'une des étapes les plus importantes du processus de pré-entraînement est le redimensionnement d'images entrantes. Cette étape va nous permettre de rendre toutes les images homogènes ; nous avons appliqué un algorithme pour désigner une taille de 100*100 pixels pour toutes les images destinées à l'entraînement de notre modèle.

4.4.3 Augmentation de données

Comme précisé dans le chapitre précédent, pour réaliser une augmentation, il faut appliquer des transformations ou procéder à une simulation. Nous avons plusieurs paramètres tels que la rotation, le décalage en largeur, le décalage en longueur, le retournement horizontal et vertical en utilisant les fonctions de pré-entraînement de la bibliothèque Keras (ImageDataGenerator).

Cependant pour évaluer la robustesse des différentes méthodes d'augmentation des images sur notre modèle, nous allons dans un premier temps effectuer une classification sans l'intervention de ces techniques, et dans un second temps nous ferons appel à la transformation de données pour évaluer son influence sur notre modèle.

4.5 Architecture du premier modèle sans augmentation de données

Notre réseau reçoit en entrée des images de tailles $100 * 100$ pixels. Notre architecture comprend *trois* (3) couches de convolutions (*CONV2D*) qui génèrent des *features map* respectivement au nombre de 8, 16 et 32 grâce à des noyaux de convolution de tailles $5 * 5$ pixels. Chacune des couches de convolution est suivie d'une couche de pooling (*MaxPooling2D*) réduisant les tailles des plans de convolution reçus tout en préservant au maximum les caractéristiques importantes. Nous choisissons de faire progresser nos couches de pooling d'un pas de 2 pixels horizontalement et verticalement. Vu le type de nos différentes images, il ne serait pas intéressant de progresser sur un pas de 3 pixels au risque de supprimer d'importantes caractéristiques.

Avec nos images MSTAR, on s'est abstenu d'utiliser l'opération du *zero padding* du fait que les meilleurs pixels sont localisés au centre des images et donc cela n'aurait pas d'importance dans notre modèle. Après une mise à plat de nos pixels grâce à un *flatten layer*, les informations sont maintenant transmises aux dernières couches qui sont les couches denses ou *fully – connected*. Trois (3) couches *denses* ont été utilisées et contiennent respectivement 128, 84 et 10 neurones.

La couche dense finale de 10 neurones est notre couche de sorties et les 10 représentent nos 10 classes. A la différence des deux premières *fully – connected layers* (les couches cachées) qui utilisent des modules de corrections *ReLU*, la dernière couche utilise le module *softmax* qui permettra de donner la probabilité d'appartenance d'une image aux classes existantes.

```
inputs = Input(shape = (100, 100, 3))
x = Conv2D(8, kernel_size = (5, 5), activation = 'relu')(inputs)
x = MaxPooling2D(pool_size = (2, 2))(x)
x = Conv2D(16, kernel_size = (5, 5), activation = 'relu')(x)
x = MaxPooling2D(pool_size = (2, 2))(x)
x = Conv2D(32, kernel_size = (5, 5), activation = 'relu')(x)
x = MaxPooling2D(pool_size = (2, 2))(x)
x = Flatten()(x)
x = Dense(120, activation = 'relu')(x)
x = Dense(84, activation = 'relu')(x)
x = Dense(10, activation = 'softmax')(x)
cnn_model = Model(inputs = inputs, outputs = x)
```

4.5.1 Résumé du modèle

La (figure 4.4) nous permet de jeter un coup d'œil sur notre modèle et les différentes formes de sorties.

```
! cnn_model.summary()
Model: "model_10"
```

Layer (type)	Output Shape	Param #
input_11 (InputLayer)	[(None, 100, 100, 3)]	0
conv2d_30 (Conv2D)	(None, 96, 96, 8)	608
max_pooling2d_30 (MaxPooling)	(None, 48, 48, 8)	0
conv2d_31 (Conv2D)	(None, 44, 44, 16)	3216
max_pooling2d_31 (MaxPooling)	(None, 22, 22, 16)	0
conv2d_32 (Conv2D)	(None, 18, 18, 32)	12832
max_pooling2d_32 (MaxPooling)	(None, 9, 9, 32)	0
flatten_10 (Flatten)	(None, 2592)	0
dense_30 (Dense)	(None, 120)	311160
dense_31 (Dense)	(None, 84)	10164
dense_32 (Dense)	(None, 10)	850

```

Total params: 338,830
Trainable params: 338,830
Non-trainable params: 0
```

FIGURE 4.4 – Résumé du modèle

4.5.2 Entraînement du modèle

On peut résumer les paramètres de l'entraînement de notre modèle ainsi : le modèle s'entraîne 50 fois sur l'ensemble de données. Chaque fois qu'il s'entraîne, les données sont divisées en lots sur lesquels il se focalise un à un.

```

Epoch 1/50
400/400 [=====] - 27s 65ms/step - loss: 1.7064 - accuracy: 0.3493 - val_loss: 1.2199 - val_accu
acy: 0.5984
Epoch 2/50
400/400 [=====] - 20s 51ms/step - loss: 1.0468 - accuracy: 0.5942 - val_loss: 0.7453 - val_accu
acy: 0.7219
Epoch 3/50
400/400 [=====] - 21s 53ms/step - loss: 0.7536 - accuracy: 0.7026 - val_loss: 0.3768 - val_accu
acy: 0.8953
Epoch 4/50
400/400 [=====] - 23s 56ms/step - loss: 0.5775 - accuracy: 0.7774 - val_loss: 0.3391 - val_accu
acy: 0.8625
Epoch 5/50
400/400 [=====] - 23s 56ms/step - loss: 0.4596 - accuracy: 0.8277 - val_loss: 0.1797 - val_accu
acy: 0.9609

```

FIGURE 4.5 – Entrainement du modèle 1

4.5.3 Visualisation des résultats de l'entraînement

Ici nous montrons les tracés de perte et de précision sur les ensembles d'entraînement et validation en fonction du nombre d'époque.

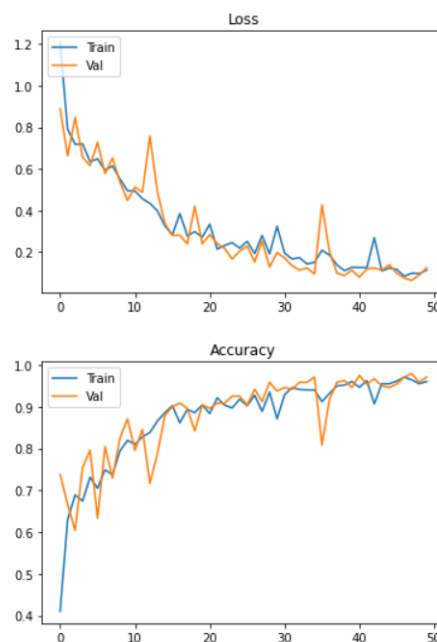


FIGURE 4.6 – Taux de précision(accuracy) et de perte(loss) du model 1 sur les données d'entraînement(train) et de validation (val)

Les graphiques nous montrent que la précision de la validation et de la formation sont très proches et le modèle a atteint une précision de plus de 90%.

Ici, malgré ce taux de précision intéressant on remarque une instabilité au niveau des deux courbes. Il faut savoir que la courbe de la formation a tendance à joindre des points bien spécifiques, d'où une apparence en dent de scie. Il est clair à ce effet, qu'il a trop bien appris les caractéristiques de nos images et donc les maîtrise parfaitement.

Ce aspect se présente de même sur notre jeux de validation et sous entend qu'elle n'est pas en effet représentative en terme de nombre. La preuve lorsqu'on essaie sur un jeux de test que nous avons à notre disposition, nous constatons que la différence de précision

entre la precision de l'entrainement et celle du test est bien perceptible : un signe de surapprentissage sur le jeux de formation.

```
34/34 [=====] - 0s 12ms/step - loss: 1.0904 - accuracy: 0.6038
[1.0903894901275635, 0.6037735939025879]
```

FIGURE 4.7 – Evaluation de notre modele sur un jeux de test

En effet sur 34 images, nous avons une énorme perte de 1.090 et une precision de 0.60. Cela n'est pas du tout de bons résultats. Il y'a trop de perte d'informations.

Voyons si on obtient des résultats acceptables avec une augmentation de données.

4.6 Formation du modèle après une augmentation par transformation des images

4.6.1 Augmentation des données

Le surajustement se produit généralement lorsqu'il y'a un petit nombre d'exemples de formation. L'augmentation des données consiste à générer des données d'entrainement à partir des exemples existants en les augmentant à l'aide de transformations aléatoires qui produisent des images d'apparence crédible. Cela permet d'exposer le modèle à d'avantage d'aspects des données et de mieux généraliser.

Nous implémentons l'augmentation des données à l'aide des couches de prétraitement de keras par exemple on a : *tf.keras.layers.randomFlip* qui retourne les images horizontalement ou verticalement, *tf.keras.layers.RandomRotation* qui permet une rotation, *tf.keras.layers.Zoom* etc.

Visualisons quelques exemples sur un élément de notre ensemble de données MSTAR.

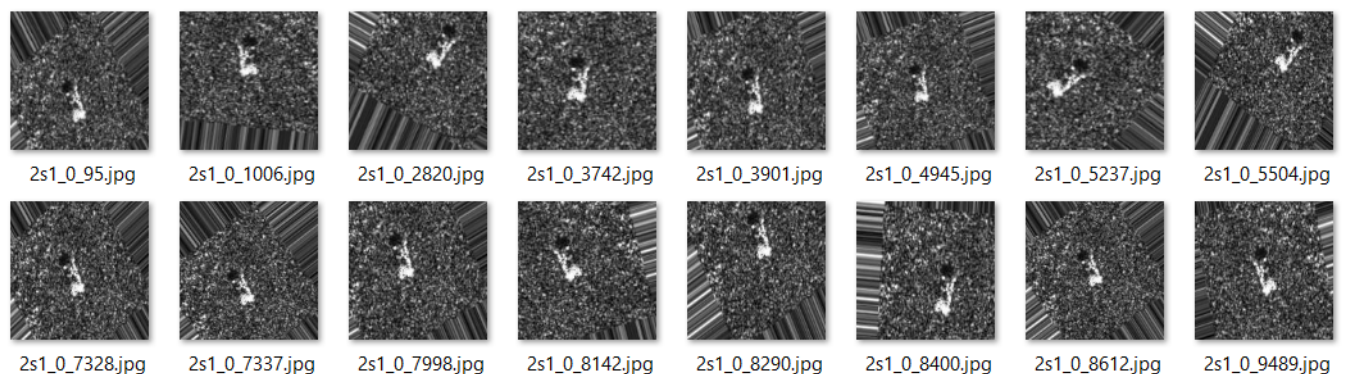


FIGURE 4.8 – Image du vehicule type 2S1 sous des transformations

4.6.2 Entraînement

Avant d'entraîner notre modèle sur l'ensemble de données augmenté, nous allons mettre toutes les chances de notre côté en introduisant dans notre architecture une couche "d'abandon" ou *Dropout* qui sera chargée de supprimer de manière aléatoire un certain nombre d'unités de sortie pendant le processus d'apprentissage.

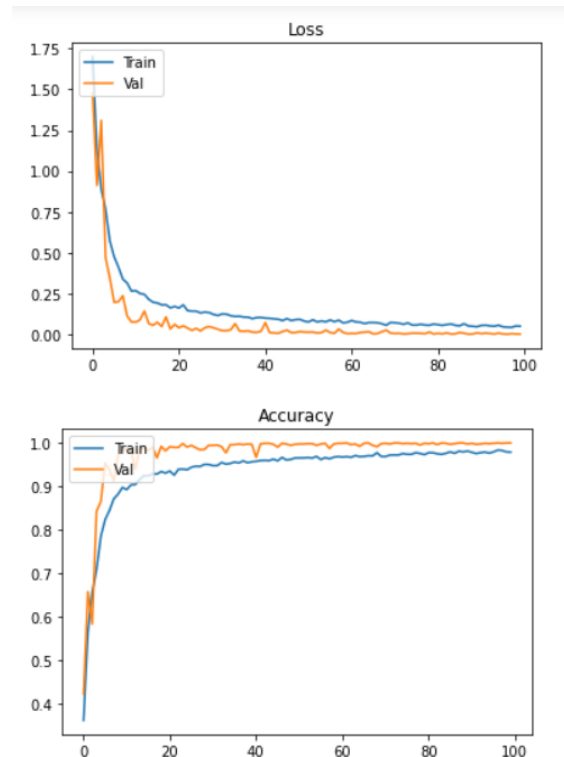


FIGURE 4.9 – courbes de précisions et de pertes sur les données d'entraînement et de validation

4.6.3 Évaluation du modèle

Après avoir appliqué l'augmentation des données et le Dropout, il y'a moins de surajustement qu'auparavant. En effet, nous constatons que la courbe de précision des données d'entraînement est plus régulière par rapport à l'expérience précédent (figure 4.9). Les graphes présentent de même une très bonne précision de 95% sur les données d'entraînement et plus de 97% sur les données de validation.

Utilisons ce modèle pour classer des images qui n'étaient pas incluses dans l'ensemble d'apprentissage ou de validation pour voir le comportement de notre modèle.

```
cnn_model.evaluate(test_generator)

Found 640 images belonging to 10 classes.
40/40 [=====] - 3s 85ms/step - loss: 0.0031 - accuracy: 1.0000

[0.0031379014253616333, 1.0]
```

FIGURE 4.10 – Prédiction sur de nouvelles données

Avec une précision parfaite de 100% et une perte de seulement 0.3% sur les données de test il est bien évident que notre modèle est bien généralisable.

4.7 Conclusion

Nous avons présenté dans ce chapitre l'implémentation d'un modèle de classification des images radars sur la base des réseaux de neurones convolutifs profonds, nous avons détaillé les étapes d'implémentation du système modèle ainsi que tous les tests effectués. Nous avons vu que le nombre de données a un réel impact sur la classification mais que des solutions très efficaces restent disponibles.

Le processus de classification des images radars est bien simple. C'est la même procédure pour classer des images optiques à la différence que celle destinées aux images radars nécessite vivement une technique d'augmentation de données.

Conclusion générale

En récapitulatif, nous avons vu que les images radars restent en générale indiscernables, incohérentes à la compréhension humaine. Elles échappent aux sens tellement elles sont complexes. Les machines quant à eux n'ont pas l'œil humaine. Une forme pour nous, n'est qu'un ensemble de pixels pour la machine. Les réseaux de neurones profonds tombent à point pour nous aider à la reconnaissance de ces types d'images qui peuvent être de source spatiale ou terrestre, aidant dans la surveillance, la sécurisation et la prévision. Il faut dire que nous ne pouvons plus nous en passer car le radar perce là où l'optique se limite.

Les réseaux de neurones convolutifs profonds donnent ainsi de bons résultats en terme de reconnaissance d'images radars. Cependant nous rencontrons un problème qui n'est pas prêt à se résoudre si tôt ; notamment le manque important d'images radars libres. Il est alors nécessaire quand on se retrouve face à un problème de classification d'images radars par les DCNN, d'impliquer une méthode d'augmentation de données.

Il existe heureusement plusieurs approches d'augmentation de données à savoir la transformation des images dont nous avons fait fit dans ce document, ainsi que des méthodes de régularisation qui sont très faciles à intégrer dans les modèles neuronaux.

En perspective, nous envisageons creuser le domaine des GANs qui peut s'avérer plus prometteur que les autres techniques. Ce sont des modèles très difficiles à implémenter. Il est important de pouvoir installer un bon équilibre entre le générateur et le discriminateur. Le but avec ce dernier sera de l'étudier et construire un simulateur de données radars fonctionnel et très performant.

Bibliographie

- [1] Antoine d'Acremont. Réseaux de neurones profonds pour la classification d'objets en imagerie infrarouge : apports de l'apprentissage à partir de données synthétiques et de la détection d'anomalies. Réseau de neurones [cs.NE]. ENSTA Bretagne - École nationale supérieure de techniques avancées Bretagne, 2020. Français. NNT : 2020ENTA0006. tel-03370137.
- [2] A. Krizhevsky, I.Sutskever, et G. E Hinton, " Imagenet classification with deep convolutional neural networks", dans Advances in neural information processing systems, pp. 1097-1105,2012.
- [3] Aurélien Hamouti, "Analyse d'images et reconnaissance d'objets pour les besoins de la Police cantonale". Présenté à Genève, le 24 février 2020 à Haute École de Gestion de Genève (HEG-GE) en vue de l'obtention du Bachelor HES.
- [4] Ayoub Karine. Traitement et classification parcimonieuse des images radar pour l'aide à la reconnaissance de cibles. Traitement du signal et de l'image [eess.SP]. ENSTA Bretagne - École nationale supérieure de techniques avancées Bretagne ; Université Mohammed V (Rabat). Faculté des sciences, 2018. Français. NNT : 2018ENTA0013. ?tel-02096884v2.
- [5] Cariust : L'histoire de la reconnaissance aérienne, 22 janvier 2018.
- [6] C.-A. Deledalle, L. Denis, S. Tabti, and F. Tupin, "MuLoG, or how to apply Gaussian denoisers to multi-channel SAR speckle reduction ?" IEEE Transactions on Image Processing, vol. 26, no. 9, pp. 4389–4403, 2017.
- [7] Deep learning et agriculture, une Une étude de la Chaire AgroTIC - Novembre 2018.

-
- [8] Cristian Coman, Rene Thaens. A Deep Learning SAR Target Classification Experiment on MSTAR Dataset, NATO Communications and Information AgencyThe Hague, THE NETHERLANDS email :cristian.coman@ncia.nato.int email : rene.thaens@ncia.nato.int
- [9] Di Zhang et al 2018 IOP Conf. Ser. : Mater. Sci. Eng. 322 072001 .
- [10] El Moujahid Yazid et Oubalkas Khalil | L'apport du logiciel ERDAS IMAGINE dans le filtrage et la classification des images sentinel-1, institue Agronomique et veterinaire Hassan II.
- [11] E.R Keydel,S.W. Lee, et J.T.Moore, "MSTAR extended operating conditions : A tutorial" ,in Proc.of 3rd SPIE Conf.Algorithms SAR Imagerytext, 1996, vol. 2757,pp. 228242.
- [12] France inte : La quête du Graal des chercheurs en intelligence artificielle. France Inter [en ligne]. [Consulté le 02 juin 2022]. Disponible à l'adresse : <https://www.franceinter.fr/emissions/cest-deja-demain/cest-deja-demain-24-janvier-2018>
- [13] L'apprentissage profond avec Python, French edition copyright ©2020 by machine-learning.fr.
- [14] Gérald PETITJEAN : Introduction aux réseaux de neurones.
- [15] Haipeng Wang, Member, IEEE, Sizhe Chen, Student Member, IEEE, Feng Xu, Senior Member, IEEE and Ya-Qiu Jin, Fellow, IEEE, "APPLICATION OF DEEP-LEARNING ALGORITHMS TO MSTAR DATA", Key Laboratory for Information Science of Electromagnetic Waves (MoE), Fudan University, Shanghai 200433, China.
- [16] LesPuysDuWeb : Parlon histoire : La quête du Graal[en ligne]. [Consulté le 02 juin 2022]. Disponible à l'adresse : <https://youtu.be/8NJ4w5XipwE>.
- [17] Iterative Weighted Maximum Likelihood Denoising with Probabilistic Patch-Based Weights, Charles-Alban Deledalle, Student Member, IEEE, Loïc Denis and Florence Tupin, Senior Member, IEEE, 2009.

-
- [18] Lortic B." et Lortic M.C. : L'imagerie radar, ORSTOM, Département SUD, U.R. UF.... Chef de travaux, EHESS, CAMS.
- [19] Najeh Maher : Filtrage et analyse des images radar, mémoire, université Laval, Mars 1997.
- [20] ORGANISATION DES NATIONS UNIES L'alimentation et l'agriculture Rome, Principe d'imagerie radar, 1991.
- [21] Patrice Wira : Réseaux de neurones artificiels, architectures et applications, Université de Haute Alsace, Laboratoire MIPS (Modélisation, Intelligence, Processus, Systèmes), Avril 2009.
- [22] Radu Horaud and Olivier Monga : Vision par ordinateur, outils fondamentaux. HERMES, 14 rue Lantiez 75017 Paris, 1993.
- [23] Ryan Melzer, William M. Severa, Mark Plagge, and Craig M. Vineyard : "Exploring Characteristics of Neural Network Architecture Computation for Enabling SAR ATR", Sandia National Laboratories, 1515 Eubank SE, Albuquerque, NM, United States .
- [24] Sonia Tabti : Modélisation des images par patchs pour leur restauration et leur interprétation. Applications à l'imagerie SAR. Traitement du signal et de l'image [eess.SP]. Télécom ParisTech, 2016. Français.tel-01585676.
- [25] Sonia Tabti : Thèse modélisation des images par patchs pour leur restauration et leur interprétation Applications à l'imagerie SAR, présenté en Juin 2016.
- [26] Tăriță Laura-Ionela : Traitement des images RSO interférométriques à haute résolution spatiale pour la surveillance des barrages et digues par télédétection radar, GRENOBLE 2011.
- [27] Xiao-Jiao Mao, Chunhua Shen, Yu-Bin Yang : Image Restoration Using Convolutional Auto-encoders with Symmetric Skip Connections, Aout 2016.
- [28] Yves Blanchard and Jacques Darricau : Histoire du Radar, part I : La naissance du Radar, Article in Revue de l'Electricité et de l'Electronique · January 2003.