ROYAUME DU MAROC
Université Mohammed V - RABAT -
جامعة محمد الخامس – الرباط –
Faculté des sciences
كلية العلوم
ÉCOLE NATIONALE D'INGÉNIEURS DE BREST
CENTRE D'ETUDES DOCTORALES - SCIENCES ET TECHNOLOGIES

N° d'ordre : CT33

THESE

En vue de l'obtention du : ***DOCTORAT***

Structure de Recherche : Laboratoire de Recherche en Informatique et Télécommunications
Discipline : Sciences de l'ingénieur
Spécialité : Informatique et télécommunications

En cotutelle avec l'Ecole Nationale d'Ingénieurs de Brest, France
Présentée et soutenue à Brest, le 29/06/2021 par :

Abdelouahid BEN TAMOU

Reconnaissance d'espèces de poissons dans des images vidéo sous-marines

Abdel-Ouahab BOUDRAA	PU, Ecole Navale, Brest, France	Président
Fabrice MERIAUDEAU	PU, Université de Bourgogne - IUT Le Creusot, France	Rapporteur/ Examinateur
Eric MOREAU	PU, Université de Toulon - SeaTech, France	Rapporteur/ Examinateur
Mohammed SADGAL	PES, Université Cadi Ayyad - Faculté des Sciences Semlalia, Maroc	Rapporteur/ Examinateur
Kamal NASREDDINE	MDC, Ecole Nationale d'Ingénieurs de Brest, France	Co-encadrant de thèse
Lahoucine BALLIHI	PH, Université Mohammed V de Rabat - Faculté des sciences, Maroc	Co-encadrant de thèse
Abdesslam BENZINOU	MDC HDR, Ecole Nationale d'Ingénieurs de Brest, France	Directeur de thèse
Salma MOULINE	PES, Université Mohammed V de Rabat - Faculté des sciences, Maroc	Directrice de thèse

Année Universitaire : 2020/2021

✉ Faculté des Sciences, avenue Ibn Battouta, BP. 1014 RP, Rabat –Maroc
☎ 00212 (05) 37 77 18 76 🖷 00212(05) 37 77 42 61; http://www. fsr.um5.ac.ma

A celui qui a attendu avec patience
le fruit de ces efforts inlassables,
mais il nous a quittés avant de le savourer,
A mon cher père

Remerciements

Cette thèse a été réalisée en cotutelle entre l'Université Mohammed V de Rabat (UM5R) et l'Ecole Nationale d'Ingénieurs de Brest (ENIB).

Le travail de recherche présenté dans ce manuscrit a été effectué au laboratoire LRIT CNRST URAC 29, à la Faculté des Sciences de Rabat (FSR), tout d'abord sous la direction du Feu de **Mr. Driss ABOUTAJDINE**, Professeur de l'Enseignement Supérieur à la FSR, puis sous la co-direction de **Mme. Salma MOULINE**, Professeur de l'Enseignement Supérieur à la FSR, et le co-encadrement de **Mr. Lahoucine BALLIHI**, Professeur Habilité à la FSR, et au laboratoire Lab-STICC UMR CNRS 6285, sur le site de l'ENIB, sous la direction de **Mr. Abdesslam BENZINOU**, Maître de Conférences HDR à l'ENIB, et le co-encadrement de **Mr. Kamal NASREDDINE**, Maître de Conférences à l'ENIB.

Tout d'abord, je tiens à remercie **Mr. Driss ABOUTAJDINE**, Professeur de l'Enseignement Supérieur à la FSR, d'avoir accepté de diriger cette thèse. Je remercie également **Mme. Salma MOULINE**, Professeur de l'Enseignement Supérieur à la FSR, d'avoir accepté de continuer à co-diriger le travail de cette thèse. Je remercie énormément mon directeur de thèse **Mr. Abdesslam BENZINOU**, Maître de Conférences HDR à l'ENIB, de m'avoir donné l'opportunité de faire cette thèse en cotutelle. Je vous remercie Monsieur de m'avoir consacré votre temps et votre énergie pour encadrer et orienter mes recherches. Je vous suis aussi reconnaissant pour votre excellent accompagnement depuis le début jusqu'à la fin de cette thèse. Votre bienveillance et vos conseils précieux m'ont beaucoup aidé pour avancer dans mon travail, et bien au-delà.

Je tiens à remercie mon co-encadrant **Mr. Lahoucine BALLIHI**, Professeur Habilité à la FSR. Je vous remercie aussi de m'avoir encadré lors de mon stage de Master et de m'avoir proposé pour cette cotutelle de thèse. Je suis également reconnaissant à mon co-encadrant **Mr. Kamal NASREDDINE**, Maître de Conférence à l'ENIB, de m'avoir suivi, guidé et aidé tout au long de ma thèse. Je vous remercie pour votre disponibilité, votre rigueur et vos précieuses remarques.

Je remercie **Mr. Abdel-Ouahab BOUDRAA**, Professeur des Universités à l'Ecole Navale à Brest, de m'avoir fait l'honneur de présider le jury de ma thèse.

Je remercie vivement **Mr. Fabrice MERIAUDEAU**, Professeur des Universités à l'Université de Bourgogne, d'avoir accepté de rapporter et examiner ce mémoire de thèse. Mers remerciements vont également à **Mr. Eric MOREAU**, Professeur des Universités à l'Université de Toulon/Sea tech, d'avoir accepté de juger la qualité de mon travail en tant que rapporteur et examinateur.
Je remercie aussi **Mr. Mohammed SADGAL**, Professeur de l'Enseignement Supérieur

à l'Université Cadi Ayyad de Marrakech, d'avoir accepté d'être rapporteur et examinateur de cette thèse.

Je remercie chaleureusement mes collègues de laboratoire LRIT et Lab-STICC particulièrement Hamza, Soufiane, Yassine, Marwa, Jacquline, Dimitrios, Noor, Morann, Alex, Mohamed, Ramez, Naima et Safae pour leur amitié, leur soutien et pour tous ces moments passés ensemble. Mes remerciements vont également à l'ensemble des enseignants du département électronique de l'ENIB qui m'ont aidé tout au long de mon année d'ATER à l'ENIB.

Un grand merci à mes amis Hamza, Abdelilah et Reda pour votre présence à mes côtés au cours de ces dernières années. Je vous remercie pour tous ces moments passés ensemble durant nos études universitaires.

Finalement, il y a des personnes que je ne saurais jamais remercier assez. Mes chers grands-parents pour votre amour, vos prières et vos sacrifices. Un énorme merci pour mes parents qui ont toujours cru en moi et qui m'ont énormément soutenu. Un grand merci à vous mes frères et sœurs pour votre encouragement et votre soutien. Ma reconnaissance va aussi aux autres membres de ma famille sans oublier ma chère tante Rahma qui nous a quittés, je n'oublierai jamais ta gentillesse et ton sourire.

Résumé —

L'objectif de cette thèse est d'élaborer des méthodes permettant la reconnaissance automatique d'espèces de poissons dans des images vidéo sous-marines. Nous privilégions les approches modernes de l'apprentissage profond (deep learning), notamment les réseaux de neurones convolutifs (CNNs).

Nous proposons une approche robuste pour la détection de poissons dans des vidéos sous-marines. Cette approche consiste à combiner deux réseaux parallèles afin de fusionner les caractéristiques liées à l'apparence et au mouvement du poisson. Ensuite, nous développons des méthodes d'identification d'espèces de poissons basées sur l'apprentissage par transfert. Finalement, la classification d'espèces de poissons est posée dans un cadre de classification par apprentissage progressif, et ce, de deux manières différentes. D'une part, nous proposons une approche de classification hiérarchique basée sur la taxonomie des espèces, qui permet de classifier les poissons en famille puis en espèce. D'autre part, nous proposons un nouveau modèle basé sur le principe de l'apprentissage incrémental pour améliorer les performances sur les classes (espèces) difficiles à identifier. Au début le modèle se focalise à bien apprendre les espèces difficiles, puis apprend progressivement les autres espèces avec une bonne stabilité.

Mots clés : vidéo sous-marine, détection de poisson, classification d'espèce de poisson, apprentissage profond, réseaux de neurones convolutifs.

Abstract —

The objective of this thesis is to develop tools and methods for automatic recognition of fish species in underwater video images. We focus on modern deep learning approaches, in particular convolutional neural networks (CNNs).

We propose a robust approach for the detection of fish in underwater video images. This approach consists in combining two parallel networks in order to fuse the features related to the appearance and the movement of the fish. Next, we develop methods for fish species identification based on transfer learning. Finally, fish species classification is posed in a progressive learning classification framework in two different ways. On the one hand, we propose a hierarchical classification approach based on species taxonomy, which allows to classify fishes into families and then into species. On the other hand, we propose a new model based on the principle of incremental learning to improve the performance on the classes (species) difficult to identify. At the beginning, the model focuses on learning the difficult species well, and then gradually learns the other species with a good stability.

Keywords : underwater video, fish detection, fish species classification, deep learning, convolutional neural networks.

Table des matières

Table des figures

Liste des tableaux

Table des sigles et acronymes

ACP	*Analyse en Composantes Principales*
ANN	*Artificial Neural Network*
AP	*Average Precision*
BRUV	*Baited Remote Underwater Video*
CAE	*Contractive AutoEncoder*
CNN	*Convolutional Neural Networks*
DAE	*Denoising AutoEncoder*
DBM	*Deep Boltzmann Machine*
DBN	*Deep Belief Networks*
DEM	*Deep Energy Models*
DNN	*Deep Neural Network*
DOV	*Diver-Operated Video*
ELM	*Extreme Learning Machine*
FCN	*Fully Convolutionnal Network*
HOG	*Histogram of Oriented Gradients*
ILSVRC	*Imagenet Large Scale Visual Recognition Challenge*
IoU	*Intersection over Union*
mAP	*mean Average Precision*
MSE	*Mean Squared Error*
NMS	*non-maximum suppression*
PM	*Précision Moyenne*
PMC	*Perceptron Multicouches*
R-CNN	*Regions with CNN features*
RBM	*Restricted Boltzmann Machine*
ReLU	*Rectified Linear Units*
RGB	*Red, Green, Blue*
RoI	*Region of Interest*
RPN	*Region Proposal Network*
RUV	*Remote Underwater Video*

SAE	*Sparse AutoEncoder*
SSD	*Single Shot Detector*
SURF	*Speeded Up Robust Features*
SVM	*Support Vector Machine*
TOWV	*TOWed Video*
UVC	*Underwater Visual Census*
YOLO	*You Only Look Once*

Introduction générale

Contexte et motivations

L'océan couvre environ 71% de la surface de la Terre avec un volume total de 1,37 milliard de kilomètres cubes et une profondeur moyenne de l'ordre de 3700 à 3800 mètres. Il abrite la majorité des espèces vivantes sur notre planète grâce à ses trois dimensions, en particulier la profondeur qui joue un rôle très important dans la répartition des espèces. L'océan contribue aux énergies renouvelables (énergie marémotrice) (Melikoglu 2018) et non renouvelables (gisements de gaz et de pétrole) (Wilberforce et al. 2019). C'est aussi une réserve importante de métaux (fer, nickel, cobalt, manganèse, or, cuivre, platine, argent, etc.) (Olafsdottir, Sverdrup et Ragnarsdottir 2017 ; Scott 2011). Il est également considéré comme un réservoir naturel des ressources en nourriture, en particulier halieutiques. En plus, l'océan est un espace économique, il sert à transporter des marchandises (Virdin et al. 2021). Aussi, il est considéré comme un milieu culturel d'écotourisme et de loisirs (pêche récréative, baignades, sports nautiques) (Picard 2015 ; Thys et al. 2016).

Actuellement, 240 000 espèces marines sont découvertes [1] dont environ 20 000 espèces de poissons, mais le nombre d'espèces qui existent dans l'océan est beaucoup plus important. Le poisson est l'une des ressources importantes pour l'homme, en particulier comme nourriture. Les poissons sont pêchés ou élevés dans des étangs ou dans des cages dans l'océan par des pêcheurs commerciaux, ou exposés dans des aquariums grand public.

Les progrès réalisés dans l'imagerie optique sous-marine conduisent de plus en plus à l'utilisation de ces systèmes dans des applications de surveillance, d'observation et/ou d'exploration. Par exemple, l'Ifremer a développé depuis 2007 des stations vidéo sous-marines dont 1500 ont pu être déployées dans les lagons de Nouvelle-Calédonie pour l'observation des habitats et des peuplements sous-marins dans des zones non protégées ou dans des réserves marines. Depuis 2010, quelques centaines de stations vidéo ont été mises en place dans le parc marin de la Côte Bleue. D'autres suivis vidéo ont récemment eu lieu dans

1. https://wwz.ifremer.fr/Expertise/Eau-Biodiversite/Biodiversite-Marine

la plupart des autres aires marines protégées (AMP) françaises. Ces nombreux espaces [2] doivent faire l'objet d'un suivi régulier des espèces pour fournir des indicateurs sur la dynamique spatiotemporelle des ressources et sur l'état de santé de la biodiversité côtière. Ces indicateurs servent d'éléments de pilotage aux politiques de gestion et de protection.

La vidéo sous-marine dispose d'atouts notables comme une haute résolution, une facilité d'interprétation et surtout une forte miniaturisation à faible coût. Malgré ces développements, le traitement automatique des enregistrements vidéo est encore très rare du fait de la complexité de l'information sous-marine (Hou et al. 2018). C'est pourtant une technologie pleine de promesses, en particulier pour le déploiement des robots et des observatoires sous-marins, dans un contexte de suivi pérenne des écosystèmes côtiers. Localiser, recenser et exploiter les populations marines nécessitent de recueillir en continu de la connaissance sur le milieu marin. La vidéo permet de surveiller les communautés aquatiques de l'écosystème sans en perturber le fonctionnement. C'est la raison essentielle pour laquelle cette technique est maintenant préférée aux techniques traditionnelles d'observation et de comptage en plongée sous-marine. En outre, la surveillance vidéo permet de réaliser en peu de temps un grand nombre d'observations réutilisables par la suite. Le maillon manquant aujourd'hui est cet outil automatique qui facilite l'analyse des images collectées.

Les données collectées par ces vidéos sous-marines peuvent être utilisées dans de nombreuses applications. Nous nous intéressons dans cette thèse à la reconnaissance d'espèces de poissons dans le milieu marin naturel. Cette application a été étudiée pour promouvoir des applications commerciales et environnementales telles que la pisciculture, la surveillance météorologique et la surveillance des quotas de pêche. Elle aide à comprendre l'écosystème marin, ce qui est vital pour étudier les problèmes qui affectent le milieu marin, tels que la pollution (Johannes 1975), la surpêche (Robinson et al. 2017), le braconnage (Roberts 1995) et le changement climatique (Daufresne et Boet 2007).

Les techniques de la vision par ordinateur et de l'apprentissage automatique peuvent aider les biologistes à observer les écosystèmes marins où l'annotation manuelle est trop coûteuse. Elles peuvent également les aider à faire des interprétations de haut niveau, comme le comptage des poissons, la distribution des espèces et l'étude des comportements des poissons. Les scientifiques marins peuvent bénéficier de ces analyses automatiques sans avoir besoin de compétences de programmation spécialisées. Malgré ces avantages, les tra-

2. Onze millions de kilomètres carré de mers sous souveraineté française ; les aires marines protégées représentaient 20% des eaux françaises en 2020.

FIGURE 1 – Exemples d'images sous-marines issues de différentes vidéos de l'ensemble de référence "*LifeClef 2015 Fish*". Ces images illustrent la forte variabilité naturelle dans un environnement marin non contraint. Nous observons entre autres, des arrière-plans complexes, la présence de plusieurs poissons simultanément, la dynamique restreinte et la variation de luminosité.

vaux réalisés dans ce domaine restent très peu du fait de la complexité de l'environnement marin. Cet environnement est considéré un défi pour la vision par ordinateur à cause de plusieurs facteurs, nous en citons principalement le changement fréquent de la luminosité, la limitation de la visibilité, la complexité du fond marin (coraux, algues, mouvement des plantes aquatiques, ...), et la diversité des espèces. Dans cet environnement, le poisson se déplace librement dans toutes les directions, il peut aussi être occulté partiellement par l'habitat ou par d'autres poissons et être confondu avec d'autres espèces à cause de la similitude en forme et en texture. La figure 1 illustre des exemples d'environnement marin naturel avec différents fonds marins plus au moins complexes.

L'apprentissage profond[3] est une approche d'apprentissage automatique qui est largement appliquée dans différentes tâches de la vision par ordinateur. Cet outil connait un grand succès grâce à ses résultats impressionnants (GOODFELLOW et al. 2016). Malgré son efficacité dans l'analyse automatique d'images, très peu de travaux l'ont utilisé pour les images sous-marines à cause des défis du milieu marin cités précédemment.

Dans cette thèse, nous nous intéressons aux images issues de caméras sous-marines posées dans des environnements marins naturels. Les images obtenues sont souvent de mauvaise qualité avec des arrière-plans très complexes. L'objectif essentiel de cette thèse est

3. *Deep learning* dans la littérature anglo-saxonne.

la mise au point d'une chaîne de traitement et d'analyse d'images vidéo sous-marines pour la reconnaissance automatique d'espèces de poissons. Cet outil est destiné à des aquariums et à des stations d'observation pour le suivi des aires marines protégées (AMP) et des dispositifs de concentration de poissons (DCP). Une autre application possible concerne les passes à poissons gérées par l'ONEMA [4]. Dans ce travail de thèse nous voulons en particulier tester l'apport de l'apprentissage profond pour la détection des poissons et la classification de leurs espèces. Cette thèse a bénéficié du soutien financier de la Région Bretagne (dispositif ARED).

Contributions de la thèse

Le travail principal de cette thèse sera donc de détecter et d'identifier l'espèce de poisson (vivant) dans une image sous-marine en utilisant l'apprentissage profond. Les contributions se résument dans ce qui suit :

— Tout d'abord, nous proposons une approche pour la détection de poissons dans des images vidéo sous-marines. Cette approche est basée sur la fusion de deux réseaux profonds en parallèles. Un premier réseau extrait les caractéristiques d'apparence de chaque image vidéo couleur. Ces caractéristiques peuvent être de type texture, forme et couleur. Tandis que l'autre réseau extrait les caractéristiques de mouvement à partir des images successives. Les caractéristiques de mouvement peuvent être très pertinentes. Le poisson apparaît dans plusieurs images d'une vidéo, et peut changer de direction et de posture en nageant, ce qui a également un impact sur la représentation des caractéristiques. Nous exploitons cette information temporelle en plus de l'information de l'apparence pour améliorer les performances de la détection.

— Ensuite, nous abordons le problème de classification d'espèces de poissons. L'apprentissage profond requiert des bases de données de grande taille pour une meilleure performance. Toutefois, les bases d'images de poissons disponibles sont de petite taille. Pour surmonter ce problème, nous utilisons l'approche d'apprentissage par transfert dans différentes stratégies tout en abordant diverses problématiques (choix de l'espace colorimétrique, élimination ou non d'arrière-plan, et manière d'augmentation

4. Office National de l'Eau et des Milieux Aquatiques.

artificielle de données). Pour l'augmentation artificielle de données, nous proposons d'augmenter le nombre d'images en utilisant un nouveau critère basé sur les courbes de perte durant l'apprentissage et durant la validation.

— Afin d'améliorer les performances en classification, nous proposons deux approches basées sur un apprentissage progressif. Cet apprentissage permet à un système d'apprendre sur des sous-ensembles de données. Nous proposons d'abord la classification hiérarchique basée sur la classification taxonomique. Elle permet de classer les poissons dans un taxon plus commun (par exemple en famille), puis dans un taxon plus spécifique (espèce). Nous proposons également un système qui intègre des classes d'espèces de manière incrémentale. Nous partons d'un sous-ensemble qui contient un nombre limité d'espèces de poissons. Nous proposons de commencer par les espèces les plus difficiles. Ensuite, le système apprendra progressivement pour atteindre de meilleures performances tout en gardant le système stable lors de l'introduction d'une nouvelle espèce.

— Les approches proposées dans cette thèse sont évaluées sur deux bases sous-marines de référence : la base d'images "Fish Recognition Ground-Truth" [5] et la base de vidéos "LifeClef 2015 Fish" [6]. Les deux bases contiennent des images de poissons de différentes couleurs, textures, positions, tailles et orientations. En détection de poissons, nous avons obtenu une F-mesure de 83,16% et une mAP de 73,69% sur la base de vidéos "LifeClef 2015 Fish". Nous avons également pu obtenir des taux de classification de 99,84% et de 81,31% sur les bases "Fish Recognition Ground-Truth" et "LifeClef 2015 Fish" respectivement. Ces résultats très prometteurs surpassent ceux de l'état de l'art.

Organisation du document

Ce document est organisé principalement en deux grandes parties.

— La première partie, "Généralités", vise à présenter le cadre général du travail proposé dans cette thèse. Dans le chapitre 1, nous faisons un état de l'art sur les techniques d'observation sous-marine et sur la reconnaissance d'espèces de poissons. Le chapitre

5. `https://groups.inf.ed.ac.uk/f4k/GROUNDTRUTH/RECOG/`
6. `www.imageclef.org/lifeclef/2015/fish`

2 a pour objectif de donner au lecteur un état de l'art sur les réseaux de neurones et l'apprentissage profond, en particulier les réseaux de neurones convolutifs.

— La deuxième partie, "Méthodologie et validation" présente le travail réalisé dans cette thèse. Dans le chapitre 3 intitulé "Détection de poissons dans des images vidéo sous-marines par fusion de réseaux CNN parallèles", nous proposons une nouvelle approche de fusion hybride pour la détection de poissons dans des vidéos sous-marines. Cette approche est comparée à l'état de l'art des approches employées. Nous proposons ensuite dans le chapitre 4, "Classification d'espèces de poissons dans des images vidéo sous-marines", d'utiliser l'apprentissage par transfert pour extraire des caractéristiques et/ou ré-entraîner plus finement un système préalablement entraîné sur une autre base de données. Nous proposons dans le même chapitre un nouveau critère pour augmenter artificiellement le nombre d'images d'un ensemble d'entraînement réduit. Le chapitre 5, "Apprentissage progressif pour la classification d'espèces de poissons", vise à améliorer les performances en classification en faisant apprendre un système de manière hiérarchique ou incrémentale.

— Dans la conclusion, nous récapitulons les réalisations effectuées par ce travail de thèse. Les nouveautés apportées aux applications biologiques et à la vision par ordinateur sont exposées avec les principaux résultats obtenus. Enfin, nous évoquerons les perspectives que nous proposons pour la présente étude.

Première partie

Généralités

Chapitre 1

Etat de l'art en reconnaissance d'espèces de poissons

Sommaire

1.1 Introduction

La reconnaissance automatique d'espèces de poissons devient de plus en plus un sujet majeur en vision par ordinateur. Ce chapitre est consacré à une présentation de l'état de l'art sur (1) les techniques d'observation de la biodiversité sous-marine (Mallet et Pelletier 2014), et (2) les approches de la vision par ordinateur proposées pour exploiter les données acquises en matière de détection de poissons et d'identification de leurs espèces (Yang et al. 2020).

Tout d'abord, nous commençons par souligner en section 1.2 les enjeux de l'environnement sous-marin, en particulier les récifs coralliens. Puis, nous présentons en section 1.3 les différentes techniques utilisées pour observer et suivre les écosystèmes marins. Nous décrivons ensuite dans la section 1.4 les travaux proposés dans la littérature pour la détection et la classification d'espèces de poissons dans des images sous-marines. En section 1.5, nous présentons les bases d'images de référence utilisées pour l'expérimentation des travaux de cette thèse ; travaux que nous discutons et positionnons par rapport à la littérature en section 1.6. Finalement, nous terminons ce chapitre par une conclusion en section 1.7.

1.2 Enjeux de l'environnement sous-marin

L'environnement sous-marin, en particulier les récifs coralliens, abrite des écosystème diversifiés et riche en biodiversité. Ces récifs sont composés d'assemblages de coraux, d'algues et d'éponges. Cette structure complexe offre un habitat idéal pour de nombreuses espèces notamment pour se protéger et se nourrir (Brandl et al. 2018). Pour cela, les récifs coralliens abritent entre 1 et 3 millions d'espèces et 25% de la totalité des espèces de poissons marins (Allsopp et al. 2008).

Les récifs coralliens protègent aussi efficacement les côtes de l'érosion ; les barrières de corail servent en quelque sorte de digues face aux grandes vagues océaniques (Harris et al. 2018). En plus de leur intérêt écologique, ils fournissent des services économiques. En effet, au moins 30 millions de personnes en dépendent directement sur les littoraux et dans les communautés insulaires (Rogers, Blanchard et Mumby 2018 ; Wilkinson 2004). Ils fournissent l'essentiel de la production en poissons et des sources de revenus et des moyens de subsistance (Hughes et al. 2003).

Alors que l'environnement sous-marin (en particulier les récifs coralliens) présente une importance écologique et économique, il est désormais menacé (Gordon et al. 2018) en particulier par la pollution (Johannes 1975), la surpêche (Robinson et al. 2017) et le changement climatique (Leggat et al. 2019). Ces facteurs détruisent l'écosystème et accélèrent la perte d'espèces de coraux et de poissons y vivent (D'agata et al. 2014). Il est désormais nécessaire de suivre l'évolution des ces écosystèmes en vue d'identifier, voire d'anticiper les possibles dégradations écosystèmiques menaçantes (Hughes et al. 2017). Ce suivi s'effectue par l'observation puis l'estimation de la diversité et de l'abondance des espèces de poissons pour comprendre la structure des communautés et la dynamique des récifs coralliens (Jackson et al. 2001). Les techniques traditionnelles d'observation des écosystèmes sont destructives et/ou n'assurent pas un suivi continu de la biodiversité sous-marine. Il est important d'adopter des techniques plus avancées, non destructives et qui assurent une continuité de suivi des écosystèmes.

1.3 Techniques d'observation sous-marine

En écologie marine, différentes techniques sont utilisées pour observer et analyser la biodiversité sous-marine. Ces techniques peuvent être regroupées en trois grandes catégories : les techniques par extraction (destructives), les techniques visuelles par plongée sous-marine, et les techniques visuelles par vidéo sous-marine.

1.3.1 Techniques par extraction

Elles consistent à extraire des échantillons du milieu sous-marin pour faire l'analyse. Ces techniques destructives ont été les premières techniques d'observation de la biodiversité sous-marine. Elles ont été utilisées principalement pour étudier les poissons, les organismes macro-benthiques et la faune.

La pêche est l'une des techniques la plus destructive, en particulier le chalutage de fond de mer (Engel et Kvitek 1998) (figure 1.1). Cette technique, basée sur l'utilisation d'énormes filets trainés par des navires, racle les fonds marins et détériore les habitats et les organismes du fond de mer (Jennings et al. 2001 ; Poiner et al. 1998 ; Watson, Revenga et Kura 2006). De plus, cette technique fournit des informations sur l'espèce

pêchée mais pas sur les autres espèces. Les techniques de pêche peuvent varier d'une espèce à l'autre en fonction des conditions météorologiques (Trenkel et Cotter 2009) et des navires (Pelletier 1991).

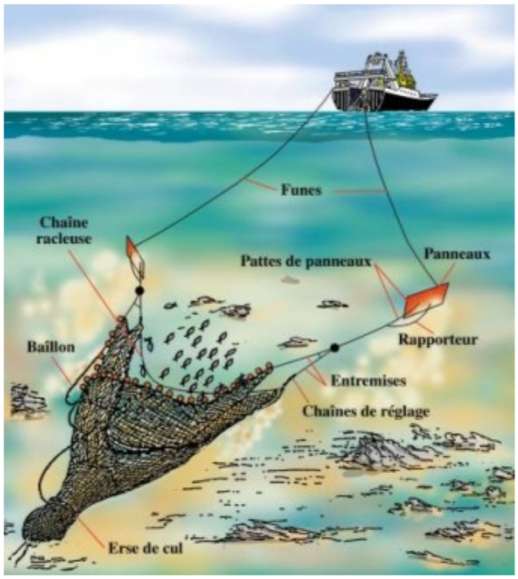


Figure 1.1 – Illustration d'un chalutage de fond de mer (Deschamps 2003). Le chalut de forme conique est remorqué par un navire en étant relié par des câbles en acier appelés funes. Des panneaux divergents situés en avant du chalut permettent son ouverture horizontale.

Une autre technique consiste à utiliser des produits létaux ou anesthésiants pour collecter les poissons (Fernandes et al. 2017 ; Priborsky et Velisek 2018). Cette technique appelée extraction par produit chimique (ou par empoisonnement) est moins destructive pour les habitats que la technique par pêche (Robertson et Smith-Vaniz 2008). Elle fournit de bons résultats en termes d'espèces observées à l'image de la technique visuelle par plongeur (Ackerman et Bellwood 2000 ; Dibble 1991). Cette technique ne sélectionne qu'une partie de l'assemblage de poissons (Robertson et Smith-Vaniz 2008). Elle est donc davantage utilisée pour les inventaires et les observations à petite échelle que pour le suivi.

Les techniques destructives ont un réel impact sur la biodiversité, ce qui n'est pas souhaitable dans le contexte du suivi des stratégies de conservation. De plus, ces techniques ne permettent pas d'étudier les comportements d'espèces marines et l'interaction avec leurs habitats.

1.3.2 Recensement visuel par plongée sous-marine

Dans les zones peu profondes, les techniques de comptage visuel sous-marin (UVC [1]) sont utilisées depuis plus de 70 ans pour observer, analyser et suivre les poissons, les organismes macro-benthiques et les habitats (Brock 1954). Ces techniques sont généralement effectuées par un ou plusieurs plongeurs et considérées comme fiables et rentables (Thresher et Gunn 1986). Durant la mission, les plongeurs dénombrent, identifient, estiment la taille des poissons et analysent leurs comportements. Les comptages peuvent être effectués de trois manières :

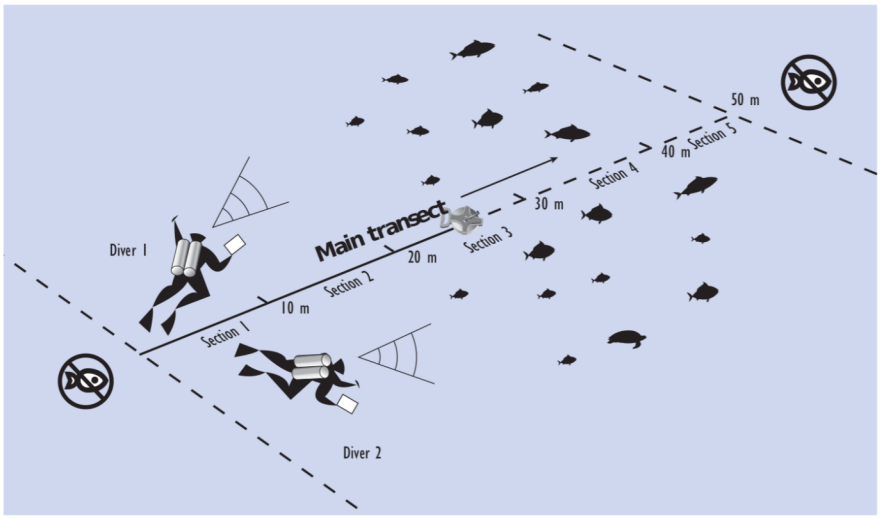


Figure 1.2 – Illustration du comptage visuel sur transect (Labrosse, Kulbicki et Ferraris 2002).

— **Parcours aléatoire** : dans cette méthode, une zone d'étude est définie et les plongeurs nagent librement dans cette zone (Jones et Thompson 1978). Cette méthode est particulièrement adaptée pour recenser les petites espèces librement approchées par les plongeurs.

— **Transect** : Un transect est une zone rectangulaire dont la longueur et la largeur sont clairement définies. Dans cette méthode (figure 1.2), les plongeurs se déplacent le long d'une ligne et observent les espèces à une certaine distance entre 3 et 5 mètres de la ligne (Brock 1954 ; Buckland et al. 2001 ; Burnham, Anderson et Laake 1980 ; David 2005). C'est l'une des méthodes les plus couramment utilisées

1. UVC : *Underwater Visual Census* dans la littérature anglo-saxonne.

et elle est bien adaptée aux études de population, en particulier celles évaluant les ressources halieutiques à des fins commerciales ou alimentaires (Bozec et al. 2011 ; Kulbicki et Sarramégna 1999).

— **Comptage statique** : un plongeur stationnaire observe des poissons face à lui ou au cours de rotation sur lui-même (Bohnsack et Bannerot 1986) (figure 1.3). Cette méthode est plus rapide que le transect (Facon et al. 2016) et fournit une meilleure approximation de la densité de poissons (Colvocoresses et Acosta 2007). Elle est particulièrement recommandée pour l'étude d'une espèce ou d'un petit groupe d'espèces, notamment dans des milieux très hétérogènes (Chateau et Wantiez 2005 ; Wantiez, Chateau et Le Mouellic 2006).

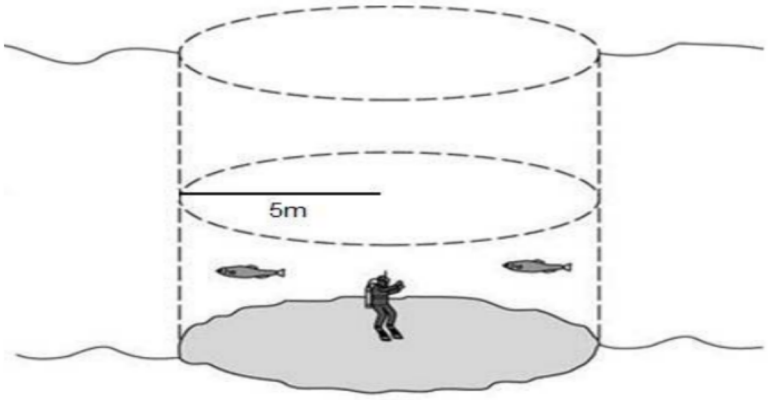


Figure 1.3 – Illustration du comptage visuel statique (Jennings, Kaiser et Reynolds 2001).

La technique UVC est la technique d'observation la plus couramment utilisée car elle est peu coûteuse et relativement rapide ; elle peut aussi détecter des espèces cachées et laisse intact le milieu naturel observé (Dickens et al. 2011 ; Kulbicki et al. 2010). Néanmoins, l'UVC se heurte à de nombreuses limitations, nous en citons notamment les suivantes :

— les caractéristiques environnementales telles que la visibilité ou la clarté de l'eau doivent être suffisantes pour l'analyse visuelle (Brock 1982 ; MacNeil et al. 2008a,b) ;
— la détectabilité des poissons est influencée par la complexité de l'habitat (Edgar et Barrett 1999), de l'abondance des espèces et de leurs caractéristiques telles que la taille, les apparences physiques et les comportements (Bernard et al. 2013 ; Bozec et al. 2011 ; Edgar, Barrett et Morton 2004 ; Kulbicki 1998 ; MacNeil et al. 2008a,b ; Willis 2001) ;

— l'état de l'océan et plus généralement les conditions météorologiques influencent les compagnes de comptage ;
— les observations sont influencées par la physiologie du plongeur, en particulier en cas de plongée en apnée. Les observateurs sont limités par leur incapacité à rester sous l'eau sans respirer. L'utilisation de l'équipement de plongée implique des limites de profondeur et de temps qui ne peuvent être dépassées sans mettre le plongeur en danger. Les plongeurs sont également limités par leur incapacité à résister au froid et à la fatigue ;
— la présence du plongeur perturbe le comportement des poissons en provoquant une réaction de fuite ou d'attraction (Chapman et al. 1974 ; Dickens et al. 2011).

1.3.3 Recensement visuel par vidéo sous-marine

Ces dernières années, les techniques de vidéo sous-marine ont été de plus en plus utilisées pour observer la macrofaune et l'habitat dans les écosystèmes marins. Les progrès technologiques concernant les caméras vidéo, l'autonomie de la batterie et le stockage de l'information rendent désormais ces techniques accessibles à la majorité des utilisateurs. Il existe différentes techniques de vidéo sous-marine, développées depuis les années 1950 pour suivre et étudier la biodiversité sous-marine, que l'on peut les classer en trois catégories (Mallet et Pelletier 2014) :

— la vidéo opérée par des plongeurs (dite méthode de transect vidéo) ;
— la vidéo tractée ;
— la vidéo à distance [2] (sans plongeur).

1.3.3.1 Vidéo opérée par des plongeurs

La technique vidéo opérée par des plongeurs (DOV [3]) ou méthode de transect vidéo consiste en un ou plusieurs plongeurs qui parcourent un transect matérialisé sur le fond marin en filmant devant et au-dessous d'eux (figure 1.4) (Cruz, Kikuchi et Leão 2008 ; Lam et al. 2006 ; Rogers et Miller 2001). Parfois, le plongeur est remorqué pour enregistrer l'habitat benthique le long de longs transects (Carleton et Done 1995 ; Kenyon

2. Le terme à distance utilisé ici désigne une technique qui n'exige pas la présence de plongeurs.
3. DOV : *Diver-Operated Video* en anglais.

et al. 2006 ; Vogt, Montebon et Alcala 1997). Les DOVs sont aussi influencées par la physiologie du plongeur, mais les observations se font à partir de données collectées sur ordinateur au bureau, plutôt que sur le terrain. Cela permet de filmer sur de longues distances tout en minimisant le temps de plongée par rapport aux comptages visuels (Pelletier et al. 2011).

Figure 1.4 – Exemple de la transect vidéo (vidéo opérée par un plongeur DOV) (Goetze et al. 2019).

Dans (Bortone, Martin et Bundrick 1994), les auteurs ont proposé un protocole dans lequel le plongeur tourne et enregistre des images simulant la technique du comptage statique (Bohnsack et Bannerot 1986). Les DOVs ont également été utilisées pour étudier le comportement des poissons par (Krohn et Boisdair 1994) et (Hall et Hanlon 2002).

1.3.3.2 Vidéo tractée

La technique de vidéo tractée (TOWV [4]) (Assis, Narvaez et Haroun 2007 ; Rende et al. 2015) consiste à remorquer à faible vitesse un système stable équipé d'une caméra. La figure 1.5 illustre un exemple d'un système de caméra utilisé en vidéo tractée. Cette

4. TOWV : *TOWed Video* en anglais.

technique filme le long d'un transect de taille et de trajectoire prédéfinies (30 m à 20 km) (Mallet et Pelletier 2014). Elle est destinée principalement à étudier les espèces marines et les épifaunes (Foveau, Haquin et Dauvin 2017 ; Sward, Monk et Barrett 2019). Elle est aussi utilisée pour caractériser, quantifier et évaluer les changements dans la flore benthique (herbiers, macro-algues, coraux, ...) et dans la faune (Schaner, Fox et Taraborelli 2009 ; Underwood et al. 2018).

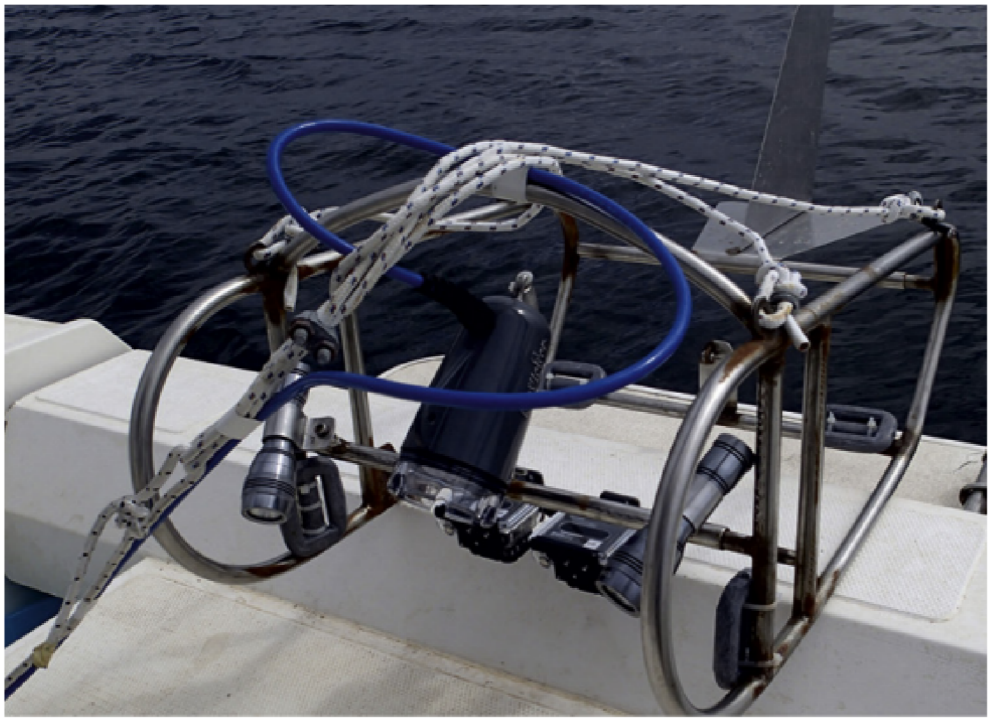


Figure 1.5 – Exemple d'un système de vidéo tractée (Rende et al. 2015).

1.3.3.3 Vidéo à distance

Les premiers travaux sur l'utilisation de la vidéo sous-marine à distance (RUV [5]) en l'environnement côtier remontent aux années 1950 (Barnes 1955). Classiquement, les caméras vidéo sont posées en plongée ou depuis la surface sur un substrat sous-marin (figure 1.6). Cette technique est plus fréquemment utilisée pour suivre le mouvement et le comportement des poissons sans perturbation humaine (Bortone, Martin et Bundrick 1991 ;

5. RUV : *Remote Underwater Video* en anglais.

LaFond 1968). Les systèmes RUVs présentent des conceptions diverses et des caractéristiques techniques différentes et peuvent avoir des capteurs supplémentaires. Ils peuvent être distingués en termes d'autonomie et de fonctionnement. D'un côté, nous trouvons des systèmes reliés qui utilisent des câbles pour l'énergie (Holt 1967 ; Tyne et al. 2010), pour le transfert de données (Aguzzi et al. 2011 ; Gibson, Atkinson et Gordon 2012) ou pour le contrôle des instruments (Kronengold et al. 1964 ; Kumpf et Lowenstein 1962). D'un autre côté, nous trouvons des systèmes autonomes qui ne sont reliés ni à un navire ni à une plate-forme (Chabanet et al. 2012 ; Pelletier et al. 2012).

(a) Système BRUV (Currey-Randall et al. 2020)

(b) Système STAVIRO (Pelletier et al. 2012)

Figure 1.6 – Exemples de systèmes vidéo sous-marine à distance (RUV).

Une variante des systèmes RUVs consiste à utiliser une caméra avec une source d'appât pour attirer les poissons (figure 1.6-(a)). On parle alors de vidéo sous-marine à distance appâtée (BRUV [6]) (Currey-Randall et al. 2020 ; Heagney et al. 2007). Les espèces attirées dépendent de l'appât utilisé (Harvey et al. 2007 ; Stobart et al. 2007 ; Wraith 2007). Les principales différences entre les systèmes BRUVs concernent l'orientation du système par rapport au fond de mer, en horizontal (EIIis et DeMartini 1995 ; Heagney et al. 2007) ou en vertical (Babcock et al. 1999 ; Willis et Babcock 2000) ; l'abondance et la composition des espèces observées en dépendent (Langlois et al. 2006 ; Wraith 2007). Les systèmes BRUVs ont également été utilisés avec la lumière infrarouge pour étudier les poissons nocturnes (Bassett et Montgomery 2011). Les études comparatives

6. BRUV : *Baited Remote Underwater Video.*

de (Lowry et al. 2012), (GHAZILOU, SHOKRI et GLADSTONE 2019) et (Colton et Swearer 2010) ont montré que les méthodes BRUVs et UVCs sont complémentaires. En effet, les UVCs permettent d'observer une plus grande diversité d'espèces alors que les BRUVs permettent d'attirer certaines espèces non observées en UVC.

Les systèmes RUVs sont souvent équipés d'une seule caméra fixe, mais ils peuvent être équipés d'une caméra rotative comme dans le système STAVIRO [7] (Pelletier et al. 2012) (figure 1.6-(b)). Ce système à 360^o effectue plusieurs rotations et fournit des images panoramiques et une zone beaucoup plus étendue qu'avec les systèmes fixes. Les systèmes RUVs peuvent aussi être équipés d'un système de stéréo-vision (Harvey et al. 2012a,b ; Langlois et al. 2012 ; Langlois, Harvey et Meeuwig 2012). La stéréo-vision utilise simultanément deux caméras pour enregistrer la même scène permettant de mesurer la distance et la taille des individus (Gibson, Atkinson et Gordon 2016 ; Harasti et al. 2017). Cette technique fournit des estimations plus précises de la distance et de la longueur des poissons que l'estimation visuelle des les plongeurs (Harvey, Fletcher et Shortis 2002 ; Harvey et al. 2004) ou les systèmes à une seule caméra (Harvey et al. 2002).

1.3.4 Synthèse comparative

Une étape cruciale dans le suivi de la biodiversité est le choix de la technique d'observation la plus appropriée. Ce choix est un compromis entre l'objet de l'étude, les moyens disponibles et la précision requise (Rotherham et al. 2007). Chaque technique a ses propres avantages et inconvénients reportés dans la table 1.1 (Mallet et Pelletier 2014). Le choix d'une technique plutôt qu'une autre se fait en fonction des caractéristiques générales et de performances avérées, mais les changements technologiques doivent également être pris en compte. Les coûts d'investissement et d'exploitation sont aussi deux paramètres cruciaux.

Les techniques les plus souvent utilisées pour observer et suivre la biodiversité sous-marine s'appuient sur les UVCs et la pêche (Mallet et Pelletier 2014). La vidéo sous-marine dispose toutefois d'atouts notables. Peu coûteuse en termes de coût et de temps, elle permet de réaliser un grand nombre d'observations réutilisables à tout moment. Elle permet aussi de surveiller les communautés aquatiques de l'écosystème sans en perturber

7. STAVIRO : *STAtion VIdéos Rotative*.

Technique	Méthode	Avantages	Inconvénients
Destructive	Pêche	- Extractif - Ne nécessite pas de plongeur - Observation possible à grande profondeur - Participation possible des pêcheurs	- Capturabilité (sélectif) - Destruction de communautés et des habitats. - Pas d'observation comportementale - Impossible à grande échelle
	Extraction chimique		
Visuelle	UVC mobile (transect)	- Non extractif - Largement utilisé - Participation possible de volontaires - Protocoles simplifiés	- Effet de plongeur - Limitation de durée d'observation et de profondeur. - Nécessite un plongeur formé à l'identification et au comptage d'espèces - L'état de l'océan et les conditions météorologiques.
	UVC fixe (comptage statique)		
Vidéo	DOV	- Non extractif - Ne nécessite pas de plongeur scientifique	- Présence du plongeur - Limitation de durée d'observation et de profondeur. -Limitation de nombre de plongées par jour. - Tous les effets associés à la présence d'un plongeur sous l'eau - Durée de l'analyse d'image
	TOWV	- Non extractif - Ne nécessite pas de plongeur - Possibilité de travailler en eau profonde - Mise en œuvre rapide - Large couverture spatiale - Participation éventuelle de personnel non scientifique	- Complexe en milieu récifale - Mobilité restreint - Peut perturber l'écosystème en raison du bruit des navires - Gestion de grands ensembles de données - Durée de l'analyse d'image
	RUV	- Non extractif - Méthode la moins invasive - Durée d'observation constante - Ne nécessite pas de plongeur - Observation possible à grande profondeur - Mise en œuvre rapide - Participation éventuelle de personnel non scientifique	- Champs de vision restreint et immobile. - Durée de l'analyse d'image - Gestion de grands ensembles de données
	BRUV	- Non extractif - Augmentation de l'abondance des poissons observée grâce à l'appât - Durée d'observation constante - Ne nécessite pas de plongeur - Possibilité de travailler en eau profonde - Participation possible de personnel non scientifique	- Modification du comportement - Modification de la composition - Effet inconnu du panache d'appât - Durée d'observation relativement longue - Durée de l'analyse d'image - Gestion de grands ensembles de données

TABLE 1.1 – Comparaison des principaux avantages et inconvénients des techniques d'observation de l'environnement sous-marin (MALLET et PELLETIER 2014).

le fonctionnement. C'est un avantage majeur par rapport aux UVCs qui nécessitent d'avoir un plongeur dans le milieu. En outre, cette technique est préférable pour suivre avec une couverture spatiale suffisante, et une large gamme de profondeurs, des zones vastes comme les aires marines protégées (AMPs), les parcs marins ou le patrimoine mondial. Sa mise en œuvre est également facile et les vidéos enregistrées peuvent être utilisées par des non-spécialistes.

Cependant, les techniques actuelles d'observation par vidéo sous-marine nécessitent des experts humains pour analyser les données collectées dont la quantité augmente rapidement. Il se pose alors le problème d'analyser de façon efficiente et objective ces grandes quantités de données. L'analyse automatisée des images collectées, objet de la présente thèse, conduirait à détecter et à classifier automatiquement les espèces de poissons dans les images vidéo sous-marines. La section suivante présente les travaux de l'état de l'art concernant la détection et la classification d'espèces de poissons.

1.4 Reconnaissance d'espèces de poissons

Les données collectées par les techniques d'observation sous-marine font l'objet de traitement et d'analyse. Nous distinguons ici deux types de méthodes de traitement et d'analyse : les méthodes visuelles dites traditionnelles et les méthodes automatiques. Cette section présente une revue des techniques de reconnaissance d'espèces de poissons dans les domaines de biologie marine et de vision par ordinateur. Nous décrivons dans un premier temps les limites des méthodes traditionnelles. Ensuite, nous présentons les techniques par apprentissage automatique destinées à la reconnaissance d'espèces de poissons.

1.4.1 Reconnaissance visuelle

Nous présentons ici les méthodes traditionnelles de reconnaissance d'espèces de poissons classiquement pratiquées en biologie marine. Afin d'identifier un organisme vivant, les biologistes se basent sur la classification scientifique des espèces appelée aussi la classification biologique. Cette classification est une méthode scientifique qui permet de classer et regrouper les organismes vivants ayant des ressemblances et des caractères en commun -au niveau biologique, phénotypique ou physiologique- en entités appelées taxons (Keat-

Chuan Ng et al. 2017). Ce système de classification repose sur une hiérarchie de sept taxons définie de la façon suivante : règne, embranchement, classe, ordre, famille, genre, et espèce. C'est généralement les deux derniers niveaux, genre et espèce, qui sont utilisés pour décrire les organismes, formant leurs noms scientifiques binomiaux. Un exemple de classification d'une espèce de poisson est illustré dans la figure 1.7 suivante.

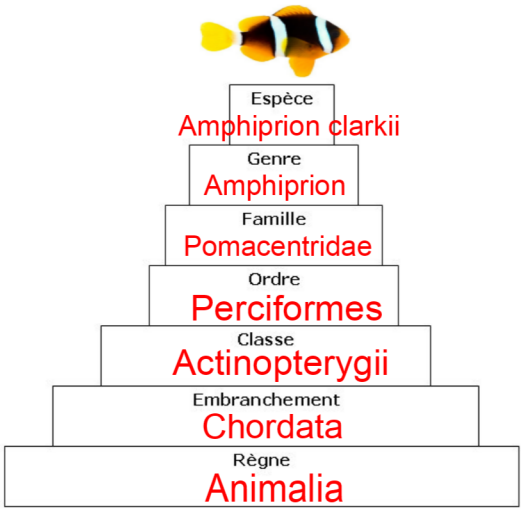


Figure 1.7 – Un exemple de l'arbre de taxonomie de l'espèce Amphiprion Clarkii.

Dans le contexte de la classification ichtyologique, les biologistes marins identifient les poissons à partir de leurs caractéristiques ichtyologiques telles que la morphologie, les parasites, la cytogénétique, l'immunogénétique, etc. (Begg et Waldman 1999 ; Keat-Chuan Ng et al. 2017 ; Miller 1972 ; Strauss et Bond 1990). La figure 1.8 montre des exemples de formes de nageoires caudales qui sont utilisées pour identifier les poissons.

Les systèmes d'enregistrement vidéo sous-marine produisent de grandes quantités de données. L'analyse visuelle de ces données peut être longue, subjective, coûteuse et sujette aux erreurs conformément à l'expérience des observateurs, en particulier pour les espèces d'apparence similaire (Benedetti-Cecchi et al. 1996 ; Edgar, Barrett et Morton 2004 ; Sale et Sharp 1983). Par conséquent, il est apparu opportun d'utiliser des techniques de vision par ordinateur pour traiter et analyser automatiquement les données enregistrées.

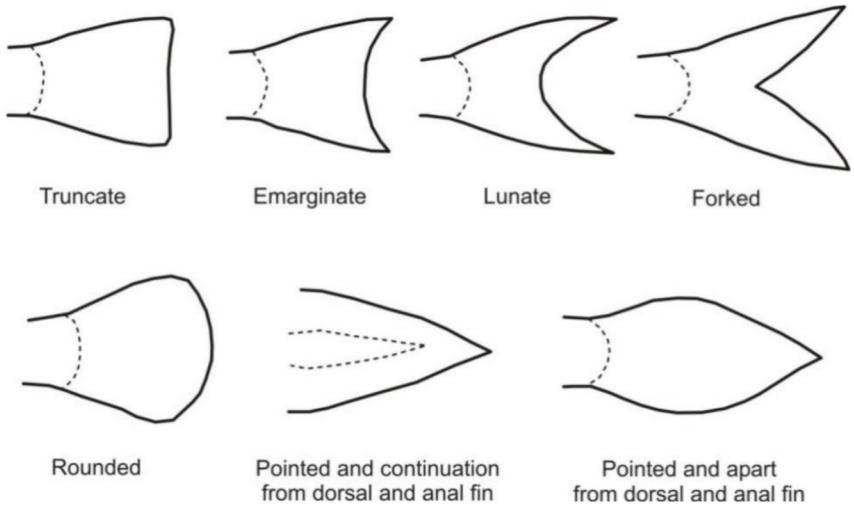


FIGURE 1.8 – Exemples de formes de nageoires caudales utilisées en reconnaissance d'espèces de poissons (KEAT-CHUAN NG et al. 2017).

1.4.2 Reconnaissance automatique

Cette section présente des méthodes de traitement d'images d'apprentissage automatique pour la reconnaissance d'espèces de poissons dans des enregistrements vidéo sous-marins. Généralement, le processus de reconnaissance consiste en trois étapes suivantes : le prétraitement des images, la détection de poissons et la classification de leurs espèces.

1.4.2.1 Prétraitement des images

Le prétraitement des images est l'une des étapes les plus critiques, y compris le choix d'espace colorimétrique, la suppression du bruit, l'augmentation artificielle de données, et l'élimination d'arrière-plan. L'image capturée à partir de l'environnement réel contient des nuisances telles que du bruit et un arrière-plan relativement complexe. Il est dès lors essentiel d'effectuer un prétraitement en vue d'améliorer les performances de la détection et de la classification. Nous citons ici quelques techniques de prétraitement les plus utilisées :

— Choix d'espace colorimétrique : pour le choix d'espace colorimétrique, la majorité des travaux en reconnaissance d'espèces de poissons ont utilisé l'espace colorimétrique

RVB (Rouge, Vert, Bleu), appelé aussi RGB[8] (Qin et al. 2016). Quelques travaux ont utilisé ou testé d'autres espaces colorimétriques comme celui des niveaux de gris (Salman et al. 2016) ou l'espace TSI (Teinte, Saturation, Intensité) appelé aussi HSI[9] (Sun et al. 2018).

— Suppression du bruit : parce qu'un bruit intense peut affecter les performances de l'algorithme de reconnaissance (Abe et al. 2017), des travaux ont proposé d'utiliser des filtres pour éliminer le bruit tels que le filtre gaussien (Sun et al. 2018) ou le filtre médian (Jin et Liang 2017).

— Recadrage et redimensionnement : le recadrage d'image a été utilisé pour éliminer les informations redondantes de l'arrière-plan (Salman et al. 2020). En plus, l'image est redimensionnée à une taille fixe en tant qu'entrée d'un réseau d'apprentissage profond (Qin et al. 2016 ; Sun et al. 2018).

— Augmentation artificielle de données : cette technique consiste à générer des nouvelles données à partir de données existantes. Elle est largement utilisée dans les réseaux de neurones profonds. Il existe plusieurs façons pour augmenter artificiellement le nombre d'images d'apprentissage de poissons telles que le retournement, la rotation, le flou, la translation, et/ou le changement d'intensité lumineuse (Qin et al. 2016 ; Sun et al. 2018).

— Soustraction du fond : le but de la soustraction du fond est de séparer le premier plan (des objets) de l'arrière-plan. Dans (Qin et al. 2016), les auteurs ont utilisé des masques binaires pour supprimer l'arrière-plan avant d'attaquer un réseau neuronal profond.

1.4.2.2 Détection automatique de poissons

Dans la littérature, plusieurs approches ont été proposées pour la détection automatique de poissons. Nous pouvons les regrouper en deux principales catégories : des approches traditionnelles basées sur la modélisation d'arrière-plan et des approches basées

8. RGB : *Red, Green, Blue.*
9. HSI : Hue Saturation Intensity.

sur l'apprentissage profond.

i. Approches par modélisation de l'arrière-plan

Empruntées au domaine de vidéosurveillance, ces approches sont généralement basées sur le mouvement pour modéliser l'arrière-plan. Il existe différentes techniques utilisées notamment la soustraction d'images et l'élimination d'arrière-plan.

La soustraction d'images est une méthode de détection non paramétrique qui sépare les régions de premier plan et d'arrière-plan en soustrayant les valeurs de pixels de deux images successives de vidéo (Lan et al. 2014). C'est la méthode de détection la plus simple, mais elle est très sensible aux changements d'illumination. De même, elle ne peut pas être utilisée avec une caméra en mouvement, et elle ne permet pas d'identifier une cible stationnaire ou lente. Il est également difficile avec cette méthode de mettre à jour l'image d'arrière-plan en temps réel.

La détection par soustraction d'arrière-plan est une méthode basée sur la modélisation de l'arrière-plan et effectue une différence entre l'image courante et le modèle d'arrière-plan. Elle est utilisée pour segmenter la région en mouvement et permet la détection et la distinction des objets dynamiques à partir de caméras statiques. Afin d'éviter l'influence des changements d'éclairage, l'image d'arrière-plan doit être continuellement mise à jour en fonction de l'image courante. Plusieurs algorithmes de soustraction d'arrière-plan ont été proposés pour la détection de poissons, y compris l'algorithme de modèle de mélange de gaussiennes[10] (Hsiao et al. 2014 ; Salman et al. 2019 ; Spampinato et al. 2008), l'algorithme d'estimation par noyau[11] (Shevchenko, Eerola et Kaarna 2018) et les extracteurs d'arrière-plan visuel[12] (Shevchenko, Eerola et Kaarna 2018). Ces techniques sont basées sur le mouvement du poisson, elles nécessitent que le poisson soit en mouvement la plupart du temps, et que les caméras doivent être fixes. En outre, leurs performances sont profondément affectées par différents facteurs, notamment le changement d'éclairage, l'arrière-plan dynamique, la stabilité de la caméra et l'occlusion.

Les approches traditionnelles par modélisation de l'arrière-plan sont considérées comme approches d'apprentissage peu profond (Bengio 2009). Cela signifie qu'elles sont incapables de modéliser des environnements sous-marins dans situations comme l'arrière-plan

10. GMM : *Gaussian Mixture Model.*
11. KDE : *Kernel Density Estimation.*
12. ViBe : *Visual background extractors.*

avec des textures complexes, l'arrière-plan dynamique, le changement fréquent de luminosité, la mauvaise visibilité, le faible contraste, le camouflage des poissons avec l'arrière-plan (en raison de la similitude des couleurs et des textures) ou encore l'occlusion des poissons.

ii. Approches profondes

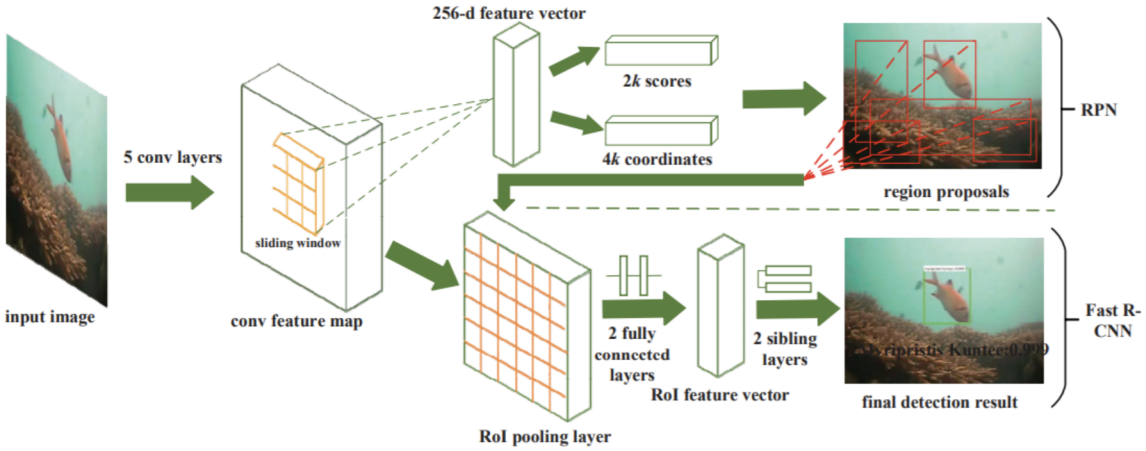


FIGURE 1.9 – L'architecture proposée par (LI et al. 2016) pour la détection de poissons utilisant le modèle Faster R-CNN (REN et al. 2015).

Ces dernières années, les chercheurs en apprentissage automatique se sont concentrés sur les caractéristiques apprises par l'apprentissage profond, en particulier avec les réseaux de neurones convolutifs (CNNs [13]) (LECUN et al. 1998). Les CNNs ont montré leur efficacité dans la reconnaissance d'objets (KRIZHEVSKY, SUTSKEVER et HINTON 2012), la détection d'objets (GIRSHICK 2015 ; GIRSHICK et al. 2014 ; REN et al. 2015), l'étiquetage de scènes (FARABET et al. 2012) et la traduction linguistique (SUTSKEVER, VINYALS et LE 2014). Les CNNs sont capables d'extraire des caractéristiques de haut niveau à partir de données non linéaires en transformant les données d'entrée de bas niveau à travers plusieurs niveaux de représentation.

Les travaux les plus récents utilisent des détecteurs d'objets classiques préexistants, basés sur les CNNs, pour la détection automatique de poissons. Le principe et l'architecture de ces détecteurs classiques seront détaillés dans le prochain chapitre. (LI et al. 2015) ont appliqué Fast R-CNN [14] (GIRSHICK 2015) sur des images sous-marines de poissons

13. *Convolutional Neural Netwroks* dans la littérature anglo-saxonne.
14. Fast R-CNN : *Fast Regions with CNN*.

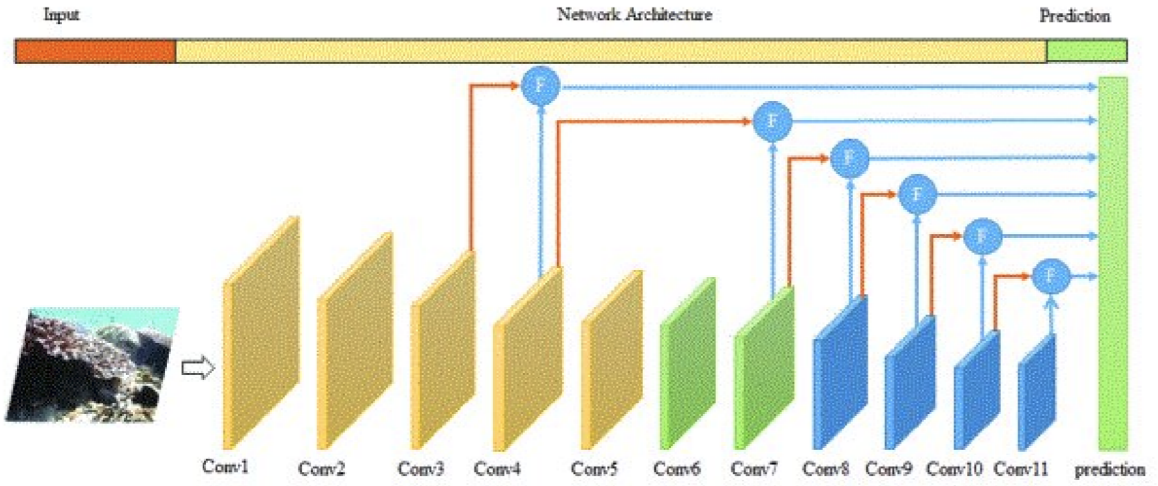


FIGURE 1.10 – L'architecture FFDet proposée par (SHI, JIA et CHEN 2018). Elle utilise le détecteur SSD et combine des caractéristiques extraites de différentes couches.

pour détecter et identifier leurs espèces. Ils ont atteint une précision moyenne (mAP [15]) de 81,40% sur la base d'images "*LifeCLEF 2014 Fish*" [16]. Ils ont également accéléré la détection, dans (LI et al. 2016), en utilisant le détecteur Faster R-CNN (REN et al. 2015) avec ZFNet [17] (ZEILER et FERGUS 2014) comme illustré dans la figure 1.9. Ils ont atteint une mAP de 82,70% sur la même base d'images. Par la suite, ils ont amélioré la mAP de 7,25% dans (LI, TANG et GAO 2017) en utilisant PVANet (HONG et al. 2016) avec Faster R-CNN. (MANDAL et al. 2018) ont utilisé Faster R-CNN avec trois réseaux convolutifs différents ZFNet, CNN-M [18] (CHATFIELD et al. 2014) et VGGNet [19] (SIMONYAN et ZISSERMAN 2014b) pour détecter 50 espèces de poissons et crustacés. Ils ont atteint une mAP de 82,40% sur des vidéos sous-marines capturées à partir de plusieurs plages et estuaires d'Australie. (ZHUANG et al. 2017) ont utilisé le détecteur SSD [20] (LIU et al. 2016) avec PVANet pour détecter les poissons. (SHI, JIA et CHEN 2018) ont présenté FFDet [21] qui utilise le détecteur SSD et combine des caractéristiques extraites de différentes couches (figure 1.10). Ils ont pu atteindre une mAP de 62,83% sur 7 514 instances de poissons issues

15. mAP : *mean Average Precision.*
16. `https://www.imageclef.org/2014/lifeclef/fish`
17. ZFNet : *Zeiler and Fergus Network.*
18. CNN-M : *CNN Medium.*
19. VGGNet : *Visual Geometry Group Network.*
20. SSD : *Single Shot Detector.*
21. *FFDet : Fused Fish Detection.*

de l'ensemble de données "*SeaClef 2016*" [22]. (Sung, Yu et Girdhar 2017) ont utilisé le détecteur YOLO [23] (Redmon et al. 2016) et ont obtenu une précision moyenne de 65,20% sur 93 images sous-marines de poissons.

D'autres travaux ont introduit des approches hybrides basées sur les CNNs et des méthodes traditionnelles comme GMM. (Jäger et al. 2016) ont utilisé la soustraction d'arrière-plan pour obtenir des propositions de boîtes englobantes. Ensuite, ces propositions sont passées dans un CNN pour extraire des caractéristiques. Enfin, une machine à vecteurs de support (SVM [24]) binaire classifie les propositions en deux classes poisson ou arrière-plan. (Zhang et al. 2016) ont proposé une détection non supervisée de poissons. Tout d'abord, ils ont généré et étiqueté automatiquement des propositions de régions en utilisant la segmentation de flux de mouvement et la recherche sélective (Uijlings et al. 2013). Ensuite, un CNN a été utilisé pour classer les propositions comme poisson ou arrière-plan. (Salman et al. 2020) ont concaténé l'image d'entrée en niveaux de gris avec le flux optique et le résultat de GMM afin d'alimenter un Faster R-CNN standard (figure 1.11). Ils ont obtenu une F-mesure de 87,44% sur la base d'images "*Fish4Knowledge Complex Scenes*" [25] et de 80,02% sur la base d'images "*LifeCLEF 2015 Fish*" [26].

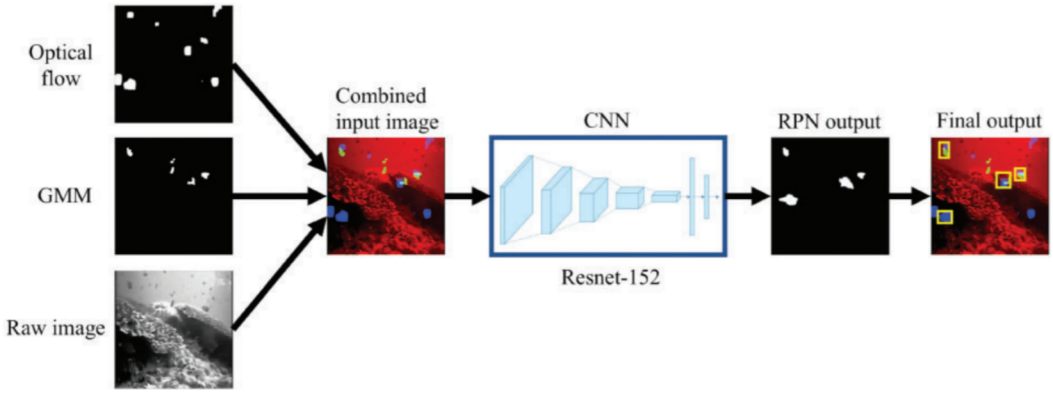


Figure 1.11 – Architecture proposée par (Salman et al. 2020) pour la détection de poissons. Le système est entraîné sur des images combinant les sorties de l'algorithme GMM, le flux optique et les images vidéo en niveaux de gris. Ceci est analogue à une image RGB à trois canaux.

22. https://www.imageclef.org/lifeclef/2016/sea
23. YOLO : *You Only Look Once*.
24. SVM : *Support Vector Machine*.
25. https://groups.inf.ed.ac.uk/f4k/F4KDATASAMPLES/INTERFACE/DATASAMPLES/search.php
26. www.imageclef.org/lifeclef/2015/fish

1.4.2.3 Classification automatique d'espèces de poissons

Dans la littérature, nous distinguons deux grandes catégories de travaux sur la classification d'espèces de poissons : la classification d'espèces de poissons morts et celle de poissons vivants. Cette dernière peut également subdiviser en deux sous-catégories selon l'environnement : contraint ou naturel.

i. Classification d'espèces de poissons morts

La première application de la classification d'espèces de poissons morts était dans le domaine de l'industrie de la pêche. (Strachan 1993 ; Strachan et Kell 1995 ; Strachan, Nesvadba et Allen 1990) ont proposé un système de classification utilisé à bord des navires de pêche. Le but est de classifier des poissons selon leurs espèces et de les envoyer vers des chaînes de traitement différentes. Pour cela, une caméra vidéo couleur est placée au-dessus d'un tapis roulant où les poissons pêchés défilent comme illustré dans la figure 1.12. Ce système utilise un éclairage artificiel qui se fait par une lampe diffuse et un rétro-éclairage. Pour la classification d'espèces, ils ont extrait des caractéristiques de couleur, longueur et largeur de la silhouette. Ensuite, ils ont utilisé l'analyse discriminante canonique comme méthode de classification. Cette méthode construit une fonction mathématique polynomiale qui permet de discriminer les espèces. Ils ont réussi à reconnaître neuf espèces avec un taux de reconnaissance moyen de 99%.

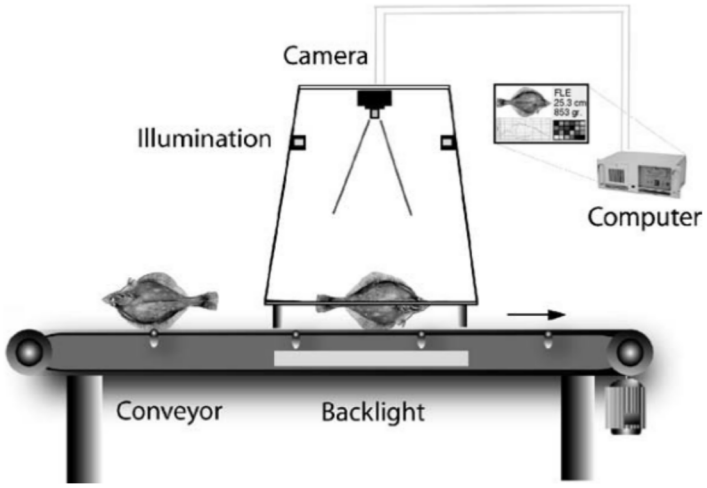


Figure 1.12 – Système de reconnaissance de poissons pêchés à bord d'un navire (White, Svellingen et Strachan 2006).

De même, (Loonis, MENARD et Demko 1996) ont proposé un système de reconnaissance de poissons morts dans une usine de traitement de poisson. Ils ont aussi utilisé le même système d'éclairage et des caractéristiques de la forme, de la couleur et de la texture. Ensuite, ils ont adopté un algorithme génétique pour sélectionner les caractéristiques les plus pertinentes. Finalement, un réseau de neurones est utilisé pour la classification. Ce système a permet de reconnaître sept espèces avec un taux de reconnaissance moyen de 90% en utilisant seulement 96 images en apprentissage.

(Zion, Shklyar et Karplus 1999) ont proposé une méthode basée sur les moments invariants pour reconnaître trois espèces de poissons morts. Ils ont pris des images de poissons dans différentes positions et orientations et sous différentes conditions de luminosité. Ils ont construit un arbre de décision en se basant sur le rapport longueur-largeur du poisson et la forme de la nageoire caudale. Ils ont obtenu un taux de reconnaissance moyen de 93,67% sur les trois espèces.

(Larsen, Olafsdottir et Ersbøll 2009) ont présenté une méthode de classification de trois espèces de poissons (la morue, l'aiglefin et le merlan) basée sur la forme et la texture en utilisant un modèle actif d'apparence (Cootes, Edwards et Taylor 2001) sous un éclairage contrôlé. Ils ont classé les caractéristiques à l'aide de l'analyse discriminante linéaire rapportant un taux de reconnaissance de 76% sur une base de 108 images de poissons.

ii. Classification d'espèces de poissons vivants

Dans cette catégorie, nous distinguons deux cas de classification d'espèces de poissons selon l'environnement : contraint (ex. en aquariums) ou naturel (ex. en mer, lac ou rivière).

- Classification en environnement contraint

Dans un environnement contraint, les caractéristiques environnementales telles que l'éclairage et l'arrière-plan sont contrôlés. Dans ce contexte d'identification d'espèces de poissons, la zone de recherche est limitée et des connaissances préalables sont déjà fournies telles que le nombre d'espèces et la forme des poissons.

(Benson et al. 2009) ont utilisé la classification de Haar pour détecter et classifier l'espèce de poisson "*Scythe Butterfly*" de l'aquarium Birch de San Diego. Leur méthode dépend fortement de l'arrière-plan de l'image et de l'angle sous lequel l'image est prise. Ils

ont réussi à classifier 92 images de poissons sur 100 images de test. Le système proposé est évalué uniquement sur une espèce de poisson. L'ajustement des paramètres pour une seule espèce de poisson est simple, mais rien n'indique que leur algorithme est adapté à l'identification d'autres espèces.

(Khalifa, Taha et Hassanien 2018) ont proposé un système d'identification pour les aquariums. Ce système identifie huit familles de poissons ainsi que 191 espèces. Le système proposé est un réseau de neurones convolutifs (LeCun et al. 1998). Il se compose de quatre couches, deux couches convolutives et deux couches entièrement connectées (figure 1.13). Sur un ensemble de 277 images de poissons, ils ont pu atteindre un taux moyen de reconnaissance de 85.59% .

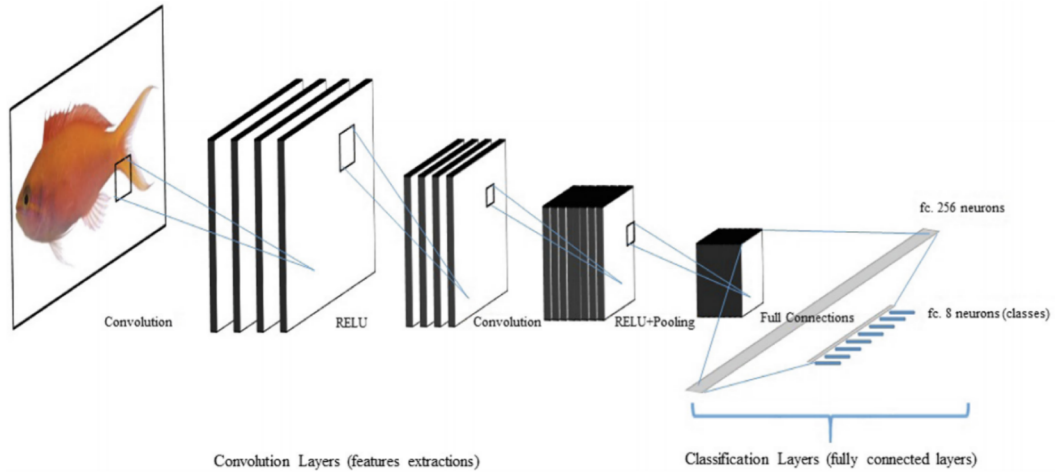


Figure 1.13 – L'architecture neuronale profonde proposée par (Khalifa, Taha et Hassanien 2018).

- Classification en environnement naturel

Dans ce type d'environnement, les poissons nagent librement et l'arrière-plan peut être changeant. Le système de reconnaissance doit aussi traiter les transformations affines et les distorsions telles que l'échelle, la rotation, les changements d'illumination et le flou.

Les premiers travaux ont utilisé des techniques traditionnelles pour reconnaître des poissons vivants dans le milieu marin naturel. (Spampinato et al. 2010) ont proposé un système automatique de classification d'espèces afin d'aider les biologistes marins à comprendre le comportement des poissons. Ils ont combiné deux types de caractéristiques pour classifier des poissons : les caractéristiques de texture et les caractéristiques de forme.

Une transformation affine est également appliquée aux images acquises pour représenter le poisson en 3D. Le système est testé sur une base contenant 360 images de dix espèces différentes. Ils ont atteint une précision moyenne d'environ 92%.

(Cabrera-Gámez et al. 2015) ont calculé différents descripteurs locaux comme l'histogramme des gradients orientés (HOG [27]), les modèles binaires locaux (LBP [28]), les modèles binaires locaux uniformes ($LBPu^2$) [29] et les modèles de gradient locaux (LGP [30]). Afin d'améliorer les résultats de classification, ils ont adopté une approche de fusion de niveaux de score où la première couche est composée d'un ensemble de classifieurs conçus selon les descripteurs choisis, tandis que le classifieur de deuxième couche prend en entrée les scores de la première couche. Ils ont testé leur approche sur la base de validation "*LifeCLEF 2015 Fish*" composée de 20 vidéos annotées ; ils ont atteint une précision moyenne de 40,41%.

Les techniques basées sur les SVMs peuvent être considérées comme des classifieurs plats du fait qu'elles classifient toutes les classes en même temps et cela en utilisant les mêmes caractéristiques pour l'ensemble des classes. Parfois, il pourrait être judicieux de choisir des caractéristiques spécifiques selon les classes ; cela est pris en compte dans les techniques dites à arbre de classification hiérarchique. L'idée est de séparer progressivement l'ensemble des images en sous-classes, avec pour chaque nœud de l'arbre un jeu de caractéristiques propres. L'inconvénient principal de cette structure est l'accumulation d'erreurs car si une erreur est commise à un nœud, elle va nécessairement entraîner de nouvelles erreurs dans les nœuds enfants. (Huang, Boom et Fisher 2012) ont extrait 66 types de caractéristiques : couleur, forme et texture de différentes parties du poisson. Ensuite, ils ont proposé une méthode de classification hiérarchique appelée "*Balance-Guaranteed Optimized Tree (BGOT)*" sensée minimiser le problème d'accumulation d'erreurs. Ils ont obtenu une précision moyenne de 95% sur une base contenant 3179 images de poissons de dix espèces différentes.

(Szűcs, Papp et Lovas 2015) ont élaboré un système pour détecter, classer et suivre les poissons en vidéos sous-marines. Pour la classification, ils ont catégorisé les poissons détectés avec un classifieur de vecteurs c-support. Le classifieur a utilisé des descripteurs de haut niveau, qui sont basés sur les vecteurs de caractéristiques robustes accélérées (SURF [31])

27. HOG : *Histogram of Oriented Gradients.*
28. LBP : *Local Binary Patterns.*
29. $LBPu^2$: *Uniform Local Binary Patterns.*
30. LGP : *Local Gradient Patterns.*
31. SURF : *Speeded Up Robust Features.*

extraits dans chaque objet (Bay, Tuytelaars et Van Gool 2006). Ils ont atteint une précision moyenne de 51% sur l'ensemble de test de la base "*LifeCLEF 2015 Fish*".

(Villon et al. 2016) ont présenté deux méthodes pour reconnaître les poissons de corail dans des vidéos sous-marines HD. La première méthode repose sur une approche traditionnelle en deux étapes : l'extraction des caractéristiques HOG et l'utilisation d'un classifieur SVM. La deuxième méthode est basée sur l'apprentissage profond en utilisant l'architecture de GoogleNet (Szegedy et al. 2015). Ils ont comparé les deux méthodes et ont trouvé que l'apprentissage profond est plus efficace que HOG+SVM.

(Li et al. 2015) ont appliqué les réseaux convolutifs de type Fast R-CNN (Girshick 2015) pour détecter et reconnaître les espèces de poissons. Ils ont obtenu une précision moyenne de 81,4% sur la base de référence "*LifeCLEF 2014 Fish*" qui contient 24277 images regroupées en 12 classes.

(Salman et al. 2016) ont créé un CNN de trois couches de convolution pour extraire des caractéristiques et alimenter des classifieurs standard comme le SVM et le k plus proches voisins (KNN [32]). Ils ont obtenu une précision moyenne de 96,75% sur un ensemble de test de 7500 images de poissons issues de la base "*LifeCLEF 2015 Fish*". (Qin et al. 2015) ont proposé un CNN avec trois couches convolutives entraîné à partir de zéro sur la base de référence "*Fish Recognition Ground-Truth*", ils ont atteint une précision moyenne de 98,57% . Ils ont aussi proposé dans (Qin et al. 2016) une architecture profonde hybride avec des méthodes traditionnelles pour extraire des caractéristiques d'images de poissons (figure 1.14). Dans cette architecture, l'analyse en composantes principales (ACP [33]) est utilisée dans deux couches convolutives, suivie d'un hachage binaire dans la couche non linéaire et d'un histogramme par blocs dans la couche de sous-échantillonnage. Ensuite, un sous-échantillonnage spatial par pyramide (SPP [34]) (Grauman et Darrell 2005) est utilisé. Enfin, la classification est effectuée avec un SVM linéaire. Par rapport à leur premier travail (Qin et al. 2015), l'architecture profonde hybride proposée n'a amélioré la précision moyenne que de 0,07%.

(Sun et al. 2016) ont appliqué deux architectures profondes, PCANet [35] (Chan et al.

32. KNN : *K-Nearest Neighbor*.
33. ou PCA pour *Principal Component Analysis* en anglais.
34. SPP : *Spatial Pyramid Pooling*.
35. PCANet : *PCA Network*.

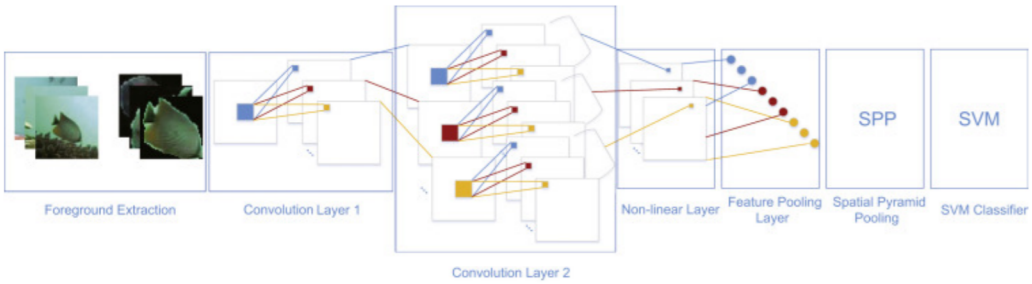


FIGURE 1.14 – L'architecture profonde hybride proposée par (QIN et al. 2016).

2015) et NIN [36] (LIN, CHEN et YAN 2013) pour extraire des caractéristiques à partir des images sous-marines. Un classifieur SVM linéaire est utilisé pour la classification. Ils ont testé leur modèle sur la base "*LifeCLEF 2015 Fish*" et ont obtenu un taux moyenne de reconnaissance de 69,84% avec l'architecture NIN et de 77,27% avec l'architecture PCANet.

(JÄGER et al. 2016) ont utilisé des caractéristiques extraites des activations de la $7^{ème}$ couche cachée du modèle pré-entraîné AlexNet (KRIZHEVSKY, SUTSKEVER et HINTON 2012), sans ré-entraîner le modèle sur la base d'images de poissons, et ont alimenté un classifieur SVM multi-classe. Ils ont atteint une précision moyenne faible de 66% sur la base *"SeaCLEF 2016"*. (SUN et al. 2018) ont proposé d'extraire les caractéristiques de poissons d'un CNN profond pré-entraîné, en utilisant l'apprentissage par transfert (PAN et YANG 2009), et l'augmentation artificielle de données pour surmonter le problème d'un ensemble d'entraînement insuffisant (figure 1.15). Ils ont atteint une précision moyenne de 99,68% sur la base de référence "*Fish Recognition Ground-Truth*" en ré-entraînant AlexNet avec l'augmentation artificielle de données et en utilisant un classifieur SVM.

1.5 Les bases d'images et de vidéos de référence

Les enregistrements vidéo sous-marines permettent de disposer d'importants volumes de données brutes à des fins d'études. Ces données doivent non seulement être labellisées, mais elles doivent aussi rester disponibles. Au niveau national, on déplore encore le manque de bases de données fonctionnelles. Il faut dire que l'annotation des images sous-marines demande beaucoup de temps et un haut niveau de qualification des experts annotateurs. Les données collectées restent en grande partie non annotées, et les jeux de données labellisées

36. NIN : *Network In Network*.

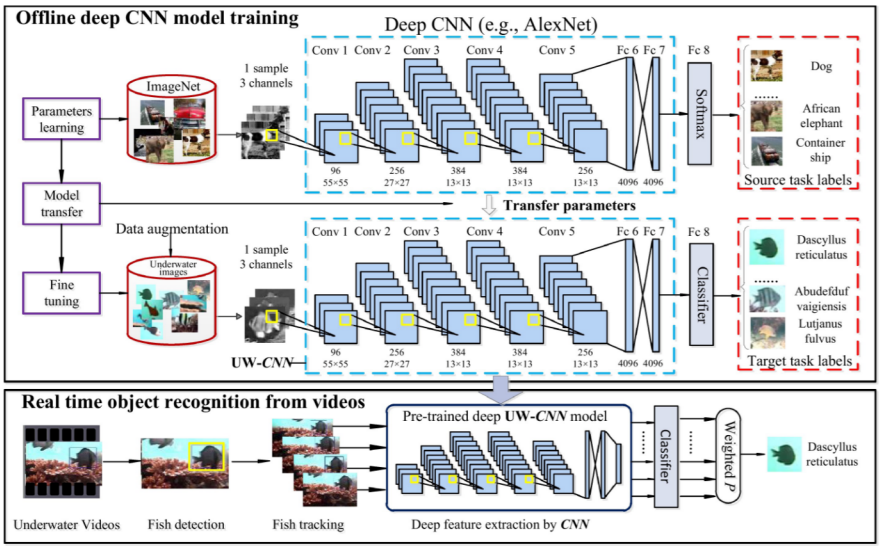


FIGURE 1.15 – Architecture, utilisant l'apprentissage par transfert et l'augmentation artificielle de données, proposée par (SUN et al. 2018).

partagés sont souvent très limités.

Dans cette thèse, nous évaluons les approches proposées sur deux bases de références sous-marines. Les deux bases contiennent des images de poissons de différentes couleurs, textures, positions, tailles et orientations. Les deux sont issues du projet européen "*Fish4-Knowledge*" F4k[37] (BOOM et al. 2012b). Au cours de ce projet de cinq ans, un vaste ensemble de plus de 700 000 vidéos sous-marines avec plus de 3 000 espèces de poissons a été collecté à Taiwan, le plus grand environnement de biodiversité de poissons au monde.

1.5.1 Base d'images "Fish Recognition Ground-Truth" (FRGT)

"*Fish Recognition Ground-Truth*" est une base d'images de poissons vivants acquise à partir d'un ensemble de vidéos sous-marines capturées en milieu marin naturel. Il y existe un total de 27 370 images de poissons distribuées sur 23 classes (une classe par espèce). Les espèces de poissons sont étiquetées manuellement avec l'aide de biologistes marins. La figure 1.16 montre des exemples des 23 espèces et la table 1.2 montre la distribution des espèces dans la base. Les images et les masques de poissons sont simultanément fournis. Ces images de poissons ont différentes tailles allant d'environ 20x20 à environ 200x200

37. https://groups.inf.ed.ac.uk/f4k/

pixels. De plus, cette base présente un réel déséquilibre des classes[38]. L'effectif des espèces les plus fréquentes est environ 1000 fois celui des moins fréquentes (table 1.2). C'est un défi d'obtenir une grande précision sur l'ensemble des classes de la base.

FIGURE 1.16 – Exemples d'images de 23 espèces de poissons issues de la base d'images "*Fish Recognition Ground-Truth*".

ID	Espèce	Effectif	ID	Espèce	Effectif
DR	Dascyllus reticulatus	12112	PM	Pomacentrus moluccensis	181
PD	Plectroglyphidodon dickii	2683	ZS	Zebrasoma scopas	90
CC	Chromis chrysura	3593	HM	Hemigymnus melapterus	42
AC	Amphiprion clarkia	4049	LF	Lutjanus fulvus	206
CL	Chaetodon lunulatus	2534	SB	Scolopsis bilineata	49
CT	Chaetodon trifascialis	190	S	Scaridae	56
MK	Myripristis kuntee	450	PV	Pempheris vanicolensis	29
AN	Acanthurus nigrofuscus	218	ZC	Zanclus cornutus	21
HF	Hemigymnus fasciatus	241	NN	Neoglyphidodon nigroris	16
NS	Neoniphon samara	299	BU	Balistapus undulates	41
AV	Abudefduf vaigiensis	98	SF	Siganus fuscescens	25
CV	Canthigaster valentine	147		Total	27370

TABLE 1.2 – Distribution des espèces de poissons dans la base d'images "*Fish Recognition Ground-Truth*".

Nous soulignons ici que pour cette base d'images de poissons nous avons utilisé la validation croisée à 7-blocs. Nous avons divisé la base en 5/7 pour l'apprentissage, 1/7 pour la validation et 1/7 pour le test.

38. Le nombre d'instances d'une classe (espèce) dépasse fortement celui des instances des autres classes (espèces).

1.5.2 Base de vidéos "LifeClef 2015 Fish" (LCF-15)

"*LifeClef 2015 Fish*" est une base de vidéos sous-marines. L'ensemble d'entraînement proposé est constitué de 20 vidéos annotées manuellement, avec une liste de 15 espèces de poissons. La figure 1.17 montre des exemples des 15 espèces et la table 1.3 montre la distribution des espèces dans la base. Chaque vidéo est étiquetée manuellement et approuvée par deux experts spécialisés. Au total, l'ensemble d'entraînement contient plus de 9000 annotations (espèce et boîte englobante pour chaque annotation) et plus de 20 000 images. Les données de cette base sont aussi déséquilibrées (les effectifs des classes sont très différents). Par exemple, le nombre d'échantillons pour l'espèce "*Dascyllus reticulatus*" est environ 40 fois celui de l'espèce "*Chaetodon speculum*". Comme pour la base FRGT, les images de poissons ont également différentes tailles allant d'environ 20x20 à environ 200x200 pixels.

L'ensemble de test comprend 73 vidéos annotées. Nous soulignons que l'effectif de trois espèces de poissons est nul, il n'y a pas d'occurrences dans l'ensemble de test (table 1.3). Ceci est délibérément fait pour évaluer la capacité des méthodes à rejeter les faux positifs.

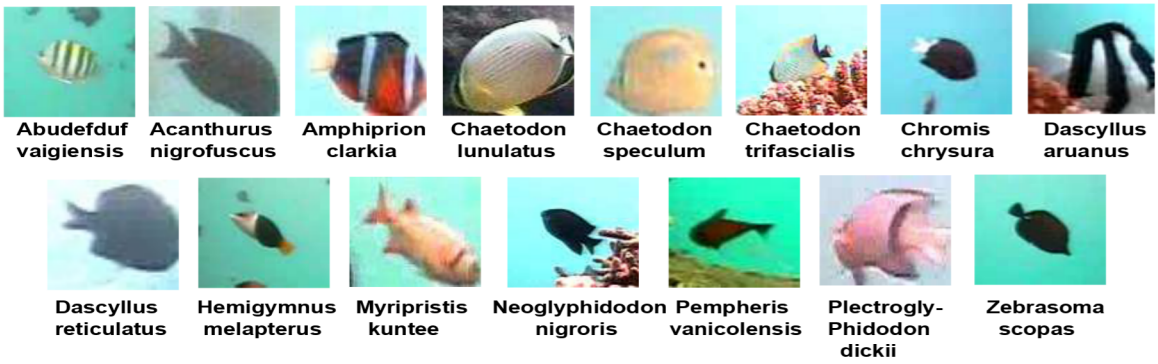


FIGURE 1.17 – Exemples d'images de 23 espèces de poissons issues de la base de vidéos "*LifeClef 2015 Fish*".

Par rapport à la base d'images FRGT, la base LCF-15 contient des images et des vidéos sous-marines de plus mauvaise qualité car floues et faiblement contrastées, avec une illumination non uniforme et des couleurs très atténuées. Avec un récif corallien très riche et des plantes en mouvement, l'arrière-plan est plus complexe et dynamique (SALMAN et al. 2016).

ID	Espèce	Apprentissage	Validation	Test
AV	Abudefduf vaigiensis	349	87	94
AN	Acanthurus nigrofuscus	2244	561	129
AC	Amphiprion clarkia	2677	669	553
CL	Chaetodon lunulatus	2969	742	1876
CS	Chaetodon speculum	130	32	0
CT	Chaetodon trifascialis	545	136	1319
CC	Chromis chrysura	3086	772	24
DA	Dascyllus aruanus	1422	355	2013
DR	Dascyllus reticulatus	5066	1267	4898
HM	Hemigymnus melapterus	285	71	0
MK	Myripristis kuntee	2597	649	118
NN	Neoglyphidodon nigroris	171	43	1643
PV	Pempheris Vanicolensis	838	210	0
PD	Plectrogly-Phidodon dickii	2355	589	676
ZS	Zebrasoma scopas	274	69	187
	Total	25008	6252	13530

TABLE 1.3 – Distribution des espèces de poissons dans la base "*LifeClef 2015 Fish*".

1.6 Discussion et positionnement de la thèse

La vidéo sous-marine est de plus en plus utilisée en biologie marine pour surveiller et suivre les écosystèmes marins. Avec les progrès technologiques des deux dernières décennies, le domaine de l'imagerie sous-marine a connu une grande avancée en matière d'équipements d'acquisition de données (netteté et résolution des images, étanchéité, autonomie de la batterie, capacité de stockage, miniaturisation, mode de connection, ...). Le traitement automatique de données vidéo reste cependant très rare à cause d'une information sous-marine trop complexe à décrire et à analyser. En effet, l'environnement marin pose de grands défis en vision par ordinateur de par les dégradations qui entachent les images et qui réduisent la visibilité du signal d'intérêt. Les phénomènes d'atténuation de la lumière, par absorption et diffusion, dues à l'eau pure s'amplifient avec la turbidité de l'eau causée par la présence de matières organiques dissoutes et de particules en suspension. En milieu corallien, les défis deviennent encore plus grands à cause de la diversité de cet habitat complexe (algues, coraux, éponges), de mouvement des plantes aquatiques, et de la variété de poissons qu'on y rencontre. Dans une tâche de reconnaissance d'espèces de poissons, le poisson se déplace librement dans toutes les directions, il peut aussi se cacher derrière des rochers et des algues. A cela s'ajoutent les problèmes de chevauchement entre des poissons passant devant la caméra, et de similitude de forme, de couleur et de texture

entre différentes espèces.

Comme nous l'avons vu, de nombreux travaux ont abordé le problème de la reconnaissance automatique d'espèces de poissons dans un milieu marin naturel. Cette reconnaissance est composée de deux étapes : 1) la détection de poissons qui vise à détecter et discriminer les poissons de l'arrière-plan, 2) la classification d'espèces de poissons qui vise à identifier l'espèce de chaque poisson détecté (Sun et al. 2018).

Avec l'arrivée de l'apprentissage profond (ou deep learning), en particulier les réseaux de neurones convolutifs CNNs, de nombreux travaux se sont intéressés à étudier l'apport des CNNs à la résolution de différentes tâches en vision par ordinateur. Dans cette thèse, nous nous penchons sur l'utilisation des algorithmes CNN pour la reconnaissance d'espèces de poissons dans des images vidéo sous-marines.

Les approches proposées dans la littérature pour la détection de poissons sont à un seul réseau, puisqu'elles sont constituées d'un seul détecteur autonome (Li et al. 2015 ; Mandal et al. 2018 ; Salman et al. 2020 ; Shi, Jia et Chen 2018 ; Zhuang et al. 2017). La détection par fusion intègre des informations provenant de plusieurs modalités (par exemple RGB, profondeur, infrarouge, et thermique) (Guerry, Le Saux et Filliat 2017) ou de plusieurs espaces (par exemple spatial et temporel) (Peng et Schmid 2016). L'intérêt de cette fusion est l'amélioration de la robustesse de la prédiction, en se basant sur des informations complémentaires et en s'assurant que le système reste opérationnel même en cas de perte d'une source d'information. A notre connaissance, à l'exception de (Salman et al. 2020), ces approches de détection de poissons ne prennent pas non plus en compte l'information de mouvement du poisson qui pourrait pourtant être utile pour la tâche de détection. Dans cette thèse, nous allons proposer un détecteur à réseaux parallèles qui fusionne deux architectures CNN afin d'améliorer la robustesse de la détection de poissons.

Pour la classification d'espèces de poissons, les travaux de l'état de l'art utilisant les CNNs ont exploité des images en niveaux de gris ou en couleurs, mais uniquement dans l'espace RGB ou TSI, bien qu'il ait été montré en vision par ordinateur l'intérêt d'utiliser d'autres espaces colorimétriques pour la classification d'objets (Kasaei et al. 2020 ; Kim, Park et Jung 2018). Dans nos travaux de thèse, nous allons tester différents espaces couleurs pour la classification d'espèces de poissons dans des images vidéo sous-marines. Nous allons aussi soulever la question de la nécessité d'éliminer l'arrière-plan des images de poissons avant la classification. Le fond marin est très riche, notamment dans les récifs

coralliens, et susceptible fournir des caractéristiques qui perturbent la classification.

D'un autre côté, l'entraînement des algorithmes CNN exige un très grand volume de données. Or, la plupart des bases d'images sous-marines disponibles ont des classes de petites tailles. Nous trouvons dans la littérature des travaux qui ont tenté d'augmenter artificiellement les données des classes pour améliorer les performances du modèle d'apprentissage. En général, ils augmentent les données pour toutes les classes même si la base d'images est déséquilibrée (Sun et al. 2018). Certains travaux ont augmenté les données uniquement pour les classes dont l'effectif est inférieur à un seuil afin d'équilibrer la base d'images (Qin et al. 2016). Cependant, l'augmentation artificielle de données nécessite plus de ressources de mémoire et de processeur. Par conséquent, il pourrait être nécessaire de procéder à une augmentation de données uniquement pour les classes difficiles à classifier. Dans cette thèse, nous proposons de tester différentes options d'augmentation artificielle de données dans un cadre d'apprentissage par transfert (Pan et Yang 2009).

Une classification multi-classe classique basée sur CNN génère à la sortie une probabilité pour chaque classe. Ensuite, le taux le plus élevé détermine la classe d'objet d'entrée. Avec cette structure de classification, le CNN traite toutes les classes de la même manière. Or, certaines classes sont naturellement plus susceptibles d'être mal classées que d'autres, en particulier pour les classes ayant moins d'effectifs ou étant difficiles à classifier. Dans certaines applications, les classes de la base d'entraînement peuvent être regroupées en des catégories comme par exemple des taxons. Ensuite, le modèle CNN est entraîné sur chaque catégorie de manière hiérarchique (Sali et al. 2020) ou incrémental (Masana et al. 2020). Dans notre application, la classification taxonomique est la principale classification scientifique des espèces. Elle permet de regrouper les espèces ayant des ressemblances et des caractères communs dans un même taxon. Cette classification nous inspire de proposer un apprentissage progressif et hiérarchique qui classifie d'abord les poissons en familles puis en espèces. Nous pouvons également regrouper les espèces selon leurs degrés de difficulté. Les espèces qui sont difficiles à identifier ont besoin d'un traitement particulier lors de l'apprentissage du modèle CNN. Nous proposons de construire un classifieur CNN en partant d'abord des espèces difficiles à classifier. Au début le modèle se focalise à bien apprendre les espèces difficiles, puis apprend progressivement de manière incrémentale les autres espèces. Nous cherchons à maintenir le modèle stable lors de l'introduction des nouvelles espèces à apprendre tout en gardant des performances élevées sur les anciennes espèces déjà apprises.

1.7 Conclusion

Dans ce chapitre, nous avons présenté différentes techniques d'observation de la biodiversité sous-marine. L'analyse visuelle de données collectées par les enregistrements vidéo sous-marins est coûteuse et nécessite de l'expérience et beaucoup de temps. L'analyse automatique des vidéos sous-marines est un besoin avéré des acteurs de l'écologie marine. Nous avons présenté également les différentes approches proposées dans la littérature pour traiter automatiquement les images sous-marines, en particulier les méthodes de pré-traitement, de détection et de classification. Comme plein d'autres secteurs, la reconnaissance d'espèces de poissons a aussi commencé à bénéficier des développements récents de l'apprentissage profond dans le domaine de la vision par ordinateur. Cette approche a montré son potentiel et son efficacité pour la reconnaissance d'objets naturels dans des environnements plus au moins complexes. Finalement, nous avons positionné les outils et méthodes proposés et développés dans le cadre de cette thèse par rapport aux travaux de l'état de l'art.

Dans le chapitre suivant, nous allons faire un état de l'art sur les réseaux de neurones et l'apprentissage profond.

Chapitre 2

Réseaux de neurones et apprentissage profond

Sommaire

2.1 Introduction

Les réseaux de neurones profonds constituent une des approches d'intelligence artificielle. Ils sont devenus en quelques années des outils précieux dans des domaines très divers de l'industrie et des services dont la vision par ordinateur, la compréhension de la parole, l'analyse de langages naturels ou le diagnostic médical (Abbas, Ibrahim et Jaffar 2019 ; Guo et al. 2016). Nous présentons dans ce chapitre un état de l'art de ces réseaux de neurones profonds. En section 2.2, nous rappelons la définition et les propriétés des réseaux de neurones, nous décrivons aussi les architectures neuronales et les types d'apprentissage les plus utilisés. Ensuite, nous présentons dans la section 2.3 l'historique de l'apprentissage profond et les évolutions techniques dans ce domaine. Dans notre travail nous nous intéressons en particulier aux réseaux de neurones convolutifs (LeCun et al. 1998) qui sont détaillés dans la section 2.4. Finalement, la section 2.5 est consacrée aux architectures profondes récemment développées en vision par ordinateur pour réaliser les deux tâches clés de détection et de classification d'objets.

2.2 Réseau de neurones

Un réseau de neurones artificiel (ANN[1]) est un ensemble d'unités de calcul élémentaires interconnectées dont chacune est appelée neurone s'inspirant du système nerveux biologique (McCulloch et Pitts 1943). Ce réseau est destiné à réaliser des tâches complexes dans différent types d'application : classification, détection, segmentation, étiquetage et régression. Les domaines d'application des réseaux de neurones sont nombreux, on cite par exemple le domaine de sécurité (reconnaissance faciale ou des empreintes, identification des spams, etc.), médical (détection des cellules cancéreuses, etc.), militaire (détection des mines, analyse des images satellitaires, etc.), de l'écologie (études de biodiversité, etc.) et du divertissement (jeux vidéo, etc.) (Ramírez-Quintana, Chacon-Murguia et Chacon-Hinojos 2012).

Pour bien comprendre le principe des réseaux de neurones, nous commençons par la description du neurone biologique avant de décrire les architectures neuronales les plus utilisées.

1. ANN : *Artificial Neural Network dans la littérature anglo-saxonne.*

2.2.1 Neurone biologique

Un neurone est une cellule du système nerveux composée de trois grands éléments (figure 2.1). Le corps cellulaire ou le soma contient le noyau qui traite les informations qui lui parviennent. Les dendrites, qui sont nombreuses et ramifiées, sont les récepteurs principaux du neurone par lesquelles il reçoit les informations arrivées des autres neurones vers le soma. L'axone est un prolongement efférent du soma, il conduit l'information traitée par le soma jusqu'aux dendrites d'autres neurones qui sont à une distance du corps cellulaire variant de 1 mm à plus de 1 m. La transmission entre les neurones s'effectue via des espaces intercellulaires appelés synapses. Un neurone est considéré le maillon élémentaire de la chaîne de transmission de l'information dans le système nerveux (Dudel 1983).

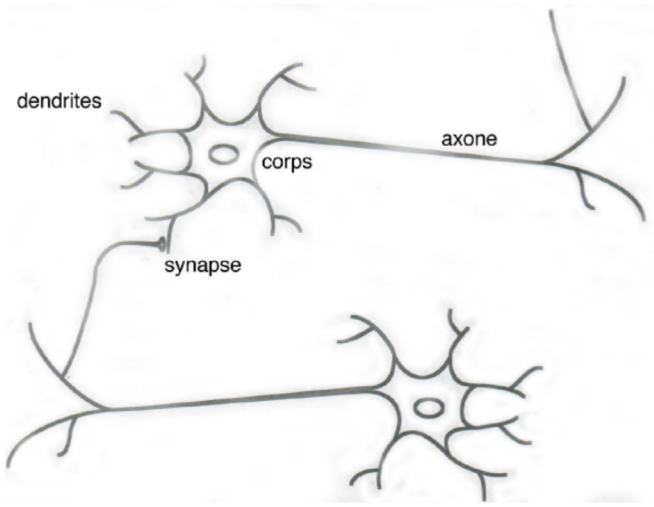


Figure 2.1 – Schéma d'un neurone biologique (Medina-Santiago et al. 2017).

Ces réseaux de neurones biologiques sont capables de réaliser de nombreuses tâches comme la reconnaissance (faciale, vocale, d'objet, ou des émotions), l'apprentissage, et la mémorisation, etc.

2.2.2 Neurone formel

Le premier modèle de neurone formel est proposé par (McCulloch et Pitts 1943) (figure 2.2). Il est basé sur une représentation mathématique simplifiée du neurone biologique. A l'instar de son homologue biologique, il est le maillon élémentaire de la transmission de

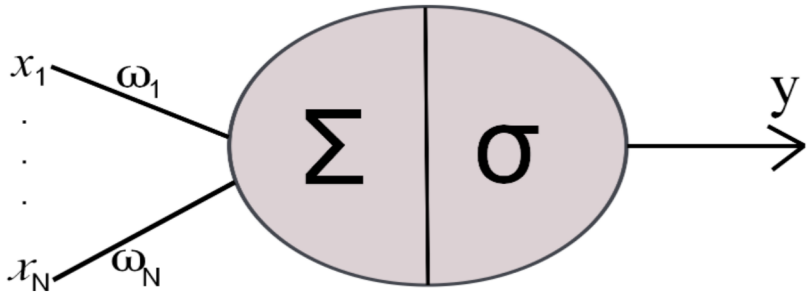


FIGURE 2.2 – Modèle de neurone de (McCULLOCH et PITTS 1943).

l'information au sein du réseau artificiel.

Mathématiquement, le neurone formel effectue une sommation pondérée des signaux entrants $x_i \in \mathbb{R}^n$ provenant d'autres neurones par leurs poids associés ω_i. Le résultat est alors transformé par une fonction d'activation (ou de transfert) σ qui calcule la sortie y du neurone. Ce modèle peut être traduit par l'équation ci-dessous :

$$y = \sigma(\sum_{i=1}^{N} \omega_i.x_i + b) = \sigma(W^T.X + b) \tag{2.1}$$

où N est le nombre d'entrées et b est le biais qui s'appelle aussi le seuil d'activation du neurone.

La fonction de transfert σ définit le type du neurone ; elle peut prendre différentes formes comme, par exemple, la fonction à seuil, la fonction linéaire, la fonction tangente hyperbolique ou la fonction sigmoïde. Le choix du type de fonction dépend de l'utilisation du réseau et de la nature de la sortie y : continue, discrète ou binaire. (DUCH et JANKOWSKI 1999) présentent une étude plus détaillée de différentes fonctions de transfert.

2.2.3 Architectures neuronales

Les neurones formels sont connectés les uns aux autres et organisés en couches pour former un réseau de neurones. Ils peuvent être totalement connectés, dans ce cas tous les neurones sont reliés les uns aux autres, ou localement où le neurone n'est relié qu'à ses voisins les plus proches sur le réseau. Généralement, deux grands modèles de réseaux sont distingués :

— les réseaux non bouclés (dit aussi statiques ou acycliques) ;
— les réseaux bouclés (dit aussi dynamiques ou récurrents).

2.2.3.1 Réseaux de neurones non bouclés

Un réseau de neurones non bouclé est un réseau en couches dans lequel l'information se propage uniquement de la couche d'entrée vers la couche de sortie sans jamais revenir en arrière. Il s'appelle aussi réseau à propagation avant [2]. Nous y distinguons trois types de couches :

— couche d'entrée : elle reçoit les valeurs d'entrée (du réseau) provenant de l'extérieur ;
— couche cachée : elle est constituée de neurones qui reçoivent des résultats de traitement des neurones de la couche précédente et effectuent leurs traitements puis transmettent les résultats aux neurones de la couche suivante. Un réseau peut avoir une ou plusieurs couche(s) cachée(s) voire aucune ;
— couche de sortie : elle fournit le résultat des traitements effectués vers l'extérieur.

Parmi les modèles les plus populaires de ce type de réseaux de neurones, nous nous intéressons au perceptron (Rosenblatt 1958), et plus précisément sa version multicouche (Rumelhart, Hinton et Williams 1986).

i. Perceptron simple

Le perceptron simple ou monocouche est un classifieur binaire. Il a été inventé par (Rosenblatt 1958), c'est le réseau de neurones le plus simple. Il est constitué d'une seule sortie booléenne à laquelle toutes les entrées sont connectées par des liens pondérés. Le réseau est entraîné pour ajuster le poids de chaque lien jusqu'à qu'il puisse discriminer les deux classes.

ii. Perceptron multicouche

Un perceptron multicouche (PMC [3]) est un réseau composé de plusieurs couches successives. Il est composé d'une couche d'entrée, d'une ou plusieurs couche(s) cachée(s) et d'une couche de sortie (figure 2.3). Chaque neurone d'une couche est relié à tous les neurones de la couche suivante. Il n'y a pas de connexions à l'intérieur d'une même couche. Le nombre

2. *Feedforward network* en anglais.
3. Ou MLP pour *Multi-Layer Perceptron* (en anglais).

de neurones dans la couche d'entrée et de sortie dépend du problème à traiter, mais le choix du nombre de couches cachées ainsi que leurs nombres de neurones sont déterminés par un compromis entre performance et vitesse d'apprentissage.

Le perceptron multicouche est utilisé principalement pour des problèmes d'approximation (Attali et Pagès 1997), de prédiction (Koskela et al. 1996) et de classification (Li et al. 2020).

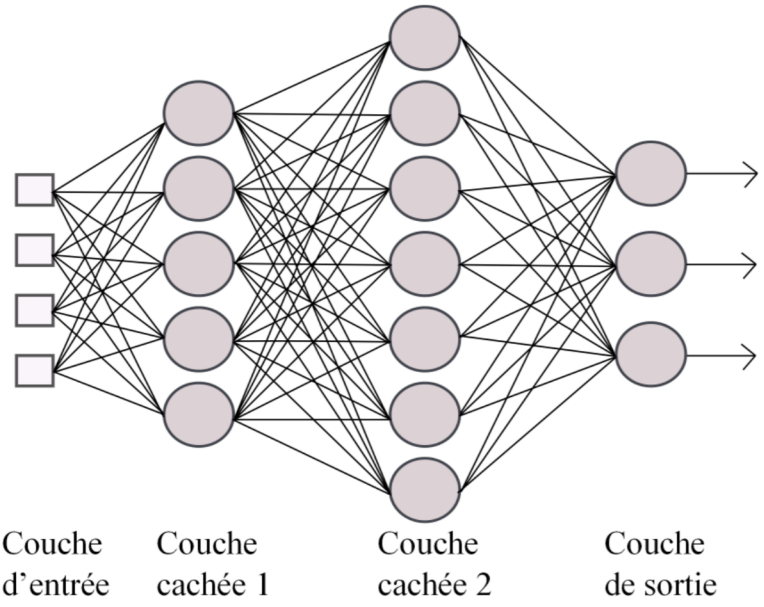


Figure 2.3 – Exemple de perceptron multicouche avec une couche d'entrée, deux couches cachées et une couche de sortie. L'information se propage uniquement dans le sens de la couche d'entrée vers la couche de sortie.

2.2.3.2 Réseaux de neurones bouclés

Contrairement aux réseaux de neurones non bouclés dont chaque neurone possède une connexion vers les neurones de la couche suivante, les réseaux de neurones bouclés ont une architecture générale dont les neurones peuvent posséder des connexions vers n'importe quel neurone, même un retour vers le point de départ du réseau. Ces réseaux sont généralement utilisés pour des tâches de traitement du signal (Sak et al. 2015) ou comme mémoire associative (Sak, Senior et Beaufays 2014).

2.2.4 Apprentissage des réseaux neuronaux

Avant son exploitation, le réseau de neurones doit suivre à un processus d'entraînement pour estimer ses différents paramètres. L'apprentissage ou l'entraînement d'un réseau neuronal est un processus itératif qui consiste à ajuster les paramètres du réseau (poids, biais) jusqu'à l'obtention du comportement désiré, et ce, en lui fournissant des exemples à apprendre.

Selon l'objectif poursuivi et la forme des exemples de la base d'entraînement, on distingue deux grandes catégories d'apprentissage :

— **Apprentissage supervisé** : l'objectif de l'entraînement est prédéterminé via la définition d'une cible à prédire (les exemples d'entraînement sont des couples d'entrée et sortie désirée). Le processus permet d'associer une sortie désirée à chaque entrée, puis modifie les paramètres du réseau progressivement jusqu'à ce que l'erreur entre les sorties calculées par le réseau et les sorties désirées soit minimisée.
— **Apprentissage non supervisé** : dans cet apprentissage, la sortie désirée n'est pas prédéterminée et les exemples d'entraînement sont formés uniquement de valeurs d'entrées. L'objectif de cet apprentissage est de détecter les similarités et les différences entre les exemples. Au cours de l'entraînement, le point de convergence n'est pas connu, le réseau de neurones recherche ce point et y converge.

Il existe plusieurs algorithmes d'apprentissage. Le choix de l'algorithme dépend de la forme de données d'apprentissage, de l'architecture du réseau, de la tâche attendue, et bien d'autres critères. Parmi ces algorithmes on cite : l'algorithme de rétro-propagation (LeCun et al. 1998 ; Rumelhart, Hinton et Williams 1986), la méthode Quasi-Newton (Dennis et Moré 1977), et l'algorithme de BFGS [4] (Yuan 1991). Nous nous intéressons plus particulièrement à l'algorithme de rétro-propagation du gradient et à ses versions modifiées améliorant ses performances et réduisant sa complexité.

4. BFGS : *Broyden–Fletcher–Goldfarb–Shanno.*

2.2.4.1 Algorithme de rétro-propagation du gradient

L'algorithme de rétro-propagation ou de propagation arrière[5] est l'algorithme le plus populaire parmi les méthodes d'apprentissage des réseaux de neurones. Il consiste à minimiser l'erreur calculée par une fonction de perte entre la sortie calculée par le réseau à la dernière couche et la sortie désirée ; la minimisation est effectuée en modifiant les poids du réseau via un calcul de gradient de l'erreur pour chaque neurone, en partant de la couche finale vers la couche initiale. Considérons un réseau de neurones de type PMC, on note $W^{(i-1,i)}$ la matrice des poids entre la couche $i-1$ et la couche i, et $y^{(i)}$ la sortie de la couche i qui vaut :

$$y^{(i)} = \sigma(p^{(i)}) \quad \text{où} \quad p^{(i)} = W^{(i-1,i)}.y^{(i-1)} \tag{2.2}$$

où σ est la fonction d'activation.

On définit une fonction de coût, appelée aussi fonction de perte[6] E qui calcule la différence entre les sorties calculées par le réseau à la dernière couche, s, et les sorties désirées, d. Il y a de nombreuses fonctions de perte comme l'erreur quadratique moyenne (MSE[7]) ou l'entropie croisée. La MSE est très utilisée, elle est calculée par la formule suivante :

$$E = \frac{1}{N}\sum_{n=1}^{N}(s_n - d_n)^2 \tag{2.3}$$

où N est le nombre de neurones dans la couche de sortie (la dernière couche du réseau). En utilisant une fonction de transfert continue et dérivable, on calcule la dérivée partielle de la fonction de perte E par rapport à un poids $\omega_{kl}^{(i-1,i)}$ du réseau entre le neurone k de la couche $i-1$ et le neurone l de la couche i :

$$\frac{\partial E}{\partial \omega_{kl}^{(i-1,i)}} = \frac{\partial E}{\partial y_l^{(i)}}.\frac{\partial y_l^{(i)}}{\partial p_l^{(i)}}.\frac{\partial p_l^{(i)}}{\partial \omega_{kl}^{(i-1,i)}} \tag{2.4}$$

Soit :

$$\frac{\partial E}{\partial \omega_{kl}^{(i-1,i)}} = \frac{\partial E}{\partial y_l^{(i)}}.\sigma'(p_l^{(i)}).y_k^{(i-1)} \tag{2.5}$$

5. *Back-propagation algorithm.*
6. *Loss function.*
7. MSE : *Mean Squared Error.*

La valeur $\frac{\partial E}{\partial y_l^{(i)}}$ est calculée de la façon suivante :

$$\frac{\partial E}{\partial y_l^{(i)}} = \begin{cases} \delta_l = \frac{\partial E}{\partial s_l} = 2(s_l - d_l) & \text{si i est la couche de sortie.} \\ \delta_l^{(i)} = \sum_m \frac{\partial E}{\partial y_m^{(i+1)}} . \frac{\partial y_m^{(i+1)}}{\partial p_m^{(i+1)}} . \frac{\partial p_m^{(i+1)}}{\partial y_l^{(i)}} & \\ = \sum_m \delta_m^{(i+1)} . \sigma'(p_m^{(i+1)}) . \omega_{lm}^{(i,i+1)} & \text{si i est une couche cachée.} \end{cases} \tag{2.6}$$

Cette équation est une formule récursive permettant de calculer de façon itérative le gradient de l'erreur à la couche i à partir du gradient à la couche $i+1$, d'où le nom de rétro-propagation du gradient. L'erreur entre la sortie désirée et la sortie calculée est minimisée en mettant à jour les poids avec la formule suivante :

$$\Delta W^{(i-1,i)} = -\eta \frac{\partial E}{\partial W^{(i-1,i)}} \tag{2.7}$$

où η représente le taux d'apprentissage [8] ; c'est un nombre positif qui contrôle la variabilité des poids durant l'entraînement. Le choix de la valeur du taux d'apprentissage est crucial (LeCun et al. 2012) car une petite valeur équivaut à des variations faibles qui peuvent garantir une certaine stabilité, mais rendent l'entraînement du réseau lent, tandis qu'une valeur relativement élevée peut amener à un entraînement plus rapide mais peut mener aussi à un processus d'entraînement instable. C'est pourquoi, il existe certaines méthodes comme AdaDelta [9] (Bergstra et Bengio 2012 ; Schaul, Zhang et LeCun 2013 ; Zeiler 2012) qui font varier la valeur de η de façon optimale au cours de l'entraînement.

L'algorithme de rétro-propagation du gradient comporte des inconvénients notables (LeCun et al. 2012), en particulier pour les réseaux multicouche. La courbe de la fonction de perte correspondante est typiquement non-quadratique, non-convexe et de grande dimension avec de nombreux minima locaux et/ou des régions plates. Ainsi, une fois les gradients sont nuls, le réseau sera bloqué dans un minimum qui peut être un minimum local. Il n'y a aucune formule qui garantit que le réseau converge vers une bonne solution, que la convergence soit rapide ou qu'elle soit atteinte. Pour rendre l'optimisation plus performante, il existe d'autres variantes améliorées de la méthode de descente de gradient, qui tentent de trouver une bonne solution tout en diminuant le temps de convergence (Ruder

8. *Learning rate.*
9. *Adaptive learning rate.*

2016). Nous citons notamment :

— **L'apprentissage stochastique**[10] : pour mettre à jour les poids du réseau, on calcule à chaque itération le gradient pour l'erreur réalisée sur un seul exemple d'entraînement choisi (par exemple aléatoirement) à partir de la base.
— **L'apprentissage par lot**[11] : pour mettre à jour les poids, le gradient est calculé pour l'erreur réalisée sur toute la base d'apprentissage. Il traite tous les exemples de la base d'apprentissage simultanément en une seule passe (c'est-à-dire dans un grand lot).
— **L'apprentissage par mini-lots**[12] : cette variante est la plus utilisée car il s'agit d'un compromis entre les deux méthodes précédentes. Au lieu de calculer le gradient de tous les exemples ou le gradient d'un seul exemple, la méthode calcule à chaque itération le gradient sur plusieurs exemples considérés comme un mini-lot. Les données d'entraînement sont donc traitées en plusieurs passes (autrement dit en mini-lots).

Il existe une autre façon d'améliorer l'algorithme de rétro-propagation qui est la méthode du momentum, qui consiste à ajouter les gradients successifs calculés à chaque itération. L'équation (2.7) devient :

$$\Delta W^{(i-1,i)}(t) = -\eta \frac{\partial E_t}{\partial W^{(i-1,i)}} + \alpha \Delta W^{(i-1,i)}(t-1) \tag{2.8}$$

où α est le momentum. Cette méthode accélère la convergence et stabilise la trajectoire du gradient lorsqu'elle a tendance à osciller en ralentissant les changements de direction.

2.2.4.2 Sous- et sur-apprentissage

L'objectif d'un réseau de neurones est de trouver une fonction de prédiction /approximation F entre des exemples x_i et leurs cibles y_i tel que $y_i \approx F(x_i)$. L'entraînement du réseau a pour but de généraliser la fonction de prédiction sur de nouvelles données. Pour que la prédiction soit la plus proche possible de la réalité, le réseau apprend sur des données étiquetées appelées données d'apprentissage. En effet, durant l'apprentissage, la fonction

10. *Stochastic learning.*
11. *Batch learning.*
12. *Mini-batch learning.*

F capte toutes les propriétés et corrélations présentes dans la base d'entraînement. Pour cela, les exemples de cette base doivent être nombreux et variés pour mieux représenter le problème, et pour que la fonction de prédiction soit mieux généralisable sur des données non vues dans la base d'apprentissage.

Le sous-apprentissage [13] (Jabbar et Khan 2015) désigne le fait que le modèle entraîné s'adapte mal sur les exemples d'entraînement. Autrement dit, le modèle entraîné n'arrive même pas à capturer les corrélations présentes dans la base d'entraînement. Par conséquent, l'erreur calculée par la fonction de perte en phase d'apprentissage reste grand. Bien évidemment, le modèle entraîné ne se généralisera pas bien non plus sur les données qu'il n'a pas vues lors de sa phase d'entraînement. Finalement, le modèle ne sera viable car les erreurs de prédictions seront grandes.

Le sur-apprentissage [14] (Hawkins 2004) désigne le fait que le modèle entraîné se spécialise trop sur les exemples d'entraînement, mais se généralise mal sur de nouvelles données. En d'autres termes, le modèle donne de très bonnes prédictions sur les données d'entraînement, mais il prédit mal sur les données qu'il n'a pas vues lors de sa phase d'entraînement.

Différentes techniques ont été proposées pour lutter contre le sur-apprentissage. On cite en particulier :

— l'arrêt prématuré [15] (Caruana, Lawrence et Giles 2001 ; Sarle 1996) qui consiste à utiliser deux partitions de données. En plus de la partition des données d'entraînement, on crée une partition de données de validation sur laquelle l'erreur de prédiction est calculée. L'entraînement est alors arrêté dès que l'erreur de validation commence à accroître ;
— la régulation des poids [16] (Bos et Chug 1996) qui consiste à pénaliser l'erreur de la fonction de perte en ajoutant un terme de régularisation $\lambda \sum_i \omega_i$ qui conduit à minimiser les valeurs des poids ;
— l'injection de bruit (Zur et al. 2009) qui consiste à ajouter du bruit aléatoire dans les vecteurs d'entrée.

13. *Underfitting.*
14. *Overfitting.*
15. *Early stopping.*
16. *Weight decay.*

2.3 Apprentissage profond

L'apprentissage profond ou l'apprentissage en profondeur (Goodfellow et al. 2016) est un sous-ensemble de l'apprentissage automatique, qui permet de transformer les données brutes en une représentation abstraite via des architectures hiérarchiques composées de plusieurs couches de représentation. Il a connu un grand succès ces dernières années par ses nombreuses applications dans les domaines de l'intelligence artificielle comme, par exemple, le traitement de langue et la vision par ordinateur. Ce succès est principalement lié à la disponibilité d'unités de calcul performantes à faible coût, à l'apparition de grandes bases de données annotées et aux avancées sur les algorithmes d'apprentissage automatique.

2.3.1 Histoire de l'apprentissage profond

Le concept d'apprentissage profond remonte à l'initiation des réseaux de neurones artificiels (Fukushima 1980 ; Fukushima, Miyake et Ito 1983). Théoriquement, il a commencé en 1980 lorsque (Fukushima 1980) a proposé le modèle Neocognitron. (LeCun et al. 1989) ont suggéré une solution pour la reconnaissance de l'écriture manuscrite en appliquant l'approche de la rétro-propagation à un réseau neuronal profond (DNN [17]). Cependant, il était pratiquement difficile de l'utiliser en raison de son énorme temps d'entraînement. Dans les deux décennies qui ont suivi, de nombreux travaux ont été menés pour résoudre ce problème de temps d'entraînement. En 2006 et 2007, des recherches prometteuses ont été réalisées par (Hinton, Osindero et Teh 2006) et (Hinton 2007). Ils ont entraîné des réseaux de croyance profonds multicouche en pré-entraînant une seule couche à la fois en tant que machine de Boltzmann restreinte non supervisée. Ensuite, ils ont utilisé la rétro-propagation supervisée pour un raffinement supplémentaire. En 2012, (Krizhevsky, Sutskever et Hinton 2012) ont remis au goût du jour la technologie d'apprentissage profond en gagnant la compétition de reconnaissance visuelle à grande échelle Imagenet ILSVRC [18] (Deng et al. 2009 ; Russakovsky et al. 2015). Les architectures profondes ont été successivement améliorées par la suite, notamment dans les travaux de (He et al. 2016 ; Simonyan et Zisserman 2014b ; Szegedy et al. 2015).

En 2015, "*AlphaGo*", un programme qui a appris par l'apprentissage profond à jouer au

17. DNN : *Deep Neural Network*.
18. ILSVRC : *Imagenet Large Scale Visual Recognition Challenge*.

jeu de Go bat le champion européen *Fan Hui* par 5 parties à 0. En 2016, il bat le champion du monde *Lee Sedol* par 4 parties à 1.

La révolution de l'apprentissage profond est principalement liée à l'amélioration de la puissance de traitement des ordinateurs, à l'apparition de nouvelles bases de données suffisamment grandes et riches capables d'entraîner des systèmes de grandes tailles, et aux grands progrès dans les méthodes d'optimisation.

Les architectures de l'apprentissage profond reposent sur la représentation hiérarchique des données. Elles extraient automatiquement des caractéristiques des données brutes. En terme d'analyse d'image, les niveaux de hiérarchie correspondent à la chaîne suivante "pixel → contours → combinaisons de contours" (Abbas, Ibrahim et Jaffar 2019). Une architecture profonde se compose de nombreuses couches et d'un grand nombre de neurones par couche. Elle permet de transformer les entrées en une représentation abstraite.

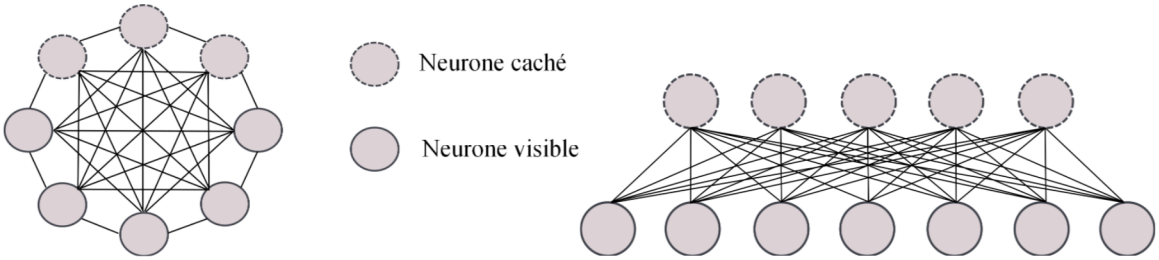
2.3.2 Techniques d'apprentissage profond

Récemment, l'apprentissage profond est intensivement étudié dans les domaines de la vision par ordinateur, différentes approches sont ainsi apparues. Elles peuvent être regroupées en trois catégories : machines de Boltzmann restreintes, auto-encodeurs et réseaux de neurones convolutifs. Dans cette section nous décrivons brièvement les deux premières catégories, et nous nous intéressons plus en détail aux réseaux de neurones convolutifs dans la section 2.4.

2.3.2.1 Machine de Boltzmann Restreinte

Une machine de Boltzmann restreinte (RBM[19]) est un réseau neuronal stochastique génératif, inventé d'abord sous le nom Harmonium par (Rumelhart et al. 1986). Une RBM est une variante de la machine de Boltzmann standard, proposée par (Hinton et Sejnowski 1986), avec la restriction qu'il n'y ait pas d'interconnexions entre les unités d'une même couche. Une RBM est composée de deux couches de neurones. La première couche contient les unités visibles et la seconde contient les unités cachées. La figure 2.4 montre la différence entre les deux modèles.

19. RBM : *Restricted Boltzmann Machine*.



(a) machine de Boltzmann (b) machine de Boltzmann restreinte

FIGURE 2.4 – Architecture d'une machine de Boltzmann (a) et d'une machine de Boltzmann restreinte (b).

En revanche, la machine de Boltzmann standard peut avoir des connexions entre des unités cachées. Elle a d'excellente capacité de représentation et de génération, mais la complexité de son entraînement est exponentiellement proportionnelle au nombre d'unités. La restriction de RBM permet d'avoir des algorithmes d'entraînement plus efficaces, en particulier l'algorithme de divergence contrastive (CARREIRA-PERPINAN et HINTON 2005). Les RBMs sont utilisées pour la réduction de dimensionnalité (TRAN, PHUNG et VENKATESH 2011), le filtrage collaboratif (SALAKHUTDINOV, MNIH et HINTON 2007) et la classification (LAROCHELLE et BENGIO 2008 ; TEH et HINTON 2001).

En utilisant les RBMs comme modules d'apprentissage, de nouvelles architectures profondes sont introduites, notamment le réseau de croyance profond (DBN [20]), la machine de Boltzmann profonde (DBM [21]) et le modèle d'énergie profond (DEM [22]). Ces trois architectures sont représentées dans la figure 2.5.

Les trois architectures sont composées d'une couche d'unités visibles et de plusieurs couches cachées. Dans le DBN, les connexions reliant les deux dernières couches cachées sont symétriques et celles des autres couches sont dirigées. Les connexions dans les couches de DBM sont toutes symétriques. Finalement, le DEM contient des unités cachées stochastiques dans la dernière couche cachée et des unités déterministes dans les autres couches.

i. Réseau de croyance profond

Le réseau de croyance profond DBN, introduit par (HINTON, OSINDERO et TEH 2006), a constitué une avancée significative dans l'apprentissage profond. C'est un modèle génératif

20. DBN : *Deep Belief Network*.
21. DBM : *Deep Boltzmann Machine*.
22. DEM : *Deep Energy Model*.

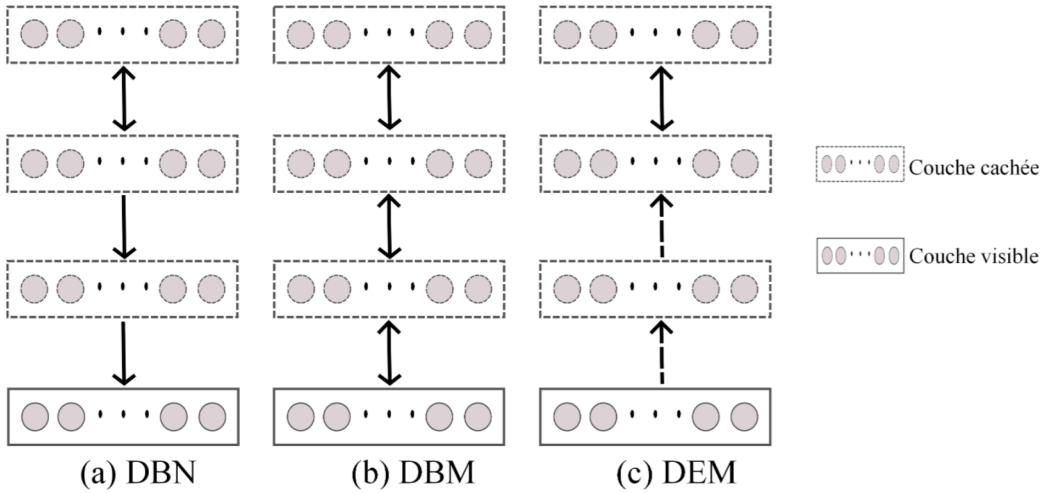


FIGURE 2.5 – Architectures profondes utilisant les RBMs. a) : Réseau de croyance profond, b) : Machine de Boltzmann profonde et c) : Modèle d'énergie profond. Les flèches représentent les connexions dirigées dans le modèle de réseau représente.

probabiliste composé de plusieurs couches de variables latentes stochastiques. Les couches cachées sont liées par des connexions dirigées à l'exception des deux dernières couches qui sont reliées par des connexions symétriques. L'entraînement du DBN s'effectue couche par couche où la projection de la couche précédente sert comme entrée de la couche suivante pour initialiser les poids du réseau ; ensuite, un apprentissage supervisé peut être appliqué pour effectuer la classification. Cette stratégie d'entraînement présente deux avantages principaux (AREL, ROSE et KARNOWSKI 2010) : 1) elle génère une initialisation correcte du réseau, ce qui aborde la difficulté de sélection des paramètres qui peut produire des optima locaux médiocres, 2) l'entraînement est non supervisé, il ne demande pas de données annotées ; pour une application spécifique, le DBN peut s'entraîner avec des données annotées limitées. Cependant, la création d'un modèle DBN est une tâche coûteuse car elle implique l'entraînement de plusieurs RBMs (BENGIO, COURVILLE et VINCENT 2013). Les DBNs sont utilisés dans différentes applications comme le traitement d'image (LEE et al. 2009), le traitement de la parole (MOHAMED, DAHL et HINTON 2009 ; SAINATH et al. 2011) et la compréhension du langage (SARIKAYA, HINTON et DEORAS 2014).

Les DBNs ont attiré l'attention des chercheurs sur l'apprentissage profond, et par conséquent, de nombreuses variantes ont été créées, nous citons notamment les DBNs parcimonieux (LEE, EKANADHAM et NG 2008) et les DBNs convolutifs (HUANG, LEE et LEARNED-MILLER 2012 ; LEE et al. 2009, 2011).

ii. Machine de Boltzmann profonde

La machine de Boltzmann profonde DBM, proposée par (Salakhutdinov et Hinton 2009), est un réseau de neurones profond génératif stochastique où toutes les connexions sont symétriques et non dirigées. Cette symétrie permet au DBM de modéliser et d'utiliser l'information de couches inférieures pour déterminer des représentations des couches supérieures plus robustes. L'entraînement des couches cachées se fait également une par une via des RBMs. Elle a été appliquée dans de nombreuses applications, on cite par exemple la reconnaissance d'expression faciale (He et al. 2013) et le traitement de données multimodale (Srivastava et Salakhutdinov 2012).

Il existe également des approches qui visent à améliorer l'efficacité des DBMs. Ces améliorations peuvent avoir lieu soit au niveau du pré-entraînement des couches cachées (Cho et al. 2013 ; Hinton et Salakhutdinov 2012), soit au niveau de l'entraînement du réseau entier (Goodfellow, Courville et Bengio 2013 ; Montavon et Müller 2012).

iii. Modèle d'énergie profond

Le modèle d'énergie profond DEM, présenté par (Ngiam et al. 2011), est une approche plus récente pour entraîner des architectures profondes. Contrairement aux DBN et DBM qui contiennent plusieurs couches cachées stochastiques, le DEM n'a qu'une seule couche cachée stochastique pour un entraînement plus efficace.

Le modèle utilise des réseaux de neurones profonds à propagation avant et est capable d'entraîner toutes les couches simultanément. Les différentes évaluations de ce modèle sur des images naturelles ont démontré que l'entraînement simultané de plusieurs couches donne de meilleurs performances par rapport à l'entraînement couche par couche. (Ngiam et al. 2011) ont utilisé la méthode hybride de Monte Carlo pour entraîner le modèle. Il existe également d'autres méthodes, notamment la divergence contrastive et la correspondance de score.

2.3.2.2 Auto-encodeur

Un auto-encodeur [23] est un réseau de neurones artificiel utilisé souvent dans l'apprentissage des caractéristiques discriminantes. Il est entraîné pour reconstruire son entrée. L'auto-encodeur est constitué de deux parties : l'encodeur et le décodeur comme illustré sur la figure 2.6.

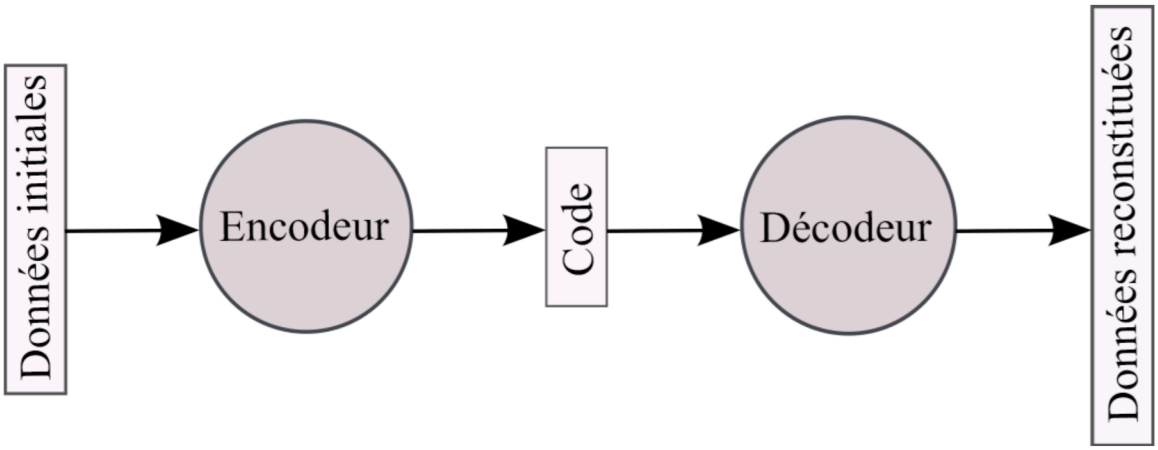


FIGURE 2.6 – Schéma de principe d'un auto-encodeur.

L'encodeur permet de transférer les données d'entrée dans un espace de caractéristiques (ayant une dimension inférieure à celui de l'espace d'entrée) afin de fournir de nouvelles représentations dites "encodées". A son tour, le décodeur reconstruit à partir de ces représentations les données initiales. Ensuite, le modèle calcule l'erreur de reconstruction entre les données reconstruites et les données initiales. Durant l'apprentissage, l'auto-encodeur ajuste ses paramètres afin de minimiser l'erreur sur les différents exemples de la base de données.

En général, un seul auto-encodeur n'est pas en mesure d'obtenir des caractéristiques discriminantes et représentatives des données d'entrée. Un auto-encodeur profond est ainsi apparu constitué de plusieurs auto-encodeurs enchaînés. Les encodées apprises par un auto-encodeur sont transmises comme entrée vers l'auto-encodeur suivant. Il a été proposé pour la première fois par (HINTON et SALAKHUTDINOV 2006), et il est encore largement étudié dans des travaux récents (JIANG et al. 2013 ; ZHANG et al. 2014 ; ZHOU et al. 2014). Un auto-encodeur profond est bien souvent entraîné en utilisant une variante de la rétro-

23. *Autoencoder* en anglais.

propagation, par exemple la méthode du gradient conjugué. Bien que cela fonctionne de manière raisonnablement efficace, ce modèle pourrait devenir inefficace si des erreurs sont présentes dans les premières couches. Cela signifie que le réseau apprendra presque toujours à reconstituer la moyenne des données d'entraînement. Une approche appropriée pour résoudre ce problème consiste à pré-entraîner le réseau avec des poids initiaux proches de la solution finale (Hinton et Salakhutdinov 2006). Il existe différentes variantes d'auto-encodeur pour améliorer sa capacité à capturer des informations importantes et à apprendre des représentations plus riches. Nous présentons brièvement trois variantes d'auto-encodeur : auto-encodeur parcimonieux, auto-encodeur débruiteur et auto-encodeur contractif.

i. Auto-encodeur parcimonieux

Un auto-encodeur parcimonieux (SAE[24]) vise à extraire des caractéristiques parcimonieuses à partir de données brutes. La parcimonie peut être obtenue soit en pénalisant les biais des unités cachées (Goodfellow et al. 2009 ; Lee, Ekanadham et Ng 2008 ; Ranzato et al. 2007), soit en pénalisant directement la sortie des activations des unités cachées (Le et al. 2011 ; Zou, Ng et Yu 2011).

ii. Auto-encodeur débruiteur

Un auto-encodeur débruiteur (DAE[25]) introduit par (Vincent et al. 2008, 2010) est un auto-encodeur classique dans lequel on vient dégrader artificiellement l'entrée par un bruit additif. Le modèle apprend à reconstituer l'entrée originale sans bruit. Durant l'apprentissage, la rétro-propagation calcule l'erreur entre la sortie débruitée calculée par le réseau et l'entrée originale sans bruit. Cette technique permet d'améliorer la robustesse du modèle. La figure 2.7 montre le processus DAE.

iii. Auto-encodeur contractif

Un auto-encodeur contractif (CAE[26]), proposé par (Rifai et al. 2011), introduit un régularisateur explicite dans la fonction d'erreur de reconstruction forçant le modèle à apprendre une fonction robuste aux légères variations des valeurs d'entrée. En effet, CAE et DAE ont une motivation similaire d'améliorer la robustesse des représentations (Bengio, Courville et Vincent 2013). Alors qu'un DAE rend le modèle robuste en ajoutant du

24. SAE : *Sparse autoencoder*.
25. DAE : *Denoising AutoEncoder*.
26. CAE : *Contractive AutoEncoder*.

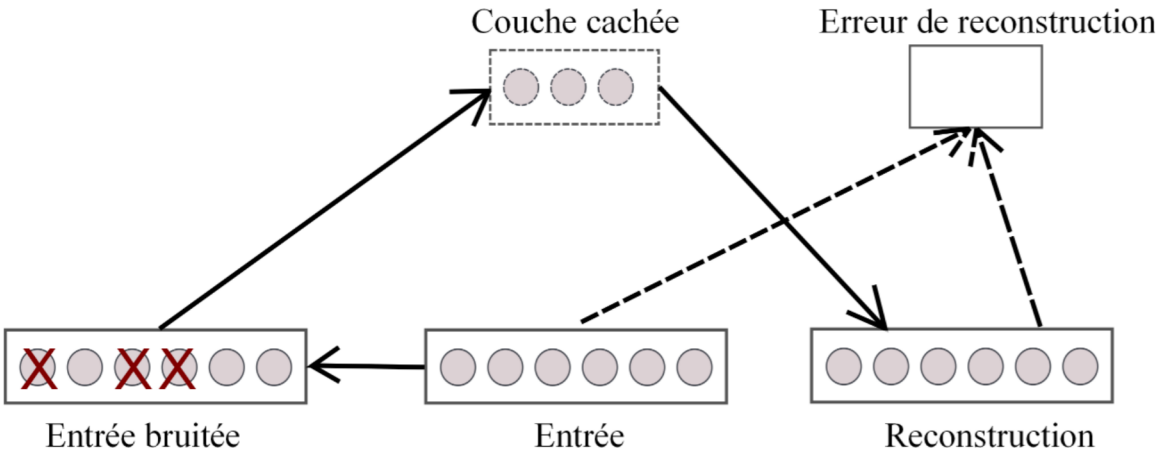


FIGURE 2.7 – Schéma de principe d'un auto-encodeur débruiteur.

bruit dans l'ensemble d'apprentissage, un CAE atteint la robustesse en introduisant une pénalité dans la fonction objective.

2.4 Réseau de neurones convolutif

Le réseau de neurones convolutif ou le réseau de neurones à convolution (CNN [27]) (LECUN et al. 1998) est l'une des approches d'apprentissage automatique la plus populaire. Ce réseau contient plusieurs couches entrainables de manière robuste, il est utilisé pour traiter des données de dimensions multiple, comme des images 2D ou 3D. Les CNNs ont connu d'énormes succès dans les applications pratiques.

Le terme "convolutif" indique que le réseau emploie l'opération linéaire de convolution. Les réseaux convolutifs sont simplement des réseaux de neurones qui utilisent la convolution à la place de la multiplication matricielle générale dans au moins une de leurs couches. L'architecture générale d'un CNN est illustrée dans la figure 2.8.

Un CNN est constitué de deux parties bien distinctes. La première partie est la partie convolutive. Elle fonctionne comme un extracteur de caractéristiques des images d'entrée. Généralement, cette partie contient trois types de couches qui sont : les couches de convolution, les couches de correction et les couches de sous-échantillonnage. Ensuite, les

27. CNN : *Convolutional Neural Network*.

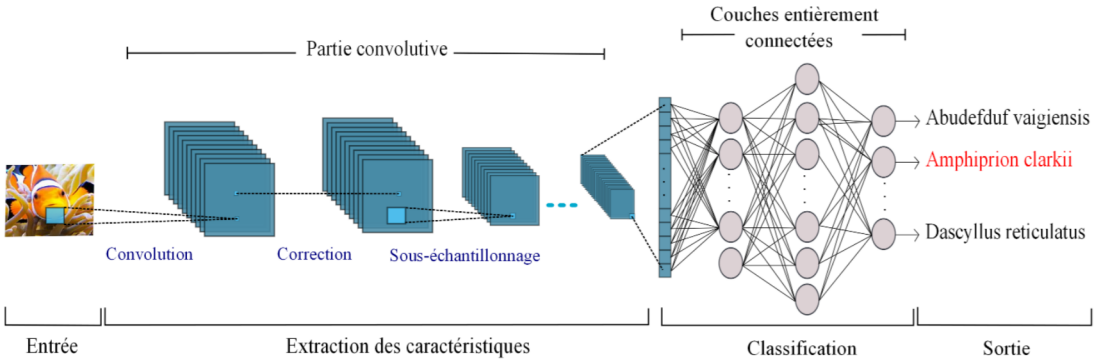


FIGURE 2.8 – Architecture générale d'un réseau de neurones convolutif.

caractéristiques extraites alimentent une deuxième partie, constituée de couches entièrement connectées. Le rôle de cette partie est de classer les images d'entrée. Pour entrainer un CNN, il y a deux phases : phase avant et phase arrière. La phase avant [28] permet de représenter l'image d'entrée avec les paramètres courants (poids et biais) dans chaque couche, puis, de calculer la fonction de coût entre la sortie calculée et la sortie désirée. La phase arrière [29] se base sur la fonction de coût et calcule le gradient de chaque paramètre. Ensuite, tous les paramètres sont mis à jour et préparés pour la phase avant suivante (avec une nouvelle image d'entrée). Après un certain nombre d'itérations des deux phases, l'apprentissage peut s'arrêter.

L'avantage d'un CNN est l'utilisation d'un poids unique associé à tous les neurones d'un même noyau de convolution. Ceci permet de réduire l'espace mémoire et améliore les performances. C'est un avantage majeur par rapport au perceptron multicouche qui considère chaque neurone indépendant et donne un poids différent à chaque signal entrant.

L'inconvénient d'un CNN est qu'il demande une grande quantité de mémoire allouée pour effectuer des calculs nécessaires ainsi que pour sauvegarder les cartes de caractéristiques générées. En outre, il nécessite de grandes bases d'entraînement pour garantir des performances élevées lors de la phase de test.

28. *Forward pass.*
29. *Backward pass.*

2.4.1 Types de couches

Un CNN est un réseau de neurones hiérarchique dont les couches de convolution alternent avec les couches de correction et de sous-échantillonnage, toutes sont suivies par quelques couches entièrement connectées.

2.4.1.1 Couches de convolution

Dans les couches de convolution, un CNN utilise différents noyaux et calcule la convolution entre chaque noyau et l'image d'entrée ou avec les cartes de caractéristiques intermédiaires pour générer des cartes de caractéristiques comme illustré sur la figure 2.9.

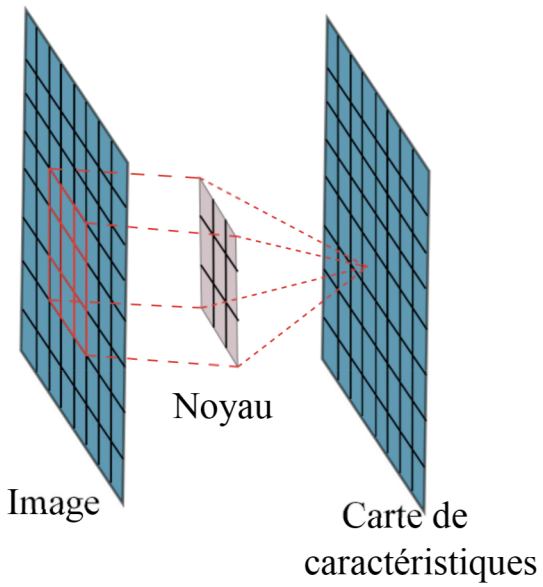


FIGURE 2.9 – Couche de convolution et carte de caractéristiques résultante.

L'opération de convolution a trois avantages principaux (ZEILER 2013) :

1. le partage du poids sur la même carte de caractéristiques réduit le nombre de paramètres du réseau ;
2. la connectivité locale apprend les corrélations entre les pixels voisins ;
3. la convolution permet d'avoir la propriété d'invariance du traitement par translation.

2.4.1.2 Couches de correction ou couches non-linéaires

Ces couches viennent après les couches de convolution afin d'améliorer l'efficacité du traitement entre les couches. La couche de correction va opérer une fonction de transformation non-linéaire sur les cartes de caractéristiques. Pour modéliser une sortie d'un neurone, on utilise généralement une fonction d'activation σ avec $\sigma(x) = tanh(x)$ ou $\sigma(x) = sigmoïde(x)$. Dans les CNNs, ces fonctions sont devenues plus lentes en termes de temps d'apprentissage avec la descente de gradient. D'autres fonctions sont apparues pour corriger ce problème, nous citons notamment la fonction unité de rectification linéaire (ReLU [30]) où $\sigma(x) = max(0; x)$. Les CNNs avec ReLU apprennent plus rapidement que leurs équivalents avec les fonctions d'activation *tanh* ou *sigmoïde*.

2.4.1.3 Couches de sous-échantillonnage

Le sous-échantillonnage [31] est utilisé pour réduire les tailles des cartes de caractéristiques en réduisant ainsi l'espace mémoire alloué et les calculs dans le réseau. La couche de sous-échantillonnage remplace la carte à une certaine position par un résumé statistique des valeurs du voisinage de cette position. Il existe plusieurs opérateurs de sous-échantillonnage, mais les deux fonctions intensité moyenne [32] et intensité maximale [33] sont les plus utilisées. La figure 2.10 montre un exemple de la fonction intensité maximale. Pour une carte de caractéristiques de 8×8, la sortie est réduite en 4×4 avec un opérateur de l'intensité maximale de 2×2 et un pas de 2.

(Boureau, Ponce et LeCun 2010) fournissent une analyse détaillée des performances des deux fonctions intensité maximale et intensité moyenne. (Scherer, Müller et Behnke 2010) ont comparé les deux opérateurs de sous-échantillonnage et ils ont trouvé que l'intensité maximale est plus efficace et peut conduire à une convergence plus rapide.

30. ReLU : *Rectified Linear Unit.*
31. *Pooling ou down sampling.*
32. *Average pooling.*
33. *Max pooling.*

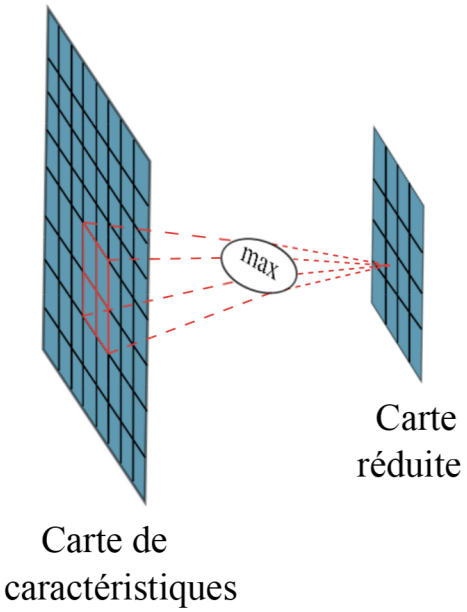


FIGURE 2.10 – L'opération de réduction de la couche de sous-échantillonnage.

2.4.1.4 Couches entièrement connectées

Ces couches viennent après plusieurs couches de convolution et de sous-échantillonnage, elles permettent de convertir les cartes de caractéristiques 2D en un vecteur de sortie 1D comme illustré sur la figure 2.11. Les couches entièrement connectées [34] fonctionnent comme un réseau de neurones traditionnel et elles contiennent 90% des paramètres d'un CNN. Elles nous permettent d'alimenter le réseau de neurones lors de la phase d'entraînement par un vecteur avec une longueur prédéfinie. Après, nous pouvons utiliser ce vecteur pour des tâches différentes telles que la classification des images. Ainsi, la couche finale possède un neurone par classe. La sortie de chacun de ces neurones utilise une fonction d'activation pour représenter la probabilité d'appartenance à la classe correspondante. L'inconvénient de cette couche est qu'elle contient plusieurs paramètres, ce qui se traduit par un effort de calcul important durant l'apprentissage.

34. *Fully connected layers.*

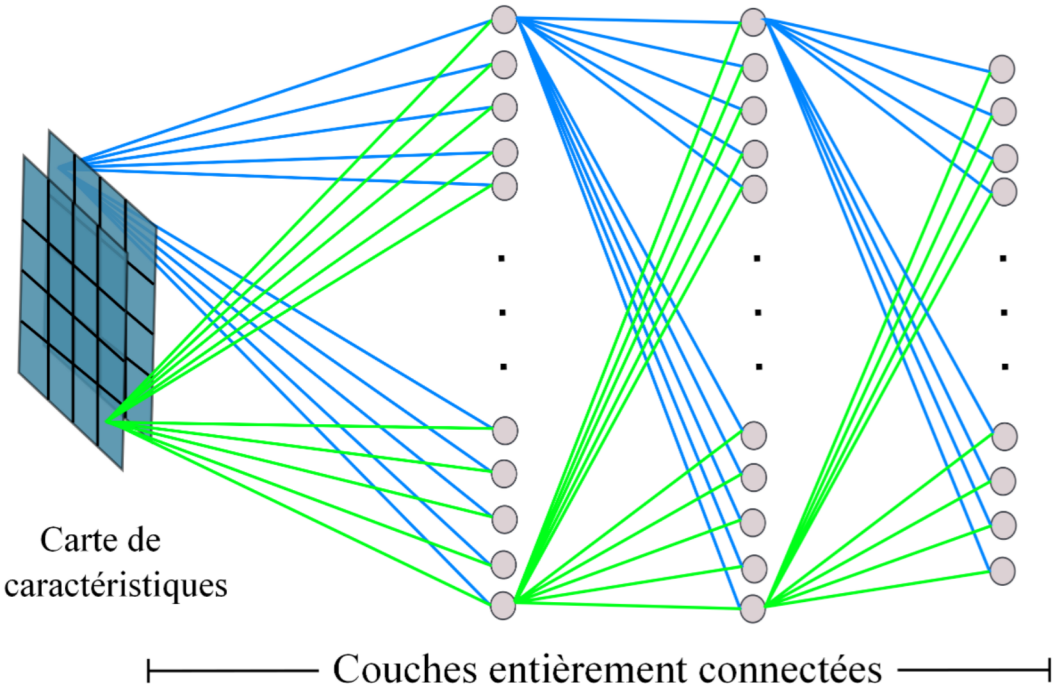


FIGURE 2.11 – L'opération des couches entièrement connectées.

2.4.2 Les architectures CNN

Nous présentons dans cette section les architectures CNN les plus couramment utilisées dans le domaine de la vision par ordinateur. Ces architectures sont devenues de plus en plus profondes avec les années et parallèlement plus performantes comme illustré dans la figure 2.12. Cette efficacité vient du développement récent des unités de traitement graphique GPU [35] qui effectuent des calculs parallèles réduisant ainsi le temps requis pour l'entraînement d'une architecture profonde.

2.4.2.1 Les architectures classiques

Une architecture classique a une structure sérielle des blocs où chaque bloc est constitué de couches de convolution, de correction et de sous-échantillonnage.

LeNet-5 : est la première architecture de CNN introduite par (LECUN et al. 1998) pour la reconnaissance de chiffres manuscrits. Elle consiste simplement en deux blocs suivis

35. *Graphical Processing Units.*

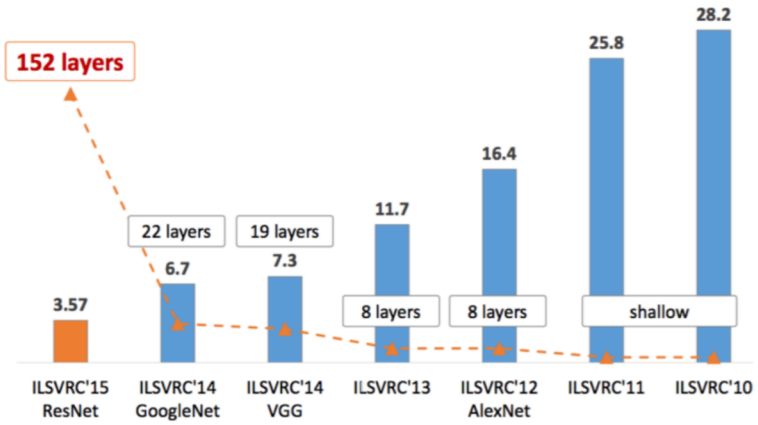


FIGURE 2.12 – Taux d'erreur (en %) de différentes architectures CNN sur la base ImageNet dans les compétitions ILSVRC de classification d'objets (entre 2010 et 2015).

de trois couches entièrement connectées (figure 2.13). L'entrée du réseau est une image en niveaux de gris de dimension 32×32. Les noyaux de filtres convolutifs utilisés sont de taille 5×5. La première couche de convolution génère 6 cartes de caractéristiques qui sont, ensuite, sous-échantillonnées avec un pas de 2. De la même manière, le deuxième bloc produit 16 cartes donnant après sous-échantillonnage des cartes de taille 5×5. Les couches entièrement connectées transforment les cartes en un vecteur et produisent successivement des vecteurs de taille 120, 84 et finalement 10, ce dernier correspondant aux 10 chiffres possibles.

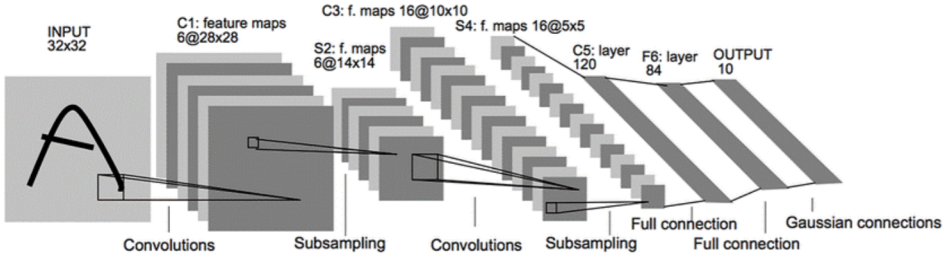


FIGURE 2.13 – Architecture LeNet-5 (LECUN et al. 1998).

AlexNet : ce CNN a été développé par (KRIZHEVSKY, SUTSKEVER et HINTON 2012). Il a permis de reprendre les études des réseaux de neurones convolutifs grâce à sa victoire lors de la compétition ILSVRC de classification d'images ImageNet. AlexNet comporte 8 couches entraînables, les cinq premières sont des couches de convolution et les trois dernières sont des couches entièrement connectées (figure 2.14). Il contient également trois couches de

sous-échantillonnage respectivement après la première, la deuxième et la dernière couche de convolution. La couche de correction utilise la fonction ReLU après chaque couche de convolution. Le nombre de noyaux et leur taille dans les cinq couches de convolution sont 96 noyaux de taille $11 \times 11 \times 3$, 256 noyaux de taille $5 \times 5 \times 48$, 384 noyaux de taille $3 \times 3 \times 128$, 384 noyaux de taille $3 \times 3 \times 192$ et 256 noyaux de taille $3 \times 3 \times 192$ respectivement. Les couches entièrement connectées possèdent 4096 neurones chacune. La dernière couche applique la fonction de normalisation exponentielle appelée "*Softmax*" et renvoie un vecteur de probabilités de taille 1000 correspondant au nombre de classes de la base ImageNet. La fonction "*Softmax*" est utilisée dans la classification multi-classe. Elle applique une certaine normalisation sur les valeurs de sortie de la couche de classification pour obtenir un vecteur de probabilités attribuant des probabilités d'appartenance de l'image d'entrée à chaque classe.

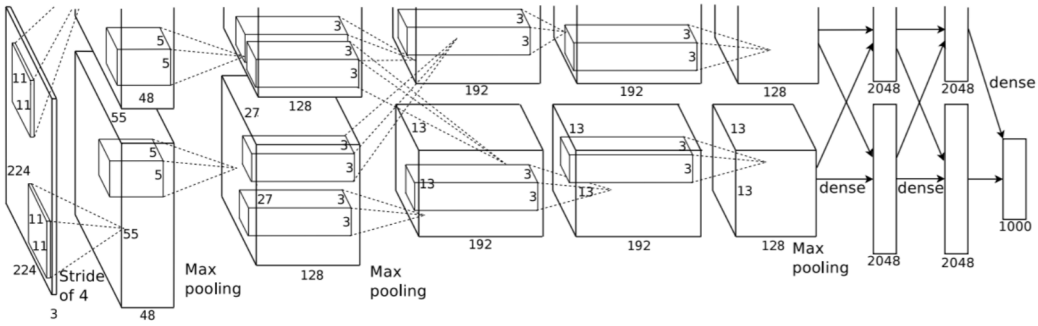


FIGURE 2.14 – Architecture d'AlexNet (KRIZHEVSKY, SUTSKEVER et HINTON 2012).

VGGNet : est une architecture de CNN proposée par (SIMONYAN et ZISSERMAN 2014b). Elle se base sur l'idée d'utiliser des séquences de convolution par bloc. VGG est constituée de plusieurs couches entraînables, 16 couches pour la version VGG-16 (figure 2.15) et 19 couches pour VGG-19. Chaque couche de convolution utilise des filtres convolutifs de taille 3×3 et a pour fonction d'activation une ReLU. Le nombre de filtres dans chaque bloc (conv1, conv2, conv3, conv4 et conv5) est 64, 128, 256, 512 et 512 respectivement. Elle contient également cinq couches de sous-échantillonnage placées à la fin de chaque bloc. Les deux premières couches entièrement connectées ont 4096 neurones chacune suivies par une couche ReLU. La dernière couche applique la fonction de normalisation exponentielle appelée "*Softmax*" et renvoie un vecteur de probabilités de taille 1000 correspondant au nombre de classes de la base ImageNet. L'inconvénient de l'architecture VGGNet est qu'elle demande énormément de mémoire à cause du nombre de ses paramètres (140 millions).

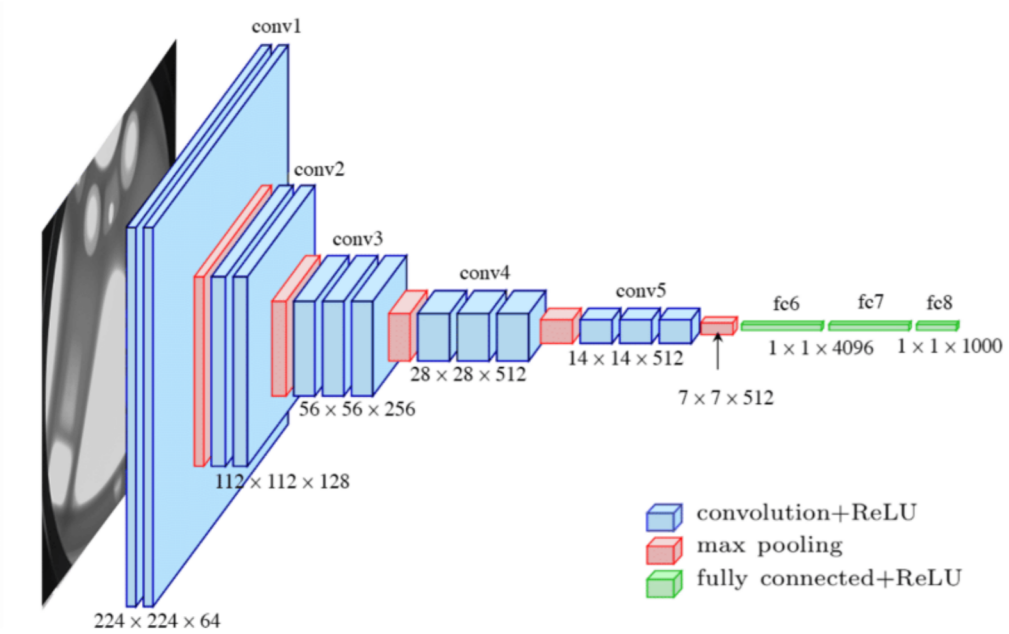


FIGURE 2.15 – Architecture de VGGNet-16 (FERGUSON et al. 2017).

2.4.2.2 Les macro-architectures

Avec le progrès des GPUs, de nouvelles architectures plus complexes et plus profondes sont apparues. Ces architectures sont composées de blocs d'opération dont les rôles sont prédéfinis.

GoogleNet : cette architecture de CNN a été introduite par (SZEGEDY et al. 2015). Sa principale contribution est l'introduction de module d'inception. Ce module (figure 2.16) effectue plusieurs convolutions exécutées en parallèle, chacune de taille différente 1×1, 3×3 ou 5×5 . GoogleNet utilise également des couches de sous-échantillonnage pour réduire la dimension des cartes de caractéristiques permettant ainsi de réaliser un gain important en temps de calcul et en espace mémoire. L'architecture finale (figure 2.17) est composée de 22 couches, mais le nombre de paramètres est réduit à 4 millions. D'autres modules d'inception ont par la suite été proposés notamment Inception V2 et V3 (SZEGEDY et al. 2016b), puis Inception V4 (SZEGEDY et al. 2016a).

ResNet : cette architecture a été proposée par (HE et al. 2016). Elle permet d'entraîner des réseaux très profonds (plus de 150 couches). Avec autant de couches le gradient

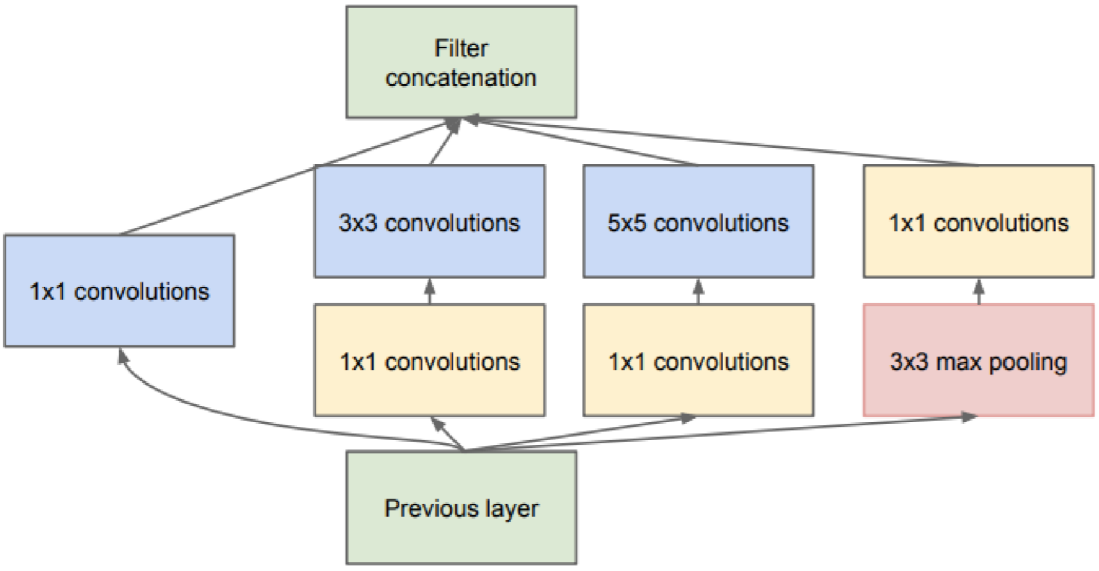


FIGURE 2.16 – Le module d'inception V1 de (SZEGEDY et al. 2015).

devient faible et ne se propage plus correctement dans les premières couches du réseau, ce qui impacte la mise à jour des paramètres. La contribution développée dans ResNet est l'introduction des connexions résiduelles pour contrecarrer ce phénomène. Une connexion résiduelle permet d'additionner l'entrée et la sortie de deux couches de convolution et de la transmettre à la couche suivante comme illustré dans la figure 2.18. Cette architecture permet de créer des réseaux très profonds de bien meilleures performances, car elle a la capacité d'extraire davantage d'information et d'avoir ainsi une analyse plus avancée des images. La figure 2.19 illustre l'architecture d'un réseau résiduel à 18 couches (OU et al. 2019).

Pour récapituler, nous montrons dans la table 2.1 un comparatif des paramètres clés des architectures de référence présentées ci-dessus. A noter que la création d'une architecture nouvelle requiert une certaine expérience, du matériel de calcul intensif et un grand jeu de données annotées.

FIGURE 2.17 – Architecture globale de GoogleNet (SZEGEDY et al. 2015). Les blocs bleus sont des convolutions, les rouges sont des opérations de sous-échantillonnage, verts sont des opérations de normalisation ou de concaténation et les jaunes sont des sorties de la fonction *"Softmax"*.

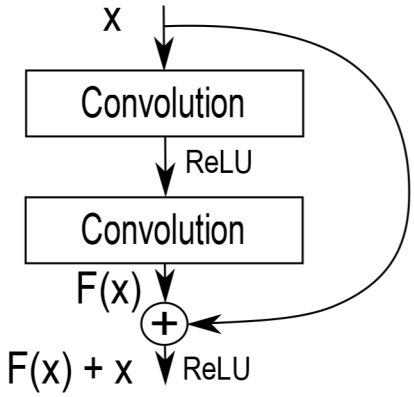


FIGURE 2.18 – Connexion résiduelle (HE et al. 2016).

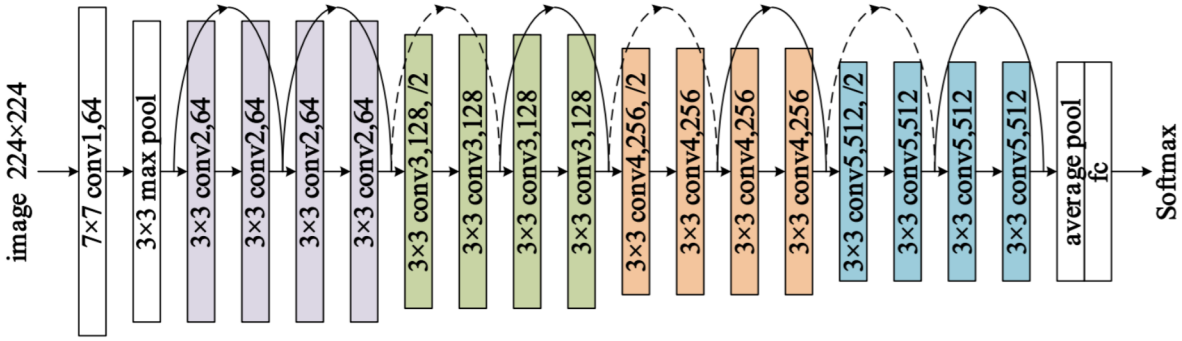


FIGURE 2.19 – Architecture du ResNet avec 18 couches (OU et al. 2019).

	LeNet-5	AlexNet	VGG16	GoogleNet	ResNet50
Année	1998	2012	2014	2014	2015
Top 5 erreur	-	15,30%	7,30%	6,67%	3,60%
Taille de filtres	5	11-5-3	3	7-5-3-1	7-3-1
Profondeur des filtres	1-16	3-256	3-512	3-1024	3-2048
Nombre de filtres par couche	6-16	96-384	64-512	64-384	64-2048
Nombre de couches convolutives	2	5	16	21	49
Nombre de couches FC	3	3	3	1	1
Nombre de paramètres	60K	61M	138M	7M	25M

TABLE 2.1 – Comparaison des architectures CNN de référence.

2.4.3 Stratégies d'entraînement

L'avantage de l'apprentissage profond par rapport à l'apprentissage superficiel[36] est qu'il peut apprendre des caractéristiques plus abstraites des données que lui sont fournies. Cependant, le nombre massif de paramètres peut conduire au sur-apprentissage souligné plus haut. En plus des techniques discutées dans la section 2.2.4.2, nous présentons ici d'autres techniques de régularisation proposées pour contrer ce problème et améliorer les performances de l'entraînement de CNNs.

2.4.3.1 Le dropout

Le dropout ou littéralement le décrochage a été proposé par (Hinton et al. 2012) et détaillé par (Baldi et Sadowski 2013). C'est une technique de régularisation très efficace pour entraîner des CNNs. Le dropout consiste, à chaque itération de la descente de gradient durant l'entraînement, à abandonner temporairement une partie aléatoire de neurones pour forcer le réseau à s'adapter à un manque d'informations et améliorer sa capacité de généralisation (figure 2.20). Les neurones abandonnés ne contribuent ni au calcul de la sortie ni à la rétro-propagation. Lors de la phase de test, tous les neurones du réseau sont utilisés (réactivation des neurones abandonnées).

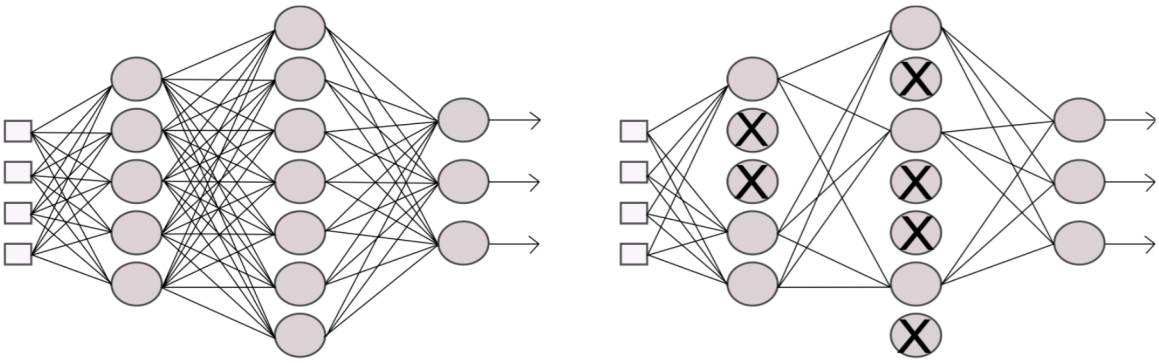


Figure 2.20 – Un exemple de dropout : A gauche : un réseau de neurones standard avec deux couches cachées. A droite : le même réseau après avoir appliqué un dropout. Les unités barrées sont abandonnées (Srivastava et al. 2014).

DropConnect (Wan et al. 2013) est une autre technique dérivée du dropout qui consiste

36. *Shallow learning.*

à abandonner aléatoirement des poids plutôt que des activations. Les expériences ont montré que le DropConnect peut obtenir de meilleurs résultats bien qu'il est relativement lent.

2.4.3.2 Augmentation artificielle de données

L'augmentation artificielle de données consiste à appliquer des transformations sur des données existantes pour générer des nouvelles données artificielles. Elle permet d'améliorer la diversité des données d'entraînement dans le but d'améliorer la généralisation et augmenter la performance du réseau. Dans le cas d'images, on peut utiliser par exemple un effet miroir, des translations, des rotations et du flou (Shorten et Khoshgoftaar 2019). La figure 2.21 montre le principe de cette technique appliquée sur des images. Nous pouvons donc dupliquer l'image d'origine autant de fois que nous avons de transformation différentes à lui appliquer. Nous pouvons en augmenter aussi davantage en croisant ces effets sur une même image, ou en y appliquant différentes intensités de l'effet, dans une certaine fourchette, pour avoir une transformation plus ou moins accentuée de l'image d'origine.

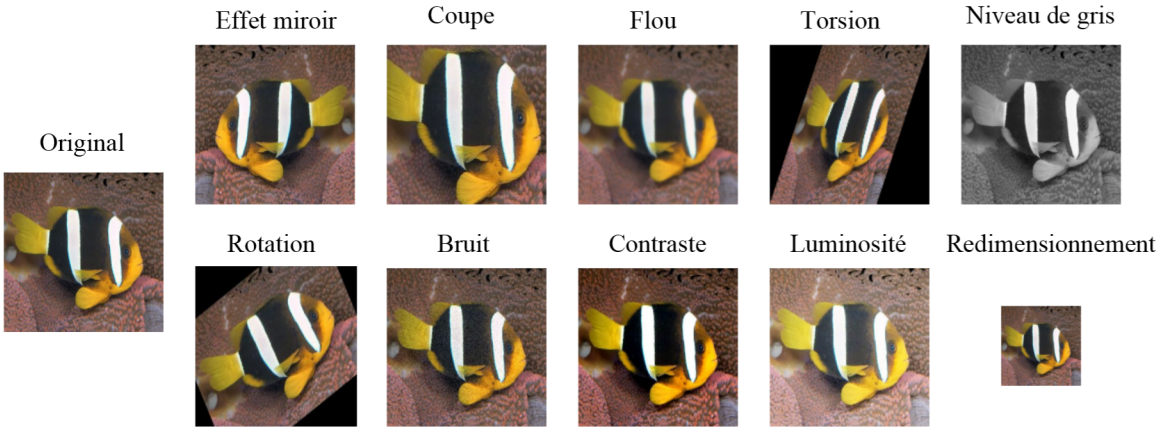


Figure 2.21 – Exemple d'augmentation artificielle de données à partir d'une image.

2.4.3.3 Initialisation avec pré-entraînement puis ré-entraînement

Le pré-entraînement (Erhan et al. 2010) consiste à initialiser l'apprentissage du réseau avec des paramètres pré-entraînés, plutôt que des paramètres pris aléatoirement. C'est une technique très utile dans les modèles basés sur les CNNs grâce à ses avantages notamment

l'accélération du processus d'entraînement et l'amélioration de la capacité de généralisation. Le modèle AlexNet (Krizhevsky, Sutskever et Hinton 2012) est entraîné sur la base ImageNet et ses paramètres sont rendus publics. De nombreuses approches ont proposé d'utiliser AlexNet comme un modèle profond de base (He et al. 2015 ; Oquab et al. 2014) et ont ré-entraîné leurs modèles pour affiner les paramètres en fonction de la tâche demandée. Il existe aussi des approches qui utilisent d'autres modèles de base comme GoogleNet (Szegedy et al. 2015), VGG (Simonyan et Zisserman 2014b) et ResNet (He et al. 2016) et donnent également de meilleures performances,.

Le ré-entrainement pour finaliser l'apprentissage ou le *"Fine-tuning"* est une étape cruciale pour affiner un modèle afin de l'adapter sur une nouvelle tâche et jeu de données. Toutes les couches du nouveau modèle sont initialisées à partir du modèle pré-entraîné à l'exception de la dernière couche de sortie où le nombre de neurones dépend du nombre de classes du nouveau jeu de données et sera donc initialisée aléatoirement. Durant le ré-entraînement, il est possible de ne ré-entraîner que cette dernière couche et geler les autres couches du modèle. Il est aussi possible de ré-entraîner plusieurs couches ou l'ensemble des couches en vue de finaliser l'apprentissage. Généralement, le modèle pré-entraîné possède déjà des paramètres quasiment optimisés, il est recommandé de les modifier faiblement à chaque itération, en adaptant un taux d'apprentissage faible, pour s'adapter en douceur à la nouvelle tâche sans écraser agressivement la connaissance déjà acquise.

Notons enfin que ces techniques de régularisation (décrites ci-dessus) ne s'excluent pas mutuellement et peuvent être combinées pour améliorer les performances.

2.5 Architectures profondes pour la classification et la détection d'objets

L'apprentissage profond a été largement adopté en vision par ordinateur dans différentes tâches telles que la classification d'images, la détection d'objets et la segmentation sémantique, qui sont des tâches clés pour la compréhension des images. Dans cette section, nous présentons brièvement les développements de l'apprentissage profond pour la classification et la détection d'objets dans des images.

2.5.1 Classification d'images

La tâche de classification d'images consiste à étiqueter les images d'entrée avec une probabilité d'appartenance à une classe d'objet particulière.

Les algorithmes d'apprentissage profond sont largement utilisés pour reconnaître les objets visuels dans de nombreuses applications de la vision par ordinateur. Pour cela, la dernière couche du réseau CNN utilise la fonction exponentielle normalisée (Equation (2.9)) appelée aussi "*Softmax*". Cette fonction applique une certaine normalisation des valeurs pour obtenir une distribution de probabilités sur les classes.

$$P = Softmax(X) = \begin{pmatrix} p_1 \\ p_2 \\ \vdots \\ p_n \end{pmatrix} \quad \text{où} \quad p_i = \frac{e^{x_i}}{\sum_{k=1}^{n} e^{x_k}} \tag{2.9}$$

$X = \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{pmatrix}$ est le vecteur des (scores) valeurs de sortie de l'avant dernière couche.

La classification par CNNs date des années 90 avec la reconnaissance des chiffres manuscrits (LeCun et al. 1998). Dans ce travail, les auteurs ont entrainé l'architecture LeNet-5 sur la base MNSIT. (Krizhevsky, Sutskever et Hinton 2012) a impulsé de nouveau la recherche sur les CNNs en gagnant la compétition ILSVRC 2012. Ils ont utilisé l'architecture AlexNet entraînée sur une grande base de données ImageNet. Après 2012, les CNNs ont toujours gagné les compétitions annuelles ILSVRC comme illustré dans la figure 2.12.

Aujourd'hui, les CNNs sont appliqués sur de nombreuses applications et jeux de données, nous citons par exemple la reconnaissance faciale (Parkhi, Vedaldi et Zisserman 2015 ; Sun et al. 2014, 2015), la reconnaissance d'actions humaines (Ji et al. 2012 ; Simonyan et Zisserman 2014a), et la reconnaissance de panneaux de signalisation (Cireşan et al. 2012 ; Lim et al. 2017).

2.5.2 Détection d'objets

La détection d'objets est une méthode permettant de détecter la présence d'une ou plusieurs classes d'objets dans une image et de localiser l'objet en question en le délimitant par une zone le plus souvent de forme rectangulaire. On parle souvent de patch pour faire référence à cette zone de l'image. Dans cette section, nous allons brièvement présenter quelques approches proposées dans la littérature pour la détection d'objets par les CNNs. Nous distinguons deux catégories d'algorithmes de détection d'objets : détecteurs à deux étages et ceux à une étage.

2.5.2.1 Détecteurs à deux étages

Ces détecteurs sont les premiers détecteurs basés CNN, ils se composent de deux modules, l'un pour la proposition de régions et l'autre pour la classification.

i. Le R-CNN

Le R-CNN [37], proposé par (Girshick et al. 2014), cherche dans un premier temps des régions d'intérêt (RoIs [38]) en appliquant un algorithme de segmentation (par exemple la recherche sélective [39] (Uijlings et al. 2013)) sur l'image d'entrée. Chaque région d'intérêt passe ensuite dans un même CNN pour l'extraction des caractéristiques qui seront utilisées enfin pour une classification de type SVM (figure 2.22).

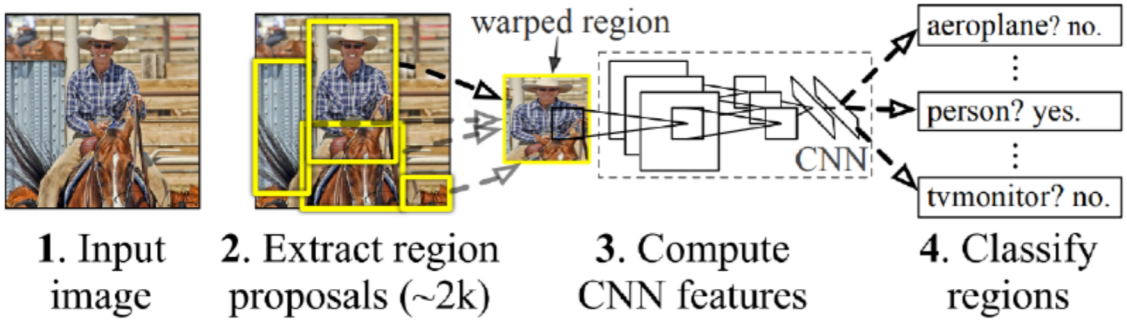


Figure 2.22 – Fonctionnement du R-CNN (Girshick et al. 2014).

37. R-CNN : *Regions with CNN features.*
38. RoI : *Region of Interst.*
39. *Selective search.*

ii. Le Fast R-CNN

L'un des problèmes conséquent du R-CNN est qu'il réalise beaucoup de calculs de carte de caractéristiques (proportionnellement au nombre de régions d'intérêt proposées). (Girshick 2015) a proposé une version plus rapide appelée le Fast R-CNN. En effet, le Fast R-CNN ne calcule qu'une seule fois les cartes de caractéristiques quelque soit le nombre de régions d'intérêt proposées. Ce qui réalise un gain de temps très important. Le Fast R-CNN consiste en cinq étapes suivantes (figure 2.23) :

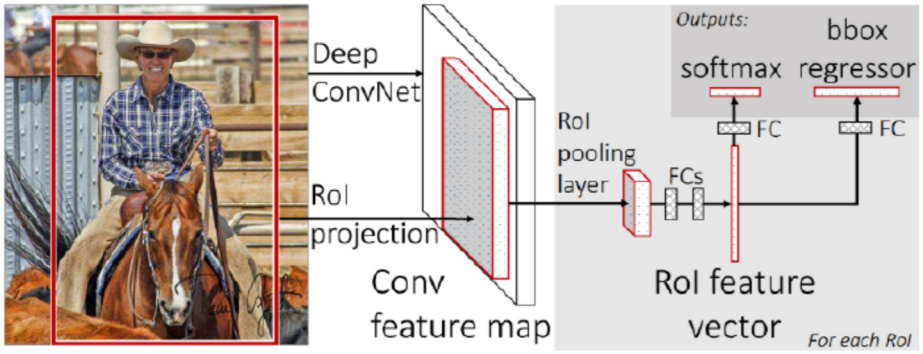


Figure 2.23 – Fonctionnement du Fast R-CNN (Girshick 2015).

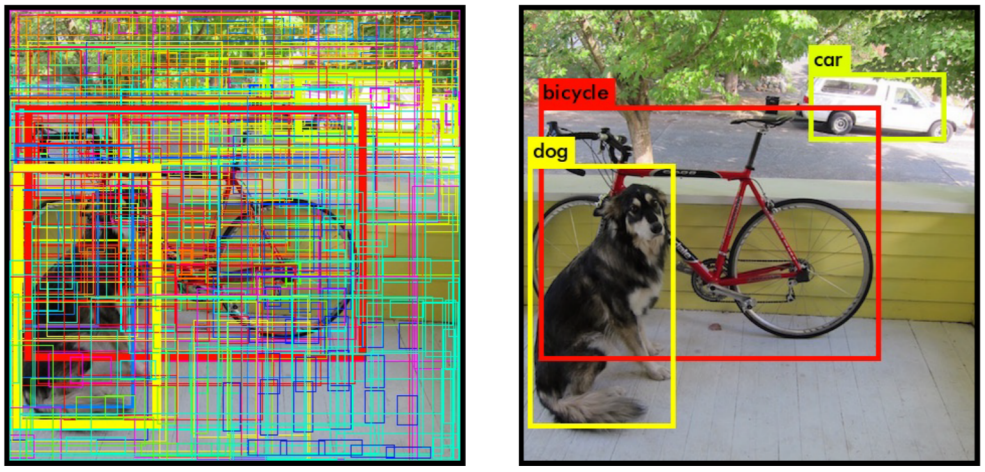
1. on passe l'image d'entrée toute entière dans un CNN pour extraire les cartes de caractéristiques sur toute l'image. Contrairement au R-CNN, on traite l'image entière d'un seul coup (pour tous les patchs), ce qui conduit à un gain en temps de calcul ;
2. on cherche toujours les régions d'intérêt avec une méthode indépendante (par exemple la recherche sélective) ;
3. on utilise ensuite une couche appelée "*RoI-Pooling*" qui va pour chacune des régions d'intérêt, proposées à l'étape 2, extraire son vecteur de caractéristiques correspondant ;
4. on classifie ces vecteurs de caractéristiques avec un réseau de neurones (au lieu d'un SVM) afin de déduire la catégorie d'appartenance de chaque patch ;
5. on utilise un autre réseau de neurones pour faire de la régression de localisation et ainsi améliorer la forme et la position de chaque patch. Ceci conduit à avoir en sortie 4 coordonnées qui vont former une boîte qui va englober l'objet en question ;
6. un même objet peut se retrouver dans plusieurs patchs. Pour éliminer les détections

redondantes, l'algorithme de suppression des non-maximums (NMS [40]) (Bodla et al. 2017) est utilisé.

La NMS consiste à :

1. trier dans l'ordre décroissant les détections selon le score du classifieur,
2. identifier la détection ayant le score le plus élevé et la mettre de côté,
3. pour chaque détection restante, supprimer cette détection si elle présente un recouvrement plus important avec la détection mise de côté,
4. reprendre à l'étape 2 avec les détections restantes.

La figure 2.24 ci-dessous illustre le principe de cette méthode.



(a) Détections avant d'appliquer la NMS

(b) Détections après l'application de la NMS

Figure 2.24 – Illustration de la NMS (Redmon et al. 2016).

iii. Le Faster R-CNN

Dans la même année, (Ren et al. 2015) ont proposé une nouvelle version du Fast R-CNN appelée le Faster R-CNN en y incorporant une stratégie interne de proposition de patchs (figure 2.25) : le réseau de proposition de régions (RPN [41]).

40. NMS : *Non-Maximum Suppression.*
41. RPN : *Region Proposal Network.*

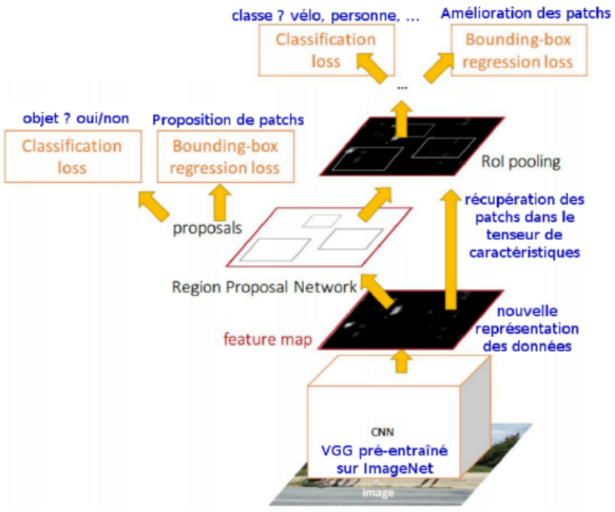


FIGURE 2.25 – Fonctionnement du Faster R-CNN (REN et al. 2015).

Réseau de proposition de régions RPN

C'est un petit réseau qui utilise les cartes d'activation générées par le CNN pour prédire un ensemble de patchs caractérisés à la fois par leurs probabilités de contenir un objet et par leurs coordonnées. Son apprentissage se fait pendant l'entraînement. Comme illustré sur la figure 2.25, le RPN profite de l'espace de caractéristiques global pour y étiqueter un nombre prédéfini de patchs sans avoir à inférer la classe finale du patch. Le RPN calcule également un score qui donne le degré d'appartenance du patch à la classe "objet" par opposition à la classe "fond".

Le reste du fonctionnement du Faster R-CNN reste relativement similaire par rapport au Fast R-CNN. Les régions proposées par le RPN ainsi que les cartes de caractéristiques utilisées alimentent ensuite des réseaux destinés à la classification et à la prédiction des coordonnées des boîtes englobantes correspondantes dans l'image d'origine.

Ce qui est intéressant avec la structure du Faster R-CNN c'est qu'elle est très modulaire. Chaque brique peut être remplacée, supprimée ou déplacée. C'est un avantage majeur pour le développement de nouvelles solutions.

iii. Mask R-CNN

Le Mask R-CNN développé par (HE et al. 2017) est une extension du Faster R-CNN. En effet, le modèle Faster R-CNN détecte des objets avec des boîtes englobantes. Le Mask

R-CNN permet de détecter des objets avec plus de précision de localisation en utilisant la segmentation d'instance. Contrairement à la segmentation sémantique qui permet d'associer à chaque pixel un label, la segmentation d'instance associe un masque et un label à chaque objet, même si ces objets appartiennent à la même classe.

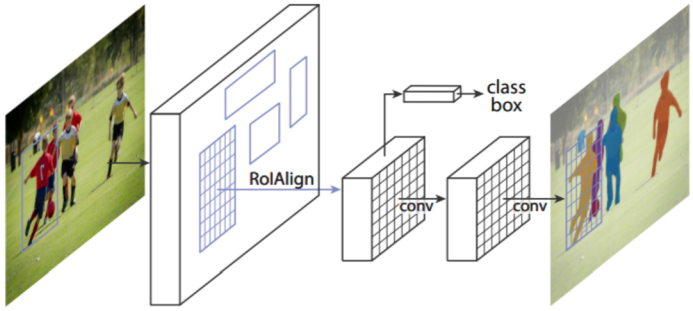


FIGURE 2.26 – Fonctionnement du Mask R-CNN (HE et al. 2017).

L'architecture de Mask R-CNN (figure 2.26) se diffère de l'architecture Faster R-CNN par l'ajout d'une branche complètement convolutive (FCN [42]) (LONG, SHELHAMER et DARRELL 2015) fonctionnant en parallèle de la classification et extrayant un masque binaire pour chaque RoI, fournissant ainsi une localisation plus précise de l'objet d'intérêt.

2.5.2.2 Détecteurs à un étage

Ces détecteurs d'objets à un étage [43] fusionnent les deux modules de base en un seul module pour prendre en compte simultanément la classification d'objet et sa localisation. OverFeat (SERMANET et al. 2013) était le premier détecteur à un étage complètement convolutionnel. Nous décrivons ici les trois modèles à un étage les plus répondus : YOLO, SSD et RetinaNet.

i. YOLO

Le principe du détecteur YOLO proposé par (REDMON et al. 2016) est de ne parcourir l'image qu'une seule fois, en la faisant passer à travers d'un CNN. Ce modèle divise l'image d'entrée en une grille de taille $S \times S$ (figure 2.27). Chaque cellule de la grille propose un nombre fixe, B, de boîtes englobantes. Ces boîtes sont caractérisées par leurs coordonnées

42. FCN : *Fully Convolutionnal Network*.
43. *Single-shot detectors*.

(x, y, w, h) et un score de présence d'objet où x et y représentent la position du centre relativement à la cellule correspondante, w et h sont respectivement la largeur et la hauteur de la boîte normalisée par la largeur et la hauteur de l'image. Par conséquent, x, y, w et h sont tous comprises entre 0 et 1. Le score reflète la probabilité que la boîte contienne un objet. Ensuite, pour chaque cellule, YOLO prédit les probabilités de C classes (une classe par type d'objet), indifféremment des boîtes. La prédiction finale de YOLO a la forme d'un tenseur $(S, S, (B \times 5 + C))$. Finalement, YOLO applique la NMS pour éliminer les détections redondantes. La figure 2.28 montre l'architecture globale de YOLO. Pour un exemple de S = 7, B= 2 et C= 20 classes, le tenseur de prédiction total est de taille $7 \times 7 \times 30$.

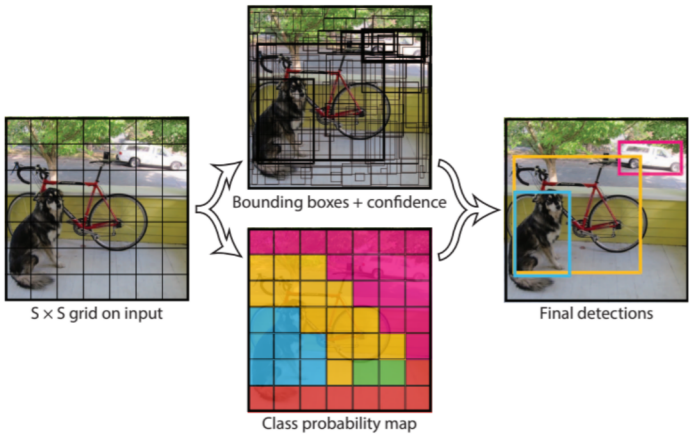


FIGURE 2.27 – Détecteur YOLO (REDMON et al. 2016), modèle de détection à un étage.

ii. SSD

Le détecteur YOLO utilise les dernières cartes de caractéristiques pour localiser les objets, ce qui rend difficile la localisation de petits objets. L'information précise de localisation est présente dans les premières couches de convolution. Le détecteur SSD développé par (LIU et al. 2016) est très proche de YOLO, mais au lieu d'utiliser une grille avec des cellules de taille fixe, il utilise des boîtes d'ancrage (ancres[44]) à différentes échelles, à l'image du Faster R-CNN. Ces ancres sont appliqués sur des cartes de caractéristiques issues de différents niveaux de couches de convolution pour ainsi améliorer l'invariance en taille de la détection (figure 2.28).

44. *Anchors* dans la littérature anglo-saxonne.

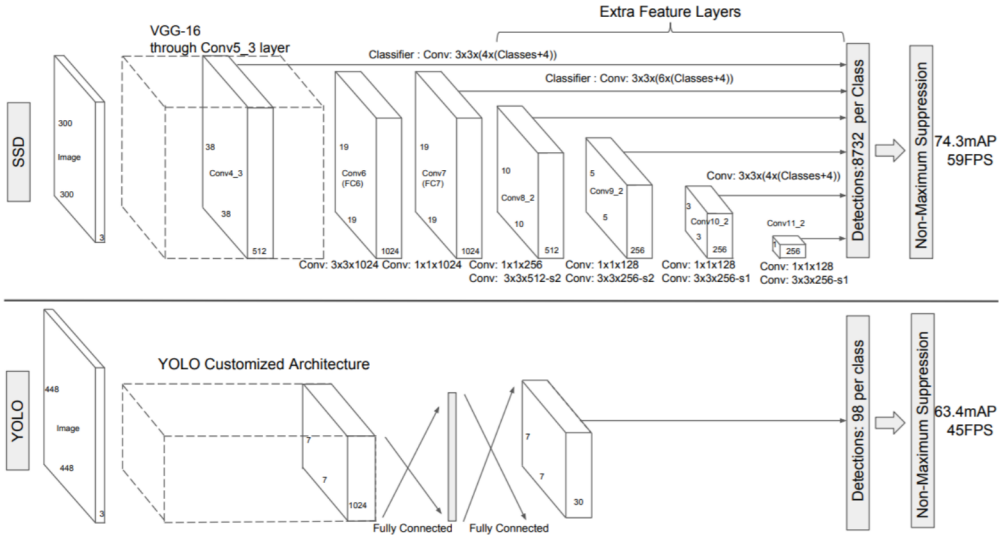


FIGURE 2.28 – Comparaison des deux architecures SSD et YOLO (LIU et al. 2016). Le modèle SSD applique des ancres et combine plusieurs cartes de caractéristiques issues de différents niveaux de couches de convolution, alors que YOLO n'utilise que les dernières cartes pour localiser les objets.

iii. RetinaNet

Pour finir, nous soulignons que les détecteurs à un étage sont plus rapides que les détecteurs à deux étages, mais ils sont moins efficaces. (LIN et al. 2017) ont proposé le détecteur RetinaNet utilisant la perte focale [45] pour améliorer les performances des détecteurs à un étage. Cette fonction de perte est obtenue en appliquant un terme de modulation à la perte d'entropie croisée afin de concentrer l'apprentissage sur des exemples difficiles. En conséquence, ils ont atteint une meilleure précision avec une vitesse plus élevée.

2.6 Conclusion

Les réseaux de neurones profonds, comme pour les réseaux de neurones classiques, reposent sur des architectures composées de neurones interconnectés. Selon l'architecture du réseau profond, différentes techniques sont apparues : machine de Boltzmann profonde,

45. *Focal loss.*

réseau de croyance profond, modèle d'énergie profond, auto-encodeur profond et réseau de neurones convolutif (CNN). Ce dernier reste le plus utilisé et tend être plus efficace pour les tâches de classification, détection, et segmentation. Grâce à ses différents types de couches (convolution, sous-échantillonnage, non linéaire, entièrement connectée, ...), CNN est capable d'extraire les informations de l'image brute et les transformer en cartes de caractéristiques pour effectuer ensuite la classification. Nous nous intéressons dans cette thèse aux réseaux de neurones convolutifs pour des tâches de reconnaissance d'espèces de poissons dans des images vidéo sous-marines.

Il existe de nombreuses architectures CNN très performantes comme AlexNet, GoogleNet, VGG, et ResNet rendant le choix difficile. Le choix devrait être un compromis entre la performance, le temps de calcul et les ressources (CPU, GPU, mémoire, ...).

Les cartes de caractéristiques fournies par les CNNs sont utilisées comme sources d'information pour des tâches de classification ou de détection. Elles peuvent être utilisées pour alimenter un classifieur de réseau de neurones intégré dans l'architecture CNN, ou pour un classifieur externe comme SVM, KNN ou arbre de décision.

Dans notre thèse, nous proposons une approche pour la détection de poissons dans des images vidéo sous-marines par fusion de réseaux CNN parallèles. Cette approche, présentée dans le chapitre suivant, fusionne des cartes de caractéristiques fournies par deux CNNs intégrés dans une architecture Faster R-CNN. Pour l'identification d'espèces de poissons, nous proposons différentes approches de classification basées sur les architectures CNN et l'apprentissage par transfert ou l'apprentissage progressif (hiérarchique et incrémental).

Deuxième partie

Méthodologie et validation

Chapitre 3

Détection de poissons dans des images vidéo sous-marines par fusion de réseaux CNN parallèles

Sommaire

3.1 Introduction

Généralement, un système automatique de reconnaissance d'espèces de poissons consiste en deux étapes : 1) la détection de poissons qui permet de localiser et discriminer le poisson de l'arrière-plan et 2) la classification d'espèces de poissons qui permet d'identifier l'espèce de chaque poisson détecté. Nous distinguons alors deux propositions : 1) Soit on construit un détecteur multi-classe effectuant les deux étapes en même temps. Ce détecteur apprendra au cours du même entraînement à faire la détection et la classification à la fois. 2) Soit on utilise dans un premier temps un détecteur mono-classe pour localiser les poissons dans les vidéos sous-marines et, ensuite, la localisation est transférée à un classifieur pour reconnaître l'espèce. Cette technique permet de séparer complètement la détection de la classification en adaptant un modèle pour chaque tâche. En effet, les bases d'entraînement sont généralement déséquilibrées. Certaines espèces sont plus fréquentes que d'autres. Cela induit ainsi un biais dans l'entraînement et réduit la performance du système à identifier les espèces. Nous allons adopter dans ce travail de thèse la seconde proposition, en construisant un détecteur pour localiser les poissons dans des images vidéo sous-marines. La problématique de la classification de poissons détectés sera traitée aux chapitres suivants.

Comme nous avons pu le voir en section 1.4.2.2 intitulée "Détection automatique de poissons", les premiers travaux sur le sujet implémentent une modélisation de l'arrière-plan en utilisant des méthodes traditionnelles telles que le modèle de mélange de gaussiennes (Hsiao et al. 2014 ; Spampinato et al. 2008). Ces méthodes ont des limites pour modéliser des fonds complexes comme le fond marin. Dans cet environnement, l'arrière-plan est complexe à cause de la diversité et du mouvement des plantes aquatiques, du faible contraste, du changement de luminosité, et de la mauvaise visibilité.

Récemment, avec l'arrivée de l'apprentissage profond, de nombreux travaux ont développé des algorithmes de détection d'objets basés sur les CNNs. Pour la détection de poissons, les travaux récents utilisent des détecteurs CNN classiques : Fast R-CNN (Li et al. 2015), Faster R-CNN (Li, Tang et Gao 2017 ; Li et al. 2016 ; Mandal et al. 2018), SSD (Shi, Jia et Chen 2018 ; Zhuang et al. 2017) et YOLO (Sung, Yu et Girdhar

2017). D'autres travaux ont introduit des approches hybrides basées sur les CNNs et des méthodes traditionnelles (Jäger et al. 2016 ; Salman et al. 2020 ; Zhang et al. 2016).

Par ailleurs, nous trouvons dans la littérature de la vision par ordinateur des stratégies pour la fusion d'informations. La fusion intègre des informations provenant de plusieurs modalités (par exemple RGB, profondeur, infrarouge, audio) ou provenant de plusieurs espaces (par exemple spatial et temporel). Le but de cette fusion est de mélanger des informations provenant de sources différentes pour mieux résoudre un problème donné. Un autre avantage de la fusion est l'amélioration de la robustesse de la prédiction et la performance du système. Elle complète l'information et garantit un système opérationnel en cas de perte d'une source d'information (dans le cas d'une fusion multimodale).

Dans ce chapitre, nous abordons la détection de poissons de récifs coralliens dans des vidéos sous-marines enregistrées en mer dans un environnement naturel sans contrainte (lumière naturelle non contrôlée, arrière-plans divers et variés, mauvaise résolution, etc.). Nous proposons pour cela deux nouvelles architectures de fusion d'informations. Le reste de ce chapitre est organisé comme suit : nous présentons tout d'abord les différentes techniques de la détection d'objets par fusion d'informations (section 3.2). En section 3.3, nous décrivons les deux architectures de fusion que nous proposons. Nous évaluons ces approches dans la section 3.4 et discutons des résultats expérimentaux. Finalement, nous terminons ce chapitre avec une conclusion en section 3.5.

3.2 Détection par fusion d'informations

La fusion d'informations consiste à utiliser simultanément plusieurs sources d'informations différentes afin d'améliorer la prédiction. Ces informations peuvent être issues de multiples modalités ou de multiples espaces. De nombreux travaux ont proposé la fusion d'informations pour différentes tâches de la vision par ordinateur. Dans la détection d'objets, beaucoup de travaux utilisent la fusion pour une détection basée sur le Faster R-CNN (Ren et al. 2015). Ce détecteur est connu pour sa robustesse dans des environnements complexes et changeants ; c'est donc un modèle très précis pour la localisation d'objets (Huang et al. 2017). Faster R-CNN se compose de trois réseaux neuronaux comme illustré dans la figure 3.1 : un réseau de base, un réseau de proposition de région (RPN) et un réseau de classification. Cette structure modulaire permet de développer de nouvelles

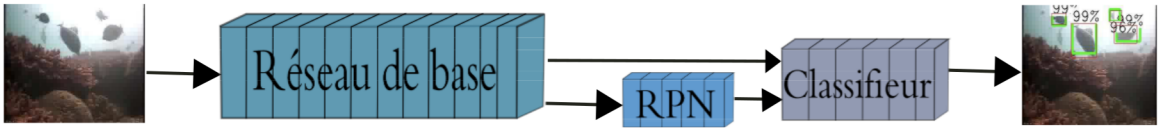


FIGURE 3.1 – Structure du détecteur Faster R-CNN composée de trois réseaux CNN : un CNN de base, un CNN de proposition de région (RPN) et un CNN classifieur.

solutions en remplaçant, supprimant ou déplaçant les modules.

Nous décrivons ici les approches de fusion multimodale, qui peuvent être regroupées en trois catégories principales : les fusions précoces, tardives et hybrides.

3.2.1 Fusion précoce

Cette fusion contient elle-même deux types de fusion selon le niveau conceptuel de l'information : fusion à bas niveau ou au niveau intermédiaire.

La fusion à bas niveau est simplement basée sur l'opérateur de concaténation. Les données brutes provenant de différentes sources sont concaténées à l'entrée du modèle. Par exemple, (FARAHNAKIAN et HEIKKONEN 2020) ont proposé de concaténer des images RGB et infrarouge pour alimenter un Faster R-CNN afin de détecter des navires maritimes (figure 3.2).

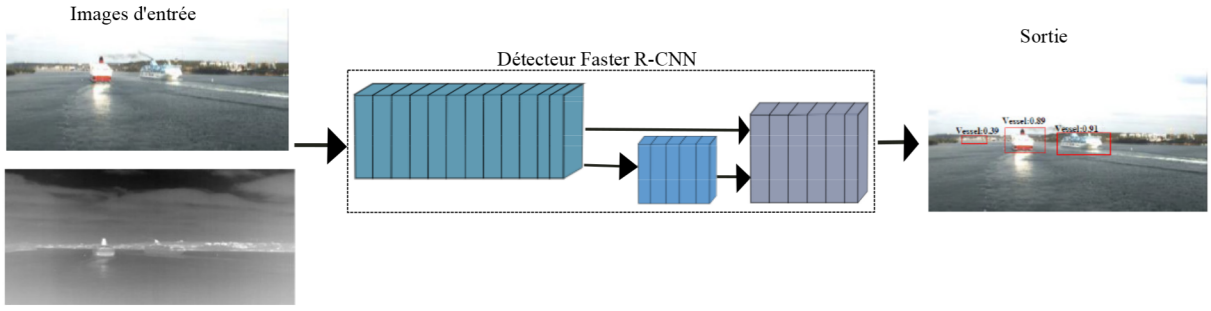


FIGURE 3.2 – Illustration de la fusion précoce à l'entrée du modèle. Ici, les images provenant de la caméra RGB et infrarouge sont concaténées avant d'alimenter un détecteur pour localiser des navires.

Dans la fusion au niveau intermédiaire, plusieurs réseaux CNN sont utilisés pour extraire séparément des caractéristiques de chaque source d'information. Ensuite, les caractéristiques sont fusionnées pour alimenter un classifieur. (GUERRY, LE SAUX et FILLIAT

2017) ont proposé la fusion en Y (figure 3.3) pour fusionner des données de couleur et de profondeur pour la détection de personnes à l'aide du modèle Faster R-CNN. Cette fusion n'utilise qu'un seul RPN et qu'un seul classifieur qui prennent en entrée la concaténation des caractéristiques extraites des deux réseaux de base parallèles. Ainsi, le RPN et le classifieur ont un espace plus riche pour détecter et classifier les objets ; néanmoins, cet espace est désormais plus grand. Le modèle exige un entraînement plus long avec plus d'exemples.

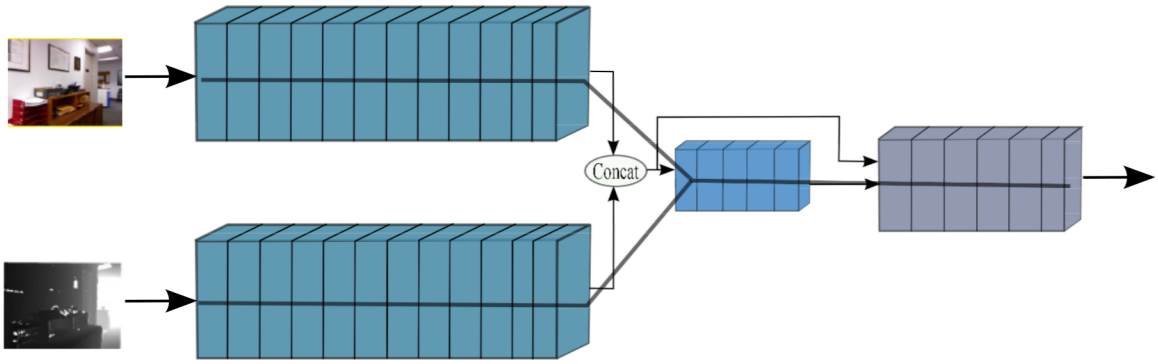


FIGURE 3.3 – Illustration de la fusion en Y. Ici, les réseaux CNN extraient les caractéristiques des images provenant de la caméra RGB et de la profondeur. Ensuite, ces caractéristiques sont fusionnées pour alimenter un RPN et un classifieur.

Un récent travail de (ZHU et al. 2020) propose un Faster R-CNN à deux réseaux CNN pour fusionner les images couleur et de profondeur pour la détection des truies en lactation (figure 3.4). Tout d'abord, les caractéristiques RGB et de profondeur sont extraites séparément à l'aide de deux CNNs. Ensuite, un seul RPN est utilisé pour générer les RoIs. Ce RPN utilise uniquement les caractéristiques de profondeur pour proposer les RoIs de profondeur. Les coordonnées générées sont projetées sur les caractéristiques RGB pour générer les RGB RoIs correspondantes. Enfin, les deux RoIs sont fusionnées pour alimenter un seul classifieur.

La fusion précoce ne nécessite qu'une seule phase d'apprentissage. Etant donné que les caractéristiques y sont fusionnées dès le départ, la fusion précoce donne aussi une représentation riche aidant à apprendre les relations entre les classes pour modéliser les interactions entre les informations. Toutefois, la dimension des caractéristiques devient plus grande augmentant le risque de sur-apprentissage.

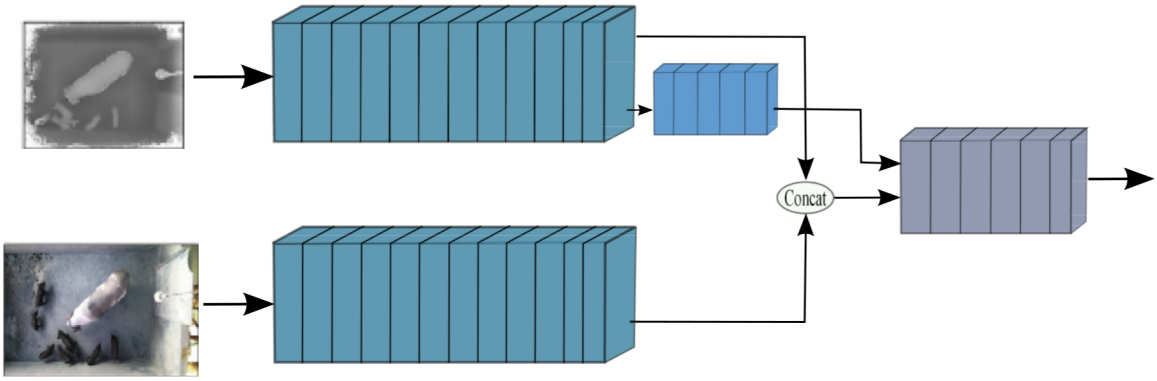


FIGURE 3.4 – Illustration du Faster R-CNN à deux réseaux parallèles de (ZHU et al. 2020). Ici, le RPN génère des RoIs en se basant uniquement sur les caractéristiques issues de l'image de profondeur. Puis, les coordonnées générées sont projetées sur les caractéristiques RGB pour générer les RGB RoIs correspondantes. Enfin, les deux RoIs sont fusionnées pour alimenter un seul classifieur.

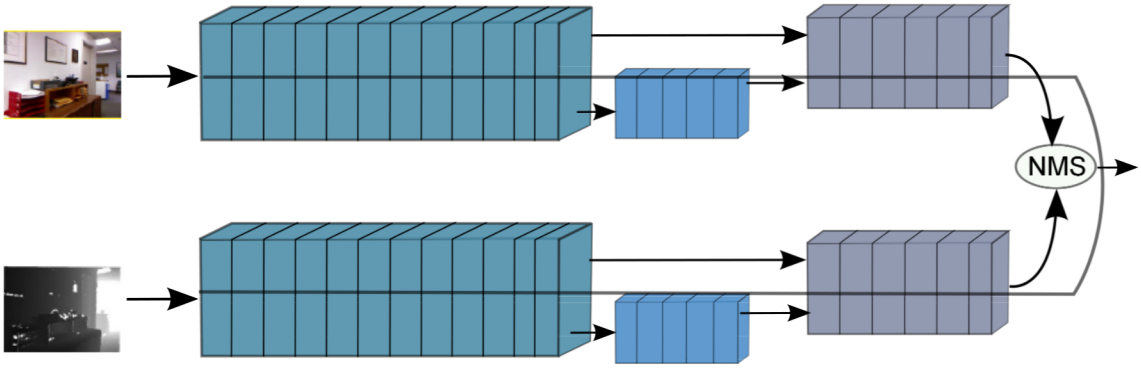
3.2.2 Fusion tardive

La fusion tardive est une fusion de haut niveau qui correspond à la fusion de différentes décisions (comme les sorties de la classification, de la détection ou de la régression). La fusion des décisions fournies par plusieurs réseaux experts peut se faire à l'aide d'un des mécanismes de fusion. De ces mécanismes nous citons la somme, la moyenne ou le maximum des scores (WANG et al. 2016), les schémas de vote (MORVANT, HABRARD et AYACHE 2014), la suppression des non-maximums (NMS) (MONKAM et al. 2018), et les modèles entraînés SVM et ELM [1] (machine d'apprentissage extrême). (GUERRY, LE SAUX et FILLIAT 2017) ont proposé deux fusions tardives comme illustré en figure 3.5 :

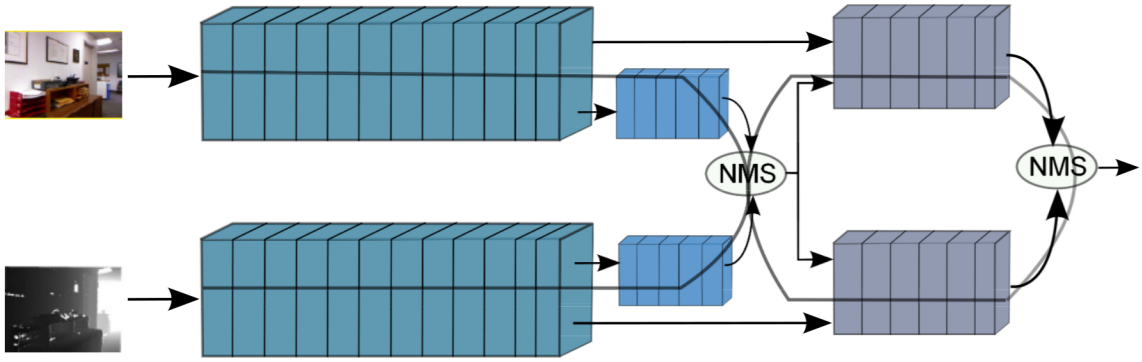
— La fusion en U est une fusion simple avec deux réseaux parallèles. Les résultats de détection fournis par les deux réseaux sont fusionnés tout à la fin via une NMS qui trie et choisit les meilleures détections d'objets.

— La Fusion en X où la NMS est placée après les RPNs, ce qui permet de mettre en commun les régions d'intérêt avant la classification par les deux réseaux. Ce partage intermédiaire permet d'échanger les détections entre les deux réseaux parallèles. Les classifieurs utilisent ces détections sans considération de leur origine. Les détections finales redondantes sont gérées par la NMS finale comme dans la fusion en U.

1. ELM : *Extrem Learning Machine*.



(a) Fusion en U



(b) Fusion en X

FIGURE 3.5 – Illustration de la fusion en U et en X (GUERRY, LE SAUX et FILLIAT 2017). En (a), la fusion en U utilise une NMS sur les sorties des classifieurs des deux réseaux parallèles. En (b), la fusion en X utilise une NMS sur les sorties des RPNs et une NMS sur les sorties des classifieurs des deux réseaux parallèles.

Il est important de souligner que dans les fusions en U et en X, les réseaux parallèles sont entraînés indépendamment l'un de l'autre. Ils ne sont fusionnés qu'à la fin de l'entraînement. Cela permettrait si besoin d'ajouter d'autres sources d'informations sans avoir à ré-entraîner les premiers réseaux. A l'encontre, la fusion précoce doit être entraînée avec toutes les données et nous ne pouvons pas y ajouter une nouvelle source d'information sans ré-entraîner l'ensemble. L'inconvénient de la fusion tardive est que chaque réseau nécessite une étape d'apprentissage supervisé distincte, ce qui signifie que cette fusion ne peut pas modéliser les éventuelles corrélations entre les informations.

3.2.3 Fusion hybride

Enfin, la fusion hybride (Poria et al. 2016 ; Wöllmer et al. 2013) est la combinaison de la fusion précoce et tardive (figure 3.6). Elle exploite les avantages des méthodes de fusion précoce et tardive dans un cadre commun et surmonte les inconvénients de chacune d'elles.

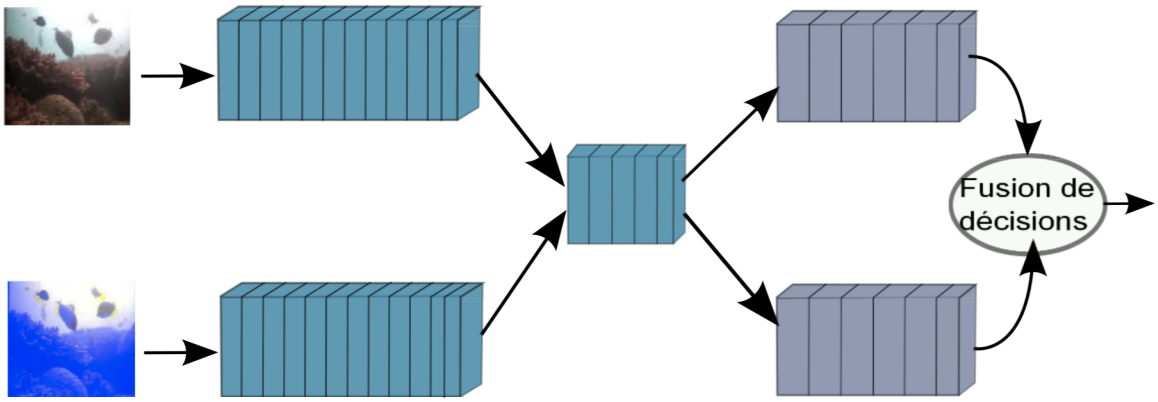


Figure 3.6 – Illustration de la fusion hybride (combinaison de la fusion précoce et tardive).

3.3 Fusion de réseaux parallèles pour la détection de poissons

A notre connaissance, les approches CNN existantes pour la détection de poissons sont basées sur un seul réseau. En outre, seuls les travaux de (Salman et al. 2020) ont proposé une stratégie de fusion basée sur la fusion bas niveau dans un détecteur à un seul réseau (figure 3.7). Ils ont concaténé la sortie de GMM, le flux optique et l'image en niveau de gris.

En nous inspirant des méthodes de fusion proposées dans la section précédente, nous proposons et développons ici deux nouvelles architectures à réseaux CNN parallèles pour la fusion d'informations (que nous appelons fusion en YU et en UY, figure 3.8). Chaque réseau CNN extrait des caractéristiques de chaque source d'information. Dans notre application de détection de poissons, un premier réseau CNN extrait les caractéristiques d'apparence de chaque image vidéo couleur, tandis que l'autre réseau CNN extrait les caractéristiques

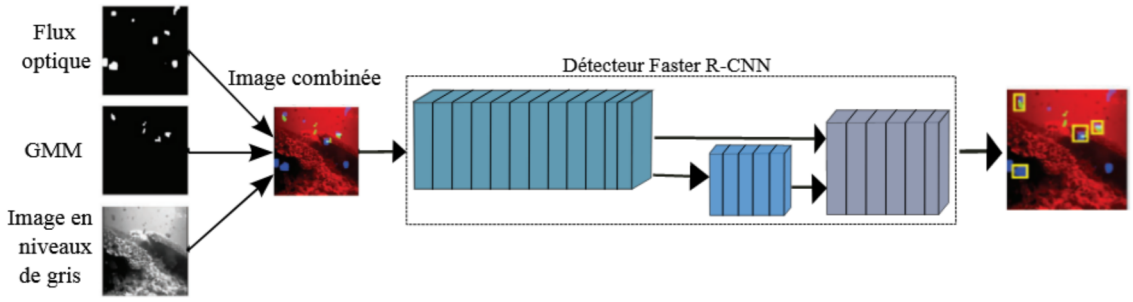


FIGURE 3.7 – Illustration du système de détection de poissons proposé par (SALMAN et al. 2020). Le système est entraîné sur des images résultats de la combinaison de la sortie de l'algorithme GMM, de flux optique et de l'image en niveaux de gris. Ceci est analogue à une image RGB à trois canaux.

de mouvement des images successives. L'entrée de ce second réseau est composée de deux images successives en niveaux de gris et du flux optique correspondant. L'objectif de ce réseau est de détecter des poissons en apprenant au système la relation qui existe entre des images successives.

3.3.1 Entrées des architectures

3.3.1.1 Entrée de couleur

Le choix du bon espace colorimétrique est crucial dans les tâches de détection automatique, en particulier dans les vidéos sous-marines où la luminosité est relativement faible. Le type de modèle de couleur d'entrée peut affecter les performances de détection. Nous choisissons l'espace colorimétrique RGB car les architectures utilisées sont déjà pré-entraînées sur des images RGB du jeu de données ImageNet (DENG et al. 2009). Les poids des filtres sont plus liés aux images RGB qu'aux autres espaces colorimétriques. De plus, dans l'environnement sous-marin, le spectre visible est modifié avec la profondeur. Les radiations de fréquences plus élevées sont les moins absorbées. Ainsi, lorsque la composante rouge disparaît déjà en eau peu profonde (5 m), la composante verte disparaît à environ 50 m et la composante bleue est absorbée à environ 60 m. En conséquence, en mer plus profonde, nous obtenons généralement une scène bleue-verte (BIANCO et al. 2015) ; pour cette raison, les composantes bleue et verte fournissent des informations beaucoup plus discriminantes que les autres composantes de différents modèles de couleurs. Nous testerons différents espaces

colorimétriques dans le prochain chapitre pour la classification d'espèces de poissons.

3.3.1.2 Entrée de mouvement

Nous utilisons le flux optique pour calculer l'entrée de mouvement. Le flux optique est un champ de déplacement 2D qui décrit le mouvement apparent des objets, des surfaces et des contours de la scène visuelle entre deux images successives. Il est calculé sur la base de l'hypothèse de constance de luminosité (BCA [2]), qui suppose que la luminosité des pixels qui se correspondent reste constante dans des images consécutives (Horn et Schunck 1981). Le flux optique est largement utilisé pour séparer le premier plan de l'arrière-plan et pour identifier les objets en mouvement. Il est ainsi largement utilisé dans les tâches de vision par ordinateur, y compris la segmentation (Tsai, Yang et Black 2016), la détection (Xu et al. 2017), la classification (Simonyan et Zisserman 2014a) et le suivi (Xiao et Jae Lee 2016).

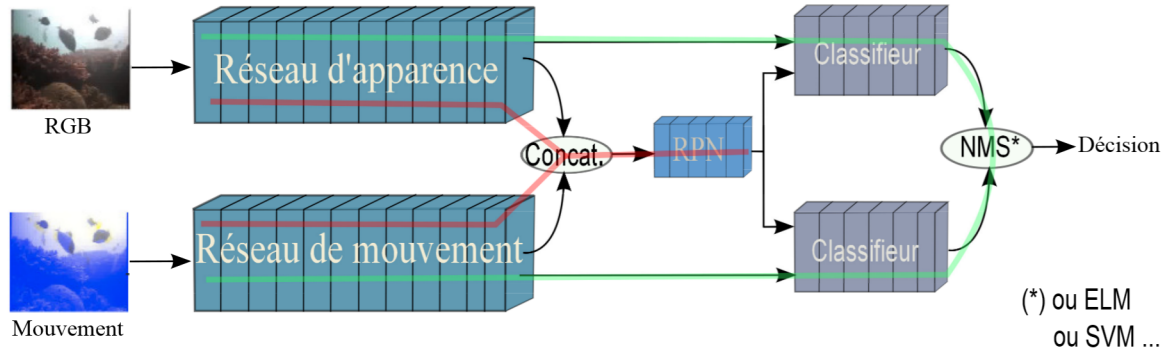
De nombreuses approches sont proposées pour estimer le flux optique (Tu et al. 2019). Dans ce travail, nous utilisons l'algorithme de variation totale [3] (avec la norme L^1) $TV-L^1$ (Zach, Pock et Bischof 2007). Cet algorithme de flux optique très populaire pour son efficacité est basé sur une méthode différentielle qui calcule la vitesse à partir des dérivées spatiales et temporelles de la luminosité de l'image.

Sous forme d'une entrée à 3 canaux, nous concaténons la sortie du flux optique avec deux images en niveaux de gris successives. Le but de cette combinaison est de détecter des poissons en apprenant au système la relation qui existe entre les images successives via un apprentissage profond. Ainsi, les deux images en niveaux de gris permettent de distinguer le mouvement du poisson du mouvement des autres objets de la scène, ce qui pourrait aussi être utile pour étudier ultérieurement le comportement des poissons.

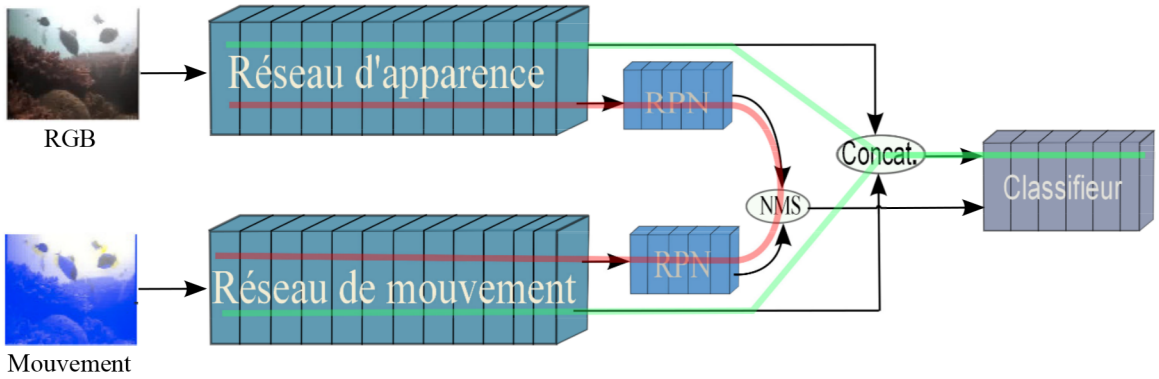
3.3.2 Architectures de détection proposées

Nous proposons deux nouvelles architectures pour la fusion de réseaux parallèles (fusion en YU et en UY). Dans notre application, l'objectif de la fusion est de mieux détecter les

2. BCA : *Brightness Constancy Assumption.*
3. $TV-L^1$: *Total variation regularization and the robust* L^1 *norm.*



(a) Fusion en YU



(b) Fusion en UY

FIGURE 3.8 – Illustration des approches de fusion proposées. (a) La fusion en YU utilise un seul RPN partagé entre deux Faster R-CNNs. (b) : La fusion en UY utilise un seul classifieur partagé entre deux Faster R-CNNs.

poissons en mouvement en utilisant les caractéristiques extraites des deux réseaux parallèles. La figure 3.8 montre les deux approches proposées pour la détection de poissons dans des images vidéo sous-marines.

— **Fusion en YU** : dans cette architecture de fusion hybride, un RPN a été partagé après deux réseaux CNN de base. Le RPN prend en entrée la concaténation des caractéristiques extraites des deux réseaux de base et génère les RoIs. Ensuite, deux classifieurs projettent ces RoIs respectivement sur les deux sorties des réseaux de base. Enfin, une phase de fusion de décisions a été placée à la fin de l'architecture pour fusionner les sorties des deux classifieurs en vue d'une meilleure détection. L'avantage de cette approche est que le RPN obtient un espace plus riche (apparence et mouvement) et peut donc mieux prédire des RoIs. L'utilisation de deux classifieurs

permet de se compléter contre le risque de détections manquées.
Nous étudions trois techniques de fusion de décisions : NMS (Guerry, Le Saux et Filliat 2017), ELM (Monkam et al. 2018) et SVM (Zha et al. 2015). La NMS est utilisée pour réduire les boîtes de détection redondantes en conservant la meilleure boîte de détection, qui a le score le plus élevé, et en supprimant les autres boîtes de détection qui se chevauchent largement. Avec ELM ou SVM, nous combinons les scores en sortie des deux classifieurs afin d'alimenter un réseau ELM ou une machine SVM conçu(e) pour reclasser chaque boîte de détection en deux classes : poisson ou non poisson. Nous avons choisi ELM et SVM car leurs processus d'entraînement ne reposent pas sur l'algorithme de rétro-propagation qui est extrêmement coûteux. De plus, ELM et SVM sont efficaces dans les tâches de classification avec une très bonne vitesse d'entraînement (Monkam et al. 2018).

— **Fusion en UY** : cette architecture de fusion précoce au niveau intermédiaire partage un seul classifieur mais utilise deux RPNs, l'un correspondant exclusivement au réseau d'apparence et l'autre au réseau de mouvement. Par conséquent, nous avons un RPN d'apparence, proposant des régions basées sur l'apparence des poissons, et un RPN de mouvement, générant des régions basées sur le mouvement des poissons. La NMS placée après les deux RPNs permet de partager les RoIs et de ne choisir que les meilleures. Ensuite, le classifieur projette ces RoIs sur la concaténation des caractéristiques extraites des deux réseaux de base. Cette technique nous permet d'obtenir un espace plus riche pour le classifieur. Ayant un classifieur unique, cette architecture a moins de paramètres à optimiser. En outre, les deux RPNs coopèrent pour générer des RoIs plus fiables.

Dans notre travail, nous utilisons l'architecture ResNet-50 (He et al. 2016) comme réseau de base pré-entraîné sur la base d'images ImageNet (Deng et al. 2009) pour générer des cartes de caractéristiques. Les cartes de caractéristiques extraites sont introduites dans le RPN pour produire des RoIs et dans le classifieur pour aboutir à une décision. Pour une image d'entrée de taille 640×480, nous utilisons quatre échelles différentes (32, 64, 128, 256) chacune avec quatre facteurs d'échelle différents $(1:1; 1:2; 2:1; \frac{2}{\sqrt{2}} : \frac{2}{\sqrt{2}})$ pour générer 16 boîtes d'ancrage (ancres ou anchors en anglais). Enfin, le réseau classifieur classe les RoIs générées en classe poisson ou non.

3.4 Résultats expérimentaux

Nous évaluons nos approches sur la base de vidéos de référence LCF-15. L'ensemble d'entraînement est constitué de 20 vidéos annotées et l'ensemble de test comprend 73 vidéos annotées. Nous définissons d'abord la métrique utilisée pour évaluer les systèmes de détection proposés (section 3.4.1). Ensuite, nous évaluons les approches de fusion en YU (section 3.4.2) et en UY (section 3.4.3) proposées dans cette thèse. Enfin, nous comparons nos approches avec des approches de l'état de l'art, notamment les approches basées sur la fusion d'informations (section 3.4.4). Nous soulignons dès à présent que nous évaluons les performances de nos approches en utilisant les métriques standards de la détection d'objets : la précision moyenne (AP [4]) et la F-mesure. Nous considérons une détection correcte si l'intersection sur l'union avec la vérité terrain est supérieure à 0,5.

Nous notons que nous avons utilisé un système informatique équipé de processeurs Intel Core-i5 avec GPU Geforce GTX 1050 Ti, installé avec 2 Go de mémoire GPU. Nous avons implémenté les approches proposées en python en utilisant Keras avec le backend de la bibliothèque TensorFlow, et l'algorithme de variation totale $TV - L^1$ pour le flux optique.

3.4.1 Métriques d'évaluation

Dans une tâche de détection, le système effectue la localisation en plus de la classification. Il est alors nécessaire de mesurer la correspondance entre les boîtes englobantes prédites par le système et les annotations de vérité terrain. Nous utilisons pour cela l'intersection sur l'union [5] appelée IoU : aire de l'intersection de deux boîtes englobantes divisée par l'aire de l'union de ces deux boîtes. L'IoU permet de mesurer le recouvrement de la boîte englobante proposée avec celle de la vérité terrain comme illustré dans la figure 3.9. Ensuite, la détection (boîte englobante proposée) est considérée correcte (vrai positif) si l'IoU est supérieur à un seuil (la plupart des travaux prennent le seuil égal à 0,5).

Nous pouvons calculer différentes métriques en se basant sur les résultats de prédiction obtenus par un système de détection et sur la vérité terrain (détectés, non-détectés, fausses alertes). Formellement, soit T le nombre total d'échantillons positifs en sortie du système de détection. Nous définissons :

4. AP : *Average Precision.*
5. IoU : *Intersection over Union.*

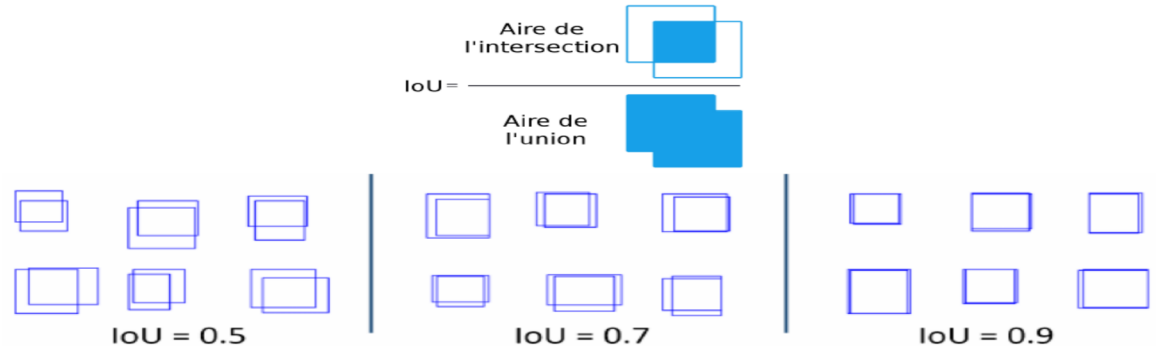


FIGURE 3.9 – Illustration de la mesure, intersection sur union, IoU. Quelques exemples d'IoU de 0.5, 0.7 et 0.9.

— **Vrais positifs** : les échantillons correctement détectés, soit VP leur nombre.
— **Faux positifs** : les échantillons détectés par erreur, soit FP leur nombre.
— **Faux négatifs** : les échantillons non détectés, soit FN leur nombre.

A partir de ces définitions, différentes métriques peuvent être calculées, notamment la précision et le rappel.

La précision d'une classe est le pourcentage de nombre d'échantillons corrects sur le nombre total d'échantillons positifs.

$$\text{Précision} = P = \frac{VP}{T} = \frac{VP}{VP + FP} \tag{3.1}$$

Le rappel d'une classe est le pourcentage de nombre d'échantillons corrects sur le nombre total d'échantillons de la classe (ce dernier étant réparti entre les vrais positifs et les faux négatifs).

$$\text{Rappel} = R = \frac{VP}{VP + FN} \tag{3.2}$$

Une mesure qui combine la précision et le rappel est la F-mesure ou F-score. Elle est calculée par :

$$\text{F-mesure} = 2.\frac{P \times R}{P + R} \tag{3.3}$$

Une autre métrique très utilisée dans la détection d'objets est la précision moyenne (AP). A partir des images de la base de test, il est possible de tracer la courbe de précision (P) en fonction du rappel (R). Cette courbe va permettre de définir la précision moyenne du modèle (AP) en calculant l'aire sous cette courbe. Elle est définie comme suit :

$$AP = \int_0^1 P(R)\,\mathrm{d}R \tag{3.4}$$

Dans le cas d'une détection multi-classe, la moyenne de la précision moyenne (mAP) est obtenue faisant la moyenne de toutes les APs sur l'ensemble des classes recherchées par le modèle.

3.4.2 Approche de fusion en YU

Nous commençons par évaluer l'approche de la fusion en YU (figure 3.8(a)). Avant d'étudier la détection finale obtenue, nous analysons le comportement du réseau RPN partagé entre les deux réseaux parallèles.

3.4.2.1 RPN standard *versus* RPN partagé

Tout d'abord, nous voulons évaluer l'apport du réseau RPN partagé sur l'amélioration des résultats de détection. Pour cela, nous comparons les résultats de notre architecture avec ceux d'un Faster R-CNN standard. Nous considérons indépendamment deux Faster R-CNNs, l'un entraîné sur l'image RGB et l'autre sur la carte de mouvement. La table 3.1 montre les performances en détection de poissons des approches comparées sur la base de vidéos de référence LCF-15. Pour la fusion en YU, la table répertorie uniquement les résultats de sortie de chaque classifieur, autrement dit, avant la fusion de décisions (figure 3.8(a)).

D'après la table 3.1, nous voyons que notre RPN conduit à de meilleurs résultats que le RPN standard qui n'est entraîné que sur l'information d'apparence ou de mouvement. Cela est grâce à l'espace de caractéristiques de notre RPN qui est entraîné à la fois sur l'information d'apparence et de mouvement. L'espace est plus riche, donc cela permet au

Architecture	Réseau	Entrée	F-mesure	AP
Un seul réseau	Apparence	RGB	77,82	64,71
	Mouvement	Mouvement	78,78	67,49
Fusion en YU	Apparence	RGB	**79,47**	**67,04**
	Mouvement	Mouvement	**80,22**	**70,50**

TABLE 3.1 – Comparaison des performances en détection de poissons (taux en %) entre le Faster R-CNN standard et notre architecture de fusion en YU, sur la base LCF-15.

RPN de mieux proposer des régions plus fiables. On note également que les modèles de mouvement sont plus efficaces que ceux d'apparence. L'algorithme de flux optique permet plus de RoIs en raison de sa sensibilité à chaque mouvement de l'image ou aux changements de luminosité. De plus, les caractéristiques de mouvement sont plus pertinentes ; en plus des informations spatiales dans les images en niveaux de gris, elles représentent également des informations temporelles telles que le mouvement des poissons, la variation de la lumière et le changement de l'arrière-plan.

La figure 3.10 montre des exemples de sorties de classifieur de chaque Faster R-CNN standard et de chaque réseau de la fusion en YU. Il est intéressant d'observer dans la première ligne que notre RPN possède une nouvelle détection alors que les deux modèles Faster R-CNN standards n'en proposent aucune. Aussi, notre RPN est en mesure de proposer de nouvelles détections qui peuvent être classées par au moins l'un de nos classifieurs (deuxième ligne). Une autre observation importante que nous pouvons tirer de ces résultats est que le RPN partagé peut également supprimer une fausse détection (troisième ligne), ce qui augmente la précision. Cependant, il peut aussi supprimer une vraie détection (dernière ligne), ce qui diminue le rappel.

3.4.2.2 Evaluation des techniques de fusion de décisions

A la fin de notre architecture de fusion en YU, nous plaçons une opération de fusion de décisions pour fusionner les sorties des deux classifieurs afin d'améliorer les performances en détection. Nous évaluons ici trois techniques de fusion : NMS, SVM et ELM. Les résultats de fusion sont présentés dans la table 3.2. Certains exemples de détection sont présentés dans la figure 3.11.

A partir de la table 3.2 et de la figure 3.11, nous pouvons voir que les trois techniques

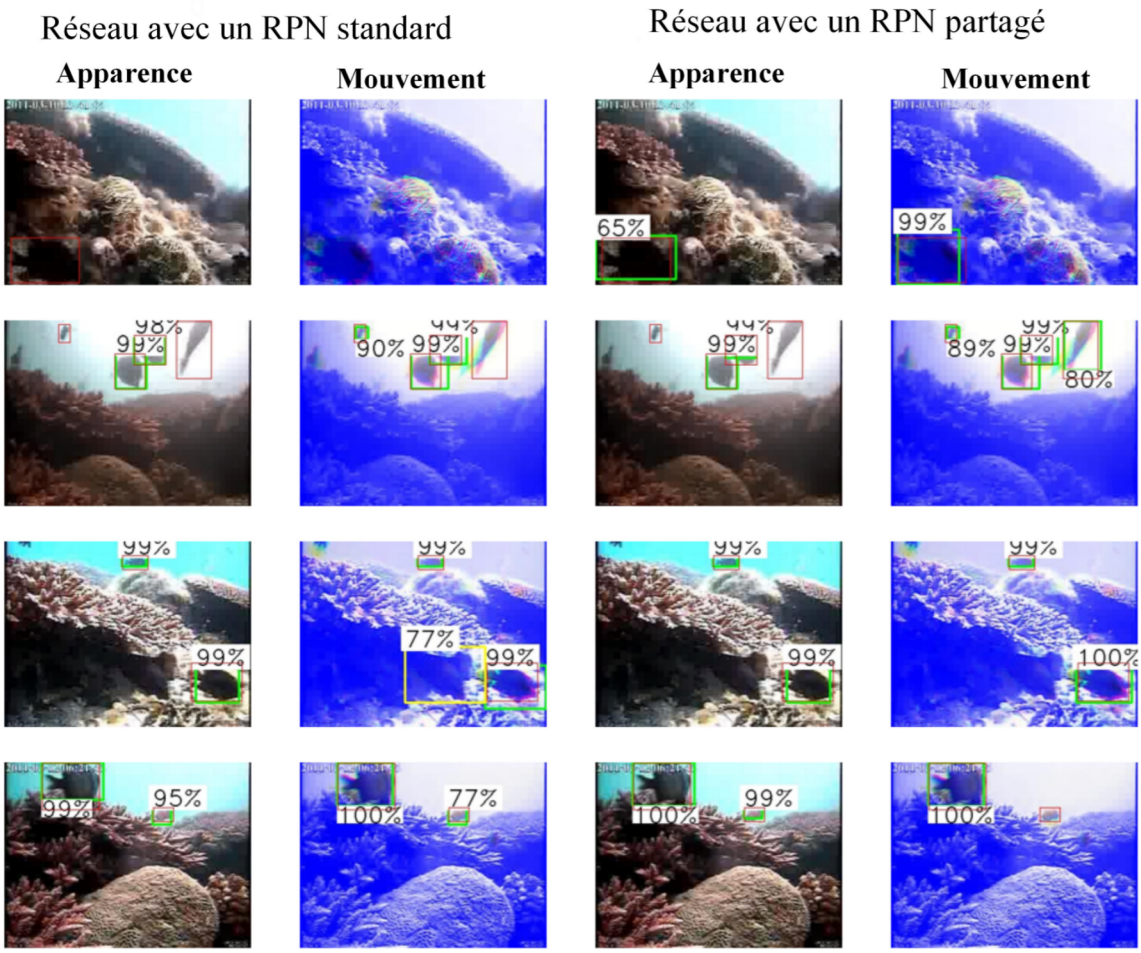


FIGURE 3.10 – Exemples de prédictions du Faster R-CNN (à un seul réseau) et de la fusion en YU. De gauche à droite : les deux premières colonnes sont des sorties du classifieur de Faster R-CNN entraîné sur des images RGB ou sur des images de mouvement. Les deux dernières colonnes sont respectivement des sorties du classifieur d'apparence et de mouvement de notre réseau parallèle en YU. Les boîtes rouges présentent les annotations vérité terrain, les boîtes vertes sont des poissons bien détectés et les boîtes jaunes sont les fausses alertes.

	Réseau	Fusion de décisions	F-mesure	AP
Fusion en YU	Apparence		79,47	67,04
	Mouvement		80,22	70,50
	Apparence + Mouvement	NMS	**83,16**	**73,69**
		SVM	81,59	70,08
		ELM	81,83	70,57

TABLE 3.2 – Performances en détection de poissons (taux en %) pour l'approche de fusion en YU, avec différentes techniques de fusion de décisions, sur la base LCF-15.

de fusion de décisions donnent de meilleurs taux F-mesure que l'utilisation de l'apparence seule ou du mouvement seul. Nous remarquons également que la technique NMS a de bien meilleures performances qu'ELM et SVM. La NMS accumule en quelque sorte les boîtes de détection issues des deux classifieurs pour une meilleure détection, augmentant ainsi la sensibilité ou le rappel (voir la première et la deuxième ligne de la figure 3.11). Aussi, nous pouvons voir dans la troisième ligne de la figure 3.11 que la NMS permet de réorganiser les boîtes par score et préserve celles de score le plus élevé. Par conséquent, la fusion NMS augmente l'AP. Cependant, avec cette technique, les fausses détections (fausses alertes) s'accumulent également (quatrième ligne), ce qui diminue la précision. D'un autre côté, ELM et SVM donnent des résultats de prédiction avec moins de faux positifs, mais certaines vraies détections sont également supprimées. Finalement, la meilleure F-mesure (**83,16%**) et la meilleure AP (**73,69%**) sont obtenues en utilisant la technique NMS.

3.4.3 Approche de fusion en UY

La stratégie de fusion en UY s'avère moins efficace que la fusion en YU pour la détection de poissons. Nous avons atteint une F-mesure de 74,12% et une AP de 62,85% sur la base de vidéos de référence LCF-15. Le problème avec cette architecture est que la détection de poissons dans un environnement sans contrainte est une tâche complexe qui conduit à un espace de caractéristiques très riche pour être traité avec un seul classifieur.

La figure 3.12 présente les courbes de précision-rappel pour les deux approches de fusion en YU et en UY. Comme nous pouvons le voir, la fusion en UY a une précision plus élevée que la fusion en YU en raison de moins de faux positifs, mais son rappel est faible. En revanche, la fusion en YU augmente le rappel en le passant de 60,12% à 76,03% sans considérablement réduire la précision de détection.

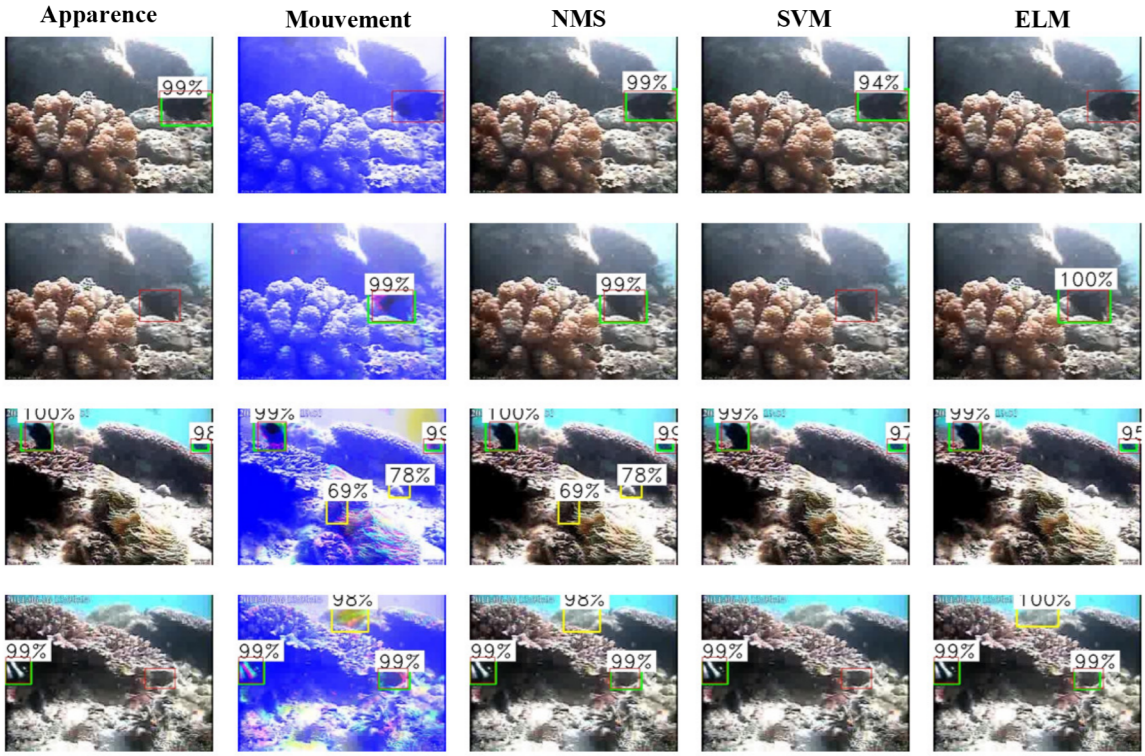
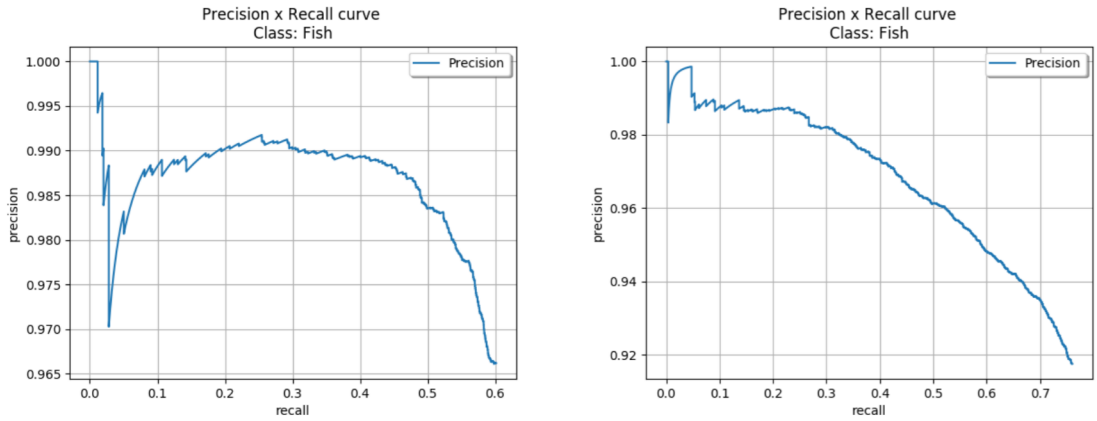


FIGURE 3.11 – Exemples de prédictions de la fusion en YU avec différentes techniques de fusion de décisions. De gauche à droite : les deux premières colonnes sont des sorties sans fusion du classifieur d'apparence et du classifieur de mouvement dans notre réseau parallèle en YU. Les trois dernières colonnes sont respectivement des sorties avec fusion NMS, SVM et ELM. Les boîtes rouges présentent les annotations vérité terrain, les boîtes vertes sont des poissons bien détectés et les boîtes jaunes sont les fausses alertes.



(a) Fusion en UY (b) Fusion en YU

FIGURE 3.12 – Courbes précision-rappel des deux approches de fusion de réseaux parallèles proposées.

3.4.4 Comparaison avec l'état de l'art

Dans cette section, nous comparons nos deux architectures avec des approches de l'état de l'art, en particulier avec des architectures basées sur la fusion d'informations. La table 3.3 présente les résultats comparatifs sur la base de vidéos de référence LCF-15.

Approche		Technique	F-mesure	AP	Architecture
Sans fusion		Faster R-CNN standard avec RGB (REN et al. 2015)	77,82	64,71	Un seul réseau
Fusion précoce	Fusion bas niveau	Approche de FARAHNAKIAN et HEIKKONEN 2020	78,78	67,49	
		Approche de SALMAN et al. 2020	80,02	-	
	Fusion au niveau intermédiaire	Fusion en Y (GUERRY, LE SAUX et FILLIAT 2017)	71,72	61,85	Réseaux parallèles
		Approche de ZHU et al. 2020	70,73	61,48	
		Fusion en UY	**74,12**	**62,85**	
Fusion tardive		Fusion en U (GUERRY, LE SAUX et FILLIAT 2017)	82,24	71,88	
		Fusion en X (GUERRY, LE SAUX et FILLIAT 2017)	**82,14**	**71,83**	
Fusion hybride		**Fusion en YU**	**83,16**	**73,69**	

TABLE 3.3 – Comparaison de performances en détection de poissons (taux en %) de nos approches de fusion et des approches de l'état de l'art, sur la base LCF-15.

A partir de la table 3.3, nous remarquons que la fusion précoce de bas niveau de l'entrée permet d'obtenir de meilleurs résultats que le Faster R-CNN standard sans fusion.

(Salman et al. 2020) ont proposé de fusionner la sortie de GMM, le flux optique et l'image en niveaux de gris. Cependant, le GMM présente de nombreux désavantages : les résultats de la segmentation ne sont pas robustes au bruit et sont sensibles aux variations d'illumination et à d'autres facteurs d'environnement tels que le mouvement des plantes aquatiques, les courants d'océan ou le tremblement de la caméra. La stratégie de la fusion au niveau intermédiaire fournit de mauvaises performances. Les autres stratégies de fusion, tardive et hybride, améliorent les performances en détection, notamment notre approche hybride de fusion en YU. Ces approches contiennent en effet deux classifieurs ce qui améliore les performances.

La figure 3.13 illustre des exemples de prédictions de la fusion en U et en X. Nous pouvons voir à partir de cette figure un avantage de la fusion en X qui a donné une nouvelle détection (deuxième ligne). Ceci est grâce à l'utilisation de la NMS intermédiaire : l'un des réseaux a trouvé une RoI qu'il n'a pas pu correctement classifier, mais l'autre réseau a pu la classifier (donnant lieu à une nouvelle détection) même si cette RoI n'a pas été proposée par ce réseau. Mais cette technique de fusion donne également de fausses alertes (troisième ligne). Dans ce cas, l'un des réseaux a trouvé une RoI qu'il a correctement classifiée mais l'autre réseau l'a mal classifiée donnant lieu à une fausse détection. La NMS intermédiaire ne conserve que les RoIs avec les scores les plus élevés et supprime les autres RoIs. Par conséquent, certaines fausses alertes de faibles scores ont été supprimées (quatrième ligne). Par contre, par ce biais certaines vraies détections ont également été supprimées (cinquième ligne).

Dans la fusion en U, il n'y a pas cet échange de RoIs mais un réordonnancement par score des détections lors de la NMS finale. Dans les exemples de la fusion en U de la figure 3.13 (première ligne), les détections provenant des deux réseaux sont accumulées pour une meilleure détection, mais avec cette fusion, de fausses alertes s'accumulent également (quatrième et dernière ligne). Cette approche améliore donc la sensibilité ou le rappel car un réseau peut compléter les détections manquées de l'autre, mais elle peut aussi diminuer la précision car les fausses détections s'accumulent également.

Contrairement à notre approche de fusion en YU, la fusion en U et en X n'entraînent pas les deux réseaux simultanément ; leurs deux réseaux sont entraînés indépendamment et ne sont fusionnés qu'à la fin de l'entraînement. Notre approche permet au RPN d'apprendre à partir d'un espace plus riche de caractéristiques d'apparence et de mouvement pour mieux proposer des RoIs.

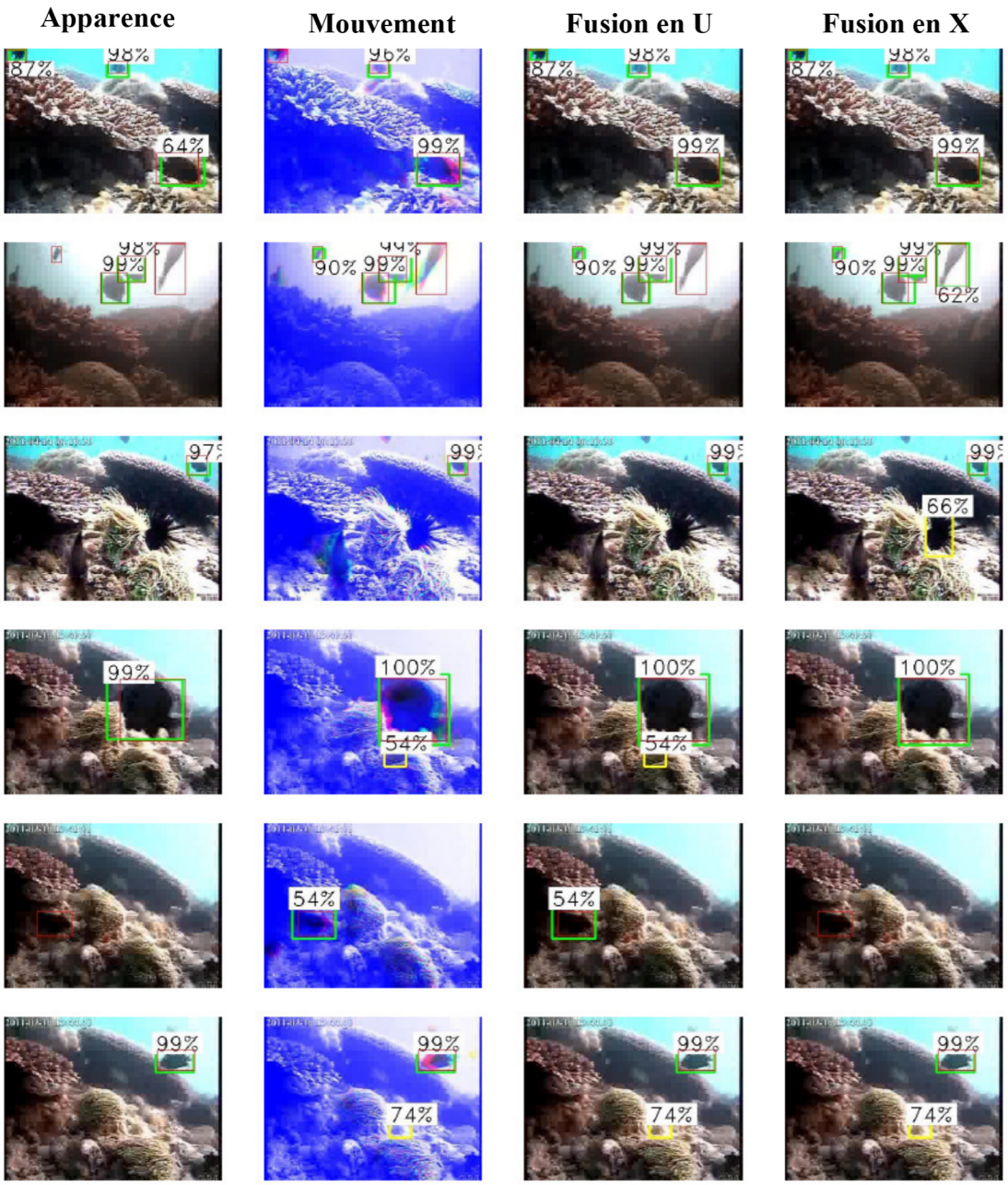


FIGURE 3.13 – Exemples de prédictions avec les fusions en U et en X. De gauche à droite : sorties du classifieur d'apparence, du classifieur de mouvement, de la fusion en U et de la fusion en X. Les boîtes rouges présentent les annotations vérité terrain, les boîtes vertes sont des poissons bien détectés et les boîtes jaunes sont les fausses alertes.

Nous pouvons voir clairement que les stratégies de la fusion au niveau intermédiaire avec un seul classifieur sont moins efficaces que les fusions tardives et hybrides avec deux classifieurs. Contrairement à notre architecture de fusion en UY, la fusion en Y et l'approche (Zhu et al. 2020) n'utilisent qu'un seul RPN et qu'un seul classifieur. L'espace de caractéristiques dans la fusion en Y est alors plus grand pour le RPN et pour le classifieur (il est plus grand pour le classifieur dans (Zhu et al. 2020)), ce qui rend l'entraînement très sensible. Dans (Zhu et al. 2020), le RPN n'utilise qu'un seul type de données pour générer des RoIs. Par conséquent, il n'utilise pas des informations complémentaires qui pourraient être pertinentes pour proposer plus de régions fiables. Ces trois architectures sont des architectures de fusion précoce, elles fusionnent des caractéristiques de poissons extraites de deux réseaux parallèles. La dimension des caractéristiques fusionnées devient grande, ce qui augmente le risque de sur-apprentissage.

Nous concluons ainsi que les fusions de réseaux parallèles avec deux classifieurs améliorent significativement les performances de détection automatique de poissons, notamment notre architecture de fusion en YU. Nous obtenons une F-mesure de **83,16%** et une AP de **73,69%**, alors que les fusions en U et en X ont respectivement des F-mesures de 82,24% et 82,14% et des AP de 71,88% et 71,83%. Par conséquent, les approches proposées surpassent les méthodes de l'état de l'art.

3.5 Conclusion

Dans ce chapitre, nous avons présenté deux nouvelles approches de fusion de réseaux parallèles, et nous les avons appliquées pour la détection automatique de poissons. Ces approches sont basées sur la fusion de deux Faster R-CNNs pour améliorer les performances de détection et de localisation. L'utilisation d'un seul RPN, ou d'un seul classifieur, partagé entre deux Faster R-CNNs permet au modèle de profiter d'un espace de caractéristiques relativement plus riche. Cet espace est constitué de la fusion des caractéristiques fournies par les deux réseaux CNN parallèles. Dans notre application, nous avons utilisé des images vidéo RGB pour capturer les caractéristiques d'apparence, et le flux optique combiné avec deux images en niveaux de gris successives pour capturer les caractéristiques de mouvement des poissons à détecter. Le but de cette fusion est de détecter des poissons en apprenant au système la relation qui existe entre les images successives. Les expériences de validation sur la base de vidéos de référence LCF-15 ont démontré que nos approches de fusion surpassent

les méthodes de l'état de l'art pour la détection automatique de poissons.

Le but de ce chapitre était de localiser les poissons dans des images vidéo sous-marines. Une fois les poissons sont détectés, l'objectif suivant est d'identifier leurs espèces. La classification d'espèces de poissons fera l'objet des chapitres suivants.

Chapitre 4

Classification d'espèces de poissons dans des images vidéo sous-marines

Sommaire

4.1 Introduction

Nous avons abordé la détection de poissons dans le chapitre précédent, nous présentons ici la deuxième partie du système de reconnaissance automatique d'espèces de poissons, c'est-à-dire la classification d'espèces de poissons. Pour cela, nous allons supposer que le poisson a été déjà détecté dans une image sous-marine et la boîte englobante le poisson est l'entrée du modèle de classification.

Les premiers travaux de classification d'espèces ont proposé d'utiliser des techniques traditionnelles telles que l'approche d'analyse discriminante (Spampinato et al. 2010), la sélection de caractéristiques (Huang, Boom et Fisher 2012), l'histogramme de gradient orienté (Cabrera-Gámez et al. 2015) et SURF (Szűcs, Papp et Lovas 2015). Les travaux récents se sont basés sur les CNNs. Certains ont appliqué les réseaux populaires pré-entraînés pour la classification d'espèces de poissons (Jäger et al. 2016 ; Li et al. 2015 ; Sun et al. 2016, 2018 ; Villon et al. 2016). D'autres ont proposé leurs propres architectures mais elles ne sont pas profondes (juste trois couches convolutives) (Qin et al. 2015 ; Salman et al. 2016). (Qin et al. 2016) ont proposé une architecture profonde hybride avec des méthodes traditionnelles (ACP, hachage binaire et histogramme par blocs) pour extraire les caractéristiques d'images de poisson. Dans ce travail, nous proposons une approche basée sur l'apprentissage profond pour la classification d'espèces de poissons capturés dans un environnement sous-marin naturel.

Les bases d'images sous-marines sont d'un nombre limité d'images d'entraînement. Par conséquence, il n'est pas recommandé d'entraîner un CNN profond à partir de zéro avec les images disponibles en raison du grand nombre de paramètres à entraîner. Pour surmonter ce problème, nous utilisons la technique de l'apprentissage par transfert (Pan et Yang 2009) pour extraire des caractéristiques et/ou ré-entraîner un réseau pré-entraîné tout en abordant plusieurs problématiques.

Les approches de l'état de l'art basées sur les CNNs pour l'identification de poissons ont utilisé des images RGB sans tester d'autres espaces colorimétriques. Pourtant, dans d'autres applications, il a été montré l'intérêt d'utiliser d'autres espaces pour la classifica-

tion d'objets (Kasaei et al. 2020 ; Kim, Park et Jung 2018). Dans notre travail, nous nous intéressons à explorer différents espaces colorimétriques afin de choisir le meilleur pour cette application.

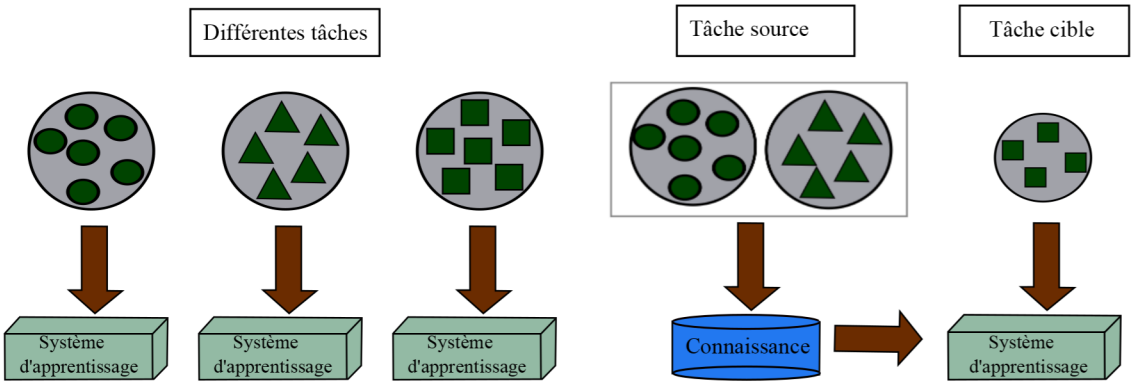
Dans (Qin et al. 2016), l'arrière-plan a été éliminé en utilisant des masques de poissons fournis dans la base d'images. L'étape d'élimination de l'arrière-plan peut n'être qu'une complexité supplémentaire sans réel intérêt pour la classification. D'un côté, le fond marin est très riche et peut fournir des caractéristiques qui perturbent la classification, d'où la nécessité d'éliminer le bruit de fond. En revanche, la richesse du fond rend très difficile la délimitation de la zone de poisson. C'est pourquoi il est tout à fait légitime de se demander si le fond doit ou non être éliminé avant la classification.

Par ailleurs, dans la littérature sur les méthodes de classification de poisson utilisant l'apprentissage profond, nous trouvons des travaux utilisant l'augmentation artificielle de données pour améliorer les performances du modèle et éviter le sur-apprentissage. Mais, en général, l'augmentation artificielle de données est appliquée sur toutes les images d'entraînement même si la base d'images est déséquilibrée (Sun et al. 2018). Parfois, le nombre d'images d'entraînement est augmenté uniquement pour les classes ayant moins d'exemples afin d'équilibrer le nombre d'exemples entre différentes classes (Qin et al. 2016). Cependant, l'augmentation artificielle de données nécessite de ressources de mémoire et de processeur. Par conséquent, il pourrait être nécessaire de procéder à une augmentation de données uniquement pour les classes qui sont difficiles à classifier. Nous proposons ici d'augmenter le nombre d'images en utilisant un nouveau critère basé sur les courbes des fonctions de perte d'apprentissage et de validation.

Le reste de ce chapitre est organisé comme suit : nous décrivons dans la section 4.2 comment nous utilisons le concept de l'apprentissage par transfert pour notre application. Ensuite, nous proposons une augmentation artificielle ciblée d'images de poissons dans la section 4.3. Dans la section 4.4, nous visualisons les filtres et les cartes de caractéristiques du modèle AlexNet que nous proposons d'utiliser. Nous présentons dans la section 4.5 notre approche proposée pour la classification d'espèces de poissons. Ensuite, nous fournissons les résultats expérimentaux sur deux bases de référence dans la section 4.6. Finalement, une conclusion du chapitre sera présentée dans la section 4.7.

4.2 Apprentissage par transfert

L'entraînement d'un CNN nécessite un très grand volume de données car il doit apprendre des millions de paramètres. Actuellement, la plupart des travaux entraînent un CNN à partir d'un modèle pré-entraîné au lieu de partir de zéro. Cette méthode, appelée apprentissage par transfert (Pan et Yang 2009), est une solution pratique pour appliquer l'apprentissage profond sans nécessiter un très grand jeu de données, ni un entraînement très long. La figure 4.1 montre la différence entre les processus d'apprentissage de la technique traditionnelle (4.1(a)) et l'apprentissage par transfert (4.1(b)). Comme nous pouvons le voir, l'apprentissage automatique traditionnel vise à apprendre chaque tâche à partir de zéro, tandis que l'apprentissage par transfert vise à transférer les connaissances de certaines tâches précédentes vers une tâche cible lorsque cette dernière a moins de données d'entraînement.



(a) Apprentissage automatique traditionnel (b) Apprentissage automatique par transfert

Figure 4.1 – Différents processus d'apprentissage : (a) l'apprentissage automatique traditionnel et (b) l'apprentissage par transfert.

Dans notre travail, nous proposons une approche basée sur l'apprentissage par transfert pour la tâche de classification d'espèces de poissons dans des images sous-marines à faible contraste et de mauvaise résolution. Nous transférons les paramètres d'un CNN entièrement entraîné sur la base ImageNet (Deng et al. 2009) et nous ré-entraînons ce CNN en utilisant une quantité limitée d'images sous-marines. Nous pouvons formaliser le problème de transfert comme suit :

— Nous considérons d'abord un domaine source $\mathcal{D}_S = \{\mathcal{X}_S, P(X_S)\}$ représentant le problème d'apprentissage d'ImageNet, où X_S est les exemples d'apprentissage

d'ImageNet, $\mathcal{X}_S$ est la sortie d'espace des caractéristiques de CNN et $P(.)$ est la distribution de probabilité marginale. Le domaine source est lié à une tâche $\mathcal{T}_S = \{\mathcal{Y}_S, f_S\}$ qui est la classification d'ImageNet avec un CNN profond et se compose de deux composantes : l'espace d'étiquettes $\mathcal{Y}_S$ et une fonction de prédiction f_S. Ici, la fonction f_S est le modèle CNN profond constituée de l'ensemble des paramètres des couches de CNN qui peuvent être appris à partir des données d'apprentissage.

— Ensuite, nous définissons un domaine cible $\mathcal{D}_C = \{\mathcal{X}_C, P(X_C)\}$ qui représente dans notre application le problème de classification d'espèces de poissons, où X_C est les exemples d'apprentissage de la base d'images de poissons, $\mathcal{X}_C$ est l'espace des caractéristiques de poissons, et une tâche d'apprentissage $\mathcal{T}_C = \{\mathcal{Y}_C, f_C\}$ qui consiste à entraîner la fonction f_C à l'aide du domaine source, où $\mathcal{Y}_C$ est l'espace d'étiquettes de poissons.

— L'apprentissage par transfert vise à aider à améliorer l'apprentissage de la fonction d'objectif cible $f_C(.)$ en $\mathcal{D}_C$ en utilisant les connaissances en $\mathcal{D}_S$ et $\mathcal{T}_S$, où $\mathcal{D}_S \neq \mathcal{D}_C$ ou $\mathcal{T}_S \neq \mathcal{T}_C$. La condition $\mathcal{D}_S \neq \mathcal{D}_C$ implique que $\mathcal{X}_S \neq \mathcal{X}_C$ ou $P(X_S) \neq P(X_C)$ et la condition $\mathcal{T}_S \neq \mathcal{T}_C$ implique que $\mathcal{Y}_S \neq \mathcal{Y}_C$ ou $f_S \neq f_C$. Dans notre application, les domaines source et cible sont différents, c'est-à-dire $\mathcal{X}_S \neq \mathcal{X}_C$ et $\mathcal{Y}_S \neq \mathcal{Y}_C$.

Généralement, les deux principaux scénarios d'apprentissage par transfert se présentent comme suit :

— Utiliser le modèle pour extraire automatiquement des caractéristiques des images. Dans ce cas, on exploite uniquement une partie du réseau pré-entraîné. On l'utilise comme extracteur de caractéristiques des images pour alimenter un nouveau classifieur, par exemple un SVM.

— Utiliser le modèle pré-entraîné pour initialiser un autre modèle qui est ensuite ré-entraîné pour finaliser l'apprentissage pour traiter le nouveau problème de classification. L'intérêt est double : on utilise une architecture optimisée avec soin par des spécialistes, et l'on profite des capacités d'extraction de caractéristiques apprises sur un jeu de données de qualité. Cette stratégie consiste en quelques sortes à prendre un système visuel déjà bien entraîné sur une tâche de classification pour le raffiner sur une tâche similaire.

Nous allons expérimenter les deux sénarios dans notre travail pour identifier les espèces de poissons dans des images sous-marines.

4.3 Augmentation artificielle d'images de poissons

Pour améliorer la performance du modèle CNN et limiter le sur-apprentissage et surmonter le problème de manque d'images suffisantes, nous utilisons la technique d'augmentation artificielle de données décrite dans la section 2.4.3.2.

Dans la plupart des travaux, les images de toutes les classes sont augmentées de la même manière, même dans le cas où l'ensemble d'entraînement est déséquilibré (Sun et al. 2018). Dans d'autres travaux, seules les images des classes ayant un nombre d'effectifs inférieur à un seuil sont augmentées afin d'équilibrer la base. Par exemple dans (Qin et al. 2016), les auteurs ont augmenté le nombre d'exemples pour les espèces qui ont moins de 300 exemples dans la base d'entraînement. Cependant, l'augmentation de données demande plus de mémoire et de ressources ainsi que le réseau demande plus de temps d'entraînement. La question qui se pose ici, faut-il vraiment augmenter le nombre d'exemples pour toutes les classes ? Si non, nous devrons augmenter les exemples de quelles classes ? Dans notre travail, nous proposons d'augmenter le nombre d'exemples en utilisant un nouveau critère basé sur les courbes de perte d'apprentissage et de validation. Cette technique consiste à augmenter uniquement le nombre d'exemples pour les classes ayant des courbes de perte non convergentes.

Nous utilisons quatre techniques d'augmentation artificielle d'images, à savoir, l'effet miroir ou le retournement horizontal, le recadrage, le redimensionnement et la rotation, qui permettent toutes de générer des images transformées à partir d'une image originale avec la même étiquette (figure 4.2). Nous retournons d'abord toutes les images de la classe considérée par l'augmentation horizontalement pour simuler des poissons nageant dans la direction opposée. Ensuite, comme les poissons peuvent se présenter à n'importe quelle distance devant la caméra, nous redimensionnons les images pour faire paraître les poissons un peu éloignés de la caméra. Nous recadrons également les images en supprimant un quart de chaque côté pour simuler l'environnement sous occlusion. Enfin, nous faisons tourner les images de poissons avec différents angles (-20^o, -10^o, 10^o et 20^o) pour le problème de classification de poissons en rotation invariante. Nous ne faisons pas de retournement vertical en raison de l'observation des poissons qui ne tournent jamais verticalement. La figure 4.2 illustre ces différentes techniques de l'augmentation artificielle de données appliquées sur une image de poisson.

FIGURE 4.2 – Exemples de différentes techniques de l'augmentation artificielle de données appliquées sur une image de poisson.

4.4 Analyse du modèle pré-entraîné AlexNet

Nous utilisons dans notre travail le modèle AlexNet pré-entraîné sur la base ImageNet. AlexNet (KRIZHEVSKY, SUTSKEVER et HINTON 2012) a été le premier réseau qui a permis à l'apprentissage profond de remonter à la surface. Il a une structure simple (8 couches profondes) qui demande moins de ressources et le rend plus rapide que d'autres réseaux comme GoogleNet (SZEGEDY et al. 2015) (22 couches profondes) et VGG (SIMONYAN et ZISSERMAN 2014b) (au moins 16 couches convolutives). Le nombre élevé de couches dans d'autres architectures rend difficile le fine-tuning des paramètres transférés, en particulier avec un nombre limité de données d'apprentissage. En effet, AlexNet est pré-entraîné sur le jeu de données ImageNet, ce qui signifie que le modèle a appris des caractéristiques riches pour une large gamme d'images. Pour montrer cette richesse de représentation, nous visualisons les filtres de convolution ainsi que leurs sorties.

4.4.1 Analyse des filtres

Nous commençons par visualiser les poids des filtres de convolution. Ceux-ci sont généralement les plus interprétables sur la première couche de convolution qui est appliquée directement sur les données brutes. La visualisation de ces poids est utile car les réseaux bien entraînés affichent habituellement des filtres lisses et sans motifs irréguliers. Les irrégularités peuvent être un indicateur d'un réseau qui n'a pas été assez entraîné pendant une durée suffisante, ou d'un réseau en sur-apprentissage à cause d'une très faible force de régularisation. La figure 4.3 montre les 96 filtres de la première couche de convolution, chacun de taille $11 \times 11 \times 3$. Le réseau a appris une variété de noyaux sélectifs en fréquence et d'orientations différentes. Les filtres codés en couleur extraient les caractéristiques de basses fréquences et ceux en gris extraient les caractéristiques des hautes fréquences. Nous voyons que les filtres sont lisses, propres et diverses ; ce qui reflète un réseau bien convergé.

FIGURE 4.3 – Visualisation des 96 filtres de la première couche de convolution (KRIZHEVSKY, SUTSKEVER et HINTON 2012).

4.4.2 Analyse des cartes de caractéristiques

Pour visualiser les cartes de caractéristiques, la technique de visualisation la plus directe est de montrer les activations du réseau pendant la phase avant. Les activations commencent habituellement à apparaître relativement denses, mais à mesure que l'apprentissage progresse, les activations deviennent généralement plus clairsemées et localisées. On

note que, avec cette visualisation, certaines cartes de caractéristiques peuvent être toutes à zéro pour de nombreuses entrées, ce qui peut indiquer des filtres morts ou des taux d'apprentissage élevés. La figure 4.4 montre les 96 cartes de caractéristiques de la première couche de convolution et la figure 4.5 illustre les 128 cartes de caractéristiques de la dernière couche de convolution en fournissant en entrée du modèle AlexNet une image de poisson avec un fond sous-marin. Nous constatons que la première couche de convolution vise à détecter des informations globales ou des caractéristiques de bas niveau comme les contours, la texture et les attributs de forme dans l'image d'entrée. Cette couche sépare également le poisson du fond. Par conséquence, nous n'avons pas besoin d'utiliser de masque pour supprimer le fond dans l'image d'entrée. Cette idée sera prouvée par nos expériences. Nous pouvons voir aussi que les sorties de cette couche sont denses mais celles de la dernière couche sont clairsemées et localisées.

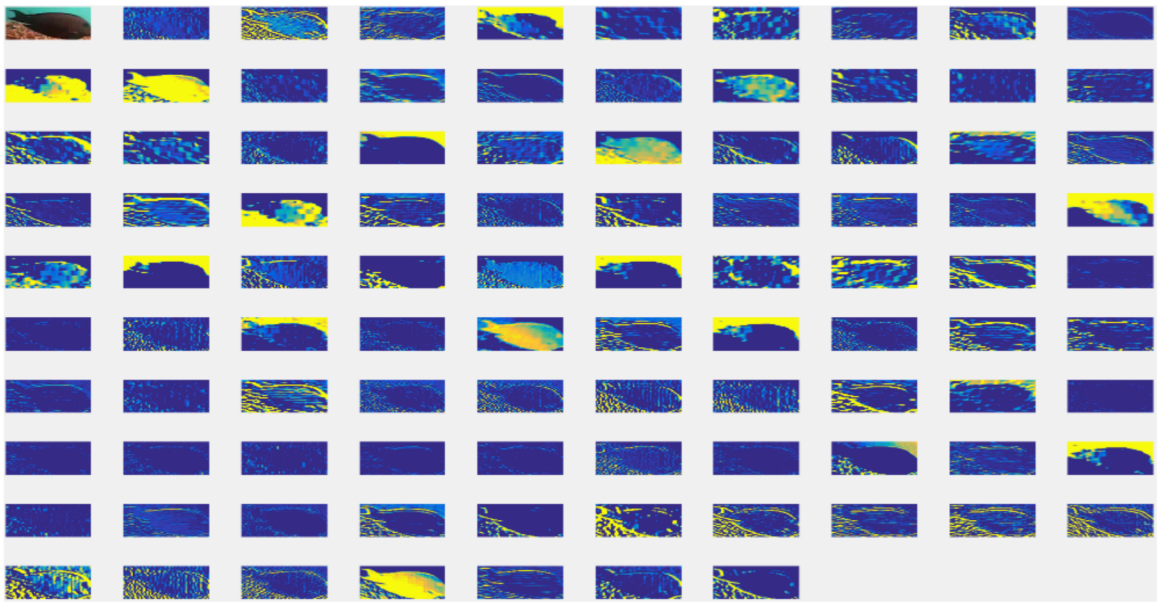


FIGURE 4.4 – La visualisation de 96 cartes de caractéristiques de la première couche de convolution pour une image d'entrée de poisson avec un fond sous-marin.

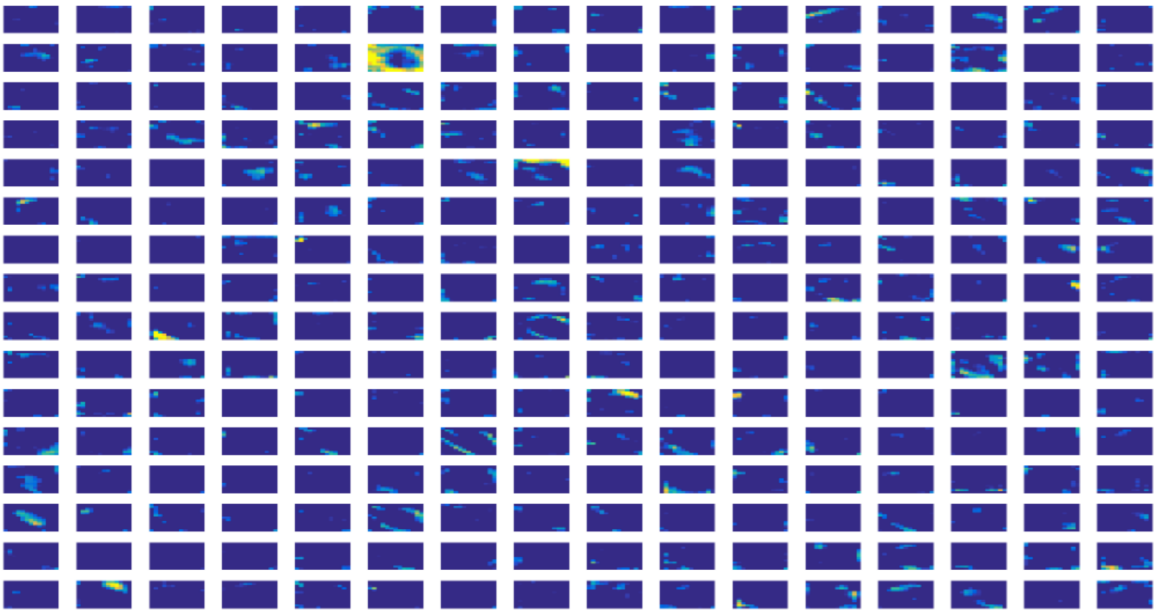


FIGURE 4.5 – La visualisation de 128 cartes de caractéristiques de la dernière couche de convolution pour une image d'entrée de poisson avec un fond sous-marin.

4.5 Modèle CNN proposé pour la classification d'espèces de poissons

Les bases d'images annotées de poissons vivant dans un environnement marin naturel ne sont pas assez grandes pour entraîner des CNNs à partir de zéro pour la classification d'espèces de poissons. De plus, d'immenses ressources de mémoire et de processeur sont nécessaires. Pour cela, nous proposons de transférer les connaissances du modèle AlexNet pré-entraîné sur la base ImageNet vers notre domaine cible, comme illustré dans la figure 4.6. Cette figure montre la structure globale de notre approche basée sur l'apprentissage par transfert du modèle AlexNet. Dans ce travail, nous proposons d'utiliser les deux modes d'apprentissage par transfert dans trois stratégies pour la classification d'espèces de poissons :

— **stratégie 1 / Extracteur de caractéristiques fixe :** nous extrayons des caractéristiques d'images de poissons en utilisant AlexNet directement sans fine-tuning sur notre base de poissons. Pour cela, nous supprimons certaines couches (par exemple $FC8$, $FC7$ ou $FC6$) et utilisons la sortie du reste du réseau comme descripteurs de

caractéristiques afin d'alimenter un classifieur SVM. Nous désignons cette stratégie par *CNN-SVM*.

— **stratégie 2 / Le fine-tuning :** nous ré-entraînons AlexNet en remplaçant les dernières couches entièrement connectées ($FC6$, $FC7$ ou $FC8$) par de nouvelles couches entièrement connectées. La couche $FC8$ est remplacée par une nouvelle couche entièrement connectée de N sorties correspondant au nombre d'espèces dans notre base d'images. Ensuite, nous ré-entraînons uniquement à partir de zéro les nouvelles couches et nous gardons les paramètres des couches précédentes. Le modèle utilise la fonction *Softmax* pour la classification. Nous désignons cette stratégie par *CNN-Soft*.

— **stratégie 3 / Combinaison du fine-tuning et l'extraction de caractéristiques :** nous ré-extrayons les caractéristiques d'images de poissons en utilisant cette fois-ci le module ré-entraîné et le classifieur SVM. Nous désignons cette stratégie par *CNN-FT-SVM*.

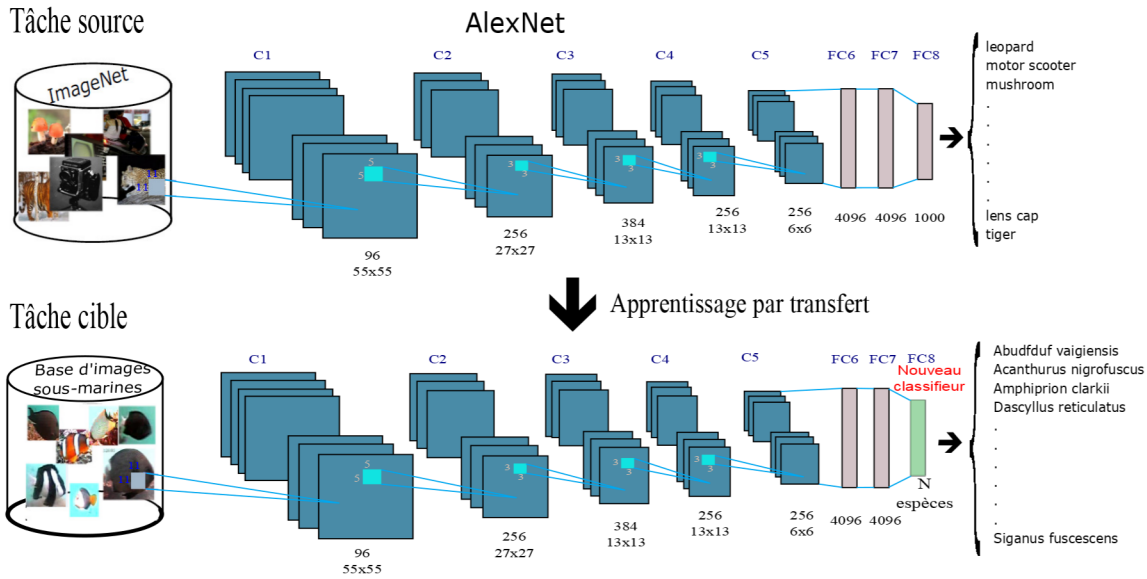


FIGURE 4.6 – L'approche proposée pour la classification d'espèces de poissons basée sur l'apprentissage par transfert.

Nous notons que nous utilisons un classifieur SVM linéaire dont les paramètres sont optimisés par la validation croisée.

4.6 Résultats expérimentaux

Dans cette section, nous présentons les résultats expérimentaux sur deux bases d'images de référence sous-marines : la base d'images FRGT (Boom et al. 2012a,b) et la base d'images LCF-15 (Joly et al. 2015). Nous évaluons l'efficacité de nos trois stratégies en adoptant deux métriques, le taux de classification ou *Accuracy* (AC) et la précision moyenne (PM).

$$AC = \frac{\sum_{i=1}^{N} \mathrm{VP}_i}{\sum_{i=1}^{N} (\mathrm{VP}_i + \mathrm{FP}_i)} \tag{4.1}$$

$$\mathrm{PM} = \frac{1}{N} \sum_{i=1}^{N} \frac{\mathrm{VP}_i}{\mathrm{VP}_i + \mathrm{FP}_i} \tag{4.2}$$

où N est le nombre de classes dans la base d'images.

Tout d'abord, nous évaluons différents espaces colorimétriques afin de choisir le meilleur espace (section 4.6.1). Ensuite, nous testons différents paramètres optimaux de l'architecture (section 4.6.2). Nous montrons ensuite dans quelle mesure certains prétraitements tels que l'augmentation artificielle de données et l'élimination de l'arrière-plan peuvent aider à améliorer les résultats (section 4.6.3). Finalement, nous comparons avec les approches de l'état de l'art (section 4.6.4).

4.6.1 Meilleur espace colorimétrique

Le choix de l'espace colorimétrique pose la principale difficulté dans le cadre de classification d'images couleurs, en particulier avec les images sous-marines où la luminosité est relativement faible. Afin de sélectionner le meilleur espace colorimétrique pour la classification d'images de poissons, nous ré-entraînons le modèle AlexNet sur huit espaces colorimétriques (RGB, niveau de gris, LAB, YCbCr, HSI, HSV, XYZ et LUV). Nous remplaçons la couche de classification $FC8$ d'AlexNet par une nouvelle couche entièrement connectée, initialisée aléatoirement de N sorties correspondant à N espèces de poissons ($N = 23$ pour la base d'images FRGT). Pour chaque espace colorimétrique choisi, nous ré-entraînons le modèle sur des images d'entrée brutes. La table 4.1 présente les résultats de

classification sur la base d'images FRGT pour les huit espaces colorimétriques considérés.

Espace colorimétrique	AC (%)	PM (%)
RGB	**99,18**	**95,49**
Niveau de gris	98,51	89,83
LAB	65,27	16,02
YCbCr	45,82	6,74
HSI	44,16	4,5
HSV	29,86	8,07
XYZ	27,64	13,09
LUV	15,27	7,54

TABLE 4.1 – Comparaison des performances en classification de poissons sur la base d'images FRGT pour différents espaces colorimétriques.

Nous pouvons voir à partir de la table 4.1 que l'utilisation d'images de poissons sous le format RGB donne de meilleurs résultats que les autres espaces colorimétriques. En effet, AlexNet a déjà été pré-entraîné sur des images RGB, ce qui signifie que les poids des filtres sont plus liés aux images RGB qu'aux autres espaces colorimétriques. De plus, dans l'environnement sous-marin, le spectre visible est modifié avec la profondeur. Les radiations de fréquences basses sont plus absorbées. Ainsi, lorsque la composante rouge disparaît déjà en eau peu profonde (5 m), la composante verte disparaît à environ 50 m et la composante bleue est absorbée à environ 60 m. En conséquence, dans la mer plus profonde, nous obtenons des scènes bleu-vert (BIANCO et al. 2015) ; pour cette raison, les composantes bleu et vert fournissent des informations beaucoup plus discriminantes. Les espaces colorimétriques basés sur l'intensité ou la luminosité comme LAB, YCbCr, HSI, HSV, XYZ et LUV produisent les performances les plus médiocres en raison de la faible luminosité dans l'environnement sous-marin. Ainsi, nous prenons l'espace colorimétrique RGB comme entrée pour les stratégies proposées dans les expériences suivantes.

4.6.2 Optimisation des paramètres

Dans cette section, nous évaluons les trois stratégies proposées en modifiant leurs options possibles. En particulier, nous comparons les performances d'utilisation de différentes couches du CNN dans les trois stratégies proposées :

— **stratégie *CNN-SVM* :** nous extrayons les caractéristiques de poissons en utili-

Stratégie	CNN-SVM-			CNN-Soft-			CNN-FT-SVM-		
Activation	pool5	FC6	FC7	FC6	FC7	FC8	pool5	FC6	FC7
AC (%)	**98,91**	98,59	98,40	99,12	99,20	**99,24**	99,19	99,32	**99,47**
PM (%)	**94,02**	93,42	92,88	93,24	93,80	**94,08**	94,79	**95,38**	94,84

TABLE 4.2 – Comparaison des performances sur la base d'images FRGT pour différentes options de couches des trois stratégies proposées.

sant l'activation de trois couches cachées (*pool*5, $FC6$ ou $FC7$) afin d'alimenter un classifieur SVM multi-classe linéaire.

— **stratégie *CNN-Soft* :** nous ré-entraînons le réseau à différentes couches. Tout d'abord, nous ré-entraînons à partir de zéro uniquement $FC8$ (*CNN-Soft-FC8*). Deuxièmement, nous ré-entraînons les couches $FC7$ et $FC8$ (*CNN-Soft-FC7*) et enfin, nous ré-entraînons toutes les couches entièrement connectées (*CNN-Soft-FC6*). Dans toutes les options de la stratégie *CNN-Soft*, $FC8$ est remplacée par une nouvelle couche entièrement connectée qui a N neurones correspondant aux N espèces de la base d'images considérée ($N = 23$ pour FRGT et $N = 15$ pour LCF-15). Dans *CNN-Soft-FC7*, la couche $FC7$ est remplacée par une nouvelle couche entièrement connectée de 4096 neurones et dans *CNN-Soft-FC6*, les couches $FC6$ et $FC7$ sont remplacées par de nouvelles couches entièrement connectées de 4096 neurones chacune.

— **stratégie *CNN-FT-SVM* :** nous utilisons le modèle ré-entraîné pour ré-extraire des caractéristiques comme dans la stratégie *CNN-SVM* à partir de l'activation des couches *pool*5, $FC6$ ou $FC7$.

La table 4.2 montre les résultats de la classification sur la base d'images FRGT. Pour la première stratégie, nous observons que la classification des caractéristiques extraites de la couche *pool*5 fournit de meilleurs résultats que la classification des caractéristiques extraites des couches $FC6$ et $FC7$. Cela est dû au fait que les couches de convolution capturent des caractéristiques universelles qui pourraient être pertinentes pour notre tâche, mais les couches entièrement connectées sont davantage liées aux détails des objets dans la base d'images ImageNet. Cependant, pour la troisième stratégie, l'étape de fine-tuning rend les couches $FC6$ et $FC7$ plus liées aux détails d'espèces de poissons contenues dans notre base d'images ; nous obtenons les meilleures ACs de **99,32%** et **99,47%** en utilisant respectivement les couches $FC6$ et $FC7$. Dans la deuxième stratégie, le fine-tuning du

modèle améliore les performances du système, en particulier lorsque nous ré-entraînons uniquement la couche $FC8$ à partir de zéro avec la fonction *Softmax*. De plus, nous pouvons voir que la classification avec SVM après le fine-tuning est légèrement meilleure que la classification avec la fonction *Softmax*.

Dans les expériences suivantes, nous n'utiliserons que *CNN-SVM-pool5*, *CNN-Soft-FC8* et *CNN-FT-SVM-FC7* et nous les désignerons respectivement par *CNN-SVM*, *CNN-Soft* et *CNN-FT-SVM*. Les résultats de ces trois stratégies sur la base d'images LCF-15 sont présentés dans la table 4.3 en termes du taux de classification et de la précision moyenne. Nous pouvons voir clairement que l'extraction des caractéristiques après l'étape du fine-tuning du modèle donne de meilleures performances que l'extraction des caractéristiques sans fine-tuning.

Métrique	CNN-SVM	CNN-Soft	CNN-FT-SVM
AC (%)	65,38	75,76	**77,29**
PM (%)	57,79	60,21	**64,43**

TABLE 4.3 – Performances en classification d'espèces de poissons des différentes stratégies proposées sur la base d'images LCF-15.

4.6.3 Prétraitement des images d'entrée

Afin d'améliorer les performances, des techniques de prétraitement telles que l'élimination de l'arrière-plan (QIN et al. 2016) et/ou l'augmentation artificielle de données (QIN et al. 2016 ; SALMAN et al. 2016) peuvent être réalisées. Les effets de ces techniques seront étudiés dans cette section.

4.6.3.1 Elimination de l'arrière-plan

Dans la base d'images FRGT, chaque image de poisson a un masque binaire. Nous proposons de tester nos stratégies sur des images de poissons avec et sans l'arrière-plan (figure 4.7). Nous donnons dans la table 4.4 les résultats de classification d'images de la base FRGT avec et sans élimination de l'arrière-plan. Nous pouvons observer que l'élimination de l'arrière-plan en utilisant les masques n'améliore pas les performances du système. Nous avons obtenu un AC de 99,45% en utilisant les images de poissons avec l'élimination de l'arrière-plan. Or, sans utiliser la technique d'élimination de l'arrière-plan, nous obtenons

un AC de 99,47%. Le modèle AlexNet a appris une variété de filtres sélectifs extrayant des caractéristiques d'image à différentes orientations et échelles (figure 4.3), rendant le modèle efficace pour extraire des caractéristiques globales comme les contours, les bords, les formes, les couleurs et les textures (figure 4.4). Ces caractéristiques aident à séparer les poissons de l'arrière-plan. L'élimination de l'arrière-plan peut être une tâche compliquée qui prend du temps et de la mémoire, surtout lorsque nous n'avons pas d'images de l'arrière-plan ou de vidéos qui facilitent la tâche de soustraction. Il est donc très important d'éviter cette étape dans de telles situations, notamment pour les applications en temps réel.

(a) (b) (c)

FIGURE 4.7 – Exemple d'élimination de l'arrière-plan. (a) : image originale, (b) : masque de poisson, (c) : poisson au premier-plan.

	Sans élimination de l'arrière-plan		Avec élimination de l'arrière-plan	
Stratégie	AC (%)	PM (%)	AC (%)	PM (%)
CNN-SVM	**98,91**	**94,02**	98,57	93,34
CNN-Soft	**99,24**	**94,08**	96,61	76,03
CNN-FT-SVM	**99,47**	94,84	99,45	**95,35**

TABLE 4.4 – Comparaison des performances sur la base FRGT avec et sans élimination de l'arrière-plan.

4.6.3.2 Augmentation artificielle de données

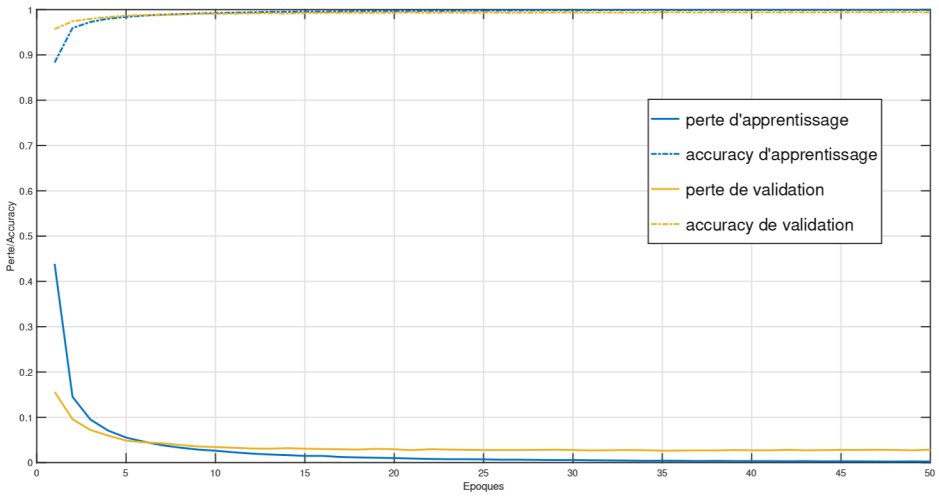
Dans cette section, nous évaluons l'effet de l'augmentation artificielle de données sur les performances de notre modèle. Nous proposons d'augmenter le nombre d'exemples pour les espèces ayant une courbe de fonction de perte non convergente.

La figure 4.8 illustre les courbes des fonctions de perte du fine-tuning AlexNet et de validation par époques sur les bases d'images FRGT 4.8(a) et LCF-15 4.8(b). L'inspection des

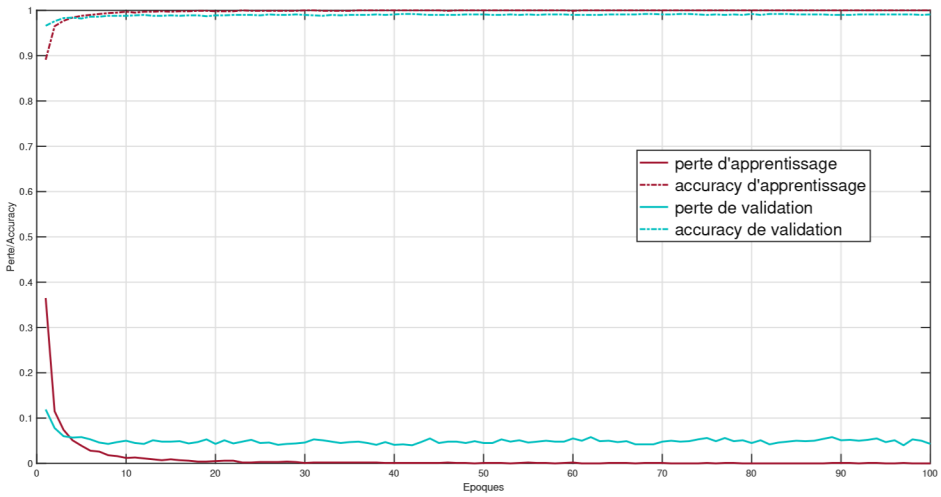
courbes des fonctions de perte d'apprentissage et de validation ne suffit pas pour conclure si le modèle a bien appris toutes les classes. En effet, sur ces graphes, nous constatons que le modèle a globalement bien convergé et ne souffre pas de sur-apprentissage. Afin de répondre à la question de la nécessité d'élargir les exemples de l'ensemble d'entraînement, nous inspectons les courbes des fonctions de perte d'apprentissage et de validation pour chaque classe. Les figures 4.9 et 4.10 illustrent les courbes des fonctions de perte de chaque espèce des bases d'images FRGT et LCF-15 respectivement, et les figures 4.11 et 4.12 montrent leurs matrices de confusion sans et avec augmentation artificielle de données.

A partir des figures 4.9(a-c) et la matrice de confusion de la figure 4.11(a), nous pouvons diviser les espèces de poissons de la base FRGT en trois catégories. La première catégorie contient les espèces les plus représentatives (DR, PD, CC, AC et CL). Leurs courbes de perte sont bien convergentes. La deuxième catégorie contient les espèces moins représentatives mais faciles à identifier (CT, MK, HF, NS, AV, CV, PM, LF, SB, S, PV, ZC et SF) ; les courbes de perte correspondantes sont également bien convergentes. La troisième catégorie comprend les espèces (AN, ZS, HM, NN et BU) qui sont moins représentatives et difficiles à identifier en raison des similitudes de forme et de couleur avec d'autres espèces. Les courbes de perte de validation pour ces espèces souffrent d'irrégularités, en particulier pour l'espèce NN. Par conséquent, nous augmentons uniquement le nombre d'exemples pour les espèces de la troisième catégorie. Les nouvelles courbes de perte de cette catégorie sont présentées dans la figure 4.9(d) et une matrice de confusion après l'augmentation artificielle de données est présentée dans la figure 4.11(b).

La figure 4.10 affiche les courbes de perte pour chaque espèce de la base LCF-15. Ici, nous n'avons que deux catégories : les espèces avec des courbes de perte de validation bien convergentes (AC, CC, CL, CS, DA et MK) (figure 4.10(a)) et les espèces dont les courbes de perte de validation souffrent d'irrégularités (AN, AV, CT, DR, HM, NN, PD, PV et ZS) (figure 4.10(b)). Dans la figure 4.12(a), nous montrons la matrice de confusion de classification sans augmentation artificielle de données. A partir de cette matrice de confusion, nous voyons que même les espèces de la première catégorie sont mal classées. Ceci est dû au fait que les images de test sont de mauvaise qualité par rapport à l'ensemble d'entraînement. Pour cette raison, nous augmentons les exemples pour toutes les espèces de cette base d'images. Les nouvelles courbes de perte ainsi que la matrice de confusion après l'augmentation artificielle de données sont données dans la figure 4.10(c) et la figure 4.12(b) respectivement.

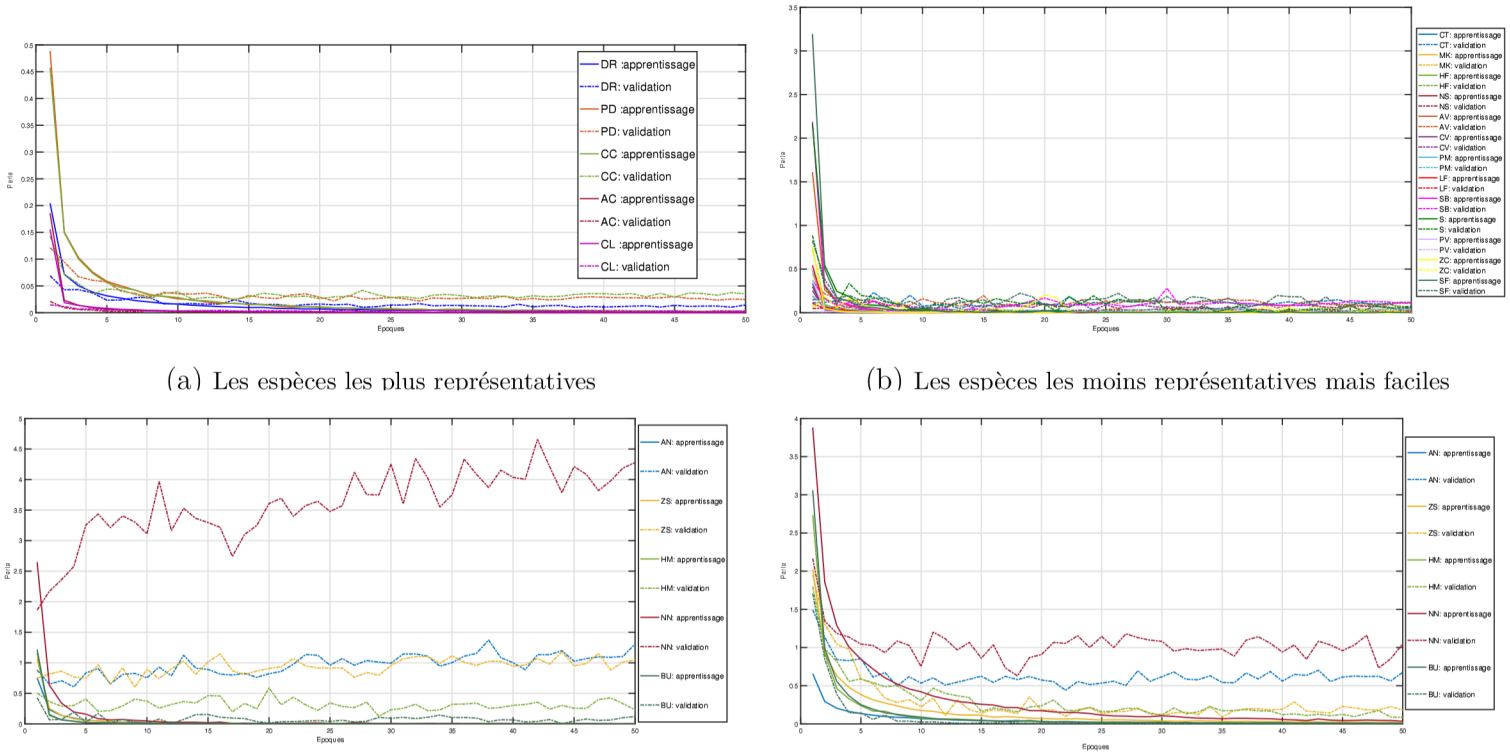


(a) Base d'images FRGT



(b) Base d'images LCF-15

FIGURE 4.8 – Courbes des fonctions de perte d'apprentissage et de validation : (a) sur la base FRGT, (b) sur la base LCF-15.



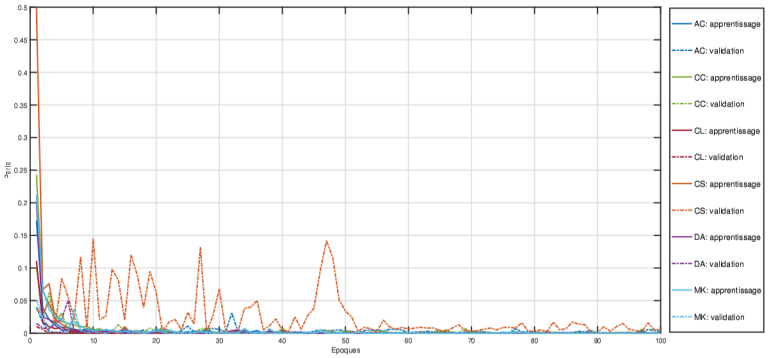
(a) Les espèces les plus représentatives

(b) Les espèces les moins représentatives mais faciles

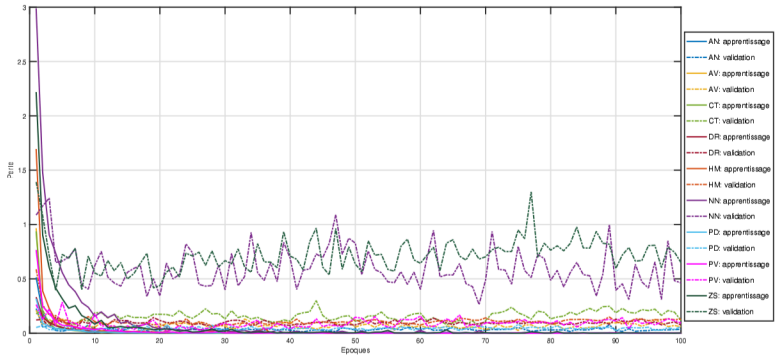
(c) Les espèces les moins représentatives et difficiles

(d) Les espèces les moins représentatives et difficiles après l'augmentation artificielle de données

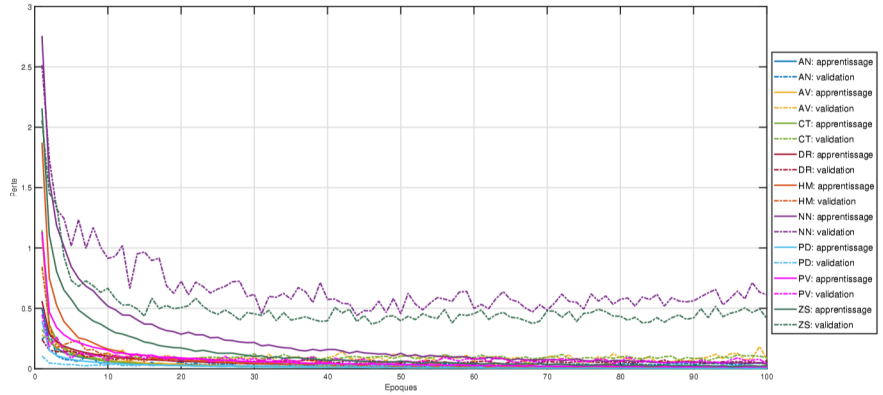
FIGURE 4.9 – Courbes des fonctions de perte d'apprentissage et de validation pour chaque espèce de la base d'images FRGT. (a) et (b) montrent des courbes bien convergentes pour les espèces les plus représentatives et certaines espèces moins représentatives qui sont faciles à identifier respectivement. (c) présente des courbes non convergentes pour des espèces moins représentatives et qui sont difficiles à identifier avec les images disponibles. (d) illustre les courbes de perte pour les espèces en (c) avec une augmentation artificielle d'images.



(a) Bien convergente

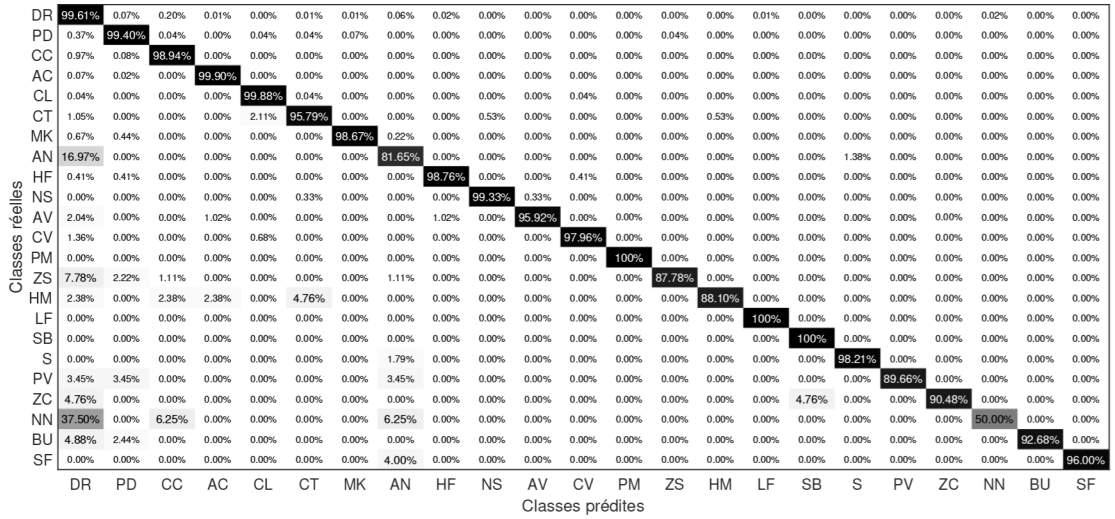


(b) Pas entièrement convergente

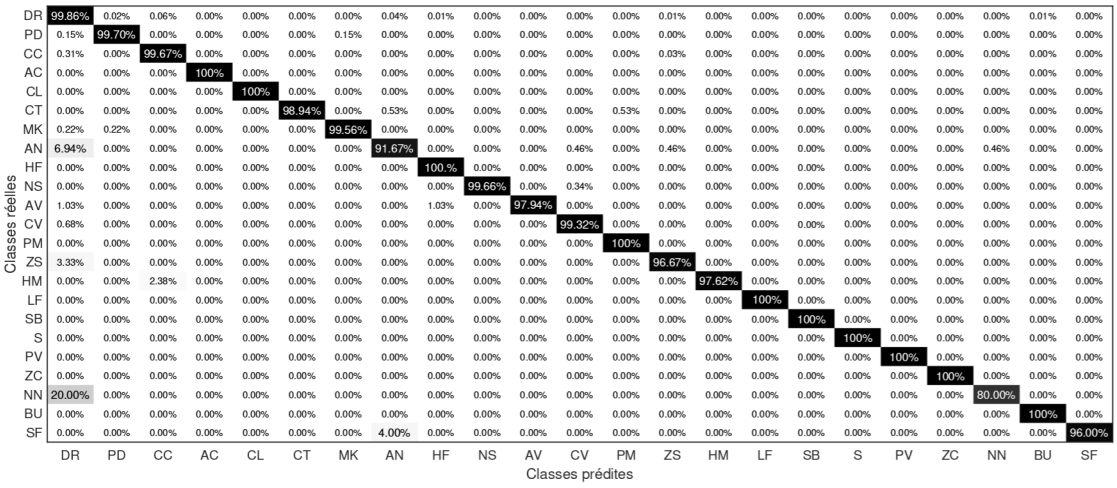


(c) Après l'augmentation de données

FIGURE 4.10 – Courbes des fonctions de perte d'apprentissage et de validation pour chaque espèce de la base d'images LCF-15. (a) montre des courbes d'espèces qui sont bien convergentes. (b) montre des courbes d'espèces qui sont non totalement convergentes en utilisant uniquement les images disponibles. (c) illustre les courbes de perte des espèces de (b) avec une augmentation artificielle de données.

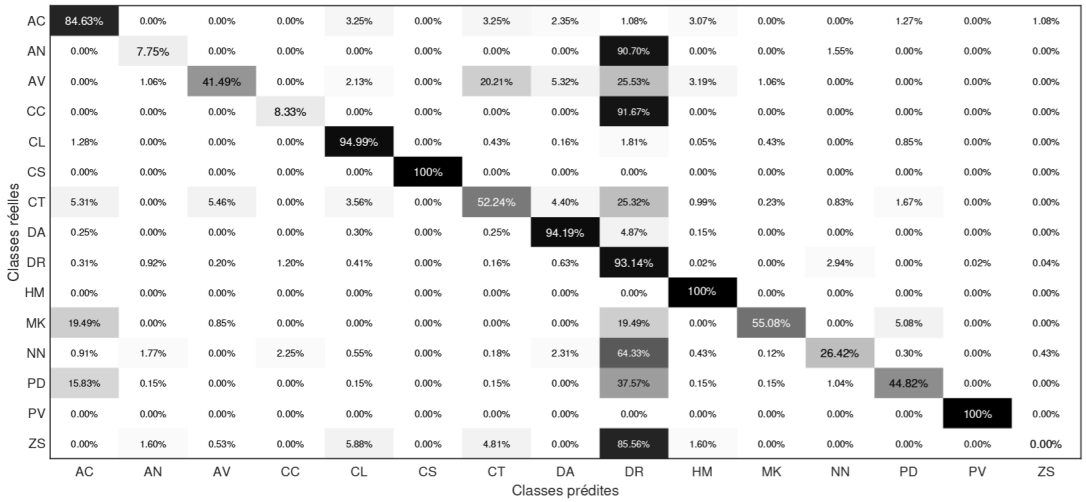


(a) Sans augmentation artificielle de données

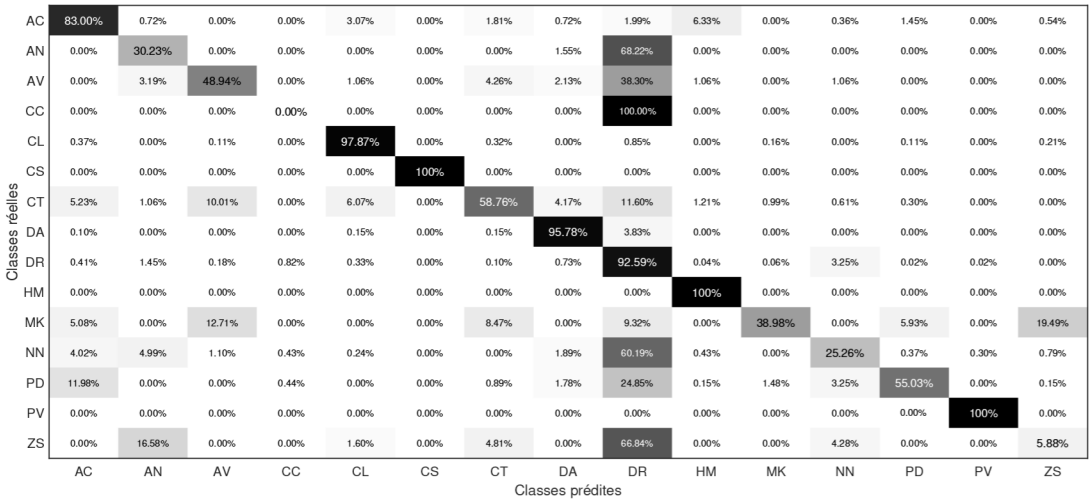


(b) Avec augmentation artificielle de données

FIGURE 4.11 – Matrices de confusion de la stratégie *CNN-Soft* sans (a) et avec (b) augmentation artificielle de données pour la base d'images FRGT.



(a) Sans augmentation de données



(b) Avec augmentation de données

FIGURE 4.12 – Matrices de confusion de la stratégie *CNN-Soft* sans (a) et avec (b) augmentation artificielle de données pour la base d'images LCF-15.

A partir des courbes de perte après l'application de la technique d'augmentation artificielle de données (figures 4.9(d) et 4.10(c)), nous pouvons observer que cette technique réduit le sur-apprentissage et améliore la généralisation. En conséquence, les performances du modèle sont améliorées. Pour la base d'images FRGT (figure 4.11), la précision a été considérablement améliorée pour les espèces qui sont difficiles à identifier : AN est amélioré de 10%, ZS de 8,89%, HM de 9,52%, NN de 30%, et BU de 7,32%. Pour la base d'images LCF-15 (figure 4.12), la précision de certaines espèces est améliorée comme pour AN (+22,95%) et PD (+10,21%).

Base de données	Stratégie	Données disponibles		Augmentation de données	
		AC (%)	PM (%)	AC (%)	PM (%)
FRGT	CNN-Soft	99,24	94,24	99,74	98,11
	CNN-FT-SVM	99,47	94,84	**99,84**	**99,73**
LCF-15	CNN-Soft	75,76	60,21	77,33	62,15
	CNN-FT-SVM	77,29	64,43	**78,95**	**65,32**

TABLE 4.5 – Comparaison des performances avec et sans augmentation artificielle de données sur les bases FRGT et LCF-15.

La table 4.5 montre les effets de l'augmentation artificielle de données sur les métriques AC et PM sur les deux bases d'images. Avec cette technique, nous obtenons les AC et PM les plus élevées de **99,84%** et **99,73%** sur la base d'images FRGT respectivement et de **78,95%** et **65,32%** sur la base d'images LCF-15 respectivement.

Nous concluons que les transformations appliquées sur des images d'entrée telles que le retournement, le recadrage, le redimensionnement et la rotation sont utiles pour réduire le sur-apprentissage sur certaines classes et peuvent considérablement améliorer la généralisation du système de classification.

4.6.4 Etude comparative avec l'état de l'art

Après avoir évalué la méthode proposée en fonction des différentes options possibles, nous effectuons maintenant une comparaison avec les méthodes de l'état de l'art.

Les tables 4.6 et 4.7 montrent la comparaison des performances de nos stratégies proposées avec celles des méthodes de l'état de l'art sur les bases d'images FRGT et LCF-15 respectivement. Dans Deep-CNN (QIN et al. 2015), un CNN avec trois couches convolu-

tives est créé et entraîné à partir de zéro. Dans DeepFish (Qin et al. 2016), les mêmes auteurs ont éliminé l'arrière-plan en utilisant les masques de poisson disponibles et ont entraîné le réseau profond avec des méthodes traditionnelles comme ACP, histogramme par blocs pour améliorer les performances de leur système de classification. Cependant, ils ont légèrement amélioré l'AC de 0,07%, alors qu'ils aient utilisé l'augmentation artificielle de données. Dans CNN-Dir (Sun et al. 2018), les auteurs ont entraîné l'architecture AlexNet directement sur les images de poissons sans l'apprentissage par transfert ni l'augmentation artificielle de données. Ils ont obtenu une faible valeur PM de 48,55%. Dans CNN-SVM (Sun et al. 2018), les auteurs ont ré-entraîné AlexNet et ils ont utilisé aussi l'augmentation artificielle de données. Ils ont obtenu cette fois-ci une valeur PM de 99,64%. Dans notre travail, sans utilisation de l'augmentation artificielle de données, nous avons obtenu une valeur AC de 99,47% et une valeur PM de 94,84%. Avec l'utilisation de l'augmentation artificielle de données appliquée uniquement sur les espèces qui sont difficiles à être classifiées, nous avons obtenu une valeur AC de **99,84%** et une valeur PM de **99,73%**, donc nous obtenons de meilleures performances par rapport aux méthodes de l'état de l'art. Nous pouvons également conclure que les réseaux entraînés avec l'apprentissage par transfert donnent de meilleurs résultats que les réseaux entraînés à partir de zéro.

A partir de la table 4.7, nous observons que les approches basées sur l'apprentissage profond donnent les meilleurs résultats que les méthodes traditionnelles. Nous pouvons également voir que l'extraction de caractéristiques à partir de CNN sans fine-tuning n'est pas efficace (*CNN-SVM* : 65,38% et *CNN-SVM* (Jäger et al. 2016) : 66%) par rapport à l'extraction de caractéristiques après le fine-tuning (*CNN- FT-SVM* : 78,95%). Ces résultats confirment la nature difficile de cette base de référence qui est marquée par des images très floues avec une confusion de fond avec les poissons et une dégradation plus élevée en termes d'intensité lumineuse.

La figure 4.13 montre certaines requêtes de la base LCF-15 qui restent mal classées par toutes nos techniques proposées. Nous pouvons expliquer ces échecs de classification par les raisons suivantes. L'image de requête peut être floue avec une faible résolution comme dans la figure 4.13(a) ; nous ne pouvons pas reconnaître l'espèce de poisson, même à l'œil. En (b), la forme du poisson résulte en fait d'un chevauchement de deux poissons. Dans (c) et (d), la position du poisson par rapport à la caméra conduit parfois à cacher des parties importantes du poisson ; dans (c) par exemple, nous ne voyons que la queue de "*Acanthurus nigrofuscus*". Dans (e), la confusion vient de la ressemblance entre les espèces

Espèce	Avec apprentissage par transfert					Sans apprentissage par transfert		
	Réentraînement plus finement		Extraction de caractéristiques			Entraînement à partir de zéro		Extraction de caractéristiques
	CNN-Soft	CNN-Soft((Sun et al. 2018))	CNN-SVM	CNN-FT-SVM	CNN-SVM((Sun et al. 2018))	CNN-Dir((Sun et al. 2018))	Deep-CNN((Qin et al. 2015))	Deep-Fish((Qin et al. 2016))
DR	99,86	99,78	99,25	99,91	100	95,12	-	99,25
PD	99,70	98,79	99,03	99,78	99,77	41,32	-	97,39
CC	99,67	99,75	98,19	99,61	99,60	81,42	-	98,24
AC	100	99,97	99,80	100	100	92,44	-	100
CL	100	100	99,92	100	100	95,15	-	100
CT	98,94	100	96,32	100	99,38	52,83	-	96,30
MK	99,56	100	98,66	100	100	84,55	-	100
AN	91,67	89,05	79,36	94,50	96,41	11,81	-	67,74
HF	100	98,15	99,59	100	100	62,03	-	100
NS	99,66	100	99,32	100	100	100	-	100
AV	97,94	100	93,88	100	100	63,16	-	92,86
CV	99,32	100	97,28	100	100	43,75	-	95,24
PM	100	96,09	99,45	100	100	48,95	-	100
ZS	96,67	85,06	84,62	100	100	8,12	-	84,62
HM	97,62	100	90,48	100	100	47,37	-	66,67
LF	100	100	100	100	100	0	-	96,55
SB	100	100	100	100	100	14,29	-	85,71
S	100	86,67	96,43	100	96,56	33,33	-	100
PV	100	100	96,43	100	100	13,89	-	100
ZC	100	100	85,71	100	100	33,33	-	100
NN	80	84,62	59,52	100	100	85,71	-	50
BU	100	95,45	92,86	100	100	8,03	-	83,33
SF	96	100	96,43	100	100	0	-	100
PM	98,11	97,10	94,02	**99,73**	99,64	48,55	-	91,91
AC	99,74	-	98,91	**99,84**	-	-	98,57	98,64

Table 4.6 – Comparaison des performances en classification d'espèces de poissons de différentes méthodes sur la base d'images FRGT.

Approche			Méthode	AC (%)
Méthode traditionnelle			SURF-SVM (Szűcs, Papp et Lovas 2015)	51
Apprentissage profond	Avec apprentissage par transfert	Réentraînement plus finement	**CNN-Soft**	**77,33**
		Extraction de caractéristiques	NIN-SVM (Sun et al. 2016)	69,84
			CNN-SVM	**65,38**
			CNN-FT-SVM	**78,95**
			CNN-SVM (Jäger et al. 2016)	66
	Sans apprentissage par transfert	Extraction de caractéristiques	PCANET-SVM (Sun et al. 2016)	77,27

Table 4.7 – Comparaison des performances en classification d'espèces de poissons de différentes méthodes sur la base d'images LCF-15.

au niveau de la forme et du motif : "*Neoglyphidodon nigroris*" est confondu avec "*Dascyllus reticulatus*". Dans (f), la luminosité dans l'environnement sous-marin est trop faible. Parfois, des annotations sur l'image (date) peuvent modifier les motifs des poissons. Par exemple, en (g), l'espèce "*Dascyllus reticulatus*" est confondue avec "*Dascyllus aruanus*".

Malgré les défis et les difficultés de cette base d'images, les CNNs avec apprentissage par transfert donnent toujours de meilleurs résultats que les méthodes traditionnelles (*SURF-SVM* (Szűcs, Papp et Lovas 2015)). En effet, les données ne sont pas linéaires en raison des défis d'environnements sous-marins naturels : variation de l'éclairage, mauvaise qualité

FIGURE 4.13 – Certaines requêtes de poissons de la base LCF-15 qui sont mal classées par toutes nos techniques proposées pour la classification automatique d'espèces de poissons.

des images sous-marines, mouvement libre de poissons, taille et forme de poissons et les fonds de corail. Le traitement d'image traditionnel n'est pas efficace pour modéliser des données non linéaires. Cependant, les CNNs sont des réseaux neuronaux paramétriques non linéaires capables d'extraire et d'apprendre des caractéristiques à partir de données d'entrée complexes.

4.7 Conclusion

Dans ce chapitre, nous avons présenté trois stratégies CNN basées sur l'apprentissage par transfert pour la tâche de classification d'espèces de poissons vivants. Nous avons utilisé le modèle AlexNet pour extraire des caractéristiques d'images de poissons avant et après le fine-tuning sur des bases d'images sous-marines. Nous avons montré que le fine-tuning améliore les performances du modèle. Nous avons également analysé les effets et les avantages de différentes options possibles sur les performances en classification, en particulier l'espace colorimétrique, l'élimination de l'arrière-plan et l'augmentation artificielle de données. Nous avons montré que les images RGB de poisson fournissent de meilleurs résultats que les autres espaces colorimétriques. L'élimination de l'arrière-plan avec des masques de poisson n'a pas amélioré les performances du système. Il a été démontré que cette opération n'est pas utile avec les réseaux CNN. De plus, nous n'avons pas augmenté les nombres d'exemples uniformément pour toutes les classes, mais nous avons proposé de les augmenter en se basant sur les courbes de fonctions de perte d'apprentissage et de validation pour une meilleure performance. Des expériences sur deux bases d'images de référence de poissons vivants sous-marines, à savoir la base Fish Recognition Ground-Truth et la base LifeCLEF 2015 Fish, ont démontré que notre approche proposée avec des options optimales surpasse différentes méthodes de l'état de l'art pour la classification d'espèces de poissons.

Chapitre 5

Apprentissage progressif pour la classification d'espèces de poissons

Sommaire

5.1 Introduction

Nous avons vu dans le chapitre précédent que la technique de l'apprentissage par transfert améliore significativement les performances du réseau CNN. Dans ce chapitre, nous nous basons sur l'apprentissage par transfert et nous proposons deux nouvelles approches pour améliorer les performances du modèle pour la classification d'espèces de poissons. Nous proposons ces deux approches avec de l'apprentissage progressif ou par groupe des classes en partant de groupes plus généraux à plus spécifiques ou de plus spécifiques à plus généraux.

La première approche, inspirée de la classification taxonomique des poissons, est une architecture CNN hiérarchique (Sali et al. 2020) qui permet de classer des poissons en familles (groupe plus général) puis en espèces (groupe plus spécifique). La deuxième approche se focalise au départ sur les classes les plus spécifiques (les classes d'espèces difficiles à identifier), ensuite, elle apprend d'une façon incrémentale de nouvelles classes (qui sont moins difficiles) sans détruire les connaissances acquises à partir d'anciennes classes (Masana et al. 2020).

Le reste de ce chapitre est organisé comme suit : nous allons décrire dans la section 5.2 la classification hiérarchique d'espèces de poissons avec les expérimentations correspondantes. Ensuite, nous allons présenter et expérimenter l'apprentissage incrémental pour la classification d'espèces de poissons dans la section 5.3. Dans la section 5.4, nous allons discuter de résultats expérimentaux. Finalement, la section 5.5 présente une conclusion du chapitre.

5.2 Classification hiérarchique d'espèces de poissons

Dans une classification multi-classe traditionnelle, un CNN est conçu pour être séquentiel et il génère à la sortie un score pour chaque classe. Ensuite, le score le plus élevé détermine la classe de l'objet d'entrée. Ainsi, le modèle CNN traite toutes les classes de la même manière. Avec cette structure de classification, certaines classes sont naturellement

plus susceptibles d'être mal classées que d'autres, en particulier pour les classes ayant moins d'exemples ou étant difficiles à classifier à cause de ressemblance avec d'autres classes. Mais dans la réalité, la propriété de l'ordre des catégories du général au spécifique existe souvent entre les classes, par exemple, le lion et le tigre peuvent généralement être regroupés en tant qu'animaux sauvages tandis que le bus et le camion sont des véhicules. Il est souvent plus facile de distinguer un lion d'un bus que d'un tigre. Cette propriété indique que la classification peut être effectuée de manière hiérarchique au lieu de traiter toutes les classes comme organisées dans une structure «plate». Lors de la classification hiérarchique, un classifieur connait d'abord qu'un lion doit être dans la catégorie des animaux sauvages, ensuite il peut être classé au niveau plus fin en tant que lion. L'un des avantages de la classification hiérarchique est que l'erreur peut être limitée à une sous-catégorie, ce qui signifie également qu'elle devrait être plus informative qu'une classification plate (Zhu et Bain 2017). Par exemple, un classifieur peut confondre un lion avec un tigre, mais il connait que cela devrait au moins être un animal sauvage.

De l'autre côté, il existe une classification scientifique des espèces basée sur des taxons. Cette classification biologique et hiérarchique permet de regrouper les espèces ayant des ressemblances et des caractères communs dans un même taxon. Cette classification nous inspire de proposer une classification hiérarchique d'espèces de poissons, basée sur CNN, en regroupant d'abord les espèces qui ont un taxon commun dans un même ensemble pour une classification efficace.

5.2.1 Approche proposée

Nous proposons un modèle CNN hiérarchique structuré en arborescence inspiré de la classification taxonomique des poissons. Notre modèle est composé de plusieurs nœuds connectés de manière arborescente. Le nœud racine est le nœud le plus élevé de l'arbre. Ce nœud prend l'image de poisson et génère des cartes de caractéristiques. Ensuite, il fait la première classification afin de classer le poisson dans une famille. Suite au résultat de cette première classification, les cartes de caractéristiques sont transmises au nœud feuille activé. Ce nœud feuille classe le poisson selon l'espèce. La figure 5.1 montre l'architecture globale de notre modèle qui contient un nœud racine et les nœuds feuilles pour un réseau de classification à deux taxons.

Dans cette architecture, tous les nœuds partagent des couches communes. Ces couches

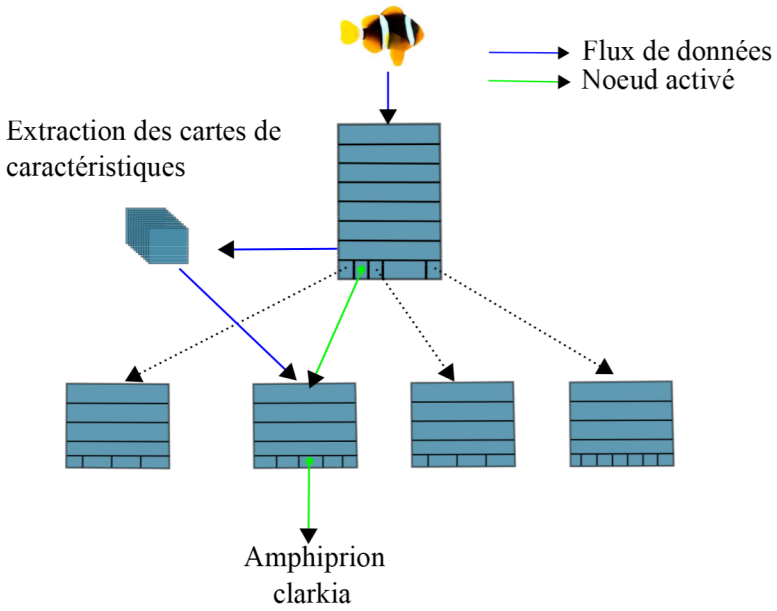


FIGURE 5.1 – Un modèle CNN hiérarchique à deux niveaux : la sortie du nœud racine est utilisée pour sélectionner le nœud feuille au niveau suivant. Illustration de l'activation du nœud *Pomacentridae* qui contient l'espèce *Amphiprion clarkia*.

extraient des cartes de caractéristiques qui vont alimenter les nœuds activés. Cela a plusieurs avantages, nous en citons notamment les suivants :

1. Les premières couches d'un CNN extraient de l'image d'entrée des caractéristiques globales ou de bas niveau, tandis que les couches supérieures extraient des caractéristiques plus localisées et spécifiques à la classe. Par conséquence, il est avantageux de partager les couches inférieures car elles sont pertinentes pour toutes les classes.

2. L'utilisation des couches partagées évite de reproduire le même réseau plusieurs fois dans chaque nœud ce qui réduit énormément le temps de calcul et les ressources de mémoire et de processeur. Cela permet d'utiliser le modèle dans des applications en temps réel.

3. Le partage des couches réduit également le nombre de paramètres de l'architecture CNN, ce qui accélère l'entraînement du CNN.

4. Les nœuds feuilles seront entraînés à être experts dans la distribution des espèces au sein de la même famille.

5.2.2 Stratégie de l'entraînement

Afin de construire une hiérarchie arborescente de catégories, nous allons regrouper les espèces qui ont des caractéristiques communes dans la même catégorie. Un classifieur sera entraîné pour classer ces espèces dans des classes plus spécifiques. Nous utilisons une approche descendante pour apprendre la hiérarchie à partir des données d'entraînement.

Au fur et à mesure que nous intégrons des nœuds dans le modèle, le nombre de paramètres de l'architecture hiérarchique croît avec augmentation de la complexité de l'entraînement et du risque de sur-apprentissage. D'autre part, le déséquilibre des images d'un mini-lot pose un problème majeur dans l'entraînement d'une architecture arborescente. Durant l'apprentissage par mini-lot, les exemples d'entraînement sont acheminés de manière probabiliste vers différents nœuds enfants. Il faut utiliser un mini-lot grand pour garantir que les gradients de paramètres dans les nœuds enfants sont estimés par un nombre suffisamment grand d'échantillons d'apprentissage (Yan et al. 2015). Toutefois, un grand mini-lot d'entraînement augmente à la fois les ressources de mémoire et ralentit le processus d'entraînement. Par conséquent, nous abordons ce problème en divisant l'entraînement en plusieurs étapes au lieu de l'entraînement dans son ensemble. En particulier, nous entraînons d'abord le nœud racine qui servira de base pour l'entraînement ultérieur des nœuds enfants.

5.2.2.1 Entraînement du nœud racine

Nous entraînons séquentiellement les nœuds de chaque niveau. Tout d'abord, le nœud racine est un CNN pré-entraîné tel que : AlexNet, VGG, GoogleNet, ou ResNet. Il est ré-entraîné sur les images du taxon le plus commun à toutes espèces afin d'extraire les caractéristiques globales qui seront l'entrée de chaque nœud feuille. Cela permet aux nœuds feuilles de se concentrer davantage sur l'entraînement des caractéristiques locales de chaque espèce. Ce nœud racine utilise la couche *Softmax* pour apprendre la corrélation entre les images d'entrée et les classer dans le taxon suivant. A la fin de cette étape, les paramètres des couches convolutives de ce nœud sont maintenues fixes.

5.2.2.2 Entraînement des nœuds feuilles

Les nœuds feuilles peuvent être entraînés indépendamment en parallèle. Chaque nœud devrait se spécialiser pour classer le poisson dans des catégories plus spécifiques. Par conséquent, l'entraînement de chaque nœud feuille n'utilise que les images des poissons du taxon correspondant. Tous les nœuds sont entraînés en utilisant l'algorithme de rétro-propagation en arrière.

5.2.2.3 La phase de test

Dans l'étape de test, une image de test est d'abord transmise au nœud racine où la couche *Softmax* produira un vecteur de scores indiquant les probabilités que l'image appartienne aux familles. Le score le plus élevé détermine le nœud de la famille vers lequel l'image de test sera acheminée. Ce processus se répète dans le nœud feuille pour attribuer l'espèce à l'image de test.

5.2.3 Résultats expérimentaux

Dans cette section, nous évaluons notre approche de la classification hiérarchique sur les deux bases d'images de poissons de référence : FRGT (Boom et al. 2012a,b) et LCF-15 (Joly et al. 2015). Nous soulignons ici que dans le nœud racine nous avons utilisé le réseau pré-entraîné ResNet50.

5.2.3.1 Base d'images de référence FRGT

La figure 5.2 illustre la classification taxonomique des poissons de la base FRGT. Nous pouvons voir que les 23 espèces de la base FRGT peuvent être regroupées en 13 familles. Les familles *Scaridae, Lutjanidae, Nemipteridae, Pempheridae, Siganidae, Zanclidae, Balistidae* et *Tetradontidae* contiennent chacune une seule espèce. Les familles *Acanthuridae, Chaetodontidae, Holocentridae* et *Labridae* contiennent chacune deux types d'espèces. Finalement, la famille *Pomacentridae* se compose de sept espèces. Le nœud racine a donc 13 sorties correspondant aux 13 familles.

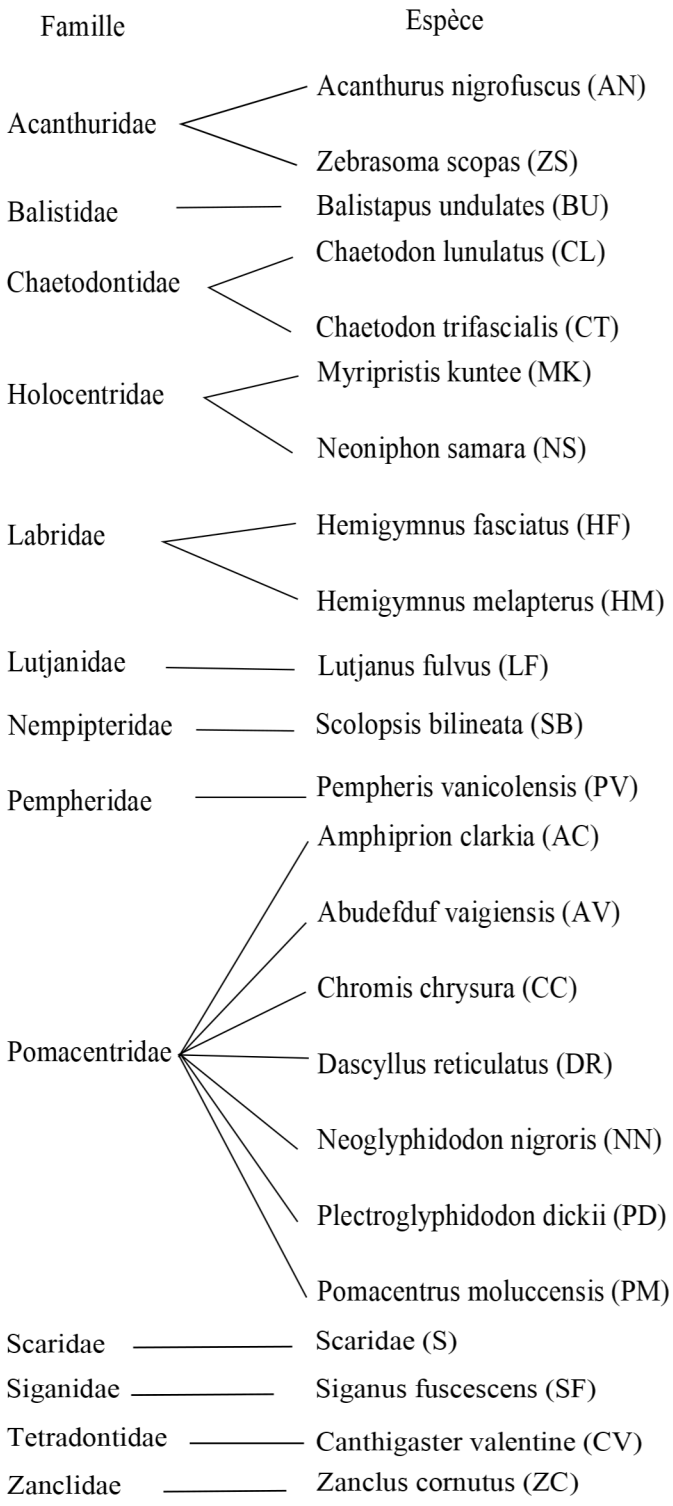


FIGURE 5.2 – Classification taxonomique d'espèces de poissons de la base d'images FRGT.

Dans une classification hiérarchique, les nœuds des premiers niveaux doivent être de bonnes performances car s'ils classifient mal les images dès le début, les nœuds enfants le feront également. La table 5.1 montre les performances des nœuds du modèle hiérarchique et la figures 5.3 illustre les courbes de la fonction de perte d'apprentissage et de validation de chaque nœud.

Nœuds	AC (%)	PM (%)
Racine	100	100
Acanthuridae	88,64	87,47
Chaetodontidae	100	100
Holocentridae	100	100
Labridae	100	100
Pomacentridae	99,44	99,68
Modèle entier	99,39	98,55

TABLE 5.1 – Performances des nœuds du modèle hiérarchique pour la base FRGT.

Nous commençons d'abord évaluer le nœud racine. Nous pouvons voir à partir de la figure 5.3 que ce nœud est performant, ses courbes se convergent rapidement. D'après la table 5.1, le nœud racine classifie tous les poissons dans leurs bonnes familles. Ceci est dû aux caractéristiques globales communes au sein de la même famille. Nous passons ensuite aux nœuds qui correspondent aux familles qui ont au moins deux espèces (*Acanthuridae, Chaetodontidae, Holocentridae, Labridae* et *Pomacentridae*). L'entraînement s'est effectué sur les images des poissons de cette famille uniquement ; le nombre de classes de sortie correspond au nombre d'espèces dans cette famille.

Nous pouvons voir d'après la table 5.1 et la figure 5.3 que le nœud *Acanthuridae* est le moins performant à cause des similitudes de forme et de couleur entre les espèces de cette famille (AN et ZS). Les autres nœuds feuilles sont performants. A partir de la matrice de confusion présentée sur la figure 5.4, nous constatons une remarque importante c'est que les erreurs de classification sont limitées au sein d'une seule famille. Cela est un avantage de la classification hiérarchique qui limite l'erreur à une sous-catégorie. Le modèle peut mal identifier une espèce mais il connait au moins sa famille ce qui rend la classification hiérarchique plus informative que la classification plate.

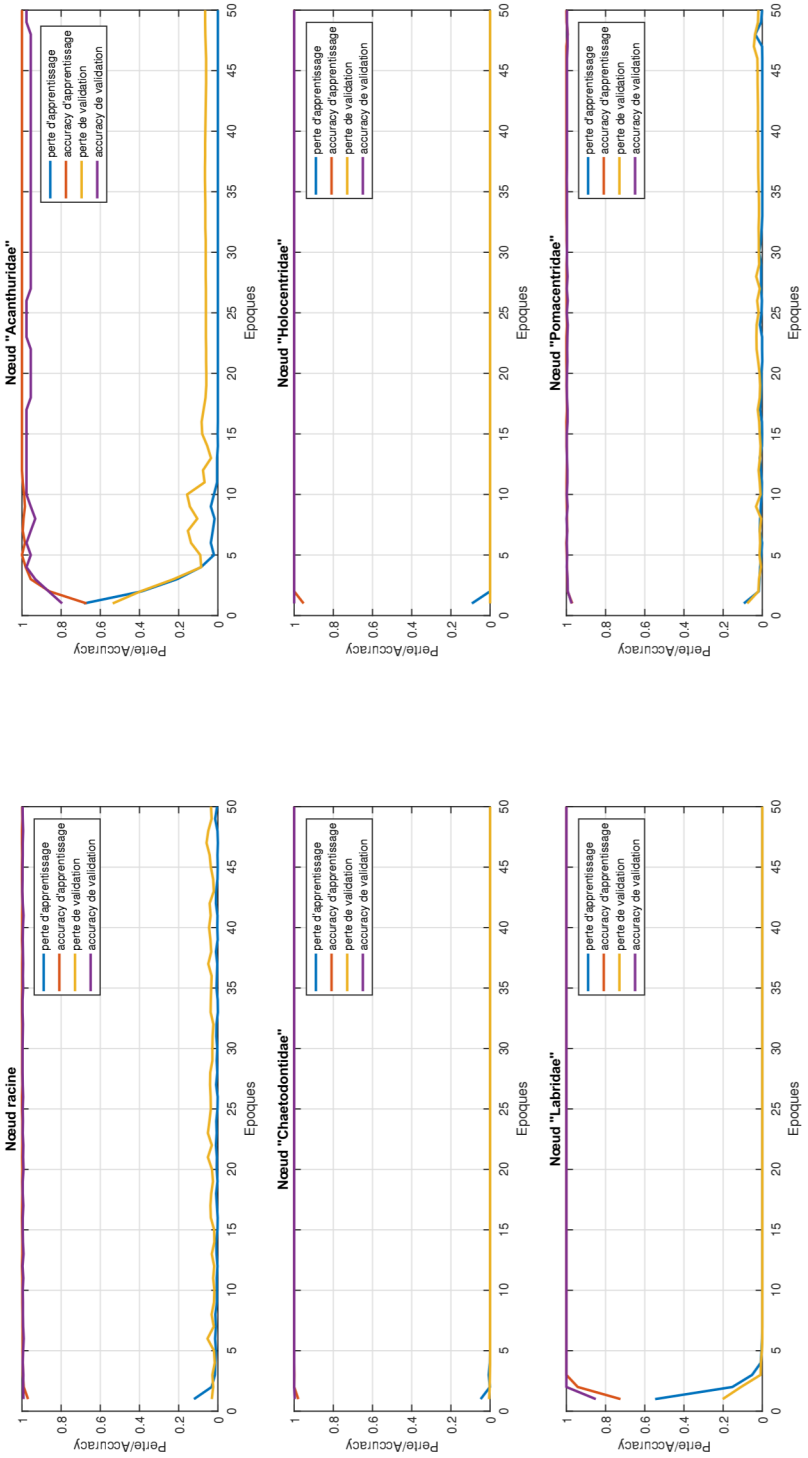


FIGURE 5.3 – Fonctions de pertes et taux de classification par époque pour chaque nœud du modèle.

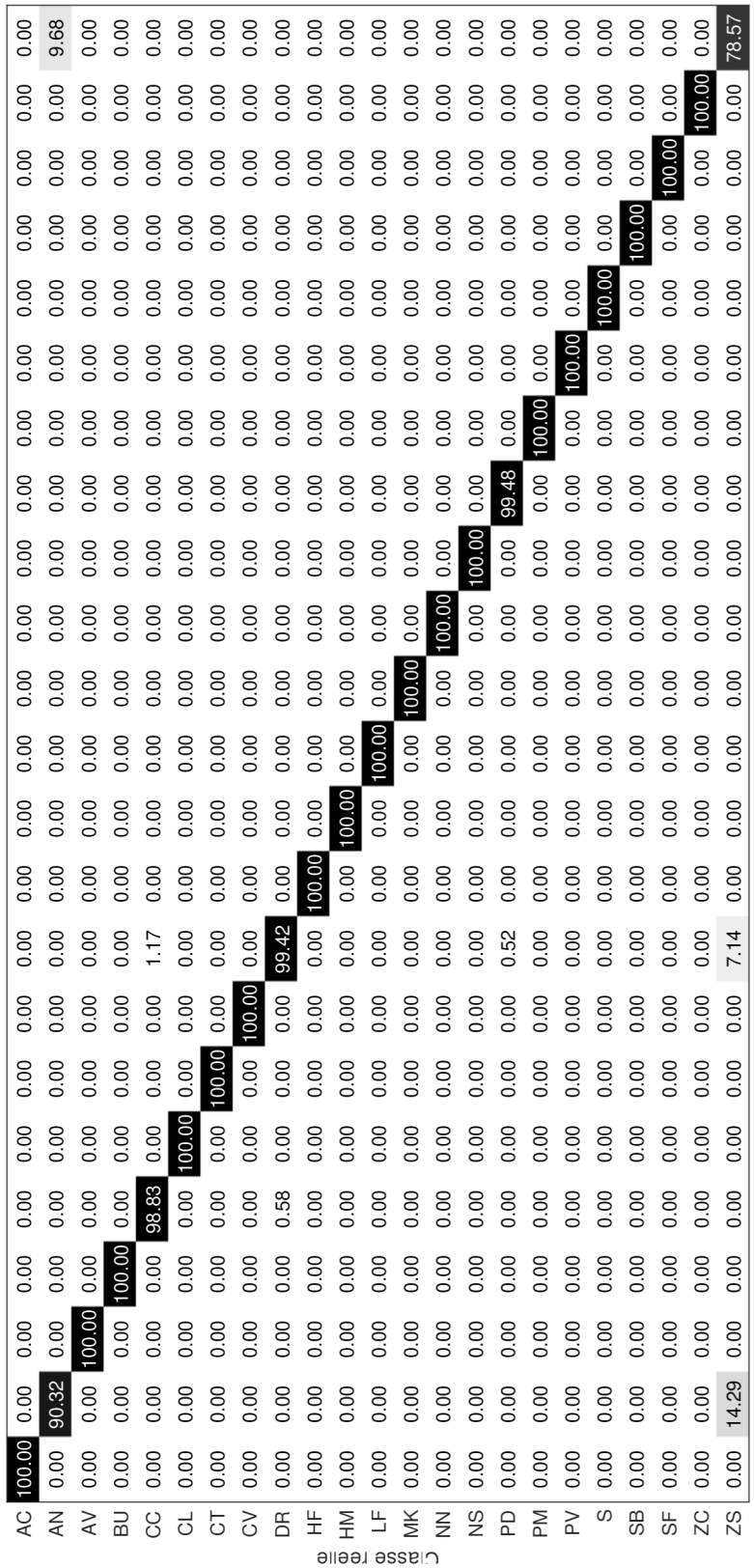


FIGURE 5.4 – Matrice de confusion du modèle entier pour la base FRGT.

5.2.3.2 Base d'images de référence LCF-15

La base LCF-15 contient 6 familles (figure 5.5). La famille *Pomacentridae* est la plus grande, elle se compose de 7 espèces. Ensuite, la famille *Chaetodontidae* contient 3 espèces et la famille *Acanthuridae* contient deux espèces. Finalement, les familles *Holocentridae, Labridae* et *Pempheridae* ne contiennent qu'une seule espèce.

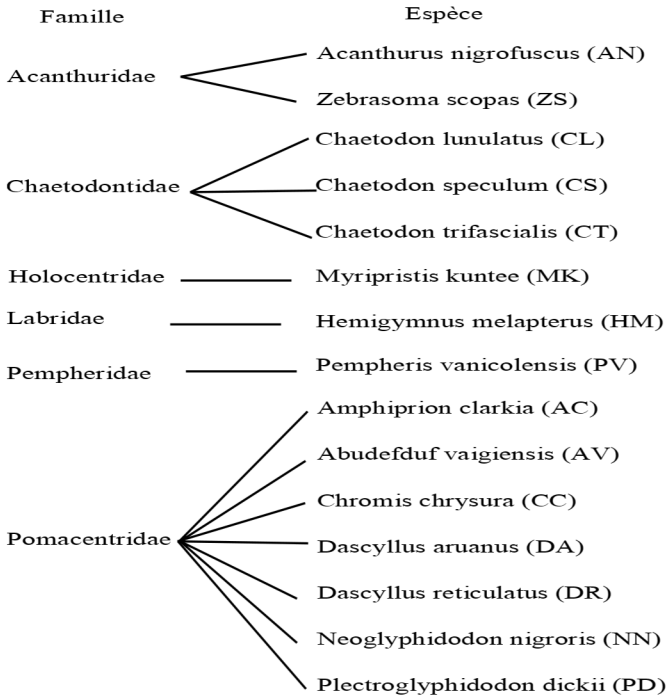
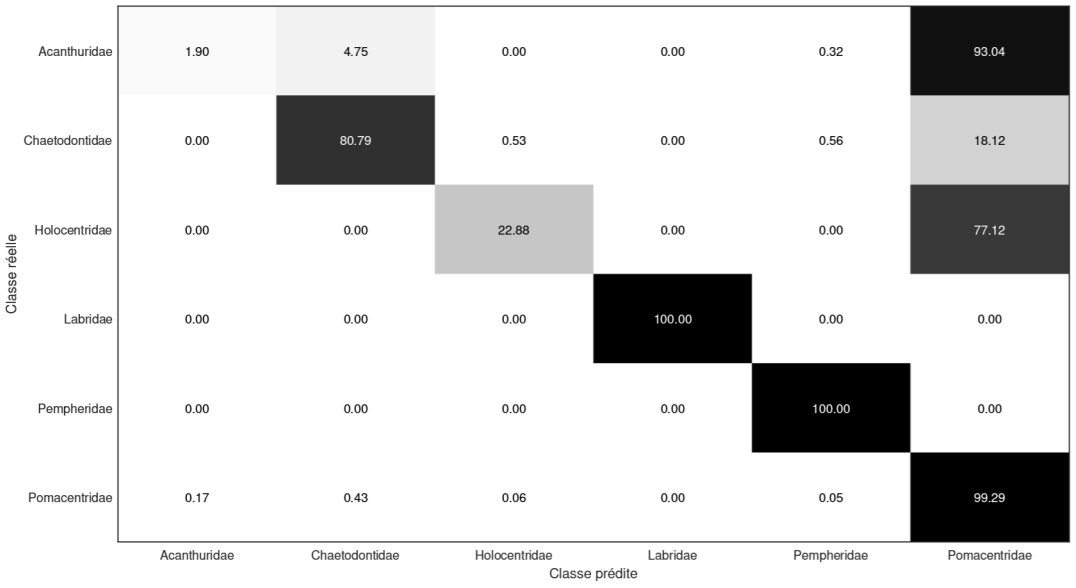


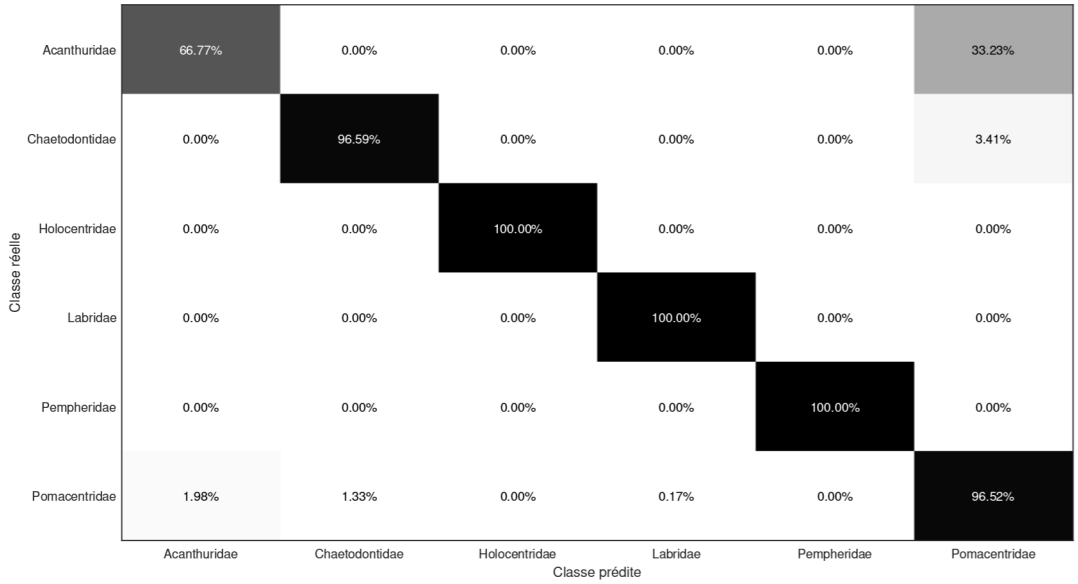
FIGURE 5.5 – Classification taxonomique d'espèces de poissons de la base d'images LCF-15.

La table 5.2 montre les performances du nœud racine du modèle hiérarchique pour la base LCF-15. Ce nœud est moins performant qu'avec la première base en particulier sans utiliser la technique d'augmentation artificielle de données ; le classifieur achève un taux de classification de 91,96%. Ceci est dû au fait que les images de test sont de mauvaise qualité par rapport aux images d'entraînement. La figure 5.6(a) illustre la matrice de confusion du nœud racine sans l'utilisation d'augmentation artificielle de données. D'après la matrice de confusion, la plupart de poissons tendent à se classifier dans la famille *Pomacentridae* car la base est déséquilibrée et la famille *Pomacentridae* est la famille la plus représentative.

Comme nous avons dit, le nœud racine doit être assez performant afin de réduire l'erreur de classification entre les familles. Pour cela, nous avons augmenté les images de poissons



(a) Sans augmentation de données



(b) Avec augmentation de données

FIGURE 5.6 – Matrice de confusion du nœud racine pour la base LCF-15 : (a) sans l'augmentation artificielle de données (b) avec l'augmentation artificielle de données.

Nœud	AC (%)	PM (%)
Racine sans augmentation de données	91,96	67,48
Racine avec augmentation de données	95,87	93,91

TABLE 5.2 – Performances du nœuds racine du modèle hiérarchique pour la base LCF-15.

en particulier pour les familles qui sont difficiles à identifier à savoir les familles *Acanthuridae, Chaetodontidae* et *Holocentridae*. Nous pouvons voir d'après la table 5.2 et la figure 5.6(b), qui représente la matrice de confusion après l'utilisation de l'augmentation de donnée, que la performance du classifieur est améliorée significativement pour les familles dont nous avons augmenté les images mais elle est un peu baissée pour la famille *Pomacentridae*. Le classifieur achève un taux de classification de 95,87%. Nous pouvons remarquer aussi qu'il y a une confusion entre les familles *Acanthuridae* et *Pomacentridae* et les familles *Chaetodontidae* et *Pomacentridae* en raison de la similitude de forme et de couleur entre les espèces de ces familles. Nous remarquons également que quelques poissons de la famille *Pomacentridae* sont classés dans *Labridae*. Finalement, tous les poissons des familles *Holocentridae* et *Pempheridae* sont bien classés.

Contrairement à la base FRGT où tous les poissons sont au moins bien classés dans leurs familles, nous avons dans cette base une erreur de classification entre les familles. Afin d'améliorer la performance de classification, nous ajoutons une classe '*Autre*' dans les nœuds feuilles (espèce) de familles *Acanthuridae, Chaetodontiade, Labridae* et *Pomacentridae* (figure 5.7). La classe '*Autre*' contient des poissons de la famille *Pomacentridae* pour les nœuds *Acanthuridae, Chaetodontidae* et *Labridae* et elle contient des poissons de *Acanthuridae* et *Chaetodontidae* pour le nœud *Pomacentridae*.

Les résultats sont rapportés dans la table 5.3. La figure 5.8 montre les matrices de confusion de chaque nœud. D'après la table 5.3 et la figure 5.8, nous observons que la performance du nœud *Acanthuridae* est la moins performante, mais en regardant les matrices de confusion des nœuds, nous remarquons, à part le classifieur de nœud *Labridae* qui atteint un taux de classification 100%, que les classifieurs identifient très bien leurs propres espèces mais ils classent mal les espèces de la classe '*Autre*'.

Finalement, la figure 5.9 illustre la matrice de confusion du modèle hiérarchique entier. Le modèle hiérarchique achève un taux de classification de 81,31% (table 5.3). D'après la matrice de confusion, la majorité des espèces sont bien classées y compris les espèces difficiles. Par conséquence, nous avons bien amélioré significativement les performances de

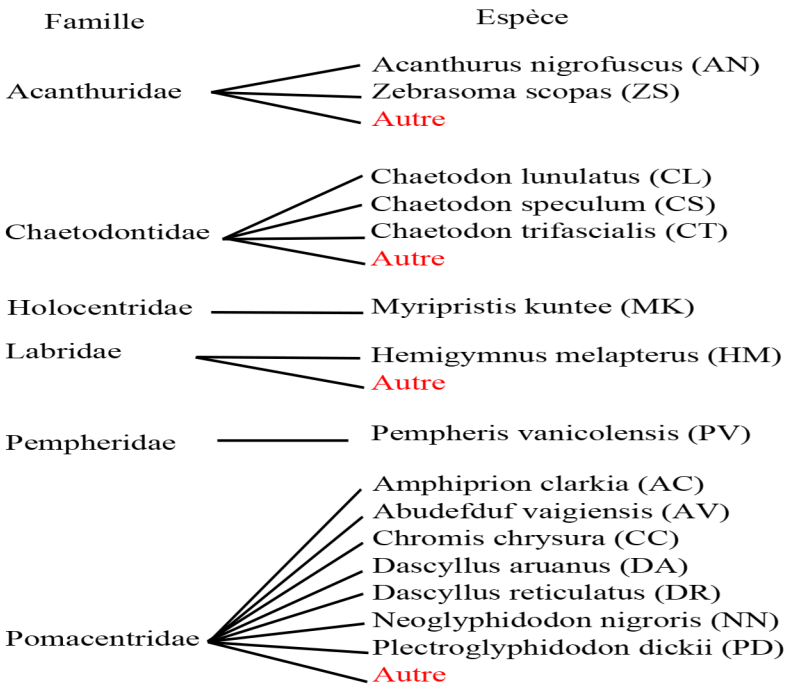


FIGURE 5.7 – Classification taxonomique d'espèces de poissons de la base d'images LCF-15 en ajoutant la classe '*Autre*'.

Nœud	AC (%)	PM (%)
Acanthuridae	55,94	65,63
Chaetodontidae	94,13	85,30
Labridae	100	100
Pomacentridae	84,01	85,07
Modèle entier	81,31	82,69

TABLE 5.3 – Performances des nœuds feuilles du modèle hiérarchique pour la base LCF-15. Les noeuds des familles contiennent une classe supplémentaire appelée '*Autre*'.

la classification par rapport à la classification plate, vue dans le chapitre précédent, dans laquelle nous avons obtenu un taux de classification de 77,33%.

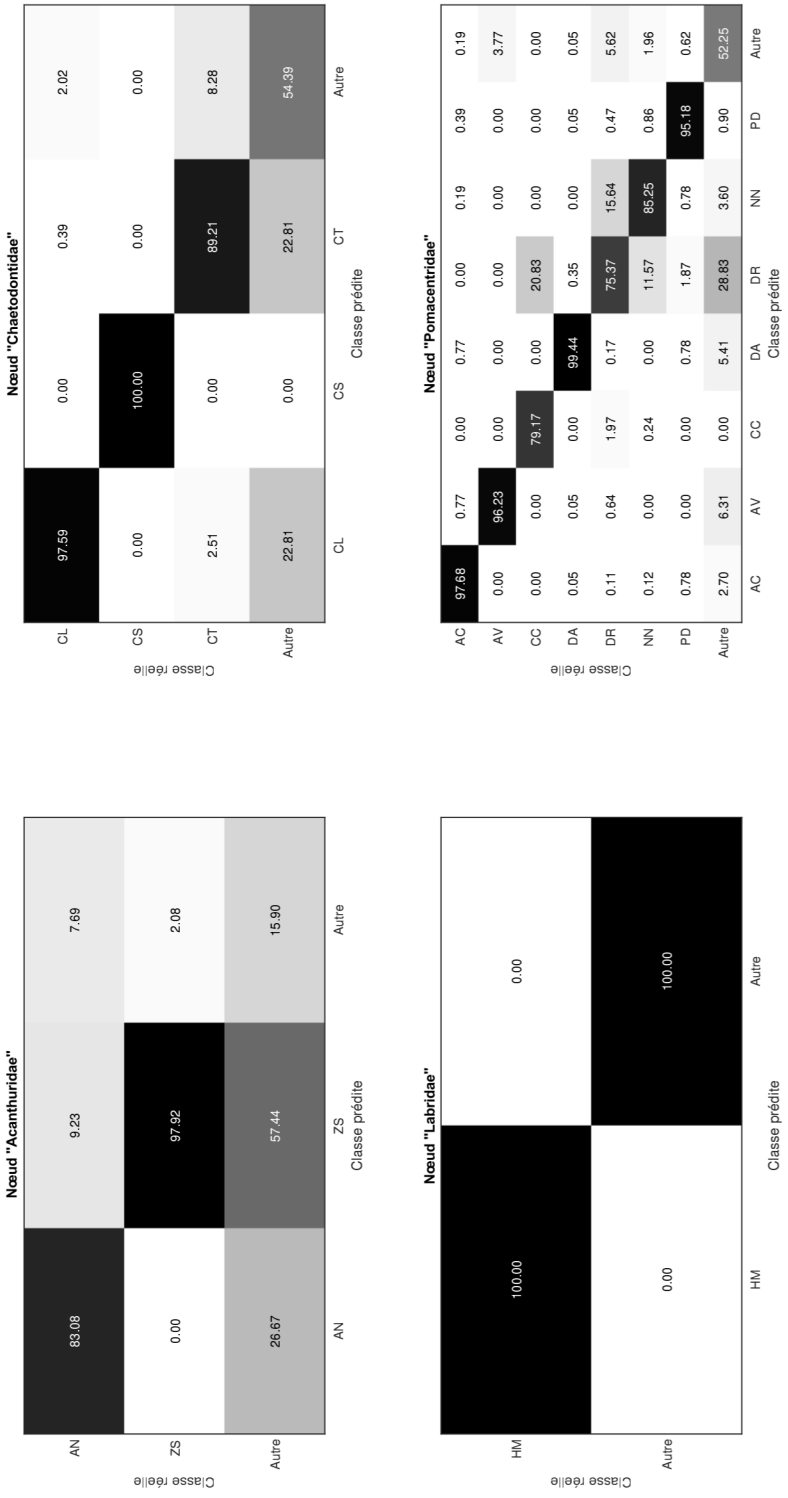


FIGURE 5.8 – Matrices de confusion des nœuds feuilles pour la base LCF-15.

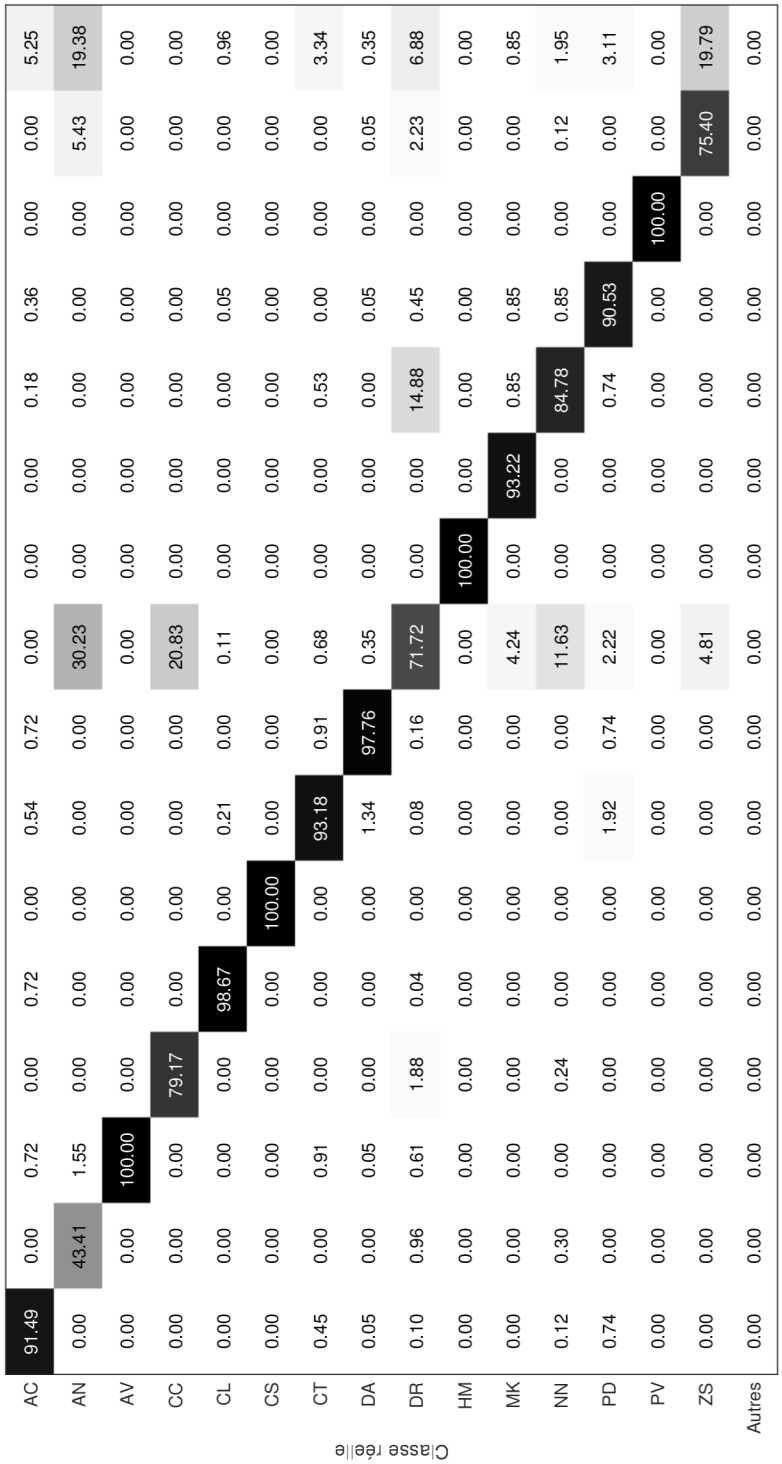


FIGURE 5.9 – Matrice de confusion du modèle hiérarchique pour la base LCF-15.

5.3 Apprentissage incrémental d'espèces de poissons

Dans la tâche de classification d'espèces de poissons, le problème principal est le nombre limité d'images dans la base. Un grand nombre d'images est généralement nécessaire pour créer un système de classification avec un taux de classification élevé. Pratiquement, il est parfois difficile d'obtenir un nombre d'images suffisant pour la classification d'espèces de poissons. En plus de ce problème s'ajoute le problème de déséquilibre de la base d'images. Généralement, les classes ayant moins d'effectifs sont plus susceptibles d'être difficiles à classer par le modèle. Nous avons vu dans le chapitre précédent la technique d'augmentation artificielle de données pour surmonter ces problèmes. Cette technique demande plus de ressources de mémoire et de calcul et n'est pas toujours suffisante.

Nous proposons un nouveau modèle basé sur le principe de l'apprentissage incrémental pour améliorer les performances sur les espèces difficiles à identifier. Au début, le modèle se focalise à bien apprendre les espèces difficiles, puis apprend progressivement les autres espèces avec une bonne stabilité.

5.3.1 Apprentissage incrémental

L'apprentissage incrémental (Shmelkov, Schmid et Alahari 2017 ; Xiao et al. 2014) est un algorithme qui permet à un modèle de recevoir et d'intégrer de nouveaux exemples sans devoir refaire un apprentissage complet. Il doit apprendre de nouvelles données sans oublier ses connaissances existantes, c'est-à-dire, sans détruire les connaissances acquises à partir de données anciennes. Un algorithme d'apprentissage incrémental est défini dans (Polikar et al. 2001) répondant aux critères suivants :

— il doit être capable d'apprendre de connaissances supplémentaires à partir de nouvelles données ;
— il ne doit pas nécessiter l'accès aux données d'origine (c'est-à-dire les données qui ont été utilisées pour apprendre le classifieur actuel) ;
— il doit préserver les connaissances déjà acquises ;
— et il doit être en mesure d'apprendre de nouvelles classes susceptibles d'être introduites avec de nouvelles données.

Ces quatre points s'appliquent pour tout problème général d'apprentissage incrémental.

Dans notre application, nous voulons utiliser le principe de l'apprentissage incrémental dans un contexte de l'apprentissage par transfert classique. En mode d'apprentissage par transfert classique, un modèle pré-entraîné est ré-entraîné sur une nouvelle base de données avec un nombre de classes prédéfinies. Le mode d'apprentissage incrémental par transfert entraîne progressivement un modèle tout en ajoutant à chaque transfert de connaissances de nouvelles classes. Le point commun entre cet apprentissage et l'apprentissage incrémental classique est que le modèle apprend de nouvelles données sans détruire les connaissances acquises à partir des données anciennes. En revanche, la différence entre eux est que l'apprentissage incrémental par transfert nécessite un réapprentissage du système sur les anciennes et les nouvelles données.

Nous pouvons distinguer principalement trois types d'algorithmes d'apprentissage incrémental :

— Stratégie architecturale (Rusu et al. 2016) : cet algorithme modifie l'architecture du modèle afin d'atténuer l'oubli, par exemple : ajouter des couches, fixer les poids...

— Stratégie de régularisation (Kirkpatrick et al. 2017 ; Li et Hoiem 2017) : on ajoute, dans la fonction de perte, des termes de perte favorisant la sélection des poids importants pour conserver les connaissances acquises. Ce type inclue également des techniques de régularisation de base telles que le décrochage et l'arrêt précoce.

— Stratégie de répétition (Hayes, Cahill et Kanan 2019) : les anciennes données sont périodiquement repassées dans le modèle pour renforcer les connexions associées à la connaissance apprise. Une approche simple consiste à stocker une partie des données d'entraînement précédentes et à les entrelacer avec de nouvelles données pour un prochain entraînement.

Nous proposons pour notre application un système basé sur la stratégie de régularisation en modifiant la fonction de perte du système.

5.3.2 Approche proposée

Nous proposons une approche qui combine l'apprentissage incrémental et l'apprentissage par transfert pour entraîner un CNN progressivement tout en ajoutant de nouvelles classes. Pour l'apprentissage des nouvelles classes, nous nous basons sur l'approche "*Learning Without Forgetting*" (Li et Hoiem 2017).

5.3.2.1 Architecture de l'approche

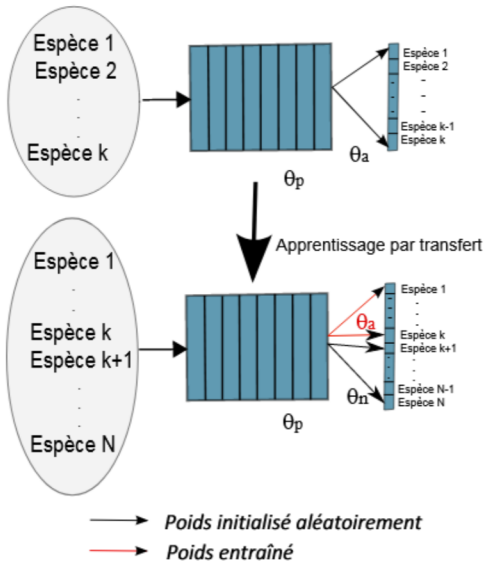


FIGURE 5.10 – Vue générale de l'approche proposée basée sur l'apprentissage incrémental. Le système initialise les poids correspondant aux nouvelles classes aléatoirement et garde les poids entraînés.

Dans notre approche illustrée sur la figure 5.10, un CNN a un ensemble de paramètres partagés θ_p (les couches convolutives), des paramètres spécifiques aux anciennes classes θ_a (les poids de neurones de la couche de sortie correspondant aux anciennes classes), et des paramètres spécifiques aux nouvelles classes initialisés aléatoirement θ_n (les poids de neurones de la couche de sortie correspondant aux nouvelles classes). Notre objectif est d'apprendre les paramètres spécifiques aux nouvelles classes θ_n et mettre à jour les paramètres θ_p et θ_a afin que le modèle entier fonctionne bien sur les anciennes et nouvelles classes.

5.3.2.2 Phase d'apprentissage

Tout d'abord, en utilisant l'apprentissage par transfert, nous entraînons le modèle sur k classes de la base d'images en utilisant un modèle pré-entraîné. L'ensemble d'entraînement est noté $\{x_i, y_i | x_i \in X_a, y_i \in Ya, i = 1, ..., k\}$ où X_a est l'ensemble des exemples et Y_a est l'ensemble des étiquettes correspondantes. A la fin de cet entraînement, nous générons les paramètres θ_p et θ_a. Dans la deuxième étape, chaque image $x_i \in X$, où $X = X_a \cup X_n$

et X_n est l'ensemble des images des nouvelles classes (de $k+1$ à N), passe par le CNN entraîné (de paramètres θ_p et θ_a) pour générer un vecteur de probabilités d'appartenance aux k anciennes classes $p_a^{(i)}$. L'ensemble $P_a = f(\theta_p, \theta_a, X)$ des probabilités sert comme des étiquettes correspondant à l'ensemble des images X ; f étant la sortie du CNN en utilisant les paramètres θ_p et θ_a. Nous allons essayer à ce que le réseau final ne bouge pas beaucoup ces prédictions.

Afin d'intégrer les nouvelles classes, nous ajoutons un nombre de neurones égal au nombre des nouvelles classes dans la couche de classification, entièrement connectés à la couche en dessous, avec des poids initialisés aléatoirement (paramètres θ_n). Le nombre de nouveaux paramètres est égal au nombre des nouvelles classes multiplié par le nombre des nœuds dans la dernière couche partagée. Nous fixons les paramètres du réseau (θ_p et θ_a) et nous entraînons le réseau pour apprendre les paramètres θ_n. Durant cette étape, nous faisons un entraînement normal où le réseau encourage les sorties calculées par le CNN $\hat{Y}_n$ à être cohérentes avec la vérité terrain Y_n. Finalement, nous entraînons conjointement tous les paramètres du modèle (θ_p, θ_a et θ_n) jusqu'à la convergence. Durant cette deuxième étape, nous voulons que l'ensemble des probabilités de sortie calculées $\hat{P}_a$ soit proche de l'ensemble des probabilités enregistrées P_a. Pour cela, nous modifions la fonction de perte du réseau en ajoutant un terme de distillation des connaissances.

5.3.2.3 Distillation des connaissances

La distillation des connaissances est une approche proposée à l'origine par (Hinton, Vinyals et Dean 2014) pour réduire la taille d'un réseau. Elle utilise deux réseaux : un réseau performant mais complexe et coûteux appelé *le maître* et un réseau plus petit appelé *l'élève*. Le réseau maître sert à entraîner le réseau élève. Ce dernier cherche à prédire les sorties du maître en imitant les probabilités assignées à chaque classe. Finalement, nous aurons donc deux réseaux qui produisent les mêmes sorties mais de tailles différentes. Cette approche permet d'obtenir un modèle élève plus léger et d'améliorer la performance. Dans notre approche, nous utilisons la distillation des connaissances pour entraîner le réseau lorsqu'on ajoute les nouvelles images sans l'oubli de l'ancienne connaissance. La distillation des connaissances permet au réseau de rapprocher ses sorties après l'intégration de nouvelles classes aux sorties du réseau avant l'intégration. Ceci peut être modélisé par une perte entropique croisée modifiée qui augmente le poids pour les probabilités les plus

petites :

$$L_{distillation}(p_a, \hat{p}_a) = -\sum_{i=1}^{l} p_a'^{(i)} \; log(\hat{p}_a'^{(i)}) \tag{5.1}$$

où l est le nombre d'étiquettes, p'_a et $\hat{p}'_a$ sont des versions modifiées des probabilités enregistrées p_a et calculées $\hat{p}'_a$:

$$p_a'^{(i)} = \frac{(p_a^{(i)})^{\frac{1}{T}}}{\sum_j (p_a^{(j)})^{\frac{1}{T}}} \quad ; \quad \hat{p}_a'^{(i)} = \frac{(\hat{p}_a^{(i)})^{\frac{1}{T}}}{\sum_j (\hat{p}_a^{(j)})^{\frac{1}{T}}} \tag{5.2}$$

où T est un paramètre appelé la température. (Hinton, Vinyals et Dean 2014) suggèrent le réglage T> 1, ce qui augmente le poids des petites valeurs et encourage le réseau à mieux apprendre les similitudes entre les classes. Nous prenons dans nos travaux T=2.

5.3.2.4 Fonction de perte totale

La fonction de perte totale (L_{Totale}) du réseau est la somme de la distillation des connaissances ($L_{distillation}$), la fonction de perte utilisée par le réseau pour apprendre les nouvelles classes (L_{perte}), et la régularisation (R).

$$L_{Totale} = \lambda_a L_{distillation}(P_a, \hat{P}_a) + L_{perte}(Y_n, \hat{Y}_n) + R(\theta_p, \theta_a, \theta_n) \tag{5.3}$$

λ_a est un poids d'équilibre de perte entre les anciennes et nouvelles classes. En augmentant sa valeur, nous favorisons l'entraînement des anciennes images par rapport aux nouvelles images. Nous prenons dans nos travaux $\lambda_a = 1$, L_{perte} est la fonction de perte d'entropie croisée et R est la régulation des poids $\lambda \sum_i \omega_i$ avec $\lambda = 0,0005$

Cette approche présente des avantages par rapport à l'apprentissage par transfert classique dans ses deux formes : sans ou avec ré-entraînement. En effet, la stratégie de l'extraction des caractéristiques sans ré-entraînement est généralement moins performante sur de nouvelles données car les paramètres partagés θ_p sont liés aux classes d'origine et ils n'ont pas appris à extraire de caractéristiques discriminantes liées à des classes nouvelles. De l'autre côté, le fine-tuning dégrade les performances sur les classes d'origine car les paramètres partagés ont réappris. L'apprentissage incrémental permet d'apprendre un réseau

sans oublier les anciennes connaissances.

5.3.3 Résultats expérimentaux

Dans cette section, nous évaluons l'apprentissage incrémental sur la base d'images de poissons de référence LCF-15 où les taux n'ont pas été suffisamment élevés avec l'apprentissage par transfert classique.

5.3.3.1 Stratégie d'apprentissage

L'apprentissage du modèle sera fait progressivement, à chaque étape nous intégrons de nouvelles classes et nous entraînons le modèle sur tous les exemples, à savoir les anciennes et les nouvelles images.

La figure 4.12(a) du chapitre 4 montre la matrice de confusion de l'apprentissage non-incrémental d'espèces de poissons sur la base LCF-15. Nous pouvons regrouper les espèces en trois groupes : groupe d'espèces avec un taux de classification faible (AN, CC, NN, ZS), groupe d'espèces avec un taux moyen (AV, CT, MK, PD) et groupe d'espèces avec un taux élevé (AC, CL, CS, DA, DR, HM, PV). Nous commençons par entraîner le modèle sur le premier groupe en utilisant un modèle ResNet50 pré-entraîné sur la base ImageNet. A la fin de cet entraînement, le modèle génère les paramètres partagés θ_{p1} et les paramètres spécifiques aux espèces du premier groupe θ_{a1}. Ensuite, nous ajoutons les classes du deuxième groupe. Afin d'intégrer ces nouvelles classes, nous ajoutons un nombre de neurones égal au nombre des classes de ce groupe dans la couche de classification. Nous initialisons aléatoirement les valeurs des poids de ces nouveaux neurones (paramètres θ_{n2}) et nous gardons les poids correspondant aux anciennes classes (θ_{p1} et θ_{a1}). Nous appliquons dans ce deuxième entraînement la nouvelle fonction de perte afin d'apprendre les nouvelles espèces tout en gardant les connaissances apprises dans l'ancien entraînement. Nous régénérons à la fin de cet entraînement les paramètres θ_{p2} et θ_{a2} et nous refaisons les mêmes procédures avec le troisième groupe.

5.3.3.2 Résultats

La figure 5.11 montre la matrice de confusion du premier modèle entraîné sur le premier groupe qui contient les espèces difficiles à reconnaître (AN, CC, NN et ZS). Le modèle identifie très bien les espèces CC et NN, suivis de l'espèce AN. L'espèce ZS reste toujours difficile à identifier. Nous obtenons un taux de classification de 92,08%. Si nous comparons les taux de classification de ces espèces avec ceux de la figure 4.12(a), nous remarquons que dans ce modèle les taux sont plus élevés. Le but de notre nouvelle approche est de maintenir ces taux élevés en ajoutant les autres espèces.

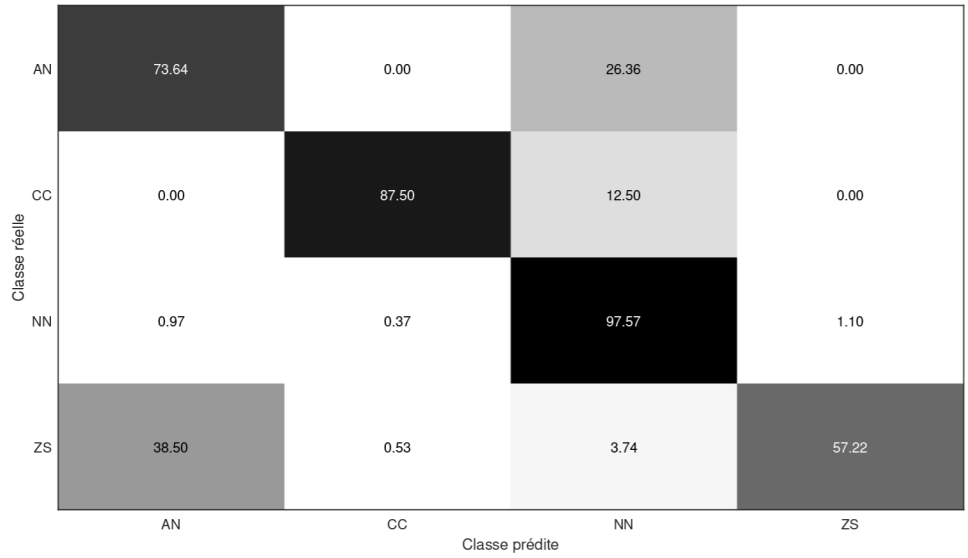


FIGURE 5.11 – Matrice de confusion sur le premier groupe de la base d'images LCF-15.

La figure 5.12 illustre la matrice de confusion du deuxième modèle entraîné sur les quatre anciennes espèces avec les quatre nouvelles espèces qui sont moyennement identifiables. Les taux de classification des anciennes espèces sont réduits mais ils restent plus élevés que ceux de l'apprentissage non-incrémental. Grâce à la fonction de perte avec la distillation de connaissances, nous avons imposé au modèle de ne pas trop oublier les connaissances requises dans le premier entraînement. Le taux de classification global a diminué seulement de 3,11% en passant de 4 à 8 espèces.

Finalement, la figure 5.13 montre la matrice de confusion pour le troisième apprentissage du modèle en ajoutant sept nouvelles espèces. Ces dernières sont les plus représentatives

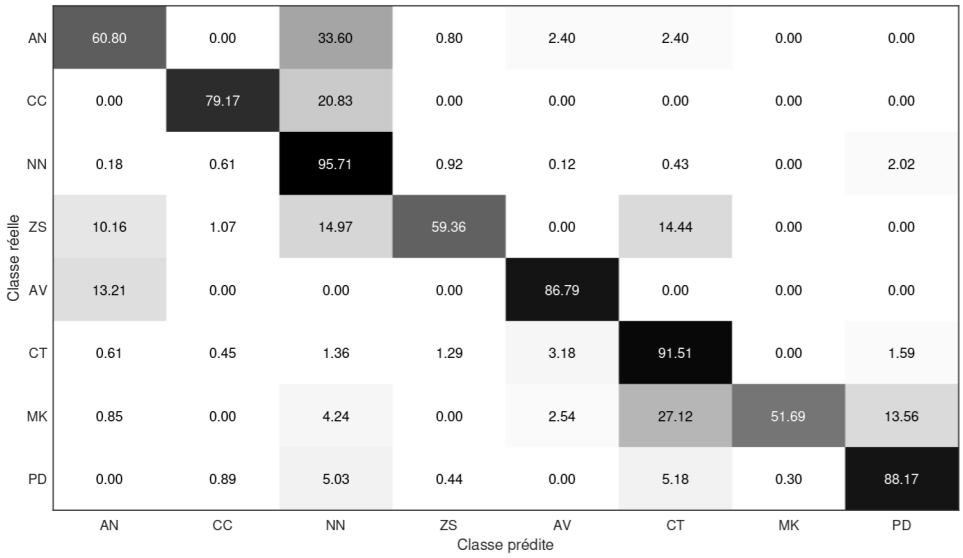


FIGURE 5.12 – Matrice de confusion sur les premier et deuxième groupe de la base d'images LCF-15.

dans la base d'images. En ajoutant ces espèces représentatives, les taux de classification des anciennes espèces sont réduits significativement ; par exemple celui de AN est passé de 60,80% à 43,41%, et celui de NN est passé de 95,71% à 62,39%. Cet oubli des anciennes connaissances est dû au fait que les nouvelles espèces sont plus représentatives et du fait de la similitude entre les anciennes et nouvelles espèces. Le taux de classification global est de 80,89%. Ce taux dépasse celui de la classification plate (77,33%).

5.4 Discussion

La figure 5.14 illustre la précision de chaque espèce de la base d'images en utilisant l'apprentissage par transfert classique et les deux approches proposées de l'apprentissage progressif. Nous y tirons ici une remarque importante ; en plus de l'amélioration des performances globales par nos approches de l'apprentissage progressif, les précisions des espèces qui sont difficiles à identifier sont assez améliorées par rapport à l'apprentissage par transfert classique. Nous avons réussi à augmenter la précision de l'espèce AN de 13,18%, CC de 62,50%, NN de 37,13%, ZS de 30,48%, AV de 39,74%, CT de 12,66%, et MK de 25,43%.

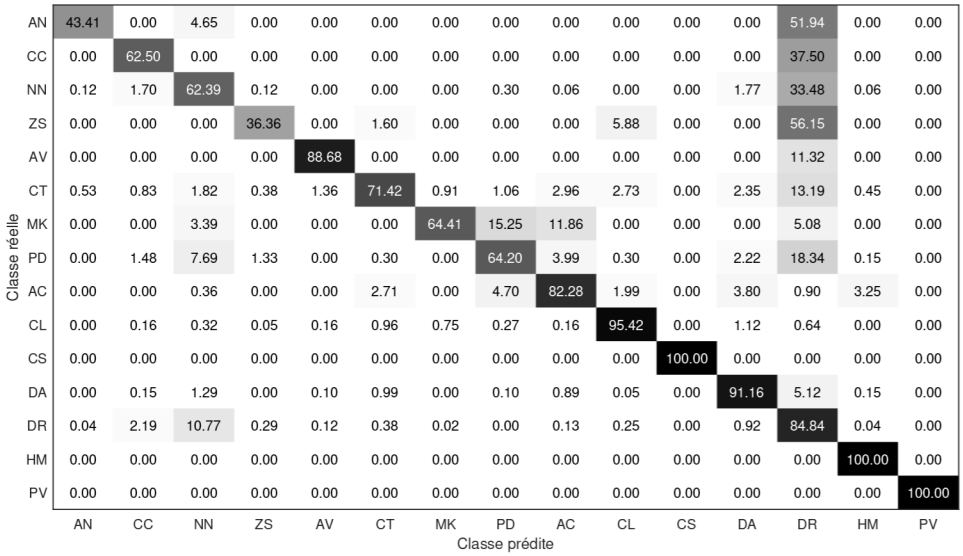


FIGURE 5.13 – Matrice de confusion sur toute la base d'images LCF-15 en utilisant l'apprentissage incrémental.

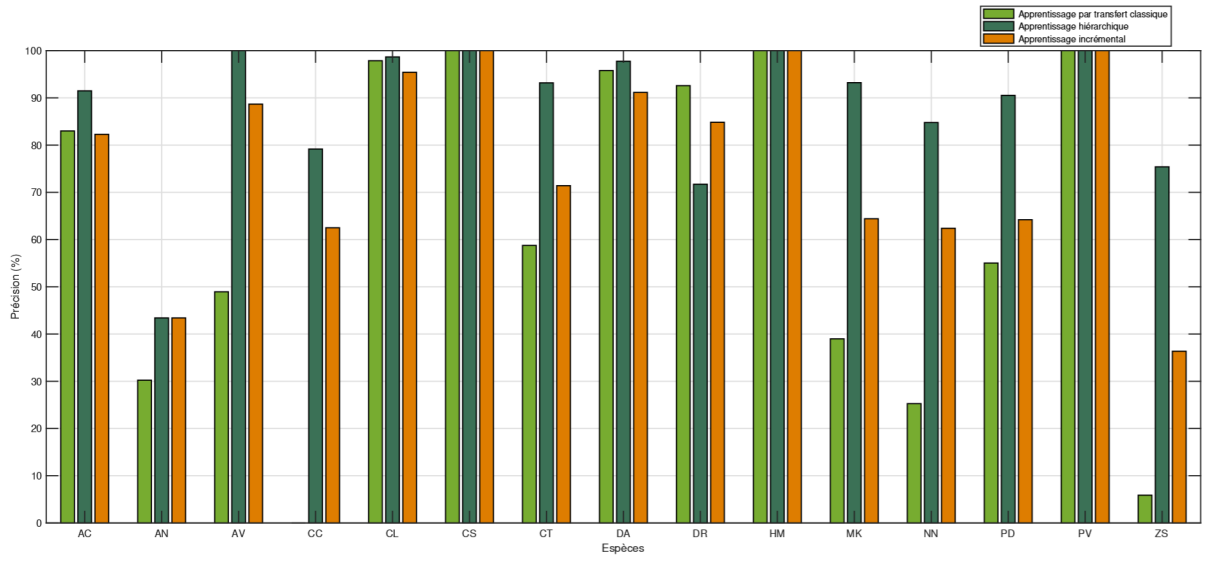
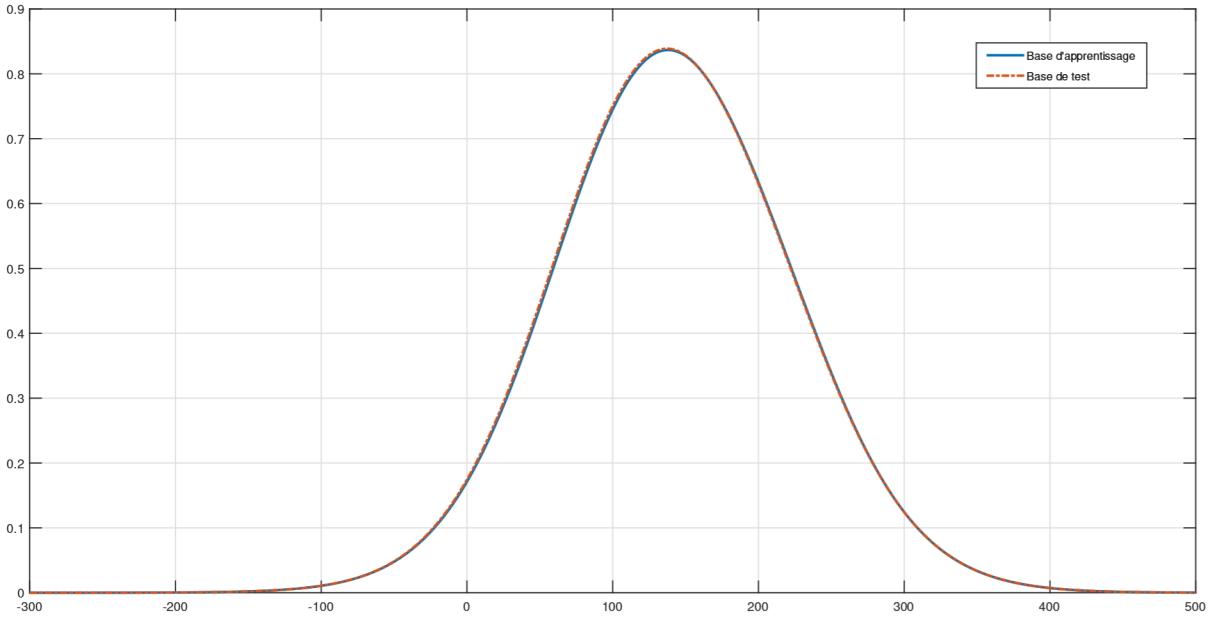


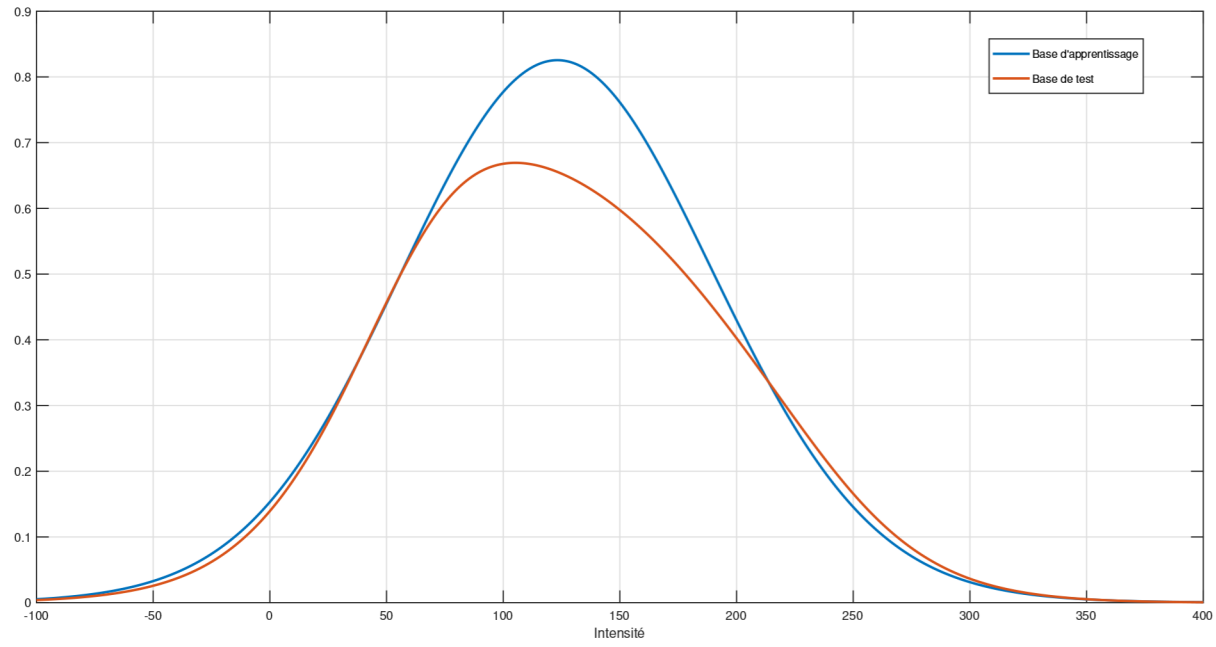
FIGURE 5.14 – Précisions de chaque espèce du modèle classique, hiérarchique et incrémental sur la base d'images LCF-15.

Pour les expérimentations, nous avons utilisé les deux bases d'images de référence FRGT et LCF-15. Nous avons vu que la base d'image FRGT fournit de très bons résultats tandis que la base LCF-15 fournit des résultats moins bons. Ceci est dû à plusieurs aspects : la mauvaise qualité des images, la résolution, le flou et les similarités entre les différentes espèces. La figure 5.15 illustre les distributions gaussiennes de l'intensité moyenne des images d'apprentissage et de test pour les deux bases d'images.

Pour la base d'images FRGT, la distribution de la base de test suit parfaitement celle de la base d'apprentissage. Ceci explique pourquoi nous avions des taux de classification élevés pour cette base d'images. Pour la base LCF-15, globalement la distribution de la base de test diffère un peu de celle de la base d'apprentissage. Pour montrer d'où vient cette différence, nous affichons les distributions gaussiennes de l'intensité moyenne pour chaque espèce comme illustré dans la figure 5.16. Nous remarquons que pour quelques espèces (AN, CC, MK,...), les distributions d'apprentissage et de test sont différentes. Ces espèces correspondent aux espèces trouvées difficiles à identifier. Ceci explique les taux de classification faibles pour ces espèces. L'ajustement de ses distributions fera l'objet d'une perspective du travail réalisé.



(a) Base FRGT



(b) Base LCF-15

FIGURE 5.15 – Distributions gaussiennes de l'intensité de la base d'apprentissage et de test : (a) la base FRGT (b) la base LCF-15.

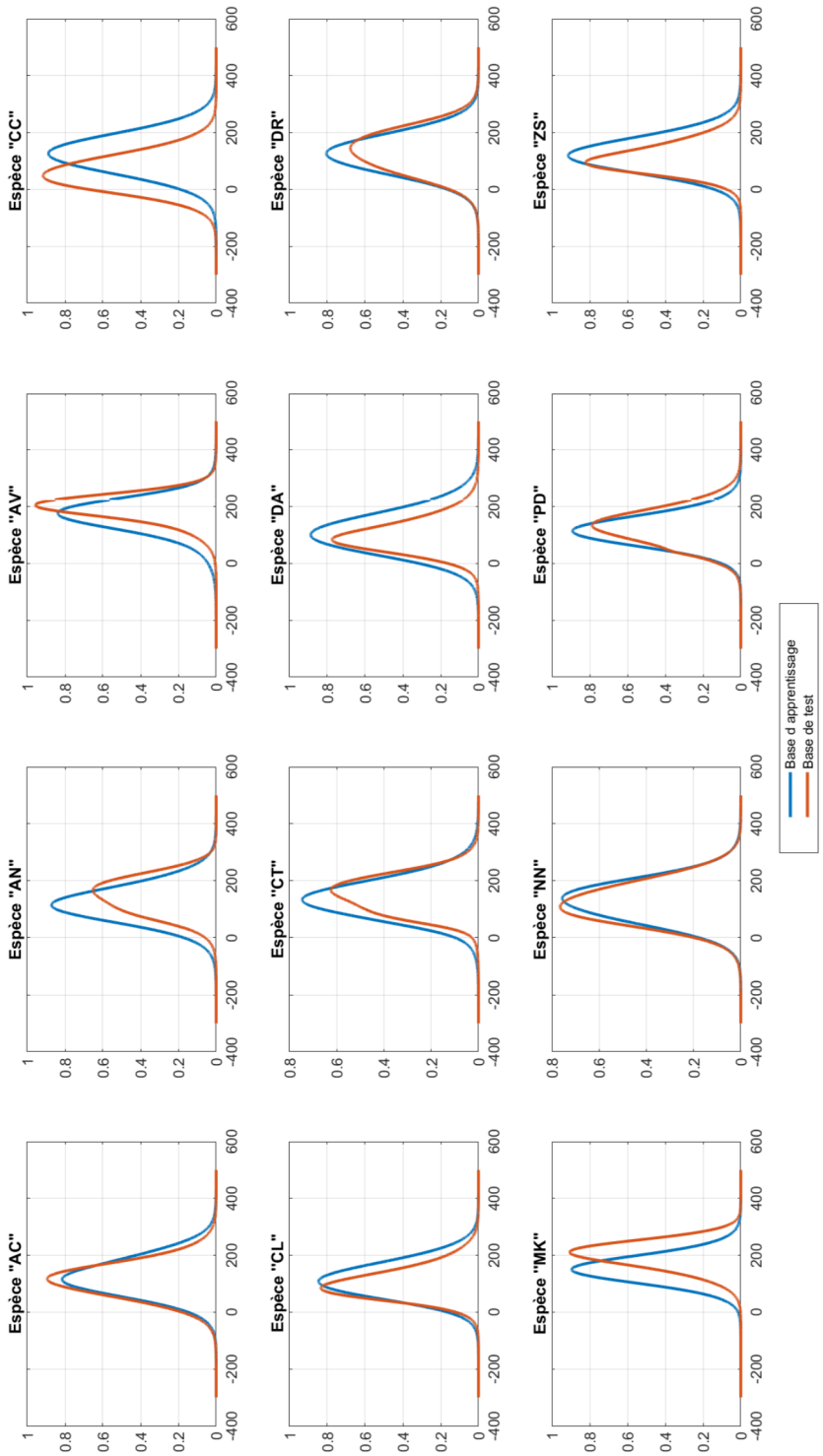


FIGURE 5.16 – Distributions gaussiennes des données de la base d'apprentissage et de test de chaque espèce de la base LCF-15.

5.5 Conclusion

Les deux approches proposées dans ce chapitre visent à améliorer les performances en classification d'espèces de poissons en utilisant l'apprentissage progressif.

La classification hiérarchique permet de classer d'abord les poissons en familles (classe générale) ensuite en espèces (classe spécifique). Avec cette approche, nous avons vu que la classification est plus informative. Les expériences sur les deux bases d'images de poissons partagent de nombreuses caractéristiques. D'abord, l'apprentissage d'un nœud sur une famille converge rapidement par rapport à l'apprentissage du modèle entier sur toute la base d'images (figure 5.3). En d'autres termes, l'apprentissage hiérarchique empêche le modèle de souffrir du problème de sur-apprentissage. Deuxièmement, l'utilisation des couches partagées pour extraire des caractéristiques de bas niveau réduit énormément le nombre de paramètres ce qui accélère le temps d'apprentissage et réduit les ressources de mémoire. Troisièmement, le modèle hiérarchique surpasse le modèle traditionnel correspondant. Cela montre que les données taxonomiques des espèces renforcent l'espace de caractéristiques de CNN. Cependant, l'inconvénient de la hiérarchie est que le modèle devient plus grand à chaque fois qu'on ajoute une nouvelle famille, ce qui demande plus de ressources de mémoire et rend l'apprentissage global plus long.

L'apprentissage incrémental permet de partir d'un nombre limité de classes (dans notre cas ce sont les espèces difficiles à identifier, classes spécifiques), ensuite nous incrémentons le nombre de classes tout en gardant des performances élevées sur les anciennes classes. L'avantage de cette approche par rapport à la classification hiérarchique est que le nombre des paramètres du modèle n'augmente pas de la même manière. Dans cette approche, nous n'ajoutons que les paramètres correspondant aux nouveaux neurones ajoutés, ce qui nous permet de gagner énormément du temps de calcul et de mémoire. L'inconvénient de cette approche est que l'apprentissage se fait sur toutes les données (les anciennes et les nouvelles), ce qui rend la phase d'entraînement de plus en plus longue.

Ces deux approches ont amélioré significativement les résultats de classification par rapport à la classification plate, surtout pour les classes difficiles.

Conclusion

Conclusion générale et perspectives

La reconnaissance d'espèces de poissons dans un environnement marin naturel est étudiée pour bien comprendre l'écosystème marin et promouvoir les applications commerciales. Cette tâche de reconnaissance est fondamentalement difficile en raison de la complexité de l'information sous-marine. Dans cette thèse, nous avons développé des approches pour la reconnaissance d'espèces de poissons dans des images sous-marines. Cette reconnaissance requiert deux étapes principales : la détection de poissons et la classification de leurs espèces.

Tout d'abord, nous avons commencé dans le chapitre 3 par la détection de poissons dans des images vidéo sous-marines par fusion de réseaux CNN parallèles. Nous avons cherché à améliorer les résultats de l'état de l'art en particulier la robustesse de la localisation de poissons en travaillant sur la fusion de différentes informations. Pour cela, nous avons proposé deux architectures à réseaux CNN parallèles qui fusionnent l'information d'apparence et de mouvement des poissons à détecter. A cette fin, en plus des images RGB, nous avons généré des cartes de mouvement en utilisant la technique du flux optique. Nos deux architectures utilisent deux Faster R-CNNs qui partagent soit le même RPN, soit le même classifieur. Ce partage d'un élément entre deux Faster R-CNNs permet de profiter d'un espace riche généré par la fusion des caractéristiques fournies par chacun des deux réseaux CNN. Nous avons pu démontrer que le partage du RPN présente de meilleurs résultats par rapport au partage du classifieur, et par rapport aux approches de l'état de l'art.

Dans le chapitre 4, nous avons proposé des approches de classification d'espèces de poissons basées sur les CNNs. Pour avoir de bonnes performances en classification avec les CNNs, nous devons avoir des bases de données de grande taille. Toutefois, les bases de données annotées disponibles sont de petite taille. Pour cela, nous nous sommes basés sur l'apprentissage par transfert pour proposer trois stratégies. Tout d'abord, le modèle CNN utilisé a servi pour extraire des caractéristiques d'images de poissons avant et après le "*fine-tuning*". Nous avons également analysé l'effet de différentes techniques d'apprentissage sur la classification, en particulier par le choix de l'espace colorimétrique, l'élimination ou non d'arrière-plan et par l'augmentation artificielle de données. Pour cette dernière, nous avons proposé un nouveau critère pour définir les classes qui ont besoin de plus d'effectif. Ce critère utilise les courbes de perte durant l'apprentissage et durant la validation. Nous

avons ainsi augmenté les effectifs uniquement pour les espèces ayant des courbes de perte non convergentes. Cette technique nous a permis d'améliorer les performances tout en utilisant le moins de ressources de mémoire et de calcul. Nous avons pu obtenir des taux de classification de 99,84% et 78,95% sur les bases d'images FRGT et LCF-15 respectivement.

Dans le chapitre 5, nous avons développé de nouvelles approches basées sur l'apprentissage progressif pour la classification d'espèces de poissons. Tout d'abord, nous avons exploité la classification taxonomique (utilisée par les biologistes pour identifier les espèces) pour construire une architecture CNN hiérarchique. Cette architecture permet de classer les poissons dans deux niveaux taxonomiques à savoir famille et espèce. Le modèle extrait d'abord les caractéristiques générales partagées par plusieurs espèces de la même famille. Ensuite, le modèle extrait les caractéristiques spécifiques discriminant les espèces appartenant à la même famille. Cette approche a permis d'atteindre des taux de classification de 99,39% et 81,31% sur les bases d'images FRGT et LCF-15 respectivement. Ces premiers résultats prometteurs encouragent à utiliser ce type de classification en familles pour certaines espèces de poissons visuellement similaires. Par ailleurs, nous avons remarqué dans la classification d'espèces de poissons qu'il y a souvent des espèces qui sont difficiles à identifier ; en particulier à cause d'une forte similarité avec d'autres espèces ou par manque de nombre d'exemples suffisant pour correctement apprendre la-dite espèce. Cette sous-catégorie nécessite une attention particulière par le modèle d'apprentissage. Nous avons proposé de focaliser le système sur cette sous-catégorie en apprenant au modèle de façon incrémentale. Le modèle CNN apprend d'abord sur cette sous-catégorie difficile. Ensuite, nous rajoutons progressivement de nouvelles espèces tout en veillant à maintenir la connaissance sur les espèces préalablement apprises. Nous avons modifié la fonction de perte du modèle pour se concentrer en particulier sur ces espèces qui posent problème. Cette approche a amélioré les performances en classification de 3,56% par rapport à l'apprentissage par transfert sur la base LCF-15.

Perspectives

Comme perspectives, nous proposons poursuivre l'étude de la reconnaissance d'espèces de poissons dans des images vidéo sous-marines. D'un point de vue méthodologique, quatre aspects vont être approfondis :

1. Les images sous-marines sont des images très bruitées à cause de plusieurs facteurs comme les phénomènes d'absorption et de diffusion des radiations, l'éclairage non uniforme, l'atténuation des couleurs et le problème de turbidité. Ces images nécessitent donc des traitements particuliers afin d'améliorer leurs résolutions. Plusieurs travaux ont proposé d'améliorer la résolution des images sous-marines en se basant sur les CNNs (Islam et al. 2020 ; Pramunendar et al. 2019 ; Zong, Chen et Wang 2020). La qualité des images d'entrée impacte fortement le comportement du réseau CNN. Pour cela, nous pourrons ajouter une étape de prétraitement avant d'attaquer un réseau CNN pour la tâche demandée. Cette étape peut toutefois engendrer un temps de traitement supplémentaire et nécessite plus d'espace mémoire, ce qui rend l'application loin d'être en temps réel. Pour surmonter ce problème, nous allons explorer la possibilité d'avoir un seul réseau CNN capable à la fois de faire toutes les tâches requises : le prétraitement d'images d'entrée, la détection et la classification.

2. Notre approche pour la détection de poissons utilise deux réseaux CNN afin de fusionner des caractéristiques issues de différents espaces d'information. Les approches multimodales ont prouvé leur efficacité et leur robustesse grâce à la complémentarité de l'information dans plusieurs applications (Farahnakian et Heikkonen 2020 ; Poria et al. 2016). Nous voudrons tester l'apport de nouvelles modalités sur notre application. Une proposition est d'utiliser la technique d'acquisition par stéréo-caméra qui peut fournir des cartes de profondeur ou d'utiliser des écho-sondeurs. Les informations issues de ces techniques peuvent alors être ajoutées comme nouvelles sources d'informations.

3. Dans la problématique de détection de poissons, nous avons englobé le poisson détecté par un rectangle défini par les coordonnées cartésiennes *(x, y)* de ses deux points haut-gauche et bas-droit. Dans le cas de proximité de deux poissons, la détection avec des rectangles englobe parfois les deux poissons dans la même boîte englobante. Nous espérons surmonter ce problème en utilisant la détection avec des ellipses. Dans ce cas, on passera de 4 à 5 paramètres à ajuster. Une ellipse est définie par les coordonnées *(x, y)* du centre, les deux axes majeur a et mineur b et l'angle de rotation θ. Une difficulté s'ajoutera ici ; le RPN devra être capable de gérer également la rotation de l'ellipse.

4. Les images d'entraînement et de test de quelques espèces de la base d'images LCF-15

ont des distributions gaussiennes différentes. Nous voulons surmonter ce problème en cherchant un algorithme qui permet d'aligner les deux distributions afin de réduire le sur-apprentissage du modèle.

D'un point de vue applicatif, les techniques de détection et de classification d'espèces de poissons peuvent contribuer à de nombreuses études marines, nous en citons principalement :

1. le suivi (tracking) de poissons (Li et al. 2018). Le suivi est une approche qui permet de suivre temporellement le poisson le long de la vidéo. Il permet d'étudier la trajectoire ainsi que le comportement du poisson (Spampinato et al. 2010) et ses interactions avec son environnement,

2. le comptage d'une ou plusieurs espèces dans une zone marine (Lainez et Gonzales 2019 ; Le et Xu 2017 ; Zhang et al. 2020). Le comptage permet d'identifier les espèces menacées, invasives ou migratrices. Il permet aussi d'étudier l'évolution d'une communauté de poissons à court et à long terme.

Enfin, nous évoquons une perspective de ce travail dans le domaine de vision par ordinateur en général. En effet, les architectures de fusion proposées pour la détection de poissons peuvent être utilisées dans d'autres applications en particulier celles qui utilisent plusieurs modalités. Les résultats obtenus par les nouvelles architectures proposées ont surpassé ceux de l'état de l'art.

Publications et bibliographie

Publications

Revues internationales

— A. BEN TAMOU, A. BENZINOU, K. NASREDDINE, "Multi-stream fish detection in unconstrained underwater videos by the fusion of two convolutional neural network detectors", Applied Intelligence, Springer, 2021, (10.1007/s10489-020-02155-8). (hal-03068449).

— A. BEN TAMOU, A. BENZINOU, K. NASREDDINE, "Incremental learning and hierarchical classification for fish species recognition", En cours de préparation.

Conférences internationales

— A. BEN TAMOU, A. BENZINOU, K. NASREDDINE, L. BALLIHI, "Underwater live fish recognition by deep learning", In : Mansouri A., El Moataz A., Nouboud F., Mammass D. (eds) Image and Signal Processing, Lecture Notes in Computer Science book series (LNCS, volume 10884), Springer, Cham, pp. 275-283, 2018.

— A. BEN TAMOU, A. BENZINOU, K. NASREDDINE, L. BALLIHI, "Transfer learning with deep convolutional neural network for underwater live fish recognition", IEEE International Image Processing, Applications and Systems Conference (IPAS'2018), Nice Sophia Antipolis, France, December 12-14, 2018.

Bibliographie

Abbas, Qaisar, Mostafa EA Ibrahim et M Arfan Jaffar (2019). « A comprehensive review of recent advances on deep vision systems ». In : *Artificial Intelligence Review* 52.1, p. 39-76 (cf. p. 45, 56).

Abe, S et al. (2017). « How many fish in a tank ? Constructing an automated fish counting system by using PTV analysis ». In : *Selected Papers from the 31st International Congress on High-Speed Imaging and Photonics*. T. 10328. International Society for Optics et Photonics, 103281T (cf. p. 24).

Ackerman, John L et David R Bellwood (2000). « Reef fish assemblages : a re-evaluation using enclosed rotenone stations ». In : *Marine Ecology Progress Series* 206, p. 227-237 (cf. p. 12).

Aguzzi, Jacopo et al. (2011). « The new seafloor observatory (OBSEA) for remote and long-term coastal ecosystem monitoring ». In : *Sensors* 11.6, p. 5850-5872 (cf. p. 18).

Allsopp, Michelle et al. (2008). *State of the World's Oceans*. Springer Science & Business Media (cf. p. 10).

Arel, Itamar, Derek C Rose et Thomas P Karnowski (2010). « Deep machine learning-a new frontier in artificial intelligence research [research frontier] ». In : *IEEE computational intelligence magazine* 5.4, p. 13-18 (cf. p. 58).

Assis, Jorge, Krupskaya Narvaez et Ricardo Haroun (2007). « Underwater towed video : a useful tool to rapidly assess elasmobranch populations in large marine protected areas ». In : *Journal of Coastal Conservation* 11.3, p. 153-157 (cf. p. 16).

Attali, Jean-Gabriel et Gilles Pagès (1997). « Approximations of functions by a multilayer perceptron : a new approach ». In : *Neural networks* 10.6, p. 1069-1081 (cf. p. 49).

Babcock, Russell C et al. (1999). « Changes in community structure in temperate marine reserves ». In : *Marine ecology progress series* 189, p. 125-134 (cf. p. 18).

Baldi, Pierre et Peter J Sadowski (2013). « Understanding dropout ». In : *Advances in neural information processing systems*, p. 2814-2822 (cf. p. 74).

Barnes, H (1955). « Underwater television and research in marine biology, bottom topography and geology ». In : *Deutsche Hydrografische Zeitschrift* 8.6, p. 213-236 (cf. p. 17).

Bassett, DK et JC Montgomery (2011). « Investigating nocturnal fish populations in situ using baited underwater video : with special reference to their olfactory capabi-

lities ». In : *Journal of Experimental Marine Biology and Ecology* 409.1-2, p. 194-199 (cf. p. 18).

Bay, Herbert, Tinne Tuytelaars et Luc Van Gool (2006). « Surf : Speeded up robust features ». In : *European conference on computer vision*. Springer, p. 404-417 (cf. p. 33).

Begg, Gavin A et John R Waldman (1999). « An holistic approach to fish stock identification ». In : *Fisheries research* 43.1-3, p. 35-44 (cf. p. 22).

Benedetti-Cecchi, Lisandro et al. (1996). « Estimating the abundance of benthic invertebrates : a comparison of procedures and variability between observers ». In : *Marine Ecology Progress Series* 138, p. 93-101 (cf. p. 22).

Bengio, Yoshua (2009). *Learning deep architectures for AI*. Now Publishers Inc (cf. p. 25).

Bengio, Yoshua, Aaron Courville et Pascal Vincent (2013). « Representation learning : A review and new perspectives ». In : *IEEE transactions on pattern analysis and machine intelligence* 35.8, p. 1798-1828 (cf. p. 58, 61).

Benson, Bridget et al. (2009). « Field programmable gate array (FPGA) based fish detection using Haar classifiers ». In : *American Academy of Underwater Sciences* (cf. p. 30).

Bergstra, James et Yoshua Bengio (2012). « Random search for hyper-parameter optimization ». In : *The Journal of Machine Learning Research* 13.1, p. 281-305 (cf. p. 52).

Bernard, ATF et al. (2013). « Observer bias and detection probability in underwater visual census of fish assemblages measured with independent double-observers ». In : *Journal of Experimental Marine Biology and Ecology* 443, p. 75-84 (cf. p. 14).

Bianco, Gianfranco et al. (2015). « A new color correction method for underwater imaging ». In : *The International Archives of Photogrammetry, Remote Sensing and Spatial Information Sciences* 40.5, p. 25 (cf. p. 97, 125).

Bodla, Navaneeth et al. (2017). « Soft-NMS–improving object detection with one line of code ». In : *Proceedings of the IEEE international conference on computer vision*, p. 5561-5569 (cf. p. 80).

Bohnsack, James A et Scott P Bannerot (1986). « A stationary visual census technique for quantitatively assessing community structure of coral reef fishes ». In : (cf. p. 14, 16).

Boom, Bastiaan J et al. (2012a). « Long-term underwater camera surveillance for monitoring and analysis of fish populations ». In : *VAIB12* (cf. p. 124, 144).

Boom, Bastiaan J et al. (2012b). « Supporting ground-truth annotation of image datasets using clustering ». In : *Proceedings of the 21st International Conference on Pattern Recognition (ICPR2012)*. IEEE, p. 1542-1545 (cf. p. 35, 124, 144).

Bortone, Stephen A, Tony Martin et Charles M Bundrick (1991). « Visual census of reef fish assemblages : a comparison of slate, audio, and video recording devices ». In : *Gulf of Mexico Science* 12.1, p. 2 (cf. p. 17).

— (1994). « Factors affecting fish assemblage development on a modular artificial reef in a northern Gulf of Mexico estuary ». In : *Bulletin of Marine Science* 55.2-3, p. 319-332 (cf. p. 16).

Bos, Siegfried et E Chug (1996). « Using weight decay to optimize the generalization ability of a perceptron ». In : *Proceedings of International Conference on Neural Networks (ICNN'96)*. T. 1. IEEE, p. 241-246 (cf. p. 54).

Boureau, Y-Lan, Jean Ponce et Yann LeCun (2010). « A theoretical analysis of feature pooling in visual recognition ». In : *Proceedings of the 27th international conference on machine learning (ICML-10)*, p. 111-118 (cf. p. 65).

Bozec, Yves-Marie et al. (2011). « Factors affecting the detection distances of reef fish : implications for visual counts ». In : *Marine Biology* 158.5, p. 969-981 (cf. p. 14).

Brandl, Simon J et al. (2018). « The hidden half : ecology and evolution of cryptobenthic fishes on coral reefs ». In : *Biological Reviews* 93.4, p. 1846-1873 (cf. p. 10).

Brock, Richard E (1982). « A critique of the visual census method for assessing coral reef fish populations ». In : *Bulletin of Marine Science* 32.1, p. 269-276 (cf. p. 14).

Brock, Vernon E (1954). « A preliminary report on a method of estimating reef fish populations ». In : *The Journal of Wildlife Management* 18.3, p. 297-308 (cf. p. 13).

Buckland, Stephen T et al. (2001). « Introduction to distance sampling : estimating abundance of biological populations ». In : (cf. p. 13).

Burnham, Kenneth P, David R Anderson et Jeffrey L Laake (1980). « Estimation of density from line transect sampling of biological populations ». In : *Wildlife monographs* 72, p. 3-202 (cf. p. 13).

Cabrera-Gámez, Jorge et al. (2015). « Exploring the use of local descriptors for fish recognition in lifeclef 2015 ». In : *CEUR Workshop Proceedings* (cf. p. 32, 114).

Carleton, JH et TJ Done (1995). « Quantitative video sampling of coral reef benthos : large-scale application ». In : *Coral Reefs* 14.1, p. 35-46 (cf. p. 15).

Carreira-Perpinan, Miguel A et Geoffrey E Hinton (2005). « On contrastive divergence learning. » In : *Aistats*. T. 10. Citeseer, p. 33-40 (cf. p. 57).

Caruana, Rich, Steve Lawrence et C Lee Giles (2001). « Overfitting in neural nets : Backpropagation, conjugate gradient, and early stopping ». In : *Advances in neural information processing systems*, p. 402-408 (cf. p. 54).

Chabanet, Pascale et al. (2012). « VideoSolo, an autonomous video system for high-frequency monitoring of aquatic biota, applied to coral reef fishes in the Glorioso Islands (SWIO) ». In : *Journal of Experimental Marine Biology and Ecology* 430, p. 10-16 (cf. p. 18).

Chan, Tsung-Han et al. (2015). « PCANet : A simple deep learning baseline for image classification ? » In : *IEEE transactions on image processing* 24.12, p. 5017-5032 (cf. p. 33).

Chapman, CJ et al. (1974). « Reactions of fish to sound generated by divers' open-circuit underwater breathing apparatus ». In : *Marine Biology* 27.4, p. 357-366 (cf. p. 15).

Chateau, Olivier et Laurent Wantiez (2005). « Comparaison de la structure des communautés de poissons coralliens d'intérêt commercial entre une réserve marine et deux zones non protégées dans le Parc du lagon sud de Nouvelle-Calédonie ». In : *Cybium* 29.2, p. 159-174 (cf. p. 14).

Chatfield, Ken et al. (2014). « Return of the devil in the details : Delving deep into convolutional nets ». In : *arXiv preprint arXiv :1405.3531* (cf. p. 27).

Cho, KyungHyun et al. (2013). « A two-stage pretraining algorithm for deep boltzmann machines ». In : *International Conference on Artificial Neural Networks*. Springer, p. 106-113 (cf. p. 59).

Cireşan, Dan et al. (2012). « Multi-column deep neural network for traffic sign classification ». In : *Neural networks* 32, p. 333-338 (cf. p. 77).

Colton, Madhavi A et Stephen E Swearer (2010). « A comparison of two survey methods : differences between underwater visual census and baited remote underwater video ». In : *Marine Ecology Progress Series* 400, p. 19-36 (cf. p. 19).

Colvocoresses, James et Alejandro Acosta (2007). « A large-scale field comparison of strip transect and stationary point count methods for conducting length-based underwater visual surveys of reef fish populations ». In : *Fisheries Research* 85.1-2, p. 130-141 (cf. p. 14).

Cootes, Timothy F., Gareth J. Edwards et Christopher J. Taylor (2001). « Active appearance models ». In : *IEEE Transactions on pattern analysis and machine intelligence* 23.6, p. 681-685 (cf. p. 30).

Cruz, Igor, Ruy KP Kikuchi et Zelinda MAN Leão (2008). « Use of the video transect method for characterizing the Itacolomis reefs, eastern Brazil ». In : *Brazilian Journal of Oceanography* 56.4, p. 271-280 (cf. p. 15).

Currey-Randall, Leanne M et al. (2020). « Optimal soak times for Baited Remote Underwater Video Station surveys of reef-associated elasmobranchs ». In : *PloS one* 15.5, e0231688 (cf. p. 18).

Daufresne, Martin et Philippe Boet (2007). « Climate change impacts on structure and diversity of fish communities in rivers ». In : *Global Change Biology* 13.12, p. 2467-2478 (cf. p. 2).

David, Hill et al. (2005). *Handbook of biodiversity methods : survey, evaluation and monitoring*. Cambridge University Press (cf. p. 13).

Deng, Jia et al. (2009). « Imagenet : A large-scale hierarchical image database ». In : *2009 IEEE conference on computer vision and pattern recognition*. Ieee, p. 248-255 (cf. p. 55, 97, 100, 116).

Dennis Jr, John E et Jorge J Moré (1977). « Quasi-Newton methods, motivation and theory ». In : *SIAM review* 19.1, p. 46-89 (cf. p. 50).

Deschamps, Gérard (2003). *Les chaluts*. Editions Quae (cf. p. 12).

Dibble, Eric D (1991). « A comparison of diving and rotenone methods for determining relative abundance of fish ». In : *Transactions of the American Fisheries Society* 120.5, p. 663-666 (cf. p. 12).

Dickens, Luke C et al. (2011). « Quantifying relative diver effects in underwater visual censuses ». In : *PloS one* 6.4, e18965 (cf. p. 14, 15).

Duch, Wlodzislaw et Norbert Jankowski (1999). « Survey of neural transfer functions ». In : *Neural Computing Surveys* 2.1, p. 163-212 (cf. p. 47).

Dudel, J (1983). « Function of nerve cells ». In : *Human physiology*. Springer, p. 3-31 (cf. p. 46).

D'agata, Stéphanie et al. (2014). « Human-mediated loss of phylogenetic and functional diversity in coral reef fishes ». In : *Current Biology* 24.5, p. 555-560 (cf. p. 11).

Edgar, Graham J et Neville S Barrett (1999). « Effects of the declaration of marine reserves on Tasmanian reef fishes, invertebrates and plants ». In : *Journal of Experimental Marine Biology and Ecology* 242.1, p. 107-144 (cf. p. 14).

Edgar, Graham J, Neville S Barrett et Alastair J Morton (2004). « Biases associated with the use of underwater visual census techniques to quantify the density and size-

structure of fish populations ». In : *Journal of Experimental Marine Biology and Ecology* 308.2, p. 269-290 (cf. p. 14, 22).

EIIis, Denise M et Edward E DeMartini (1995). « technique for indexing abundances of juvenile pink snapper ». In : *Fishery Bulletin* 93, p. 67-77 (cf. p. 18).

Engel, Jonna et Rikk Kvitek (1998). « Effects of otter trawling on a benthic community in Monterey Bay National Marine Sanctuary ». In : *Conservation Biology* 12.6, p. 1204-1214 (cf. p. 11).

Erhan, Dumitru et al. (2010). « Why does unsupervised pre-training help deep learning ? » In : *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics*, p. 201-208 (cf. p. 75).

Facon, Mathilde et al. (2016). « A comparative study of the accuracy and effectiveness of line and point intercept transect methods for coral reef monitoring in the southwestern Indian Ocean islands ». In : *Ecological Indicators* 60, p. 1045-1055 (cf. p. 14).

Farabet, Clement et al. (2012). « Learning hierarchical features for scene labeling ». In : *IEEE transactions on pattern analysis and machine intelligence* 35.8, p. 1915-1929 (cf. p. 26).

Farahnakian, Fahimeh et Jukka Heikkonen (2020). « Deep Learning Based Multi-Modal Fusion Architectures for Maritime Vessel Detection ». In : *Remote Sensing* 12.16, p. 2509 (cf. p. 92, 108, 173).

Ferguson, Max et al. (2017). « Automatic localization of casting defects with convolutional neural networks ». In : *2017 IEEE international conference on big data (big data)*. IEEE, p. 1726-1735 (cf. p. 70).

Fernandes, IM et al. (2017). « The efficacy of clove oil as an anaesthetic and in euthanasia procedure for small-sized tropical fishes ». In : *Brazilian Journal of Biology* 77.3, p. 444-450 (cf. p. 12).

Foveau, Aurélie, Sylvain Haquin et Jean-Claude Dauvin (2017). « Using underwater imagery as a complementary tool for benthos sampling in an area with high-energy hydrodynamic conditions ». In : (cf. p. 17).

Fukushima, Kunihiko (1980). « A self-organizing neural network model for a mechanism of pattern recognition unaffected by shift in position ». In : *Biol. Cybern.* 36, p. 193-202 (cf. p. 55).

Fukushima, Kunihiko, Sei Miyake et Takayuki Ito (1983). « Neocognitron : A neural network model for a mechanism of visual pattern recognition ». In : *IEEE transactions on systems, man, and cybernetics* 5, p. 826-834 (cf. p. 55).

GHAZILOU, Amir, Mohammad reza SHOKRI et William GLADSTONE (2019). « Comparison of baited remote underwater video (BRUV) and underwater visual census (UVC) for assessment of reef fish in a marginal reef in the northern Persian Gulf ». In : *Iranian Journal of Ichthyology* 6.3, p. 197-207 (cf. p. 19).

Gibson, R, R Atkinson et J Gordon (2016). « A review of underwater stereo-image measurement for marine biology and ecology applications ». In : *Oceanography and marine biology : an annual review* 47, p. 257-292 (cf. p. 19).

Gibson, RN, RJA Atkinson et JDM Gordon (2012). « Challenges to the assessment of benthic populations and biodiversity as a result of rhythmic behaviour : Video solutions from cabled observatories ». In : *Oceanography and Marine Biology : An Annual Review* 50, p. 233-284 (cf. p. 18).

Girshick, Ross (2015). « Fast r-cnn ». In : *Proceedings of the IEEE international conference on computer vision*, p. 1440-1448 (cf. p. 26, 33, 79).

Girshick, Ross et al. (2014). « Rich feature hierarchies for accurate object detection and semantic segmentation ». In : *Proceedings of the IEEE conference on computer vision and pattern recognition*, p. 580-587 (cf. p. 26, 78).

Goetze, Jordan S et al. (2019). « A field and video analysis guide for diver operated stereo-video ». In : *Methods in Ecology and Evolution* 10.7, p. 1083-1090 (cf. p. 16).

Goodfellow, Ian et al. (2009). « Measuring invariances in deep networks ». In : *Advances in neural information processing systems*, p. 646-654 (cf. p. 61).

Goodfellow, Ian et al. (2016). *Deep learning*. T. 1. 2. MIT press Cambridge (cf. p. 3, 55).

Goodfellow, Ian J, Aaron Courville et Yoshua Bengio (2013). « Joint training deep boltzmann machines for classification ». In : *arXiv preprint arXiv :1301.3568* (cf. p. 59).

Gordon, Timothy AC et al. (2018). « Habitat degradation negatively affects auditory settlement behavior of coral reef fishes ». In : *Proceedings of the National Academy of Sciences* 115.20, p. 5193-5198 (cf. p. 11).

Grauman, Kristen et Trevor Darrell (2005). « The pyramid match kernel : Discriminative classification with sets of image features ». In : *Tenth IEEE International Conference on Computer Vision (ICCV'05) Volume 1*. T. 2. IEEE, p. 1458-1465 (cf. p. 33).

Guerry, Joris, Bertrand Le Saux et David Filliat (2017). « " Look at this one" detection sharing between modality-independent classifiers for robotic discovery of people ». In :

2017 European Conference on Mobile Robots (ECMR). IEEE, p. 1-6 (cf. p. 39, 92, 94, 95, 100, 108).

Guo, Yanming et al. (2016). « Deep learning for visual understanding : A review ». In : *Neurocomputing* 187, p. 27-48 (cf. p. 45).

Hall, K et R Hanlon (2002). « Principal features of the mating system of a large spawning aggregation of the giant Australian cuttlefish Sepiaapama (Mollusca : Cephalopoda) ». In : *Marine Biology* 140.3, p. 533-545 (cf. p. 16).

Harasti, D et al. (2017). « Use of stereo baited remote underwater video systems to estimate the presence and size of white sharks (Carcharodon carcharias) ». In : *Marine and Freshwater Research* 68.7, p. 1391-1396 (cf. p. 19).

Harris, Daniel L et al. (2018). « Coral reef structural complexity provides important coastal protection from waves under rising sea levels ». In : *Science Advances* 4.2, eaao4350 (cf. p. 10).

Harvey, ES et al. (2012a). « Contrasting habitat use of diurnal and nocturnal fish assemblages in temperate Western Australia ». In : *Journal of Experimental Marine Biology and Ecology* 426, p. 78-86 (cf. p. 19).

Harvey, ES et al. (2012b). « Response of diurnal and nocturnal coral reef fish to protection from fishing : an assessment using baited remote underwater video ». In : *Coral Reefs* 31.4, p. 939-950 (cf. p. 19).

Harvey, Euan, David Fletcher et Mark Shortis (2002). « Estimation of reef fish length by divers and by stereo-video : a first comparison of the accuracy and precision in the field on living fish under operational conditions ». In : *Fisheries Research* 57.3, p. 255-265 (cf. p. 19).

Harvey, Euan et al. (2002). « A comparison of the accuracy and precision of measurements from single and stereo-video systems ». In : *Marine Technology Society Journal* 36.2, p. 38-49 (cf. p. 19).

Harvey, Euan et al. (2004). « A comparison of underwater visual distance estimates made by scuba divers and a stereo-video system : implications for underwater visual census of reef fish abundance ». In : *Marine and Freshwater Research* 55.6, p. 573-580 (cf. p. 19).

Harvey, Euan S et al. (2007). « Bait attraction affects the performance of remote underwater video stations in assessment of demersal fish community structure ». In : *Marine Ecology Progress Series* 350, p. 245-254 (cf. p. 18).

Hawkins, Douglas M (2004). « The problem of overfitting ». In : *Journal of chemical information and computer sciences* 44.1, p. 1-12 (cf. p. 54).

HAYES, Tyler L, Nathan D CAHILL et Christopher KANAN (2019). « Memory efficient experience replay for streaming learning ». In : *2019 International Conference on Robotics and Automation (ICRA)*. IEEE, p. 9769-9776 (cf. p. 156).

HE, Kaiming et al. (2015). « Spatial pyramid pooling in deep convolutional networks for visual recognition ». In : *IEEE transactions on pattern analysis and machine intelligence* 37.9, p. 1904-1916 (cf. p. 76).

— (2016). « Deep residual learning for image recognition ». In : *Proceedings of the IEEE conference on computer vision and pattern recognition*, p. 770-778 (cf. p. 55, 70, 73, 76, 100).

HE, Kaiming et al. (2017). « Mask r-cnn ». In : *Proceedings of the IEEE international conference on computer vision*, p. 2961-2969 (cf. p. 81, 82).

HE, Shan et al. (2013). « Facial expression recognition using deep Boltzmann machine from thermal infrared images ». In : *2013 Humaine Association Conference on Affective Computing and Intelligent Interaction*. IEEE, p. 239-244 (cf. p. 59).

HEAGNEY, Elizabeth C et al. (2007). « Pelagic fish assemblages assessed using mid-water baited video : standardising fish counts using bait plume size ». In : *Marine Ecology Progress Series* 350, p. 255-266 (cf. p. 18).

HINTON, Geoffrey, Oriol VINYALS et Jeffrey DEAN (2014). « Distilling the Knowledge in a Neural Network. NIPS 2014 Deep Learning Workshop ». In : *arXiv preprint arXiv :1503.02531* (cf. p. 158, 159).

HINTON, Geoffrey E (2007). « Learning multiple layers of representation ». In : *Trends in cognitive sciences* 11.10, p. 428-434 (cf. p. 55).

HINTON, Geoffrey E, Simon OSINDERO et Yee-Whye TEH (2006). « A fast learning algorithm for deep belief nets ». In : *Neural computation* 18.7, p. 1527-1554 (cf. p. 55, 57).

HINTON, Geoffrey E et Ruslan R SALAKHUTDINOV (2006). « Reducing the dimensionality of data with neural networks ». In : *science* 313.5786, p. 504-507 (cf. p. 60, 61).

HINTON, Geoffrey E et Russ R SALAKHUTDINOV (2012). « A better way to pretrain deep boltzmann machines ». In : *Advances in Neural Information Processing Systems*, p. 2447-2455 (cf. p. 59).

HINTON, Geoffrey E, Terrence J SEJNOWSKI et al. (1986). « Learning and relearning in Boltzmann machines ». In : *Parallel distributed processing : Explorations in the microstructure of cognition* 1.282-317, p. 2 (cf. p. 56).

Hinton, Geoffrey E et al. (2012). « Improving neural networks by preventing co-adaptation of feature detectors ». In : *arXiv preprint arXiv :1207.0580* (cf. p. 74).

Holt, Dean (1967). « Opportunities for research utilizing underwater TV and acoustic systems ». In : *BioScience* 17.9, p. 635-636 (cf. p. 18).

Hong, Sanghoon et al. (2016). « PVANet : Lightweight deep neural networks for real-time object detection ». In : *arXiv preprint arXiv :1611.08588* (cf. p. 27).

Horn, Berthold KP et Brian G Schunck (1981). « Determining optical flow ». In : *Techniques and Applications of Image Understanding*. T. 281. International Society for Optics et Photonics, p. 319-331 (cf. p. 98).

Hou, Minjun et al. (2018). « Joint residual learning for underwater image enhancement ». In : *2018 25th IEEE International Conference on Image Processing (ICIP)*. IEEE, p. 4043-4047 (cf. p. 2).

Hsiao, Yi-Hao et al. (2014). « Real-world underwater fish recognition and identification, using sparse representation ». In : *Ecological informatics* 23, p. 13-21 (cf. p. 25, 90).

Huang, Gary B, Honglak Lee et Erik Learned-Miller (2012). « Learning hierarchical representations for face verification with convolutional deep belief networks ». In : *2012 IEEE Conference on Computer Vision and Pattern Recognition*. IEEE, p. 2518-2525 (cf. p. 58).

Huang, Jonathan et al. (2017). « Speed/accuracy trade-offs for modern convolutional object detectors ». In : *Proceedings of the IEEE conference on computer vision and pattern recognition*, p. 7310-7311 (cf. p. 91).

Huang, Phoenix X, Bastiaan J Boom et Robert B Fisher (2012). « Underwater live fish recognition using a balance-guaranteed optimized tree ». In : *Asian Conference on Computer Vision*. Springer, p. 422-433 (cf. p. 32, 114).

Hughes, Terry P et al. (2003). « Climate change, human impacts, and the resilience of coral reefs ». In : *science* 301.5635, p. 929-933 (cf. p. 10).

Hughes, Terry P et al. (2017). « Coral reefs in the Anthropocene ». In : *Nature* 546.7656, p. 82-90 (cf. p. 11).

Islam, Md Jahidul et al. (2020). « Underwater image super-resolution using deep residual multipliers ». In : *2020 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, p. 900-906 (cf. p. 173).

Jabbar, H et Rafiqul Zaman Khan (2015). « Methods to avoid over-fitting and under-fitting in supervised machine learning (comparative study) ». In : *Computer Science, Communication and Instrumentation Devices*, p. 163-172 (cf. p. 54).

Jackson, Jeremy BC et al. (2001). « Historical overfishing and the recent collapse of coastal ecosystems ». In : *science* 293.5530, p. 629-637 (cf. p. 11).

Jäger, Jonas et al. (2016). « SeaCLEF 2016 : Object Proposal Classification for Fish Detection in Underwater Videos. » In : *CLEF (Working Notes)*, p. 481-489 (cf. p. 28, 34, 91, 114, 136, 137).

Jennings, S, MJ Kaiser et JD Reynolds (2001). *Marine Fisheries Ecology (p. 435)* (cf. p. 14).

Jennings, Simon et al. (2001). « Impacts of trawling disturbance on the trophic structure of benthic invertebrate communities ». In : *Marine Ecology Progress Series* 213, p. 127-142 (cf. p. 11).

Ji, Shuiwang et al. (2012). « 3D convolutional neural networks for human action recognition ». In : *IEEE transactions on pattern analysis and machine intelligence* 35.1, p. 221-231 (cf. p. 77).

Jiang, Xiaojuan et al. (2013). « A novel sparse auto-encoder for deep unsupervised learning ». In : *2013 Sixth International Conference on Advanced Computational Intelligence (ICACI)*. IEEE, p. 256-261 (cf. p. 60).

Jin, Leilei et Hong Liang (2017). « Deep learning for underwater image recognition in small sample size situations ». In : *OCEANS 2017-Aberdeen*. IEEE, p. 1-4 (cf. p. 24).

Johannes, RE (1975). « Pollution and degradation of coral reef communities ». In : *Elsevier Oceanography Series*. T. 12. Elsevier, p. 13-51 (cf. p. 2, 11).

Joly, Alexis et al. (2015). « LifeCLEF 2015 : multimedia life species identification challenges ». In : *International Conference of the Cross-Language Evaluation Forum for European Languages*. Springer, p. 462-483 (cf. p. 124, 144).

Jones, Robert S et M John Thompson (1978). « Comparison of Florida reef fish assemblages using a rapid visual technique ». In : *Bulletin of Marine Science* 28.1, p. 159-172 (cf. p. 13).

Kasaei, S Hamidreza et al. (2020). « Investigating the Importance of Shape Features, Color Constancy, Color Spaces and Similarity Measures in Open-Ended 3D Object Recognition ». In : *arXiv preprint arXiv :2002.03779* (cf. p. 39, 115).

Keat-Chuan Ng, C et al. (2017). « A review of fish taxonomy conventions and species identification techniques ». In : *Survey in Fisheries Sciences* 4.1, p. 54-93 (cf. p. 21-23).

Kenyon, Jean C et al. (2006). « Towed-diver surveys, a method for mesoscale spatial assessment of benthic reef habitat : a case study at Midway Atoll in the Hawaiian Archipelago ». In : *Coastal Management* 34.3, p. 339-349 (cf. p. 15).

Khalifa, Nour Eldeen M, Mohamed Hamed N Taha et Aboul Ella Hassanien (2018). « Aquarium family fish species identification system using deep neural networks ». In : *International Conference on Advanced Intelligent Systems and Informatics*. Springer, p. 347-356 (cf. p. 31).

Kim, Hyun-Koo, Ju H Park et Ho-Youl Jung (2018). « An efficient color space for deep-learning based traffic light recognition ». In : *Journal of Advanced Transportation* 2018 (cf. p. 39, 115).

Kirkpatrick, James et al. (2017). « Overcoming catastrophic forgetting in neural networks ». In : *Proceedings of the national academy of sciences* 114.13, p. 3521-3526 (cf. p. 156).

Koskela, Timo et al. (1996). « Time series prediction with multilayer perceptron, FIR and Elman neural networks ». In : *Proceedings of the World Congress on Neural Networks*. Citeseer, p. 491-496 (cf. p. 49).

Krizhevsky, Alex, Ilya Sutskever et Geoffrey E Hinton (2012). « Imagenet classification with deep convolutional neural networks ». In : *Advances in neural information processing systems*, p. 1097-1105 (cf. p. 26, 34, 55, 68, 69, 76, 77, 119, 120).

Krohn, Martha M et Daniel Boisdair (1994). « Use of a stereo-video system to estimate the energy expenditure of free-swimming fish ». In : *Canadian Journal of Fisheries and Aquatic Sciences* 51.5, p. 1119-1127 (cf. p. 16).

Kronengold, M et al. (1964). « An acoustic-video system for marine biological research. Description of the system ». In : *Marine bio-acoustics*. Pergamon Press New York, p. 11-25 (cf. p. 18).

Kulbicki, Michel (1998). « How the acquired behaviour of commercial reef fishes may influence the results obtained from visual censuses ». In : *Journal of Experimental Marine Biology and Ecology* 222.1-2, p. 11-30 (cf. p. 14).

Kulbicki, Michel et Sébastien Sarramégna (1999). « Comparison of density estimates derived from strip transect and distance sampling for underwater visual censuses : a case study of Chaetodontidae and Pomacanthidae ». In : *Aquatic Living Resources* 12.5, p. 315-325 (cf. p. 14).

Kulbicki, Michel et al. (2010). « Counting coral reef fishes : interaction between fish life-history traits and transect design ». In : *Journal of Experimental Marine Biology and Ecology* 387.1-2, p. 15-23 (cf. p. 14).

Kumpf, HE et JM Lowenstein (1962). « Undersea observation station ». In : *Sea Frontiers* 8.4, p. 198-206 (cf. p. 18).

Labrosse, Pierre, Michel Kulbicki, Jocelyne Ferraris et al. (2002). « Underwater visual fish census surveys : Proper use and implementation ». In : (cf. p. 13).

LaFond, EC (1968). « Photographic problems in oceanography ». In : *Underwater Photo-Optical Instrumentation Applications II*. T. 12. International Society for Optics et Photonics, p. 11-20 (cf. p. 17).

Lainez, Sheryl May D et Dennis B Gonzales (2019). « Automated Fingerlings Counting Using Convolutional Neural Network ». In : *2019 IEEE 4th International Conference on Computer and Communication Systems (ICCCS)*. IEEE, p. 67-72 (cf. p. 174).

Lam, Katherine et al. (2006). « A comparison of video and point intercept transect methods for monitoring subtropical coral communities ». In : *Journal of Experimental Marine Biology and Ecology* 333.1, p. 115-128 (cf. p. 15).

Lan, Yongtian et al. (2014). « Robot fish detection based on a combination method of three-frame-difference and background subtraction ». In : *The 26th Chinese Control and Decision Conference (2014 CCDC)*. IEEE, p. 3905-3909 (cf. p. 25).

Langlois, Tim et al. (2006). « Baited underwater video for assessing reef fish populations in marine reserves ». In : *Fisheries Newsletter-South Pacific Commission* 118, p. 53 (cf. p. 18).

Langlois, Timothy J et al. (2012). « Similarities between line fishing and baited stereo-video estimations of length-frequency : novel application of kernel density estimates ». In : *PLoS One* 7.11, e45973 (cf. p. 19).

Langlois, TJ, ES Harvey et JJ Meeuwig (2012). « Strong direct and inconsistent indirect effects of fishing found using stereo-video : Testing indicators from fisheries closures ». In : *Ecological Indicators* 23, p. 524-534 (cf. p. 19).

Larochelle, Hugo et Yoshua Bengio (2008). « Classification using discriminative restricted Boltzmann machines ». In : *Proceedings of the 25th international conference on Machine learning*, p. 536-543 (cf. p. 57).

Larsen, Rasmus, Hildur Olafsdottir et Bjarne Kjær Ersbøll (2009). « Shape and texture based classification of fish species ». In : *Scandinavian Conference on Image Analysis*. Springer, p. 745-749 (cf. p. 30).

Le, Jiuyi et Lihong Xu (2017). « An automated fish counting algorithm in aquaculture based on image processing ». In : *2016 International Forum on Mechanical, Control and Automation (IFMCA 2016)*. Atlantis Press, p. 358-366 (cf. p. 174).

Le, Quoc V et al. (2011). « On optimization methods for deep learning ». In : *ICML* (cf. p. 61).

LeCun, Yann et al. (1989). « Backpropagation applied to handwritten zip code recognition ». In : *Neural computation* 1.4, p. 541-551 (cf. p. 55).

LeCun, Yann et al. (1998). « Gradient-based learning applied to document recognition ». In : *Proceedings of the IEEE* 86.11, p. 2278-2324 (cf. p. 26, 31, 45, 50, 62, 67, 68, 77).

LeCun, Yann A et al. (2012). « Efficient backprop ». In : *Neural networks : Tricks of the trade*. Springer, p. 9-48 (cf. p. 52).

Lee, Honglak, Chaitanya Ekanadham et Andrew Y Ng (2008). « Sparse deep belief net model for visual area V2 ». In : *Advances in neural information processing systems*, p. 873-880 (cf. p. 58, 61).

Lee, Honglak et al. (2009). « Convolutional deep belief networks for scalable unsupervised learning of hierarchical representations ». In : *Proceedings of the 26th annual international conference on machine learning*, p. 609-616 (cf. p. 58).

— (2011). « Unsupervised learning of hierarchical representations with convolutional deep belief networks ». In : *Communications of the ACM* 54.10, p. 95-103 (cf. p. 58).

Leggat, William P et al. (2019). « Rapid coral decay is associated with marine heatwave mortality events on reefs ». In : *Current Biology* 29.16, p. 2723-2730 (cf. p. 11).

Li, Xiaojing et al. (2018). « Real-time underwater fish tracking based on adaptive multi-appearance model ». In : *2018 25th IEEE international conference on image processing (ICIP)*. IEEE, p. 2710-2714 (cf. p. 174).

Li, Xiu, Youhua Tang et Tingwei Gao (2017). « Deep but lightweight neural networks for fish detection ». In : *OCEANS 2017-Aberdeen*. IEEE, p. 1-5 (cf. p. 27, 90).

Li, Xiu et al. (2015). « Fast accurate fish detection and recognition of underwater images with fast r-cnn ». In : *OCEANS 2015-MTS/IEEE Washington*. IEEE, p. 1-5 (cf. p. 26, 33, 39, 90, 114).

Li, Xiu et al. (2016). « Accelerating fish detection and recognition by sharing CNNs with objectness learning ». In : *OCEANS 2016-Shanghai*. IEEE, p. 1-5 (cf. p. 26, 27, 90).

Li, Yanjuan et al. (2020). « PredAmyl-MLP : Prediction of Amyloid Proteins Using Multilayer Perceptron ». In : *Computational and Mathematical Methods in Medicine* 2020 (cf. p. 49).

Li, Zhizhong et Derek Hoiem (2017). « Learning without forgetting ». In : *IEEE transactions on pattern analysis and machine intelligence* 40.12, p. 2935-2947 (cf. p. 156).

Lim, Kwangyong et al. (2017). « Real-time traffic sign recognition based on a general purpose GPU and deep-learning ». In : *PLoS one* 12.3, e0173317 (cf. p. 77).

Lin, Min, Qiang Chen et Shuicheng Yan (2013). « Network in network ». In : *arXiv preprint arXiv :1312.4400* (cf. p. 34).

Lin, Tsung-Yi et al. (2017). « Focal loss for dense object detection ». In : *Proceedings of the IEEE international conference on computer vision*, p. 2980-2988 (cf. p. 84).

Liu, Wei et al. (2016). « Ssd : Single shot multibox detector ». In : *European conference on computer vision*. Springer, p. 21-37 (cf. p. 27, 83, 84).

Long, Jonathan, Evan Shelhamer et Trevor Darrell (2015). « Fully convolutional networks for semantic segmentation ». In : *Proceedings of the IEEE conference on computer vision and pattern recognition*, p. 3431-3440 (cf. p. 82).

Loonis, Pierre, Michel MENARD, Christophe Demko et al. (1996). « A new genetic algorithm for the multi-classifiers fusion optimization ». In : *Proceedings of Information Processing and Management of Uncertainty in Knowledge Based Systems*. T. 2, p. 957-961 (cf. p. 30).

Lowry, Michael et al. (2012). « Comparison of baited remote underwater video (BRUV) and underwater visual census (UVC) for assessment of artificial reefs in estuaries ». In : *Journal of Experimental Marine Biology and Ecology* 416, p. 243-253 (cf. p. 19).

MacNeil, M Aaron et al. (2008a). « Accounting for detectability in reef-fish biodiversity estimates ». In : *Marine Ecology Progress Series* 367, p. 249-260 (cf. p. 14).

MacNeil, M Aaron et al. (2008b). « Detection heterogeneity in underwater visual-census data ». In : *Journal of Fish Biology* 73.7, p. 1748-1763 (cf. p. 14).

Mallet, Delphine et Dominique Pelletier (2014). « Underwater video techniques for observing coastal marine biodiversity : a review of sixty years of publications (1952–2012) ». In : *Fisheries Research* 154, p. 44-62 (cf. p. 10, 15, 17, 19, 20).

Mandal, Ranju et al. (2018). « Assessing fish abundance from underwater video using deep neural networks ». In : *2018 International Joint Conference on Neural Networks (IJCNN)*. IEEE, p. 1-6 (cf. p. 27, 39, 90).

Masana, Marc et al. (2020). « Class-incremental learning : survey and performance evaluation ». In : *arXiv preprint arXiv :2010.15277* (cf. p. 40, 140).

McCulloch, Warren S et Walter Pitts (1943). « A logical calculus of the ideas immanent in nervous activity ». In : *The bulletin of mathematical biophysics* 5.4, p. 115-133 (cf. p. 45-47).

Medina-Santiago, A et al. (2017). « Neural network backpropagation with applications into nutrition ». In : *International Conference on Innovation in Medicine and Healthcare*. Springer, p. 46-54 (cf. p. 46).

Melikoglu, Mehmet (2018). « Current status and future of ocean energy sources : A global review ». In : *Ocean Engineering* 148, p. 563-573 (cf. p. 1).

Miller, Daniel J (1972). « Guide to the coastal marine fishes of California ». In : *Fish Bull.* 157, p. 1-235 (cf. p. 22).

Mohamed, Abdel-rahman, George Dahl et Geoffrey Hinton (2009). « Deep belief networks for phone recognition ». In : *Nips workshop on deep learning for speech recognition and related applications*. T. 1. 9. Vancouver, Canada, p. 39 (cf. p. 58).

Monkam, Patrice et al. (2018). « Ensemble learning of multiple-view 3D-CNNs model for micro-nodules identification in CT images ». In : *IEEE Access* 7, p. 5564-5576 (cf. p. 94, 100).

Montavon, Grégoire et Klaus-Robert Müller (2012). « Deep Boltzmann machines and the centering trick ». In : *Neural Networks : Tricks of the Trade*. Springer, p. 621-637 (cf. p. 59).

Morvant, Emilie, Amaury Habrard et Stéphane Ayache (2014). « Majority vote of diverse classifiers for late fusion ». In : *Joint IAPR International Workshops on Statistical Techniques in Pattern Recognition (SPR) and Structural and Syntactic Pattern Recognition (SSPR)*. Springer, p. 153-162 (cf. p. 94).

Ngiam, Jiquan et al. (2011). « Learning deep energy models ». In : *Proceedings of the 28th international conference on machine learning (ICML-11)*, p. 1105-1112 (cf. p. 59).

Olafsdottir, Anna Hulda, Harald Ulrik Sverdrup et Kristin Vala Ragnarsdottir (2017). « On the metal contents of ocean floor nodules, crusts and massive sulphides and a preliminary assessment of the extractable amounts ». In : *World resources Forum*, p. 150-156 (cf. p. 1).

Oquab, Maxime et al. (2014). « Learning and transferring mid-level image representations using convolutional neural networks ». In : *Proceedings of the IEEE conference on computer vision and pattern recognition*, p. 1717-1724 (cf. p. 76).

Ou, Xianfeng et al. (2019). « Moving object detection method via ResNet-18 with encoder–decoder structure in complex scenes ». In : *IEEE Access* 7, p. 108152-108160 (cf. p. 71, 73).

Pan, Sinno Jialin et Qiang Yang (2009). « A survey on transfer learning ». In : *IEEE Transactions on knowledge and data engineering* 22.10, p. 1345-1359 (cf. p. 34, 40, 114, 116).

Parkhi, Omkar M, Andrea Vedaldi et Andrew Zisserman (2015). « Deep face recognition ». In : (cf. p. 77).

Pelletier, Dominique (1991). « Les sources d'incertitude en gestion des pêcheries. Evaluation et propagation dans les modèles ». Thèse de doct. Institut National Agronomique Paris-Grignon (cf. p. 12).

Pelletier, Dominique et al. (2011). « Comparison of visual census and high definition video transects for monitoring coral reef fish assemblages ». In : *Fisheries Research* 107.1-3, p. 84-93 (cf. p. 16).

Pelletier, Dominique et al. (2012). « Remote high-definition rotating video enables fast spatial survey of marine underwater macrofauna and habitats ». In : *Plos One* 7.2, e30536 (cf. p. 18, 19).

Peng, Xiaojiang et Cordelia Schmid (2016). « Multi-region two-stream R-CNN for action detection ». In : *European conference on computer vision*. Springer, p. 744-759 (cf. p. 39).

Picard, David (2015). « Making ecotourism sustainable : refocusing on economic viability. Lessons learnt from the "Regional strategic action plan for coastal ecotourism development in the South Western Indian Ocean" ». In : *Journal of Sustainable Tourism* 23.6, p. 819-837 (cf. p. 1).

Poiner, IR et al. (1998). « Final report on effects of trawling in the Far Northern Section of the Great Barrier Reef : 1991-1996 ». In : (cf. p. 11).

Polikar, Robi et al. (2001). « Learn++ : An incremental learning algorithm for supervised neural networks ». In : *IEEE transactions on systems, man, and cybernetics, part C (applications and reviews)* 31.4, p. 497-508 (cf. p. 155).

Poria, Soujanya et al. (2016). « Fusing audio, visual and textual clues for sentiment analysis from multimodal content ». In : *Neurocomputing* 174, p. 50-59 (cf. p. 96, 173).

Pramunendar, Ricardus Anggi et al. (2019). « A Robust Image Enhancement Techniques for Underwater Fish Classification in Marine Environment ». In : *International Journal of Intelligent Engineering and Systems* 12.5, p. 116-129 (cf. p. 173).

Priborsky, Josef et Josef Velisek (2018). « A review of three commonly used fish anesthetics ». In : *Reviews in Fisheries Science & Aquaculture* 26.4, p. 417-442 (cf. p. 12).

Qin, Hongwei et al. (2015). « When underwater imagery analysis meets deep learning : A solution at the age of big visual data ». In : *OCEANS 2015-MTS/IEEE Washington*. IEEE, p. 1-5 (cf. p. 33, 114, 135, 137).

Qin, Hongwei et al. (2016). « DeepFish : Accurate underwater live fish recognition with a deep architecture ». In : *Neurocomputing* 187, p. 49-58 (cf. p. 24, 33, 34, 40, 114, 115, 118, 127, 136, 137).

Ramírez-Quintana, Juan A, Mario I Chacon-Murguia et Jose F Chacon-Hinojos (2012). « Artificial Neural Image Processing Applications : A Survey. » In : *Engineering Letters* 20.1 (cf. p. 45).

Ranzato, Marc'Aurelio et al. (2007). « Efficient learning of sparse representations with an energy-based model ». In : *Advances in neural information processing systems*, p. 1137-1144 (cf. p. 61).

Redmon, Joseph et al. (2016). « You only look once : Unified, real-time object detection ». In : *Proceedings of the IEEE conference on computer vision and pattern recognition*, p. 779-788 (cf. p. 28, 80, 82, 83).

Ren, Shaoqing et al. (2015). « Faster r-cnn : Towards real-time object detection with region proposal networks ». In : *Advances in neural information processing systems*, p. 91-99 (cf. p. 26, 27, 80, 81, 91, 108).

Rende, SF et al. (2015). « Advances in micro-cartography : A two-dimensional photo mosaicing technique for seagrass monitoring ». In : *Estuarine, Coastal and Shelf Science* 167, p. 475-486 (cf. p. 16, 17).

Rifai, Salah et al. (2011). « Contractive auto-encoders : Explicit invariance during feature extraction ». In : *Icml* (cf. p. 61).

Roberts, Callum M (1995). « Effects of fishing on the ecosystem structure of coral reefs ». In : *Conservation biology* 9.5, p. 988-995 (cf. p. 2).

Robertson, D Ross et William F Smith-Vaniz (2008). « Rotenone : an essential but demonized tool for assessing marine fish diversity ». In : *Bioscience* 58.2, p. 165-170 (cf. p. 12).

Robinson, James PW et al. (2017). « Fishing degrades size structure of coral reef fish communities ». In : *Global Change Biology* 23.3, p. 1009-1022 (cf. p. 2, 11).

Rogers, Alice, Julia L Blanchard et Peter J Mumby (2018). « Fisheries productivity under progressive coral reef degradation ». In : *Journal of applied ecology* 55.3, p. 1041-1049 (cf. p. 10).

Rogers, Caroline S et Jeff Miller (2001). « Coral bleaching, hurricane damage, and benthic cover on coral reefs in St. John, US Virgin Islands : a comparison of surveys with the chain transect method and videography ». In : *Bulletin of Marine Science* 69.2, p. 459-470 (cf. p. 15).

Rosenblatt, Frank (1958). « The perceptron : a probabilistic model for information storage and organization in the brain. » In : *Psychological review* 65.6, p. 386 (cf. p. 48).

ROTHERHAM, D et al. (2007). « A strategy for developing scientific sampling tools for fishery-independent surveys of estuarine fish in New South Wales, Australia ». In : *ICES Journal of Marine Science* 64.8, p. 1512-1516 (cf. p. 19).

RUDER, Sebastian (2016). « An overview of gradient descent optimization algorithms ». In : *arXiv preprint arXiv :1609.04747* (cf. p. 52).

RUMELHART, David E, Geoffrey E HINTON et Ronald J WILLIAMS (1986). « Learning representations by back-propagating errors ». In : *nature* 323.6088, p. 533-536 (cf. p. 48, 50).

RUMELHART, David E et al. (1986). « Sequential thought processes in PDP models ». In : *Parallel distributed processing : explorations in the microstructures of cognition* 2, p. 3-57 (cf. p. 56).

RUSSAKOVSKY, Olga et al. (2015). « Imagenet large scale visual recognition challenge ». In : *International journal of computer vision* 115.3, p. 211-252 (cf. p. 55).

RUSU, Andrei A et al. (2016). « Progressive neural networks ». In : *arXiv preprint arXiv :1606.04671* (cf. p. 156).

SAINATH, Tara N et al. (2011). « Making deep belief networks effective for large vocabulary continuous speech recognition ». In : *2011 IEEE Workshop on Automatic Speech Recognition & Understanding*. IEEE, p. 30-35 (cf. p. 58).

SAK, Hasim, Andrew W SENIOR et Françoise BEAUFAYS (2014). « Long short-term memory recurrent neural network architectures for large scale acoustic modeling ». In : (cf. p. 49).

SAK, Haşim et al. (2015). « Fast and accurate recurrent neural network acoustic models for speech recognition ». In : *arXiv preprint arXiv :1507.06947* (cf. p. 49).

SALAKHUTDINOV, Ruslan et Geoffrey HINTON (2009). « Deep boltzmann machines ». In : *Artificial intelligence and statistics*, p. 448-455 (cf. p. 59).

SALAKHUTDINOV, Ruslan, Andriy MNIH et Geoffrey HINTON (2007). « Restricted Boltzmann machines for collaborative filtering ». In : *Proceedings of the 24th international conference on Machine learning*, p. 791-798 (cf. p. 57).

SALE, PF et BJ SHARP (1983). « Correction for bias in visual transect censuses of coral reef fishes ». In : *Coral reefs* 2.1, p. 37-42 (cf. p. 22).

SALI, Rasoul et al. (2020). « Hierarchical Deep Convolutional Neural Networks for Multi-category Diagnosis of Gastrointestinal Disorders on Histopathological Images ». In : *arXiv preprint arXiv :2005.03868* (cf. p. 40, 140).

Salman, Ahmad et al. (2016). « Fish species classification in unconstrained underwater environments based on deep learning ». In : *Limnology and Oceanography : Methods* 14.9, p. 570-585 (cf. p. 24, 33, 37, 114, 127).

Salman, Ahmad et al. (2019). « Real-time fish detection in complex backgrounds using probabilistic background modelling ». In : *Ecological Informatics* 51, p. 44-51 (cf. p. 25).

Salman, Ahmad et al. (2020). « Automatic fish detection in underwater videos by a deep neural network-based hybrid motion learning system ». In : *ICES Journal of Marine Science* 77.4, p. 1295-1307 (cf. p. 24, 28, 39, 91, 96, 97, 108, 109).

Sarikaya, Ruhi, Geoffrey E Hinton et Anoop Deoras (2014). « Application of deep belief networks for natural language understanding ». In : *IEEE/ACM Transactions on Audio, Speech, and Language Processing* 22.4, p. 778-784 (cf. p. 58).

Sarle, Warren S (1996). « Stopped training and other remedies for overfitting ». In : *Computing science and statistics*, p. 352-360 (cf. p. 54).

Schaner, Ted, Michael G Fox et Ana Carolina Taraborelli (2009). « An inexpensive system for underwater video surveys of demersal fishes ». In : *Journal of Great Lakes Research* 35.2, p. 317-319 (cf. p. 17).

Schaul, Tom, Sixin Zhang et Yann LeCun (2013). « No more pesky learning rates ». In : *International Conference on Machine Learning*, p. 343-351 (cf. p. 52).

Scherer, Dominik, Andreas Müller et Sven Behnke (2010). « Evaluation of pooling operations in convolutional architectures for object recognition ». In : *International conference on artificial neural networks*. Springer, p. 92-101 (cf. p. 65).

Scott, Steven D (2011). « Marine minerals : their occurrences, exploration and exploitation ». In : *OCEANS'11 MTS/IEEE KONA*. IEEE, p. 1-8 (cf. p. 1).

Sermanet, Pierre et al. (2013). « Overfeat : Integrated recognition, localization and detection using convolutional networks ». In : *arXiv preprint arXiv :1312.6229* (cf. p. 82).

Shevchenko, Violetta, Tuomas Eerola et Arto Kaarna (2018). « Fish detection from low visibility underwater videos ». In : *2018 24th International Conference on Pattern Recognition (ICPR)*. IEEE, p. 1971-1976 (cf. p. 25).

Shi, Cuncun, Caiyan Jia et Zhineng Chen (2018). « FFDet : a fully convolutional network for coral reef fish detection by layer fusion ». In : *2018 IEEE Visual Communications and Image Processing (VCIP)*. IEEE, p. 1-4 (cf. p. 27, 39, 90).

Shmelkov, Konstantin, Cordelia Schmid et Karteek Alahari (2017). « Incremental learning of object detectors without catastrophic forgetting ». In : *Proceedings of the IEEE International Conference on Computer Vision*, p. 3400-3409 (cf. p. 155).

Shorten, Connor et Taghi M Khoshgoftaar (2019). « A survey on image data augmentation for deep learning ». In : *Journal of Big Data* 6.1, p. 60 (cf. p. 75).

Simonyan, Karen et Andrew Zisserman (2014a). « Two-stream convolutional networks for action recognition in videos ». In : *Advances in neural information processing systems*, p. 568-576 (cf. p. 77, 98).

— (2014b). « Very deep convolutional networks for large-scale image recognition ». In : *arXiv preprint arXiv :1409.1556* (cf. p. 27, 55, 69, 76, 119).

Spampinato, Concetto et al. (2008). « Detecting, Tracking and Counting Fish in Low Quality Unconstrained Underwater Videos. » In : *VISAPP (2)* 2008.514-519, p. 1 (cf. p. 25, 90).

Spampinato, Concetto et al. (2010). « Automatic fish classification for underwater species behavior understanding ». In : *Proceedings of the first ACM international workshop on Analysis and retrieval of tracked events and motion in imagery streams*, p. 45-50 (cf. p. 31, 114, 174).

Srivastava, Nitish et Russ R Salakhutdinov (2012). « Multimodal learning with deep boltzmann machines ». In : *Advances in neural information processing systems*, p. 2222-2230 (cf. p. 59).

Srivastava, Nitish et al. (2014). « Dropout : a simple way to prevent neural networks from overfitting ». In : *The journal of machine learning research* 15.1, p. 1929-1958 (cf. p. 74).

Stobart, Ben et al. (2007). « A baited underwater video technique to assess shallow-water Mediterranean fish assemblages : Methodological evaluation ». In : *Journal of Experimental Marine Biology and Ecology* 345.2, p. 158-174 (cf. p. 18).

Strachan, NJC (1993). « Recognition of fish species by colour and shape ». In : *Image and vision computing* 11.1, p. 2-10 (cf. p. 29).

Strachan, NJC et L Kell (1995). « A potential method for the differentiation between haddock fish stocks by computer vision using canonical discriminant analysis ». In : *ICES Journal of Marine Science* 52.1, p. 145-149 (cf. p. 29).

Strachan, Norval James Colin, Paul Nesvadba et Alastair R Allen (1990). « Fish species recognition by shape analysis of images ». In : *Pattern Recognition* 23.5, p. 539-544 (cf. p. 29).

Strauss, RICHARD E et CARL E Bond (1990). « Taxonomic methods : morphology ». In : *Methods for fish biology*, p. 109-140 (cf. p. 22).

Sun, Xin et al. (2016). « Fish recognition from low-resolution underwater images ». In : *2016 9th International Congress on Image and Signal Processing, BioMedical Engineering and Informatics (CISP-BMEI)*. IEEE, p. 471-476 (cf. p. 33, 114, 137).

Sun, Xin et al. (2018). « Transferring deep knowledge for object recognition in Low-quality underwater videos ». In : *Neurocomputing* 275, p. 897-908 (cf. p. 24, 34, 35, 39, 40, 114, 115, 118, 136, 137).

Sun, Yi et al. (2014). « Deep learning face representation by joint identification-verification ». In : *Advances in neural information processing systems*, p. 1988-1996 (cf. p. 77).

Sun, Yi et al. (2015). « Deepid3 : Face recognition with very deep neural networks ». In : *arXiv preprint arXiv :1502.00873* (cf. p. 77).

Sung, Minsung, Son-Cheol Yu et Yogesh Girdhar (2017). « Vision based real-time fish detection using convolutional neural network ». In : *OCEANS 2017-Aberdeen*. IEEE, p. 1-6 (cf. p. 28, 90).

Sutskever, Ilya, Oriol Vinyals et Quoc V Le (2014). « Sequence to sequence learning with neural networks ». In : *Advances in neural information processing systems*, p. 3104-3112 (cf. p. 26).

Sward, Darryn, Jacquomo Monk et Neville Barrett (2019). « A systematic review of remotely operated vehicle surveys for visually assessing fish assemblages ». In : *Frontiers in Marine Science* 6, p. 134 (cf. p. 17).

Szegedy, Christian et al. (2015). « Going deeper with convolutions ». In : *Proceedings of the IEEE conference on computer vision and pattern recognition*, p. 1-9 (cf. p. 33, 55, 70-72, 76, 119).

Szegedy, Christian et al. (2016a). « Inception-v4, inception-resnet and the impact of residual connections on learning ». In : *arXiv preprint arXiv :1602.07261* (cf. p. 70).

Szegedy, Christian et al. (2016b). « Rethinking the inception architecture for computer vision ». In : *Proceedings of the IEEE conference on computer vision and pattern recognition*, p. 2818-2826 (cf. p. 70).

Szűcs, Gábor, Dávid Papp et Dániel Lovas (2015). « SVM classification of moving objects tracked by Kalman filter and Hungarian method ». In : *Working Notes of CLEF 2015 Conference, Toulouse, France* (cf. p. 32, 114, 137).

Teh, Yee Whye et Geoffrey E Hinton (2001). « Rate-coded restricted Boltzmann machines for face recognition ». In : *Advances in neural information processing systems*, p. 908-914 (cf. p. 57).

Thresher, Ronald E et John S Gunn (1986). « Comparative analysis of visual census techniques for highly mobile, reef-associated piscivores (Carangidae) ». In : *Environmental Biology of Fishes* 17.2, p. 93-116 (cf. p. 13).

Thys, Tierney et al. (2016). « Tracking a marine ecotourism star : movements of the short ocean sunfish Mola ramsayi in Nusa Penida, Bali, Indonesia ». In : *Journal of Marine Biology* 2016 (cf. p. 1).

Tran, Truyen, Dinh Phung et Svetha Venkatesh (2011). « Mixed-variate restricted Boltzmann machines ». In : *Asian conference on machine learning*, p. 213-229 (cf. p. 57).

Trenkel, Verena M et John Cotter (2009). « Choosing survey time series for populations as part of an ecosystem approach to fishery management ». In : *Aquatic Living Resources* 22.2, p. 121-126 (cf. p. 12).

Tsai, Yi-Hsuan, Ming-Hsuan Yang et Michael J Black (2016). « Video segmentation via object flow ». In : *Proceedings of the IEEE conference on computer vision and pattern recognition*, p. 3899-3908 (cf. p. 98).

Tu, Zhigang et al. (2019). « A survey of variational and CNN-based optical flow techniques ». In : *Signal Processing : Image Communication* 72, p. 9-24 (cf. p. 98).

Tyne, Julian A et al. (2010). « An integrated data management and video system for sampling aquatic benthos ». In : *Marine and Freshwater Research* 61.9, p. 1023-1028 (cf. p. 18).

Uijlings, Jasper RR et al. (2013). « Selective search for object recognition ». In : *International journal of computer vision* 104.2, p. 154-171 (cf. p. 28, 78).

Underwood, Mark et al. (2018). « A portable shallow-water optic fiber towed camera system for coastal benthic assessment ». In : *OCEANS 2018 MTS/IEEE Charleston*. IEEE, p. 1-7 (cf. p. 17).

Villon, Sébastien et al. (2016). « Coral reef fish detection and recognition in underwater videos by supervised machine learning : Comparison between Deep Learning and HOG+ SVM methods ». In : *International Conference on Advanced Concepts for Intelligent Vision Systems*. Springer, p. 160-171 (cf. p. 33, 114).

Vincent, Pascal et al. (2008). « Extracting and composing robust features with denoising autoencoders ». In : *Proceedings of the 25th international conference on Machine learning*, p. 1096-1103 (cf. p. 61).

Vincent, Pascal et al. (2010). « Stacked denoising autoencoders : Learning useful representations in a deep network with a local denoising criterion. » In : *Journal of machine learning research* 11.12 (cf. p. 61).

Virdin, J et al. (2021). « The Ocean 100 : Transnational corporations in the ocean economy ». In : *Science advances* 7.3, eabc8041 (cf. p. 1).

Vogt, H, ARF Montebon et MLR Alcala (1997). « Underwater video sampling : an effective method for coral reef surveys ». In : *Proc. 8th Int. Coral Reef Symp.* T. 2, p. 1447-1452 (cf. p. 16).

Wan, Li et al. (2013). « Regularization of neural networks using dropconnect ». In : *International conference on machine learning*, p. 1058-1066 (cf. p. 74).

Wang, Yifan et al. (2016). « Two-Stream SR-CNNs for Action Recognition in Videos. » In : *BMVC* (cf. p. 94).

Wantiez, Laurent, Olivier Chateau et Soazig Le Mouellic (2006). « Initial and mid-term impacts of cyclone Erica on coral reef fish communities and habitat in the South Lagoon Marine Park of New Caledonia ». In : *Marine Biological Association of the United Kingdom. Journal of the Marine Biological Association of the United Kingdom* 86.5, p. 1229 (cf. p. 14).

Watson, R, C Revenga et Y Kura (2006). « Fishing gear associated with global marine catches : II. Trends in trawling and dredging ». In : *Fisheries Research* 79.1-2, p. 103-111 (cf. p. 11).

White, Darren J, C Svellingen et Norval JC Strachan (2006). « Automated measurement of species and length of fish by computer vision ». In : *Fisheries Research* 80.2-3, p. 203-210 (cf. p. 29).

Wilberforce, Tabbi et al. (2019). « Overview of ocean power technology ». In : *Energy* 175, p. 165-181 (cf. p. 1).

Wilkinson, Clive et al. (2004). « Status of coral reefs of the world : 2004 : summary ». In : (cf. p. 10).

Willis, Trevor J (2001). « Visual census methods underestimate density and diversity of cryptic reef fishes ». In : *Journal of Fish Biology* 59.5, p. 1408-1411 (cf. p. 14).

Willis, Trevor J et Russell C Babcock (2000). « A baited underwater video system for the determination of relative density of carnivorous reef fish ». In : *Marine and Freshwater research* 51.8, p. 755-763 (cf. p. 18).

Wöllmer, Martin et al. (2013). « Youtube movie reviews : Sentiment analysis in an audio-visual context ». In : *IEEE Intelligent Systems* 28.3, p. 46-53 (cf. p. 96).

Wraith, James A (2007). « Assessing reef fish assemblages in a temperate marine park using baited remote underwater video ». In : (cf. p. 18).

Xiao, Fanyi et Yong Jae Lee (2016). « Track and segment : An iterative unsupervised approach for video object proposals ». In : *Proceedings of the IEEE conference on computer vision and pattern recognition*, p. 933-942 (cf. p. 98).

Xiao, Tianjun et al. (2014). « Error-driven incremental learning in deep convolutional neural network for large-scale image classification ». In : *Proceedings of the 22nd ACM international conference on Multimedia*, p. 177-186 (cf. p. 155).

Xu, Dan et al. (2017). « Detecting anomalous events in videos by learning deep representations of appearance and motion ». In : *Computer Vision and Image Understanding* 156, p. 117-127 (cf. p. 98).

Yan, Zhicheng et al. (2015). « HD-CNN : hierarchical deep convolutional neural networks for large scale visual recognition ». In : *Proceedings of the IEEE international conference on computer vision*, p. 2740-2748 (cf. p. 143).

Yang, Ling et al. (2020). « Computer Vision Models in Intelligent Aquaculture with Emphasis on Fish Detection and Behavior Analysis : A Review ». In : *Archives of Computational Methods in Engineering*, p. 1-32 (cf. p. 10).

Yuan, Ya-xiang (1991). « A modified BFGS algorithm for unconstrained optimization ». In : *IMA Journal of Numerical Analysis* 11.3, p. 325-332 (cf. p. 50).

Zach, Christopher, Thomas Pock et Horst Bischof (2007). « A duality based approach for realtime tv-l 1 optical flow ». In : *Joint pattern recognition symposium*. Springer, p. 214-223 (cf. p. 98).

Zeiler, Matthew D (2012). « Adadelta : an adaptive learning rate method ». In : *arXiv preprint arXiv :1212.5701* (cf. p. 52).

— (2013). « Hierarchical convolutional deep learning in computer vision ». Thèse de doct. New York University (cf. p. 64).

Zeiler, Matthew D et Rob Fergus (2014). « Visualizing and understanding convolutional networks ». In : *European conference on computer vision*. Springer, p. 818-833 (cf. p. 27).

Zha, Shengxin et al. (2015). « Exploiting image-trained CNN architectures for unconstrained video classification ». In : *arXiv preprint arXiv :1503.04144* (cf. p. 100).

Zhang, David et al. (2016). « Unsupervised underwater fish detection fusing flow and objectiveness ». In : *2016 IEEE Winter Applications of Computer Vision Workshops (WACVW)*. IEEE, p. 1-7 (cf. p. 28, 91).

Zhang, Jie et al. (2014). « Coarse-to-fine auto-encoder networks (cfan) for real-time face alignment ». In : *European conference on computer vision*. Springer, p. 1-16 (cf. p. 60).

Zhang, Song et al. (2020). « Automatic fish population counting by machine vision and a hybrid deep neural network model ». In : *Animals* 10.2, p. 364 (cf. p. 174).

Zhou, Yingbo et al. (2014). « Is joint training better for deep auto-encoders ? » In : *arXiv preprint arXiv :1405.1380* (cf. p. 60).

Zhu, Xinqi et Michael Bain (2017). « B-CNN : branch convolutional neural network for hierarchical classification ». In : *arXiv preprint arXiv :1709.09890* (cf. p. 141).

Zhu, Xunmu et al. (2020). « Automatic recognition of lactating sow postures by refined two-stream RGB-D faster R-CNN ». In : *Biosystems Engineering* 189, p. 116-132 (cf. p. 93, 94, 108, 111).

Zhuang, Peiqin et al. (2017). « Marine Animal Detection and Recognition with Advanced Deep Learning Models. » In : *CLEF (Working Notes)* (cf. p. 27, 39, 90).

Zion, B, A Shklyar et I Karplus (1999). « Sorting fish by computer vision ». In : *Computers and electronics in agriculture* 23.3, p. 175-187 (cf. p. 30).

Zong, Xianhui, Zhehan Chen et Dadong Wang (2020). « Local-CycleGAN : a general end-to-end network for visual enhancement in complex deep-water environment ». In : *Applied Intelligence*, p. 1-12 (cf. p. 173).

Zou, Will Y, Andrew Y Ng et Kai Yu (2011). « Unsupervised learning of visual invariance with temporal coherence ». In : *NIPS 2011 workshop on deep learning and unsupervised feature learning*. T. 3 (cf. p. 61).

Zur, Richard M et al. (2009). « Noise injection for training artificial neural networks : A comparison with weight decay and early stopping ». In : *Medical physics* 36.10, p. 4810-4818 (cf. p. 54).

Résumé

L'objectif de cette thèse est d'élaborer des méthodes permettant la reconnaissance automatique d'espèces de poissons dans des images vidéo sous-marines. Nous privilégions les approches modernes de l'apprentissage profond (deep learning), notamment les réseaux de neurones convolutifs (CNNs).

Nous proposons une approche robuste pour la détection de poissons dans des vidéos sous-marines. Cette approche consiste à combiner deux réseaux parallèles afin de fusionner les caractéristiques liées à l'apparence et au mouvement du poisson. Ensuite, nous développons des méthodes d'identification d'espèces de poissons basées sur l'apprentissage par transfert. Finalement, la classification d'espèces de poissons est posée dans un cadre de classification par apprentissage progressif, et ce, de deux manières différentes. D'une part, nous proposons une approche de classification hiérarchique basée sur la taxonomie des espèces, qui permet de classifier les poissons en famille puis en espèce. D'autre part, nous proposons un nouveau modèle basé sur le principe de l'apprentissage incrémental pour améliorer les performances sur les classes (espèces) difficiles à identifier. Au début le modèle se focalise à bien apprendre les espèces difficiles, puis apprend progressivement les autres espèces avec une bonne stabilité.

Mots-clefs : vidéo sous-marine, détection de poisson, classification d'espèce de poisson, apprentissage profond, réseaux de neurones convolutifs.

Abstract

The objective of this thesis is to develop tools and methods for automatic recognition of fish species in underwater video images. We focus on modern deep learning approaches, in particular convolutional neural networks (CNNs).

We propose a robust approach for the detection of fish in underwater video images. This approach consists in combining two parallel networks in order to fuse the features related to the appearance and the movement of the fish. Next, we develop methods for fish species identification based on transfer learning. Finally, fish species classification is posed in a progressive learning classification framework in two different ways. On the one hand, we propose a hierarchical classification approach based on species taxonomy, which allows to classify fishes into families and then into species. On the other hand, we propose a new model based on the principle of incremental learning to improve the performance on the classes (species) difficult to identify. At the beginning, the model focuses on learning the difficult species well, and then gradually learns the other species with a good stability.

Key Words : underwater video, fish detection, fish species classification, deep learning, convolutional neural networks.