

RÉPUBLIQUE DU CAMEROUN

Paix - Travail - Patrie

UNIVERSITÉ DE YAOUNDÉ I

FACULTÉ DES SCIENCES

DÉPARTEMENT D'INFORMATIQUE



REPUBLIC OF CAMEROON

Peace - Work - Fatherland

UNIVERSITY OF YAOUNDE I

FACULTY OF SCIENCE

DEPARTMENT OF COMPUTER SCIENCE

MODÈLE EXPLICABLE BASÉ SUR LES AUTOENCODEURS POUR LA DÉTECTION DES CLASSES NOUVELLES

En vue de l'obtention du diplôme de Master Recherche en Informatique

Option

Science des données

Présenté par

KEUFACK DONGMO BELVIANE BERENICE

Matricule

19Y204

Supervisé par

Pr. TSOPZE NORBERT



Année académique 2023/ 2024



Université de Yaoundé 1
Département d'Informatique

MODÈLE EXPLICABLE BASÉ SUR LES AUTOENCODEURS POUR LA DÉTECTION DES CLASSES NOUVELLES

Présenté par

KEUFACK DONGMO BELVIANE BERENICE

Matricule

19Y204

Supervisé par

Pr. TSOPZE NORBERT

Année académique 2023/ 2024

DÉDICACES

Je dédie ce mémoire à mes parents.

REMERCIEMENTS

Je tiens à exprimer ma profonde gratitude envers le Divin Créateur pour la vie, la santé et l'opportunité de mener à bien ce travail. Mes sincères remerciements vont à toutes les personnes qui ont contribué, directement ou indirectement, à la réalisation de ce mémoire, notamment :

- **Pr TSOPZE NORBERT**, pour avoir accepté de m'encadrer et pour son suivi attentif. Travailler à vos côtés m'a beaucoup appris, tant sur le plan académique que personnel, en matière de rigueur et de ponctualité.
- **Les membres du jury**, pour leur disponibilité et l'honneur qu'ils m'accordent en acceptant d'évaluer ce travail.
- **L'Université de Yaoundé 1**, en particulier le Département d'Informatique, pour la qualité de l'enseignement dispensé tout au long de mon cursus académique.
- **Mes parents, KEUFACK Olivier et TEMGOUA NIZEKO Pauline Solange**, pour leur amour inconditionnel, leur confiance et leur soutien qui m'ont portés bien plus haut que je n'aurais pu l'imaginer. Vous m'avez appris à tracer ma route et à croire en mes rêves, une leçon inestimable que je n'aurais pu apprendre seul.
- **Mes frères et sœurs**, pour leur accompagnement constant, de près ou de loin.
- **La grande famille KENFACK**, pour le soutien moral qu'elle m'a apporté depuis mon arrivée à Yaoundé, contribuant ainsi à ma réussite académique.
- **Mes grand-parents : NIZEKO FABIEN, DONGMO Pauline et KENFACK Madeleine**, pour leur aide précieuse, leur soutien indéfectible et leur patience lors de mes retours tardifs des cours.
- **KEUFACK LIDWINE**, pour ses relectures attentives et ses encouragements durant la rédaction de ce mémoire.
- **Mes aînés académiques et camarades de promotion** : Farida Nchare, Germes Obiang, Bill Tanekeu, Nakam Manuella, Victor Djiembou et Landry Fordjou, pour leur soutien inestimable.

Enfin, j'adresse ma sincère reconnaissance à tous ceux et celles qui ont contribué, de près ou de loin, à la réalisation de ce travail et qui ont fait de cette expérience une réussite.

RÉSUMÉ

Ce mémoire présente une méthode de classification d'éléments dans des environnements dynamiques, basée sur une architecture modulaire utilisant des autoencodeurs. Le modèle proposé combine un modèle généraliste et des sous-modèles spécialisés pour chaque classe, permettant une adaptation continue aux nouvelles classes. Les expérimentations ont été réalisées sur des images d'espèces biologiques, plus précisément les espèces d'invertébrés marines et des oiseaux. Les résultats obtenus révèlent une Précision Moyenne Globale de 99,18 %, un Oubli Moyen de 1,75 % et une Plasticité de 95,27 % basés sur l'erreur de reconstruction MSE ; une Précision Moyenne Globale de 93,33 %, un Oubli Moyen de 7,73 % et une Plasticité de 79,22 % basés sur l'erreur de reconstruction RMSE ; une Précision Moyenne Globale de 66,86 %, un Oubli Moyen de 49,77 % et une Plasticité de 6,8 % basés sur l'erreur relative. Pour le jeu de données d'oiseaux, les résultats montrent une Précision Moyenne Globale de 99,71 %, un Oubli Moyen de 0,08 % et une Plasticité de 99,63 % basés sur l'erreur de reconstruction MSE ; une Précision Moyenne Globale de 94,81 %, un Oubli Moyen de 0,83 % et une Plasticité de 94,19 % basés sur l'erreur de reconstruction RMSE ; et une Précision Moyenne Globale de 79,01 %, un Oubli Moyen de 5,8 % et une Plasticité de 76,58 % basés sur l'erreur de reconstruction relative après l'intégration de 10 classes. L'étude met en lumière les défis et les opportunités de l'apprentissage continu, ouvrant des perspectives pour la gestion adaptative des écosystèmes marins et d'autres applications nécessitant une classification évolutive dans des environnements dynamiques.

Mots-clés : Classification multiclasse, Apprentissage continu, autoencodeurs, environnements dynamiques, architecture modulaire.

ABSTRACT

This thesis presents a method for classifying elements in dynamic environments, based on a modular architecture using autoencoders. The proposed model combines a generalist model and specialized sub-models for each class, allowing continuous adaptation to new classes. The experiments were carried out on images of biological species, specifically marine invertebrates and birds. The results obtained show an overall average precision of 99.18%, an average forgetting of 1.75% and a plasticity of 95.27% based on the MSE reconstruction error ; an overall average precision of 93.33%, an average forgetting of 7.73% and a plasticity of 79.22% based on the RMSE reconstruction error ; an overall average precision of 66.86%, an average forgetting of 49.77% and a plasticity of 6.8% based on the relative error. For the bird dataset, the results show an overall average precision of 99.71%, an average forgetting of 0.08%, and a plasticity of 99.63% based on the MSE reconstruction error ; an overall average precision of 94.81%, an average forgetting of 0.83% and a Plasticity of 94.19% based on the RMSE reconstruction error ; and an Overall Mean Accuracy of 79.01%, a Mean Oversight of 5.8% and a Plasticity of 76.58% based on the Relative Reconstruction Error after integration of 10 classes. The study highlights the challenges and opportunities of continuous learning and opens prospects for adaptive management of marine ecosystems and other applications requiring evolutionary classification in dynamic environments.

Keywords : Multiclass classification, Continual learning, autoencoders, dynamic environments, modular architecture.

TABLE DES MATIÈRES

Introduction Générale	1
1 Généralités et État de l'art	3
1.1 Introduction	3
1.2 Généralités	3
1.2.1 Environnements dynamiques	3
1.2.2 Apprentissage automatique	4
1.2.3 Autoencodeurs	4
1.2.4 Erreurs de reconstruction	5
1.3 Adaptation à l'environnement dynamique	7
1.3.1 Mécanisme d'apprentissage continu	7
1.3.2 Scénarios de l'apprentissage continu	7
1.3.3 Techniques de l'apprentissage continu	8
1.4 État de l'art	9
1.4.1 Modèles basés sur les autoencodeurs	9
1.4.2 Modèles basés sur d'autres méthodes	10
1.4.3 Modèles basés sur l'apprentissage continu pour la détection d'anomalies	10
1.5 Tableau récapitulatif de l'état de l'art	11
1.6 Bilan du chapitre	12
2 Méthode Proposée	13
2.1 Introduction	13
2.2 Méthode de conception du modèle de classification	13
2.2.1 Construction de l'architecture de l'autoencodeur de base	15
2.2.2 Spécialisation du modèle M pour obtenir les M_i	15
2.3 Intégration de nouvelles classes	16
2.3.1 Processus de détection des nouvelles classes	16
2.3.2 Mécanisme d'ajout de nouvelles classes	16
2.4 Bilan du chapitre	17
3 Expérimentations, résultats, discussions	18
3.1 Introduction	18
3.2 Présentation des données de l'expérimentation	18

3.3	Pré-traitement des données	18
3.4	Métriques d'évaluation	19
3.4.1	Précision Moyenne	19
3.4.2	Précision Moyenne Globale	19
3.4.3	Oubli Moyen	19
3.4.4	Plasticité	20
3.5	Protocole expérimental	20
3.5.1	Environnement de travail	20
3.5.2	Architectures des modèles	20
3.5.3	Division des données	21
3.5.4	Description des paramètres d'entraînement	21
3.6	Résultats et discussions	21
3.6.1	Performances des modèles généralistes	21
3.6.2	Performances des modèles spécialistes	22
3.6.3	Cas particuliers et variabilité des performances	25
3.6.4	Phase de continuité	28
3.7	Bilan du chapitre	33
4	Conclusion Générale	34
	Bibliographie	35

TABLE DES FIGURES

1.1	Structure générale d'un autoencodeur	4
2.1	Architecture du modèle	14
2.2	Modèle continu	17
3.1	Reconstructions des données par le modèle généraliste.	22
3.2	Reconstructions des données par le modèle généraliste.	22
3.3	Reconstructions des données par le modèle spécialiste sur sa classe cible pour les invertébrés marins.	23
3.4	Reconstructions des données par le modèle spécialiste sur une classe différente pour les invertébrés marins.	23
3.5	Reconstructions des données par le modèle spécialiste sur sa classe cible pour le jeu de données d'oiseaux.	24
3.6	Reconstructions des données par le modèle spécialiste sur une classe différente pour le jeu de données d'oiseaux.	24
3.7	Boîte à moustaches montrant un modèle spécialiste performant de manière variable en fonction du nombre d'images par classe.	25
3.8	Boîte à moustaches du modèle reconstruisant mieux les données d'autres classes. . .	26
3.9	Boîte à moustaches du modèle performant bien sur ses données et sur d'autres classes.	26
3.10	Boîte à moustaches montrant un modèle spécialiste performant de manière variable en fonction du nombre d'images par classe.	27
3.11	Boîte à moustaches du modèle reconstruisant mieux les données d'autres classes. . .	28
3.12	Boîte à moustaches du modèle performant bien sur ses données et sur d'autres classes.	28
3.13	Évolution des métriques d'apprentissage continu basée sur la MSE	29
3.14	Évolution des métriques d'apprentissage continu basée sur la RMSE	29
3.15	Évolution des métriques d'apprentissage continu basée sur l'erreur relative	30
3.16	Évolution des métriques d'apprentissage continu basée sur la MSE pour le jeu de données d'oiseaux	30
3.17	Évolution des métriques d'apprentissage continu basée sur la RMSE pour le jeu de données d'oiseaux	31
3.18	Évolution des métriques d'apprentissage continu basée sur l'erreur relative pour le jeu de données d'oiseaux	31
3.19	Image de la classe <i>Inachidae</i>	33
3.20	Image de la classe <i>Pyromaia tuberculata</i>	33

LISTE DES TABLEAUX

1	Liste des abréviations et leurs significations	ix
1.1	Comparaison des techniques d'apprentissage continu	8
1.2	Résumé des travaux	12
3.1	Comparaison des paramètres d'entraînement des modèles généraliste et spécialiste .	21

LISTE DES ABRÉVIATIONS

Abréviation	Signification
AA	Précision Moyenne (Average Accuracy)
AI	Intelligence Artificielle
AIA	Précision Moyenne Globale (Average Incremental Accuracy)
FM	Oubli Moyen (Forgetting Measure)
GAP	Pooling Moyen Global (Global Average Pooling)
GPU	Unité de Traitement Graphique (Graphics Processing Unit)
kNN	k-plus proches voisins (k-nearest neighbors)
MSE	Erreur Quadratique Moyenne (Mean Squared Error)
RMSE	Racine de l'Erreur Quadratique Moyenne (Root Mean Squared Error)
VAE	Autoencodeur Variationnel (Variational Autoencoder)

TABLE 1 – Liste des abréviations et leurs significations

INTRODUCTION GÉNÉRALE

Dans l'ère numérique actuelle, nous assistons à une explosion de données provenant de diverses sources telles que les capteurs, l'imagerie et les réseaux sociaux [1]. Cette abondance de données transforme profondément les secteurs scientifiques, industriels et économiques, générant des défis significatifs en termes d'analyse et d'interprétation [2]. Pour faire face à cette complexité croissante, les algorithmes d'apprentissage automatique sont devenus des outils indispensables, offrant des capacités d'analyse avancées pour extraire des informations pertinentes de ces vastes ensembles de données [1]. Des techniques d'apprentissage automatique comme les réseaux de neurones profonds, jouent un rôle crucial dans l'exploitation de ces données [3]. Cette exploitation est faite de diverses manières parmi lesquelles la classification. Elle consiste à regrouper les données par catégories. Un algorithme ou modèle de classification est entraîné sur ces données et pourra être déployé pour prédire des classes des nouvelles données. La classification est dite supervisée si les données contiennent un attribut particulier (étiquette ou classe) que l'exploitation va consister à construire un modèle pour prédire cet attribut, et non supervisée sinon. Les algorithmes classiques de classification supervisée ont généralement de bonnes performances lorsque toutes les classes sont connues à l'entraînement du modèle. Dans des environnements dynamiques, où la distribution des données peut changer après le déploiement de l'algorithme, ou encore où de nouvelles classes peuvent être détectées (par des experts du domaine) après l'entraînement du modèle, ces systèmes peuvent devenir obsolètes parce qu'ils n'ont pas été entraînés pour détecter les éléments de ces nouvelles classes. Cela peut se rencontrer des applications dans divers domaines comme la biologie où les experts continuent de découvrir de nouvelles espèces, en sécurité informatique où de nouvelles formes de menaces sont découvertes régulièrement.

Le problème traité dans ce mémoire est celui de la détection des nouvelles classes par un modèle de classification supervisée au cours son utilisation. Une nouvelle classe est une classe dans les observations ou exemples n'appartenaient pas aux données utilisées pour entraîner le modèle. Les observations de la nouvelle classe sont découvertes après le déploiement du modèle. Il est alors question de construire un modèle capable de détecter les exemples d'une nouvelle classe, et puis de se mettre à jour pour désormais tenir compte de cette nouvelle classe. Plus précisément, si le modèle a été entraîné pour classer en n catégories, après la détection de la nouvelle classe, il devra être capable de classer en $n + 1$ catégories. L'apprentissage continu (continual learning en Anglais) a été développé pour mettre à jour le modèle afin de l'adapter au changement de la distribution des données utilisées pour l'entraîner.

L'approche modulaire est l'une des approches du continual learning où le modèle est constitué

des modules (ou sous modèles) spécialisés pour une classe. Sa mise à jour est faite en ajoutant dans le modèle un module spécialisé pour reconnaître la nouvelle classe. L'approche modulaire présente comme avantage la facilité de mise à jour, mais comme les approches du continual learning peut souffrir de l'oubli catastrophique et les algorithmes de cette approche nécessitent beaucoup de ressources mémoires lorsque le nombre de sous modules s'augmente ; et d'une spécialisation excessive des sous modules [4]. Parmi les outils utilisés pour la construction des sous modules, les autoencodeurs ont permis d'obtenir des résultats dans des tâches assimilables à la classification binaire comme la détection d'anomalies, des détection des avalanches, pas sur les tâches de classification multiclasse (n-aire). Il sera donc question de construire un modèle de continual learning modulaire avec des sous modules basés sur des autoencodeurs pour la classification multiclasse.

L'approche utilisée propose une solution pour la classification dans des environnements dynamiques, où de nouvelles classes peuvent apparaître au fil du temps. Elle se décompose en deux phases principales. La première phase consiste en la construction d'un modèle de base capable d'apprendre les caractéristiques générales des données, puis sa spécialisation pour chacune des classes connues. Cette étape permet d'obtenir un ensemble de sous-modèles, chacun étant optimisé pour une classe spécifique. Le modèle global initial est constitué de l'ensemble de ces sous-modèles. La seconde phase consiste à mettre en place un mécanisme d'apprentissage continu qui permet au système de détecter automatiquement l'apparition de nouvelles classes, de créer et d'intégrer de nouveaux sous-modèles spécialisés pour ces classes. Dans la solution proposée, les sous modules sont constitués des autoencodeurs. Comparés à d'autres modèles de classification, les autoencodeurs possèdent un avantage distinctif : leur capacité à reconstruire avec une erreur faible les éléments de la classe sur laquelle ils ont été entraînés, tout en présentant des erreurs de reconstruction élevées pour les éléments inconnus. Chaque nouveau sous modèle est initialisé avec les poids du sous modèle de base, conservant ainsi les connaissances globales de la tâche . Contrairement aux modèles traditionnels qui peuvent avoir du mal à s'adapter aux nouvelles classes, les autoencodeurs offrent une méthode robuste pour détecter ou identifier de nouveaux éléments. Pour limiter l'oubli catastrophique, un mécanisme de transfert d'apprentissage est intégré : chaque nouveau modèle est initialisé avec les poids du modèle de base, conservant ainsi les connaissances globales de la tâche.

Le reste de ce mémoire est organisé comme suit : le Chapitre 1 présente une revue de la littérature et des généralités sur l'apprentissage automatique, la détection d'anomalies, et les autoencodeurs. Le Chapitre 2 décrit la méthodologie proposée, avec une explication détaillée de l'architecture du modèle et du processus d'intégration de nouvelles classes. Le Chapitre 3 discute des résultats des expérimentations, ainsi que de l'évaluation des performances du modèle. Enfin, le Chapitre 4 propose des perspectives futures pour l'amélioration et l'application du modèle dans d'autres domaines.

CHAPITRE 1

GÉNÉRALITÉS ET ÉTAT DE L'ART

1.1 Introduction

Ce chapitre a pour objectif de fournir une base théorique et un aperçu des travaux existants dans le domaine de la détection des classes nouvelles. Il se divise en deux parties principales : les généralités et l'état de l'art. Dans la première partie, nous présenterons les concepts fondamentaux nécessaires à la compréhension du sujet. Nous commencerons par une introduction aux environnements dynamiques en présentant leurs caractéristiques. Ensuite, nous introduirons les principes fondamentaux de l'apprentissage automatique, en nous concentrant sur les autoencodeurs et la détection de nouvelles classes. Nous réaliserons une revue de la littérature existante sur la détection des classes nouvelles et la gestion des environnements dynamiques, puis analyserons les approches actuelles et les innovations récentes dans le domaine, en mettant en avant les limites des méthodes existantes et les défis non résolus.

1.2 Généralités

1.2.1 Environnements dynamiques

Définition

Un environnement dynamique est un cadre ou un système où les conditions, les variables et les paramètres évoluent au fil du temps[5]. Les environnements dynamiques sont caractérisés par :

- **Variabilité temporelle** : Les conditions de l'environnement changent avec le temps, ce qui peut affecter les décisions prises par les modèles d'apprentissage.
- **Imprévisibilité** : Les changements peuvent être brusques, rendant difficile la prédiction des conditions futures.
- **Adaptabilité nécessaire** : Les systèmes opérant dans ces environnements doivent être capables de s'adapter rapidement aux nouvelles conditions pour maintenir leur performance.

Exemples d'environnements dynamiques

Comme exemples d'environnements dynamiques, nous pouvons citer :

- **Écosystèmes naturels** : La biodiversité évolue en raison du changement climatique, de l'intervention humaine, et d'autres facteurs.
- **Marchés financiers** : Les prix des actions changent constamment en fonction de nombreux facteurs comme les nouvelles économiques.

1.2.2 Apprentissage automatique

Selon Tom Mitchell, **l'apprentissage automatique (machine learning)** est une branche de l'intelligence artificielle qui permet aux systèmes d'apprendre et de s'améliorer à partir de l'expérience sans être explicitement programmés[3]. Cette discipline se divise en plusieurs approches :

- **L'apprentissage supervisé** : Le modèle apprend à partir d'exemples étiquetés, c'est-à-dire que chaque entrée de données est associée à une sortie correspondante, comme pour la classification d'images ou chaque image ou ensembles d'images est assimilées à une étiquette.
- **L'apprentissage non supervisé** : Le modèle cherche à trouver des structures ou des motifs dans des données non étiquetées, comme dans le clustering.
- **L'apprentissage par renforcement** : Le modèle apprend en interagissant avec un environnement et en recevant des récompenses ou des punitions basées sur ses actions, comme les agents autonomes dans les jeux.

Ces différentes approches sont implémentées à travers une variété de modèles d'apprentissage automatique, tels que les réseaux de neurones, les forêts aléatoires, et les machines à vecteurs de support.

1.2.3 Autoencodeurs

Définition et structure

Les autoencodeurs sont des modèles de réseaux de neurones conçus pour apprendre une représentation comprimée (ou codée) des données d'entrée[1]. Comme illustré par la Figure 1.1, ils sont composés de deux parties principales : un encodeur et un décodeur. L'encodeur prend les données d'entrée et les mappe dans un espace de dimension réduite, appelé espace latent, tandis que le décodeur reconstruit les données d'entrée à partir de cette représentation latente.

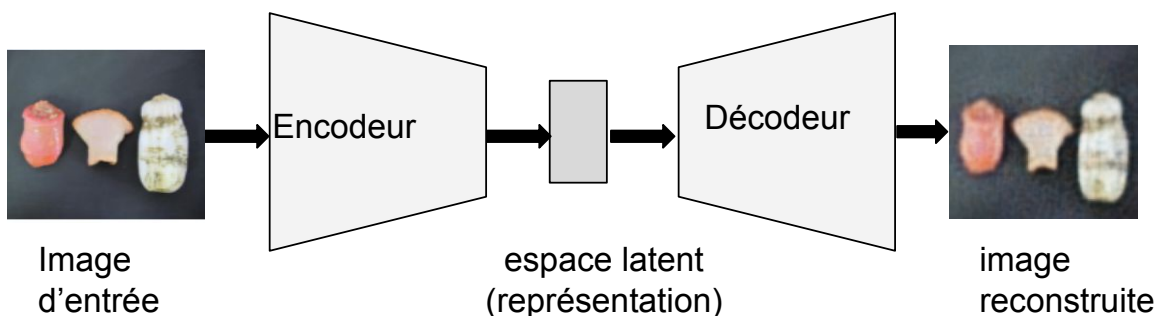


FIGURE 1.1 – Structure générale d'un autoencodeur

L'encodeur est une fonction f_θ paramétrée par θ , qui prend une entrée x et produit une représentation latente z :

$$z = f_\theta(x) \quad (1.1)$$

Où :

- x : Données d'entrée.
- z : Représentation latente.
- f_θ : Fonction de l'encodeur paramétrée par θ (poids et biais).

Le décodeur est une fonction g_ϕ paramétrée par ϕ , qui prend la représentation latente z et produit une reconstruction \hat{x} :

$$\hat{x} = g_\phi(z) \quad (1.2)$$

- \hat{x} : Reconstruction de x .
- g_ϕ : Fonction du décodeur paramétrée par ϕ (poids et biais).

L'objectif de l'entraînement d'un autoencodeur est d'ajuster les paramètres θ et ϕ de manière à minimiser l'écart entre x et \hat{x} , souvent mesuré par une fonction de perte telle que l'erreur quadratique moyenne (Mean Squared Error, MSE) .

1.2.4 Erreurs de reconstruction

Dans le cadre de l'évaluation des performances des autoencodeurs, plusieurs types d'erreurs de reconstruction sont utilisés, parmi lesquels : l'erreur quadratique moyenne (MSE), la racine carrée de l'erreur quadratique moyenne (RMSE) et l'erreur relative. Ces erreurs jouent un rôle essentiel dans la quantification de la qualité de la reconstruction des données par l'autoencodeur.

Erreur quadratique moyenne (MSE - Mean Squared Error)

Elle est calculée par la formule 1.3, elle évalue la moyenne des carrés des différences entre les attributs de l'entrée originale et ceux de l'entrée reconstruite par l'autoencodeur.

$$\text{MSE} = \frac{1}{n} \sum_{i=1}^n (y_i - \hat{y}_i)^2 \quad (1.3)$$

où y_i est la valeur réelle, \hat{y}_i est la valeur reconstruite, et n est le nombre total de données.

Avantages :

- **Simplicité** : La MSE est simple à comprendre et à implémenter.

Limites :

- **Sensibilité aux écarts** : Comme elle élève les différences au carré, elle est particulièrement sensible aux grandes erreurs, ce qui peut être utile pour pénaliser les reconstructions de mauvaise qualité.

Racine de l'erreur quadratique moyenne (RMSE - Root Mean Squared Error)

La RMSE, calculée par la formule 1.4, représente la racine carrée de la moyenne des carrés des différences entre les valeurs prédites et les valeurs réelles.

$$\text{RMSE} = \sqrt{\frac{1}{n} \sum_{i=1}^n (y_i - \hat{y}_i)^2} \quad (1.4)$$

Avantages :

- **Interprétation** : L'erreur RMSE est dans la même unité que les données d'origine, ce qui facilite son interprétation.

Limites :

- **Sensibilité aux écarts** : Comme la MSE, la RMSE est sensible aux grandes erreurs, mais elle est plus facilement interprétable.

Erreur relative

L'erreur relative définie par l'équation 1.5 mesure la moyenne des écarts absolus entre les valeurs réelles (y_i) et prédites (\hat{y}_i), normalisée par l'écart-type des valeurs réelles $\sigma(y_i)$.

$$\text{Erreur relative} = \frac{1}{n} \sum_{i=1}^n \left(\frac{|y_i - \hat{y}_i|}{\sigma(y_i) + \epsilon} \right) \quad (1.5)$$

où $\sigma(y_i)$ est l'écart-type de l'image, et ϵ est une petite constante pour éviter la division par zéro.

Avantages :

- **Comparaison relative** : Elle permet de comparer l'erreur par rapport à l'écart-type des données, ce qui est utile pour des données avec des échelles différentes.
- **Robustesse** : L'erreur relative peut être plus robuste face aux variations d'intensité entre les différentes dimensions ou canaux.

Limites :

- **Dépendance à l'écart-type** : La précision de l'erreur relative dépend fortement de la manière dont l'écart-type est calculé, et elle peut être moins intuitive à interpréter.
- **Complexité** : Le calcul de l'erreur relative est plus complexe que celui de la MSE ou de la RMSE, ce qui peut rendre l'analyse des résultats moins directe.

Fonctionnement

Le fonctionnement d'un autoencodeur, tel qu'illustré par l'algorithme 1, repose sur l'entraînement simultané de l'encodeur et du décodeur pour apprendre une représentation latente optimale des données d'entrée. L'entraînement se fait par descente de gradient, où les paramètres des réseaux sont ajustés pour minimiser l'erreur de reconstruction, qui est la différence entre les données d'entrée et leur reconstruction. Ce processus se répète jusqu'à ce que l'erreur soit minimisée, assurant que l'encodeur capture efficacement les caractéristiques essentielles des données [1].

Algorithme 1 Algorithme d'entraînement d'un autoencodeur [6]

Données : Jeu de données $x^{(1)}, \dots, x^{(N)}$

Résultat : Encodeur f_ϕ , décodeur g_θ

1: $\phi, \theta \leftarrow$ Initialiser les paramètres

2: **Répéter**

3: $E = \sum_{i=1}^N \|x^{(i)} - g_\theta(f_\phi(x^{(i)}))\|$ {Calculer la somme de l'erreur de reconstruction}

4: $\phi, \theta \leftarrow$ Mettre à jour les paramètres en utilisant les gradients de E (par ex. descente de gradient stochastique)

5: **jusqu'à** convergence des paramètres ϕ, θ

1.3 Adaptation à l'environnement dynamique

Dans un contexte où les données évoluent constamment et où de nouvelles classes peuvent émerger au fil du temps, un mécanisme d'adaptation est intégré pour maintenir la pertinence et l'efficacité du modèle de classification. Cette section détaille la stratégie mise en place pour permettre au système de s'ajuster dynamiquement aux changements de l'environnement de données.

1.3.1 Mécanisme d'apprentissage continu

L'apprentissage continu, aussi appelé continual learning ou apprentissage tout au long de la vie, est un domaine de l'intelligence artificielle qui vise à permettre aux modèles d'apprentissage automatique d'apprendre de manière continue sur de nouvelles données, en accumulant des connaissances sans oublier ce qui a été appris dans le passé [7].

L'apprentissage continu répond à un besoin essentiel des systèmes d'apprentissage modernes, nécessitant une adaptation rapide à des environnements dynamiques. Les modèles d'apprentissage automatique doivent être capables de s'adapter rapidement aux changements dans les données et dans l'environnement afin d'éviter les erreurs coûteuses et de maintenir la performance des modèles¹.

1.3.2 Scénarios de l'apprentissage continu

En fonction des caractéristiques du flux de données, les problèmes d'apprentissage continu peuvent être divisés en trois scénarios principaux, chacun ayant une solution commune : permettre au modèle de rester stable face aux connaissances anciennes tout en étant suffisamment flexible pour intégrer les nouvelles informations [8]. Voici les principaux scénarios d'apprentissage continu :

- **Apprentissage continu progressif en classe :** C'est un scénario dans lequel le nombre de classes d'une tâche de classification n'est pas fixe, mais peut augmenter au fil du temps.
- **Apprentissage continu incrémental dans le domaine :** Ce scénario se caractérise par le fait que la distribution des données change au fil du temps.
- **Apprentissage continu incrémental des tâches :** Cette technique d'apprentissage automatique entraîne un modèle à résoudre plusieurs tâches successives.

Dans le cadre de notre étude, nous nous concentrons spécifiquement sur le scénario d'apprentissage continu progressif en classe. Ce choix est motivé par la nature de notre problème, où de

1. <https://datascientest.com/apprentissage-continu-tout-savoir> consulté le 15 mai 2024

nouvelles classes peuvent émerger au fil du temps, nécessitant une adaptation dynamique de notre modèle de classification. Cette approche nous permet de traiter efficacement l'apparition de nouvelles catégories de données tout en maintenant les performances sur les classes déjà connues.

1.3.3 Techniques de l'apprentissage continu

Le tableau suivant présente trois principales approches pour l'apprentissage continu, chacune avec ses avantages et ses inconvénients spécifiques. L'approche par régularisation protège les paramètres importants pour éviter la perte de connaissances, mais nécessite un réglage précis des hyperparamètres. L'approche architecturale favorise la réutilisation des connaissances et l'ajout de nouvelles capacités, mais exige des ressources computationnelles élevées. Enfin, l'approche basée sur la mémoire de rejeu préserve les connaissances antérieures en stockant des données passées, mais peut être coûteuse en termes de mémoire et de calcul.

Techniques	Description de la méthode	Avantages	Limites
Régularisation	Protège les paramètres importants des tâches précédentes lors de l'apprentissage de nouvelles tâches pour préserver les connaissances antérieures	<ul style="list-style-type: none"> • Préserve les connaissances antérieures • Aide à éviter le sur-apprentissage 	<ul style="list-style-type: none"> • Nécessite de définir des hyperparamètres supplémentaires • Peut dégrader les performances sur la tâche courante
Architecturale	Se concentre sur l'adaptation de la structure du modèle pour de nouveaux apprentissages	<ul style="list-style-type: none"> • Favorise la réutilisation des connaissances précédemment acquises • Facilite l'ajout de nouvelles capacités sans oublier les anciennes 	<ul style="list-style-type: none"> • Nécessite des ressources computationnelles importantes
Mémoire de rejeu	Stocke un sous-ensemble représentatif des données passées pour les rejouer lors de l'apprentissage de nouvelles tâches, aidant ainsi à préserver les connaissances antérieures	<ul style="list-style-type: none"> • Aide à préserver les connaissances antérieures • Offre de bonnes performances sur les tâches précédentes 	<ul style="list-style-type: none"> • Nécessite de stocker et de gérer efficacement les données passées • Peut être coûteux en termes de mémoire et de calcul

TABLE 1.1 – Comparaison des techniques d'apprentissage continu

Après une analyse approfondie des différentes techniques d'apprentissage continu présentées dans le tableau 1.1, nous avons opté pour **une approche architecturale, plus précisément modulaire**, pour notre recherche. Cette décision est motivée par ses avantages : elle favorise la réutilisation des connaissances, facilite l'ajout de nouvelles capacités sans dégrader les performances, et s'adapte mieux aux environnements complexes. Cette approche surmonte les limitations des méthodes basées sur la régularisation, qui nécessitent un réglage précis des hyperparamètres et peuvent dégrader les performances, ainsi que celles basées sur la mémoire de rejeu, qui sont coûteuses en termes de calcul.

1.4 État de l’art

La détection de nouvelles classes représente un domaine de recherche en pleine expansion qui s’inscrit dans le contexte plus large de l’apprentissage machine adaptatif. Cette approche se distingue fondamentalement des approches traditionnelles de classification et de détection d’anomalies. Alors que les systèmes classiques opèrent dans un cadre statique avec un nombre prédéfini de classes, cette méthode permet non seulement d’identifier mais aussi d’intégrer dynamiquement de nouvelles catégories de données, sans nécessiter un réentraînement complet du modèle. Dans cette section, nous explorons les différentes approches développées pour la classification et la détection d’anomalies qui ouvrent des perspectives pour la détection et l’intégration de nouvelles classes.

1.4.1 Modèles basés sur les autoencodeurs

Les autoencodeurs ont émergé comme des outils pour la détection de nouvelles classes qui peuvent être considéré comme des anomalies dans divers domaines, grâce à leur capacité à apprendre des représentations compactes des données normales, ce qui les rend particulièrement efficaces pour identifier les écarts par rapport à ces modèles appris. Ils se distinguent par leur capacité à modéliser la distribution des données normales et à générer une erreur de reconstruction plus élevée lorsque des anomalies, considéré éventuellement comme des nouvelles classes sont détectées.

Saumya Sinha et al. ont démontré l’efficacité des Autoencodeurs Variationnels (VAE) dans la détection d’avalanches en utilisant des images SAR (Synthetic Aperture Radar). Leur méthode repose sur l’entraînement d’un VAE avec des données comprenant des anomalies, puis l’ajustement d’un seuil de détection basé sur l’erreur de reconstruction des images. Lorsque des anomalies sont présentes dans les images, comme les avalanches, l’erreur de reconstruction devient élevée, indiquant ainsi une anomalie. Cependant, l’efficacité de cette méthode est fortement influencée par la qualité des données d’entraînement et le réglage des hyperparamètres, notamment le taux d’apprentissage [9]. En outre, cette approche se limite essentiellement à une classification binaire, où l’objectif est de distinguer les anomalies des données normales, sans capacité explicite à identifier et classifier plusieurs nouvelles classes simultanément. Une des raisons pour lesquelles les VAE sont particulièrement efficaces dans ce contexte est leur capacité à capturer des distributions complexes, même dans des scénarios où les données peuvent être bruitées ou partiellement observables, comme les images SAR. Toutefois, leur dépendance à des données bien étiquetées reste un défi.

Dans le domaine industriel, les travaux de Cyriana M.A. Roelofs et al. montrent que les autoencodeurs ne sont pas seulement efficaces pour identifier les anomalies dans les systèmes de turbines éoliennes, mais également pour analyser leurs causes profondes[10]. Dans leur approche, les autoencodeurs sont utilisés pour reconstruire les caractéristiques observées des turbines à partir de données de capteurs. Les erreurs de reconstruction, servent alors à identifier les anomalies. Plus précisément, lorsque l’autoencodeur est formé à reproduire des comportements normaux, une erreur de reconstruction faible indique que le modèle a bien capté la dynamique du système. En revanche, une erreur de reconstruction élevée signale la présence d’une anomalie, car le modèle ne parvient pas à reconstruire correctement les données déviantes. Cependant, cette méthode reste limitée à une approche binaire : elle détecte les anomalies sans être capable de distinguer ni classer de multiples types d’anomalies ou de nouvelles classes. Pour affiner la détection d’anomalies et analyser leurs causes profondes, les auteurs ont utilisé des techniques complémentaires, comme la distance de Mahalanobis. En calculant cette distance, il devient possible d’identifier des points de données spécifiques responsables de l’anomalie, notamment en détectant des caractéristiques multivariées qui divergent. L’utilisation de l’erreur relative permet d’évaluer la contribution individuelle

de chaque caractéristique à l’anomalie détectée, ce qui est essentiel pour comprendre les défaillances spécifiques des sous-systèmes de la turbine. Cependant, un défi majeur avec cette méthode est le risque de sur-spécialisation. En effet, si l’autoencodeur est trop fortement entraîné sur un système particulier avec des conditions spécifiques, il peut devenir moins performant pour détecter des anomalies dans d’autres contextes ou sous d’autres conditions. Ce phénomène de sur-spécialisation limite la capacité du modèle à généraliser à d’autres systèmes industriels.

Malgré leur efficacité, ces modèles basés sur les autoencodeurs présentent une limitation majeure : leur dépendance à des jeux de données complets, où les anomalies et les comportements normaux sont déjà connus. Cette dépendance limite leur capacité à identifier de nouvelles classes, potentiellement considérées comme des anomalies, qui peuvent apparaître de manière imprévue et ne pas être présentes dans les données d’entraînement initiales.

1.4.2 Modèles basés sur d’autres méthodes

En plus des autoencodeurs, d’autres approches se sont avérées efficaces pour la détection d’anomalies.

L’Isolation Forest, développé par Hongzuo Xu et al., propose une méthode appelée Deep Isolation Forest (DIF) qui combine les réseaux neuronaux et l’algorithme d’Isolation Forest (IF) pour améliorer la détection d’anomalies complexes [11]. Le DIF repose sur un réseau neuronal qui encode les données dans un espace de représentation plus riche et adapté aux anomalies subtiles. Une fois les données transformées, l’Isolation Forest est appliqué pour diviser l’espace des données en régions distinctes, isolant efficacement les points de données anormaux. Il est important de noter que l’Isolation Forest, bien que performant pour des anomalies isolées ou évidentes, peut avoir des difficultés avec des anomalies subtiles, en raison de la manière dont il isole les points. Sa performance diminue également lorsque les données sont déséquilibrées ou qu’il y a des variations graduelles dans les comportements anormaux. Cependant, une autre limite du Deep Isolation Forest concerne sa capacité à gérer l’apparition de nouvelles classes dans des environnements évolutifs. Cette méthode reste binaire, cherchant à isoler des points anormaux sans pouvoir discriminer entre plusieurs nouvelles classes ou catégories d’anomalies. L’algorithme est conçu pour identifier les écarts par rapport aux comportements normaux, ce qui peut entraîner la détection des nouvelles classes comme des anomalies, car elles ne font pas partie des données d’entraînement initiales. Cela pose un problème d’adaptation : le modèle doit être entièrement réentraîné pour incorporer ces nouvelles classes, ce qui peut être coûteux et inefficace dans des environnements dynamiques où de nouvelles classes apparaissent fréquemment.

1.4.3 Modèles basés sur l’apprentissage continu pour la détection d’anomalies

Les approches de détection d’anomalies basées sur l’apprentissage continu ont pris de l’importance, en particulier dans le domaine de la surveillance vidéo. L’article de Keval Doshi et Yasin Yilmaz présente une méthode innovante qui combine l’extraction de caractéristiques à l’aide de réseaux neuronaux avec des techniques de détection d’anomalies basées sur des méthodes statistiques, telles que le k-nearest neighbors (kNN) [12]. Cette méthode permet de réduire la complexité de l’entraînement tout en offrant une capacité d’apprentissage adaptatif à partir de nouvelles données, ce qui est essentiel dans des environnements dynamiques où les anomalies peuvent varier rapidement. Cependant, elle présente des limites, notamment la dépendance aux données d’entraînement initiales et la nécessité d’une évaluation en temps réel.

Cependant, ces méthodes présentent encore des limitations, notamment leur dépendance aux données étiquetées pour détecter les anomalies. Elles se limitent à la détection binaire d’anomalies, ne permettant pas de distinguer entre différentes catégories ou de classifier les nouvelles classes émergentes.

La méthode que nous proposons se distingue par son approche dynamique pour intégrer de nouvelles classes, potentiellement considérées comme des anomalies. Contrairement aux approches existantes, notre modèle est conçu pour s’adapter dynamiquement à l’apparition de nouvelles classes dans un environnement évolutif, sans nécessiter de réentraînement complet comme le suggère l’article de Hongzuo Xu et al., [11] . Cette approche permet non seulement de détecter des anomalies, mais aussi d’identifier et d’intégrer de nouvelles classes au fil du temps, tout en maintenant les performances sur les classes déjà apprises. En utilisant l’erreur de reconstruction et un mécanisme d’apprentissage continu, notre modèle est capable de généraliser à des environnements complexes et en constante évolution.

1.5 Tableau récapitulatif de l’état de l’art

Le tableau 1.2 présente un résumé de l’état de l’art sur les approches de la détection d’anomalies considérées comme de nouvelles classes avec l’apprentissage automatique.

Titre de l’article	Domaine d’application	Méthodologie	Avantages	Limites
Variational Autoencoder Anomaly-Detection of Avalanche Deposits in Satellite SAR Imagery	Détection d’avalanches	Utilisation des VAE pour modéliser des distributions complexes et identifier les anomalies à partir de l’erreur de reconstruction	Modélisation des distributions complexes, capacité à traiter des images radar SAR	Sensibilité à la qualité et à la quantité des données, dépendance aux hyperparamètres du modèle, limité à la classification binaire
Autoencoder-Based Anomaly Root Cause Analysis for Wind Turbines	Analyse des causes d’anomalies dans les turbines éoliennes	Autoencodeurs combinés avec des techniques comme la distance de Mahalanobis pour identifier les anomalies spécifiques	Détection précise d’anomalies et analyse des causes, capacité à gérer les dépendances entre caractéristiques	Sensible aux données bruitées, limitation pour les environnements dynamiques et changeants, classifie uniquement entre anomalies et normalité

Titre de l'article	Domaine d'application	Méthodologie	Avantages	Limites
Isolation Forest : A Scalable Anomaly Detection Algorithm using Random Forest	Détection d'anomalies générale	Algorithme d'Isolation Forest basé sur la création de partitions aléatoires pour isoler les points anormaux	Haute évolutivité, temps de calcul rapide pour les grands ensembles de données	Moins performant pour les anomalies subtiles ou les jeux de données déséquilibrés, incapable de discriminer plusieurs types d'anomalies (classification binaire)
Continual Learning for Anomaly Detection in Surveillance Videos	Détection d'anomalies dans les vidéos de surveillance	Apprentissage continu combiné à l'extraction de caractéristiques par réseaux neuronaux et détection par kNN	Apprentissage en continu, adaptable aux environnements évolutifs, capacité d'entraînement sur de nouvelles données sans réentraînement complet	Dépendance aux données d'entraînement initiales, complexité de l'évaluation en temps réel, limitation pour l'intégration de nouvelles classes

TABLE 1.2 – Résumé des travaux

1.6 Bilan du chapitre

Ce chapitre a établi les fondations nécessaires à la compréhension de notre approche pour concevoir et mettre en œuvre un système d'apprentissage automatique destiné à la classification dans des environnements dynamiques. Nous avons examiné les concepts clés relatifs aux autoencodeurs, en mettant en avant leur capacité à reconstruire des éléments connus et à détecter de nouvelles classes à travers les erreurs de reconstruction. Une revue de la littérature a révélé l'efficacité d'un ensemble de techniques dans des tâches similaires. Cependant, malgré la puissance de ces techniques pour identifier des tendances ou classifier des éléments connus, ces algorithmes rencontrent des difficultés face à des situations nouvelles [9, 10, 11]. En effet, ces algorithmes se sont limités à des tâches de classification binaire. Ainsi, la question de notre recherche est donc : comment développer un système d'apprentissage continu ayant des sous modules basés sur les autoencodeurs pour la classification multiclasse ? Pour répondre à cette question, nous proposons une approche basée sur l'apprentissage continu, combinée à une architecture de modèle modulaire pour l'intégration continue de nouvelles classes. Le chapitre suivant présentera en détail la méthode proposée.

CHAPITRE 2

MÉTHODE PROPOSÉE

2.1 Introduction

Dans cette section, nous présentons une méthode de classification basée sur un autoencodeur pour détecter de nouvelles classes dans un environnement dynamique. Cette approche se déroule en deux phases. La première phase consiste à entraîner un modèle de base M sur les classes connues, puis à le spécialiser pour chaque classe M_i . La seconde phase, intégrant l'apprentissage continu, utilise les sous-modèles M_i pour tester et évaluer chaque nouvelle donnée avec les sous-modèles existants. Un nouveau sous-modèle M_i est créé si la donnée ne correspond à aucune classe connue, permettant au système de s'adapter progressivement, en intégrant de nouvelles classes au fil du temps.

Les sections suivantes détailleront la conception de l'autoencodeur, le processus de spécialisation, ainsi que la méthode employée pour assurer son adaptabilité dans la classification d'éléments nouveaux au fil du temps.

2.2 Méthode de conception du modèle de classification

La méthode proposée se déroule en deux étapes principales : dans un premier temps, un modèle généraliste de base, noté M , est construit par un entraînement sur l'ensemble des données disponibles, couvrant les différentes classes C_i . Ce modèle généraliste a pour objectif de capturer les caractéristiques communes aux différentes classes C_i .

Dans un second temps, ce modèle de base M est spécialisé pour chacune des classes C_i individuellement. Cela se traduit par un réentraînement de M sur les données spécifiques à chaque classe C_i , donnant ainsi naissance aux modèles spécialisés M_i .

Pour chaque modèle spécialisé M_i , un seuil d'erreur de reconstruction est alors calculé. Ce seuil est déterminé à partir des erreurs de reconstruction obtenues par M_i sur les données de la classe C_i correspondante. Le seuil est calculé selon la formule suivante :

$$\text{Seuil} = \mu + k \cdot \sigma$$

où :

- μ est la moyenne des erreurs de reconstruction,

- σ est l'écart-type des erreurs de reconstruction,
- k est un facteur multiplicatif généralement choisi en fonction du niveau de confiance souhaité.

Cette approche de calcul du seuil basée sur la moyenne et l'écart-type est préférable à l'utilisation du maximum, car ce dernier serait trop sensible aux outliers et ne refléterait pas de manière représentative la distribution des erreurs, comme le souligne l'article de Dubey et al.[13]. De plus, elle s'ajuste aux caractéristiques des données, puisque chaque classe C_i aura sa propre moyenne et son propre écart-type.

Lors de la prédiction, lorsqu'une nouvelle donnée est présentée au système, son erreur de reconstruction est évaluée par rapport à chacun des modèles spécialisés M_i . Si l'erreur est inférieure au seuil défini pour un modèle M_i , cela signifie que la donnée est bien représentée par ce modèle et appartient probablement à la classe C_i associée. Dans le cas où plusieurs modèles M_i parviennent à bien reconstruire la donnée, c'est le modèle ayant la plus faible erreur de reconstruction qui détermine la classe d'appartenance.

Enfin, si l'erreur de reconstruction de la nouvelle donnée n'est inférieure à aucun des seuils des modèles M_i , elle est alors considérée comme appartenant à une nouvelle classe, non présente parmi celles utilisées pour l'entraînement.

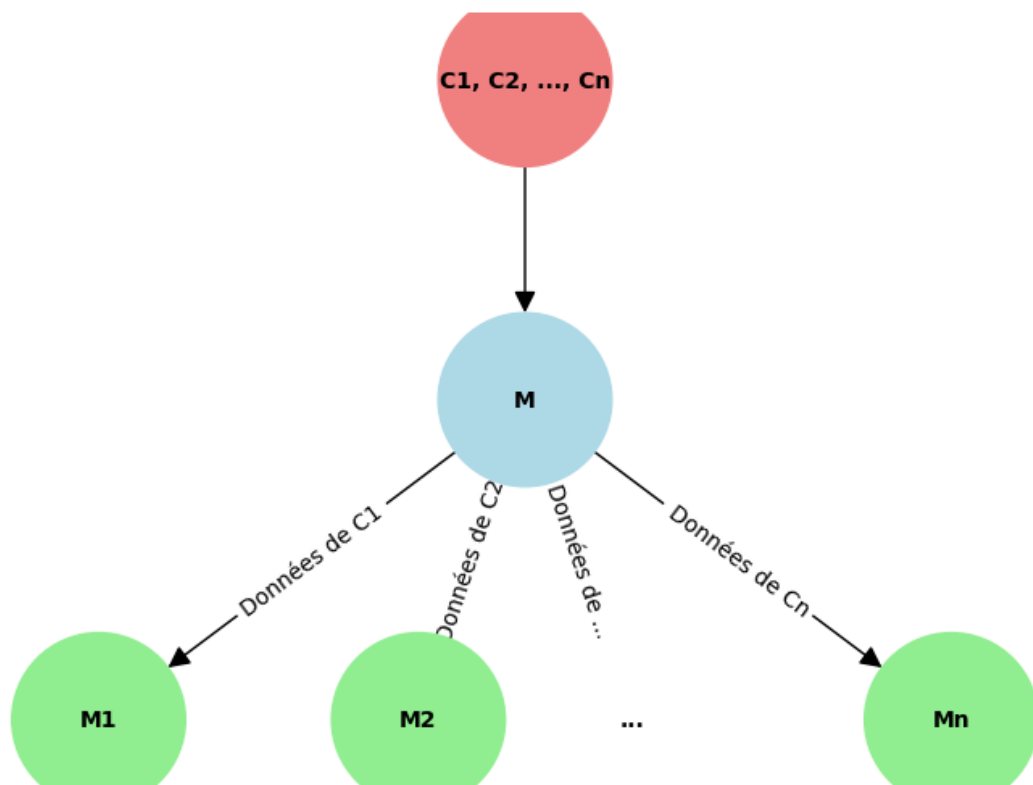


FIGURE 2.1 – Architecture du modèle

La figure 2.1 illustre ce processus où, initialement, les différentes classes de l'ensemble de données, désignées par C_1, C_2, \dots, C_n , sont regroupées pour entraîner l'autoencodeur général M . À partir de ce modèle général, des modèles spécialisés, notés M_1, M_2, \dots, M_n , sont dérivés et affinés en utilisant uniquement les données d'une seule classe spécifique correspondante.

2.2.1 Construction de l'architecture de l'autoencodeur de base

La construction du modèle de base repose sur l'architecture d'un autoencodeur. Ce choix se justifie par ses capacités à bien reconstruire les éléments d'un même ensemble, et à mal reconstruire ceux des autres ensembles.

Un autoencodeur se compose de deux parties principales :

- **Encodeur** : Cette partie prend la donnée d'entrée et la projette dans un espace de représentation de dimension réduite, appelé espace latent.
- **Décodeur** : À partir de cette représentation latente, le décodeur reconstruit la donnée d'entrée initiale en inversant le processus de l'encodeur. La reconstruction se fait en réinterprétant les caractéristiques latentes pour générer une version aussi proche que possible de la donnée d'origine.

2.2.2 Spécialisation du modèle M pour obtenir les M_i

L'architecture de base reste inchangée. Toutefois, pour chaque classe C_i , nous appliquons l'apprentissage par transfert en réutilisant les poids pré-entraînés du modèle de base M comme point de départ, plutôt que de recommencer l'apprentissage à zéro. Nous exploitons les connaissances déjà acquises par le modèle de base lors de son entraînement initial sur les classes connues, permettant ainsi d'accélérer le processus d'entraînement et d'améliorer la performance du modèle spécialisé.

L'algorithme 2 présente les étapes nécessaires pour construire et spécialiser les modèles pour chaque classe spécifique C_i à partir du modèle de base M , ainsi que le calcul des seuils d'erreur de reconstruction spécifiques à chaque classe.

Algorithme 2 Construction et spécialisation des modèles

- 1: **Données** : Ensemble de données D contenant plusieurs classes $\{C_1, C_2, \dots, C_n\}$
 - 2: **Résultats** : Modèle généraliste M , Modèles spécialisés M_i pour chaque classe C_i , Seuils d'erreur de reconstruction E_i pour chaque modèle spécialisé M_i
 - 3: $M \leftarrow$ Entraîner le modèle généraliste de base sur l'ensemble des données D pour capturer les caractéristiques communes aux classes
 - 4: **Pour** chaque classe spécifique C_i dans l'ensemble des classes $\{C_1, C_2, \dots, C_n\}$ **faire**
 - 5: $M_i \leftarrow M$ {Initialiser M_i avec les poids pré-entraînés du modèle M }
 - 6: $M_i \leftarrow$ Réentraîner M_i sur les données spécifiques à la classe C_i
 - 7: $Erreurs \leftarrow$ Calculer les différentes erreurs obtenues après le décodage de chaque donnée de la classe C_i en utilisant M_i
 - 8: $\mu \leftarrow$ Déterminer la moyenne des erreurs de reconstruction
 - 9: $\sigma \leftarrow$ Calculer l'écart-type des erreurs de reconstruction
 - 10: $E_i \leftarrow \mu + k \cdot \sigma$ {Déterminer le seuil d'erreur E_i à l'aide de la formule où k est un facteur multiplicatif choisi selon le niveau de confiance souhaité}
 - 11: **Fin Pour**
-

2.3 Intégration de nouvelles classes

2.3.1 Processus de détection des nouvelles classes

Le processus de détection des nouvelles classes est illustré par l’algorithme 3. Ce mécanisme repose sur la comparaison des erreurs de reconstruction générées par chaque sous-module de l’autoencodeur lorsqu’une nouvelle donnée est introduite. Si cette erreur dépasse un seuil préétabli, le sous-module émet un signal indiquant la possibilité d’une nouvelle classe. Toutefois, si plusieurs sous-modules reconnaissent la donnée comme appartenant à une classe existante, la classe associée à la plus faible erreur de reconstruction est choisie. Si aucun sous-module ne peut reconstruire la donnée avec une erreur suffisamment faible, elle est identifiée comme une nouvelle classe.

Algorithme 3 Processus de détection des nouvelles classes

Données : Nouvelle donnée X_{new}

Résultat : Classe associée à X_{new} ou nouvelle classe détectée

- 1: **Pour** chaque sous-module M_i de l’autoencodeur **faire**
 - 2: Calculer l’erreur de reconstruction $\mathcal{E}_i = \|X_{\text{new}} - \hat{X}_{M_i}\|$
 - 3: **Fin Pour**
 - 4: **Si** $\forall i, \mathcal{E}_i > \epsilon_i$ **alors**
 - 5: X_{new} est identifié comme appartenant à une nouvelle classe
 - 6: **sinon**
 - 7: Sélectionner la classe associée à $\min_i \mathcal{E}_i$
 - 8: **Fin Si**
-

2.3.2 Mécanisme d’ajout de nouvelles classes

Lorsque le système détecte une nouvelle classe, un nouveau module, initié à partir de M , est créé pour gérer cette classe, comme illustré par la figure 2.2. Ce module est spécialement conçu et entraîné à partir des données identifiées comme appartenant à la nouvelle classe. Une fois l’entraînement terminé, ce module est intégré dans le système global, permettant ainsi au modèle d’élargir son champ de connaissances tout en préservant les classes précédemment apprises. Cette approche modulaire assure une croissance flexible et efficace du modèle, où chaque nouvelle classe peut être intégrée sans perturber les performances globales du système.

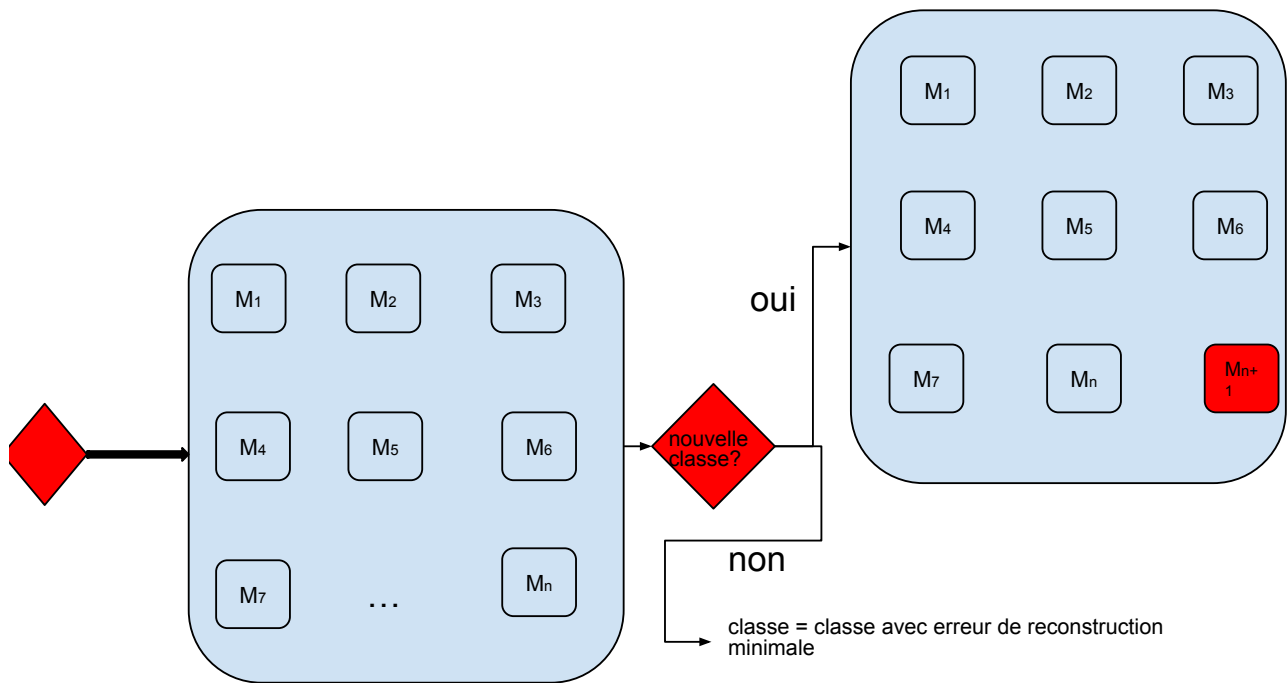


FIGURE 2.2 – Modèle continu

2.4 Bilan du chapitre

Dans ce chapitre, nous avons détaillé le processus de conception et de mise en œuvre de notre système modulaire, conçu pour l'adaptation dynamique dans des environnements complexes. Nous avons commencé par présenter l'architecture de base de l'autoencodeur, puis nous avons exploré la spécialisation de cette architecture pour chaque classe, en introduisant des sous-modules dédiés, entraînés spécifiquement pour ces classes. Nous avons également discuté de l'utilisation de l'erreur de reconstruction comme indicateur clé pour la détection des nouvelles classes et de l'intégration progressive de ces nouveaux sous-modules. Enfin, nous avons introduit des méthodes pour définir des seuils de détection précis.

CHAPITRE 3

EXPÉRIMENTATIONS, RÉSULTATS, DISCUSSIONS

3.1 Introduction

Dans ce chapitre, nous présentons les différentes étapes réalisées pour la construction du modèle utilisé lors des expérimentations, ainsi que les diverses opérations de prétraitement. Nous poursuivons par la présentation des métriques employées pour évaluer notre modèle. Ensuite, nous détaillons le protocole expérimental, avant de présenter les résultats obtenus en fonction de ces métriques.

3.2 Présentation des données de l'expérimentation

Pour les expérimentations, nous utilisons deux ensembles de données. Le premier a été collecté sur les fonds marins d'Afrique du Sud, dans le but de développer une solution automatisée de classification d'images pour identifier les espèces marines à partir de photographies¹. Chaque année, des mesures précises sont effectuées et des photographies des espèces marines sont prises, puis ces données sont ajoutées à une base de données. Le jeu de données utilisé pour l'entraînement de notre modèle est la partie train, structurée en 137 dossiers distincts, chacun correspondant à une classe d'espèce marine spécifique et contenant des images.

Le second ensemble de données est le jeu de données Caltech-UCSD Birds-200-2011 (CUB-200-2011), qui contient 200 sous-catégories d'oiseaux. Nous utilisons uniquement la partie train, composée de 5994 images, pour nos expérimentations².

3.3 Pré-traitement des données

Le prétraitement est l'ensemble des opérations appliquées aux données dans le but de les rendre exploitables par des algorithmes d'apprentissage. Nous présentons ici le prétraitement général utilisé sur nos images.

1. <https://zindi.africa/competitions/marine-invertebrates-classification-challenge> consulté le 3 octobre 2023

2. <https://www.kaggle.com/datasets/thiagojr/cub-200-2011> consulté le 17 août 2024

- **Redimensionnement** : Les images d'origine, de taille 512x384 pixels pour les images marines et de tailles variables pour le jeu de données CUB-200-2011, sont redimensionnées à une taille fixe de 128x128 pixels. Ce redimensionnement permet de diminuer les ressources nécessaires et le temps de traitement, tout en réduisant la variabilité liée à la taille, sans perdre trop d'informations visuelles pour une classification efficace.
- **Conversion en tenseur** : Cette conversion implique une normalisation des valeurs de pixels, les faisant passer de l'échelle $[0, 255]$ à $[0, 1]$.

3.4 Métriques d'évaluation

Dans le cadre de l'apprentissage continu, les performances des modèles sont généralement évaluées selon trois aspects : la performance globale (Précision Moyenne Globale) sur l'ensemble des classes apprises, la stabilité de la mémoire (Oubli Moyen) des anciennes classes, et la plasticité d'apprentissage des nouvelles classes.

3.4.1 Précision Moyenne

La précision moyenne, en anglais (Average Accuracy, AA), est calculée à l'instant n , correspondant au nombre de classes apprises jusqu'à présent. Si $P_{i,j} \in [0, 1]$ représente la précision évaluée sur l'ensemble de test de la classe j après l'apprentissage de la classe i (avec $j \leq i$), alors la précision moyenne AA_n à l'instant n est définie par :

$$AA_n = \frac{1}{n} \sum_{j=1}^n P_{n,j} \quad (3.1)$$

Cette métrique permet de mesurer la performance globale du modèle à l'instant présent, en prenant en compte toutes les classes apprises jusqu'à ce moment.

3.4.2 Précision Moyenne Globale

La précision moyenne globale, en anglais (Average Incremental Accuracy, AIA), est une mesure qui reflète l'évolution de la performance du modèle au fil du temps. Elle est calculée en prenant la moyenne des précisions moyennes AA_i obtenues jusqu'à l'instant n :

$$AIA_n = \frac{1}{n} \sum_{i=1}^n AA_i \quad (3.2)$$

Cette métrique capture non seulement la performance actuelle, mais aussi la manière dont le modèle a maintenu ses performances au cours de l'apprentissage des nouvelles classes. Une Précision Moyenne Globale élevée indique que le modèle maintient des performances constantes et solides sur toutes les classes apprises [14].

3.4.3 Oubli Moyen

L'oubli moyen, en anglais (Forgetting Measure, FM), quantifie l'oubli des connaissances précédemment acquises par le modèle lorsqu'il apprend de nouvelles classes. Pour chaque classe j apprise avant l'instant n , l'oubli est calculé comme la différence entre la meilleure précision obtenue pour cette classe et la précision actuelle :

$$f_{j,n} = \max_{i \in \{1, \dots, n-1\}} (P_{i,j} - P_{n,j}) \quad (3.3)$$

L'oubli moyen à l'instant n est ensuite calculé comme la moyenne des oublis pour toutes les classes apprises avant n :

$$\text{FM}_n = \frac{1}{n-1} \sum_{j=1}^{n-1} f_{j,n} \quad (3.4)$$

Cette métrique est essentielle pour évaluer l'oubli catastrophique dans les scénarios d'apprentissage continu, introduite par Chaudhry et al. [15].

3.4.4 Plasticité

La plasticité, en anglais (Plasticity), mesure la capacité du modèle à apprendre de nouvelles classes sans compromettre les performances sur les anciennes classes. Elle est définie par la précision obtenue sur la dernière classe ajoutée au modèle, notée P_n :

$$\text{Plasticity}_n = P_n \quad (3.5)$$

Cette métrique est cruciale pour évaluer la capacité du modèle à s'adapter à de nouvelles informations tout en conservant les connaissances acquises [16].

3.5 Protocole expérimental

3.5.1 Environnement de travail

Les expérimentations ont été menées sur une machine locale équipée d'un système d'exploitation Linux Ubuntu 22.04 LTS, d'une mémoire de 64 Go de RAM, d'un processeur Intel Core i7-7820HQ, et d'un GPU NVIDIA Quadro P4000 avec 8 Go de mémoire. Pour le développement des autoencodeurs, nous avons utilisé le langage Python et le framework PyTorch.

3.5.2 Architectures des modèles

Le modèle d'autoencodeur développé est composé de deux parties : un encodeur et un décodeur.

- L'encodeur est basé sur le modèle de réseau de neurones convolutif ResNet18, préentraîné sur la base de données ImageNet. Il extrait une représentation compacte et sémantique des données d'entrée en utilisant les couches de convolution et de pooling du modèle ResNet18. Il a été démontré par Seyed Iman Mirzadeh et al. [17] qu'en apprentissage continu, la suppression de la couche de global average pooling (GAP) permet d'obtenir de meilleures performances. Nous supprimons également la couche de classification, car pour un modèle d'autoencodeur, l'objectif n'est pas la classification mais plutôt l'extraction de caractéristiques pertinentes pour permettre au décodeur de reconstruire fidèlement les données d'entrée.
- Le décodeur, quant à lui, est chargé de reconstruire l'image à partir de cette représentation encodée. Il est composé de plusieurs couches de convolutions transposées, qui vont progressivement augmenter la taille des données jusqu'à obtenir les mêmes dimensions que l'entrée initiale. Ce processus de "décodage" permet de générer une image reconstruite à partir de la représentation compacte.

3.5.3 Division des données

Pour le modèle généraliste, les données des deux jeux de données ont été divisées en deux ensembles distincts : 80% pour l'entraînement et 20% pour la validation. En revanche, pour le modèle spécialiste, les données de chaque classe ont été réparties en trois ensembles : 70% pour l'entraînement, 15% pour la validation, et 15% pour le test.

3.5.4 Description des paramètres d'entraînement

Le tableau 3.1 présente les paramètres d'entraînement des modèles généraliste et spécialiste.

TABLE 3.1 – Comparaison des paramètres d'entraînement des modèles généraliste et spécialiste

Paramètre	Modèle Généraliste	Modèle Spécialiste
Architecture	ResNet18 préentraîné + Décodeur personnalisé	ResNet18 préentraîné + Décodeur personnalisé
Taille des Images	128x128	128x128
Ensemble de Données	Tous les sous-dossiers	Les 5 classes les plus représentées
Taille du Batch	32	Variable selon la classe (dépend du nombre d'images)
Optimiseur	Adam (learning rate = 0.001)	Adam (learning rate = 0.001)
Fonction de Perte	MSE (Mean Squared Error)	MSE (Mean Squared Error)
Nombre d'Époques	Variable selon la convergence (généralement 500)	Variable selon la convergence (généralement 500)
Stratégie de Sauvegarde	Sauvegarde du modèle avec la meilleure perte de validation	Sauvegarde des meilleurs modèles spécialisés par classe avec la meilleure perte de validation
Stratégie d'Entraînement	Entraînement global sur toutes les classes	Entraînement spécialisé par classe

3.6 Résultats et discussions

3.6.1 Performances des modèles généralistes

Les Figures 3.1 et 3.2 présentent les reconstructions produites par les modèles généralistes pour un échantillon d'images originales issues de différentes classes pour les deux jeux de données.



FIGURE 3.1 – Reconstructions des données par le modèle généraliste.



FIGURE 3.2 – Reconstructions des données par le modèle généraliste.

On observe que certaines reconstructions préservent les formes et caractéristiques générales des objets, résultant en une faible erreur de reconstruction. Cependant, pour d'autres classes, la qualité de la reconstruction est significativement dégradée, avec des images plus floues et des détails moins distincts, entraînant une erreur de reconstruction élevée.

3.6.2 Performances des modèles spécialistes

Les Figures 3.3 et 3.4 illustrent les performances du modèle spécialiste, qui a été affiné (fine-tuné) à partir du modèle généraliste pour le jeu de données des invertébrés marins. Les Figures 3.5 et 3.6 présentent les résultats sur le jeu de données d'oiseaux.

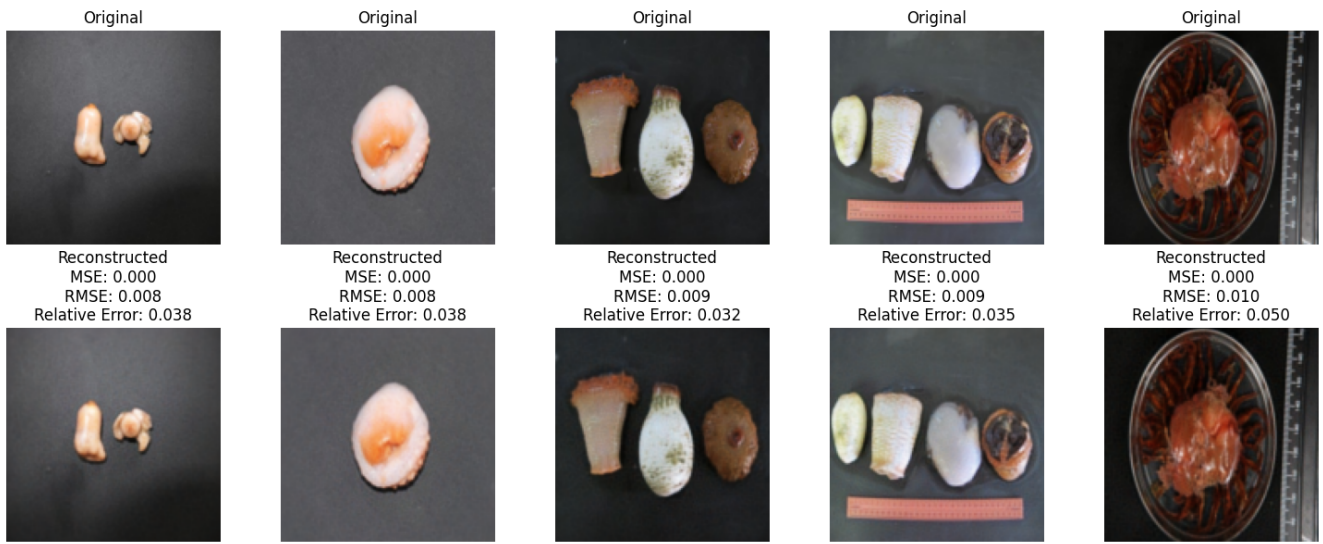


FIGURE 3.3 – Reconstructions des données par le modèle spécialiste sur sa classe cible pour les invertébrés marins.

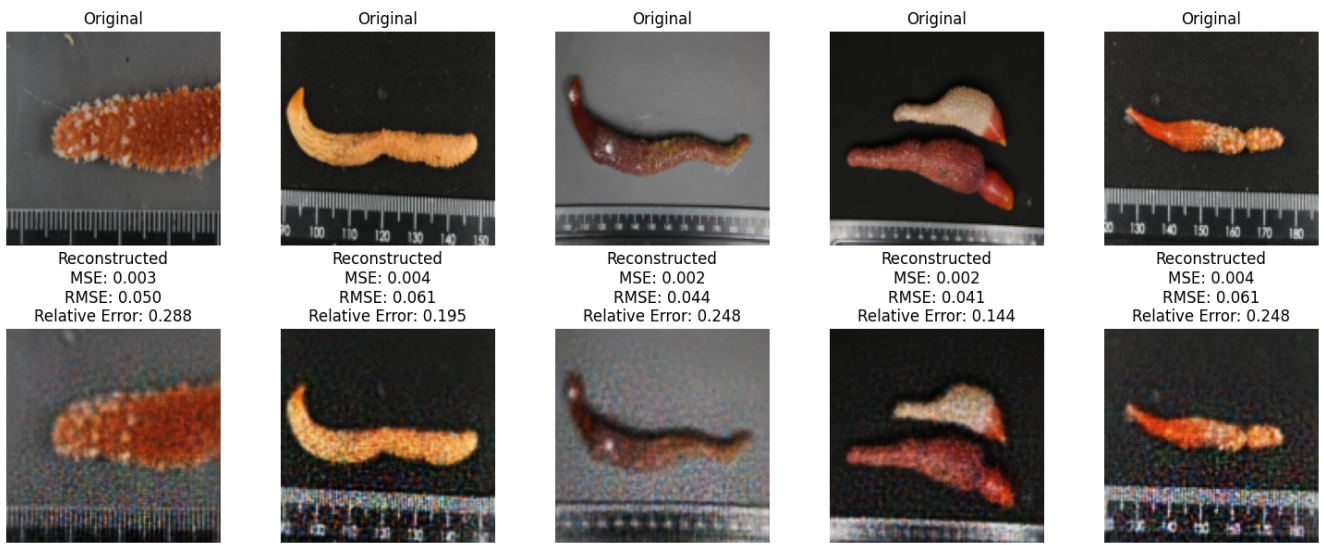


FIGURE 3.4 – Reconstructions des données par le modèle spécialiste sur une classe différente pour les invertébrés marins.

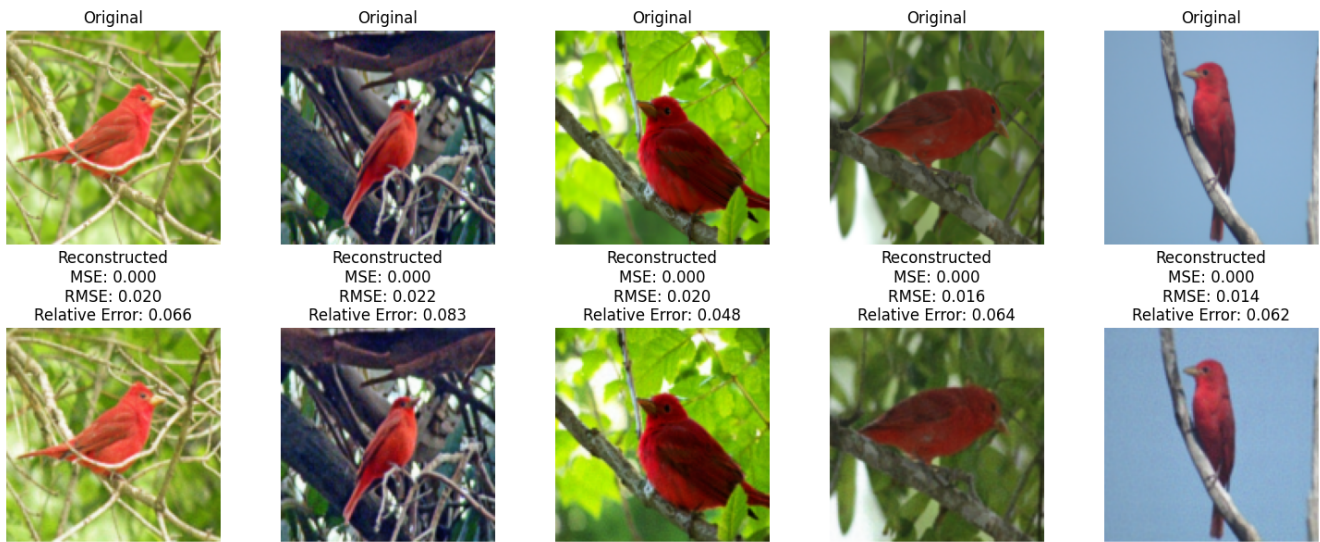


FIGURE 3.5 – Reconstructions des données par le modèle spécialiste sur sa classe cible pour le jeu de données d’oiseaux.

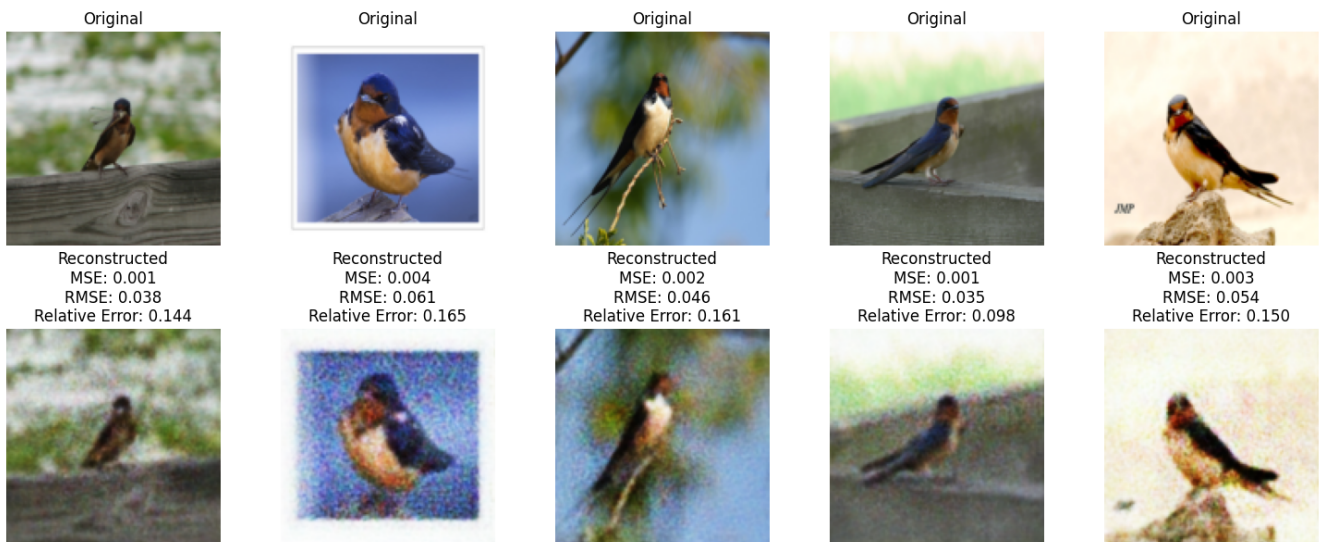


FIGURE 3.6 – Reconstructions des données par le modèle spécialiste sur une classe différente pour le jeu de données d’oiseaux.

Les modèles spécialistes démontrent une amélioration notable de la qualité de reconstruction par rapport aux modèles généralistes lorsqu’ils sont appliqués à leur classe cible. On observe des reconstructions avec une très faible erreur, reflétant une excellente capture des formes et textures spécifiques à ces classes.

En revanche, lorsqu’ils sont appliqués à une classe différente, le modèle spécialiste montre une dégradation notable de la qualité de reconstruction. Les images reconstruites présentent une perte de détails et une qualité globale inférieure, avec des erreurs de reconstruction élevées. Néanmoins, les reconstructions restent reconnaissables, ce qui suggère que les modèles conservent une partie des connaissances générales acquises lors de leur entraînement initial sur toutes les classes.

3.6.3 Cas particuliers et variabilité des performances

Jeu de données des invertébrés marins

L'analyse des performances à l'aide de boîtes à moustaches révèle des cas intéressants :

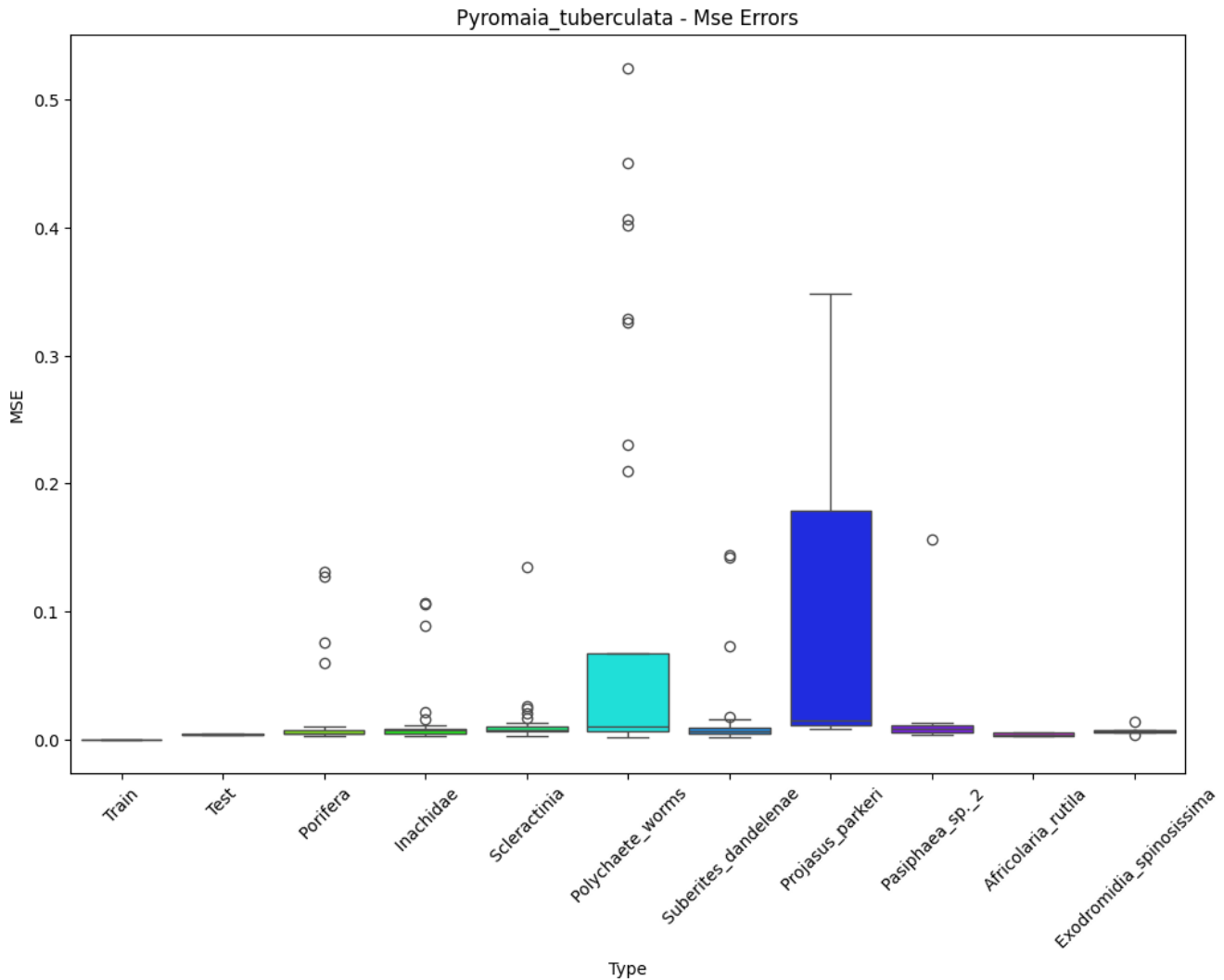


FIGURE 3.7 – Boîte à moustaches montrant un modèle spécialiste performant de manière variable en fonction du nombre d'images par classe.

Pour les classes avec moins d'images, les modèles tendent à présenter des résultats plus faibles sur les données de leurs classes spécifiques et des erreurs élevées sur les images des autres classes. Cela suggère que ces modèles sont plus aptes à reconstruire les éléments de leur propre classe, comme le montre la Figure 3.7.

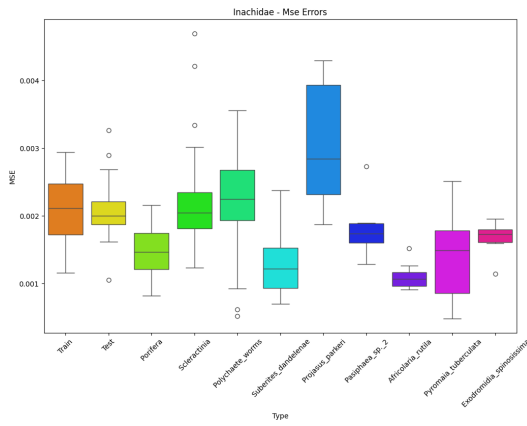


FIGURE 3.8 – Boîte à moustaches du modèle reconstruisant mieux les données d'autres classes.

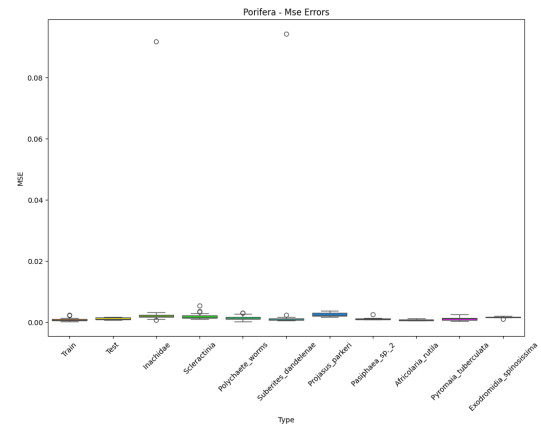


FIGURE 3.9 – Boîte à moustaches du modèle performant bien sur ses données et sur d'autres classes.

1. La Figure 3.9 montre que certains modèles réussissent à bien reconstruire les données de leur classe spécialisée tout en maintenant des erreurs faibles pour les éléments d'autres classes. Ce comportement est particulièrement observé pour les classes disposant d'un grand nombre d'images.
2. Comme illustré dans la Figure 3.8, certains modèles spécialisés reconstruisent parfois mieux les données d'une classe pour laquelle ils n'ont pas été spécialisés, tout en présentant une erreur élevée sur leurs propres données. Ce phénomène pourrait être attribué à des problèmes de "fine-grained" ou à des variations de luminosité.

Ces observations soulignent la variabilité des performances des modèles spécialisés en fonction des caractéristiques des classes et de la quantité de données disponibles pour l'entraînement.

Jeu de données d'oiseaux

L'analyse des performances à l'aide de boîtes à moustaches révèle des cas similaires :

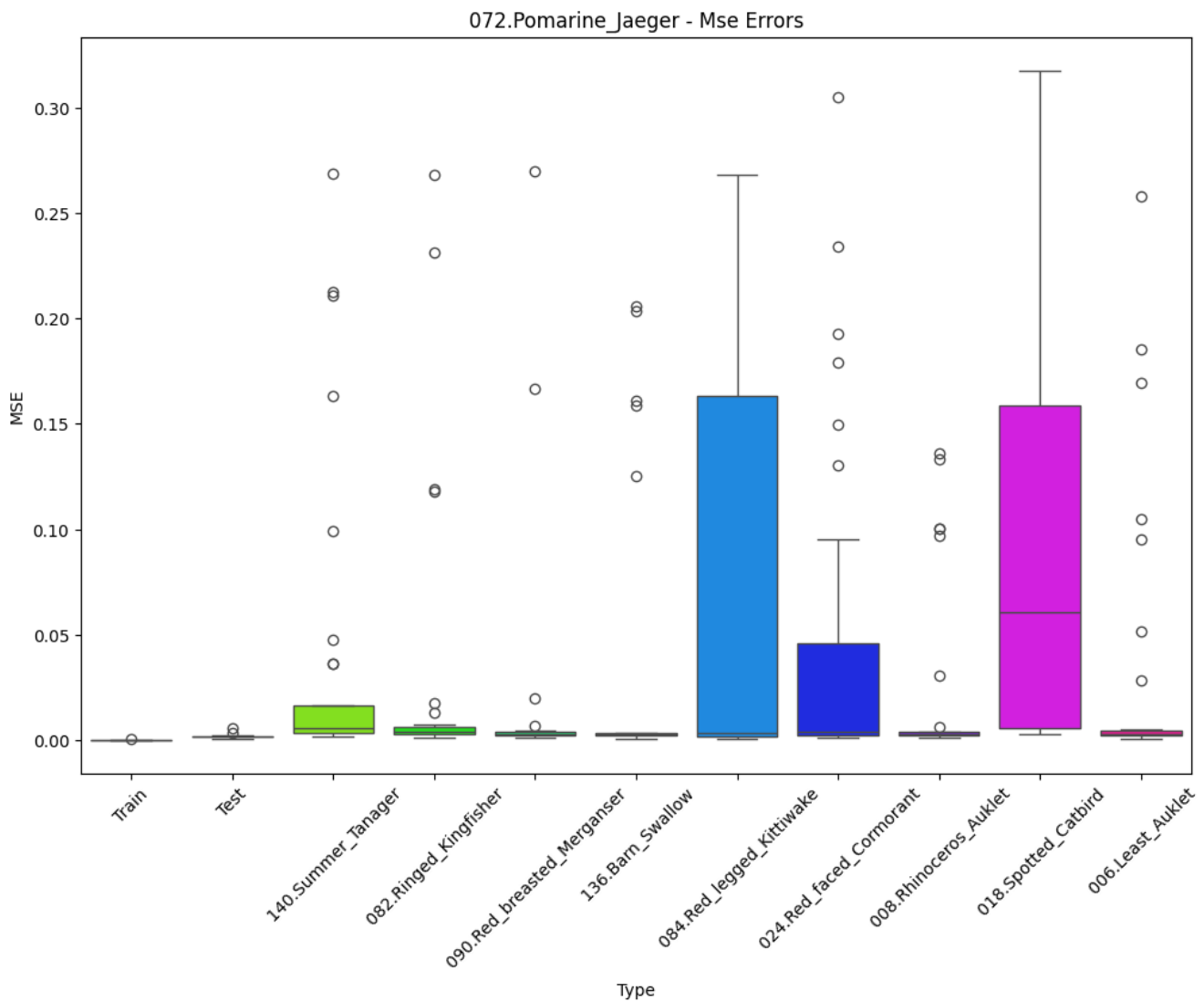


FIGURE 3.10 – Boîte à moustaches montrant un modèle spécialiste performant de manière variable en fonction du nombre d’images par classe.

Pour certaines classes, le modèle tend à présenter des résultats plus faibles sur les données de leurs classes spécifiques et des erreurs élevées sur les images des autres classes. Cela suggère que ces modèles sont plus aptes à reconstruire les éléments de leur propre classe, comme le montre la Figure 3.10.

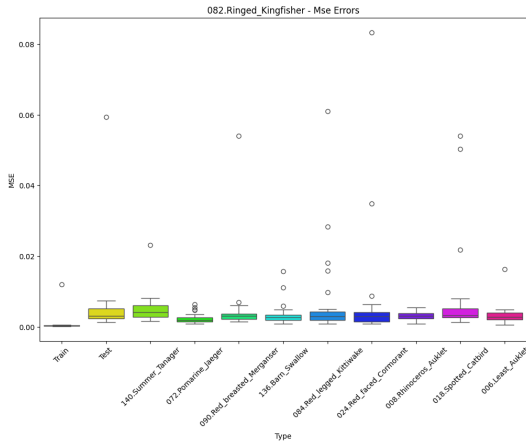


FIGURE 3.11 – Boîte à moustaches du modèle reconstruisant mieux les données d’autres classes.

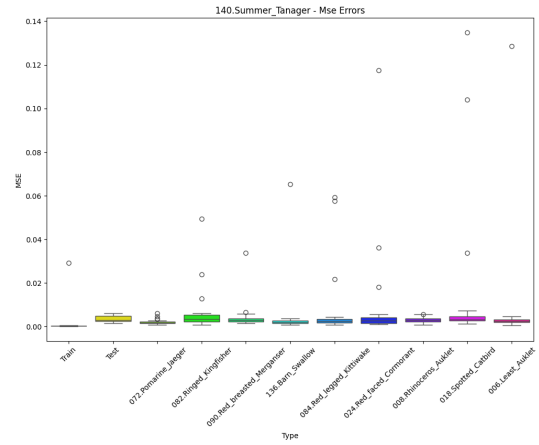


FIGURE 3.12 – Boîte à moustaches du modèle performant bien sur ses données et sur d’autres classes.

1. La Figure 3.12 montre que certains modèles réussissent à bien reconstruire les données de leur classe spécialisée tout en maintenant des erreurs faibles pour les éléments d’autres classes.
2. Comme illustré dans la Figure 3.11, certains modèles spécialisés reconstruisent parfois mieux les données d’une classe pour laquelle ils n’ont pas été spécialisés, tout en présentant une erreur élevée sur leurs propres données.

3.6.4 Phase de continuité

La phase de continuité du modèle d’apprentissage continu a été analysée à travers plusieurs métriques. Ces métriques nous permettent de comprendre comment le modèle s’adapte à l’ajout progressif de nouvelles classes, tout en maintenant ou en ajustant ses performances sur les classes déjà apprises.

Analyse des performances par classe

L’analyse des performances du modèle pour les premières classes ajoutées, en termes d’erreurs quadratiques moyennes (MSE), de la racine des erreurs quadratiques moyennes (RMSE), et d’erreurs relatives par classe montre une variabilité notable entre les différentes classes. Nous avons testé la classe Scleractinia (une classe de notre jeu de données d’invertébrés marins) sur les sous-modèles existants dans notre modèle (modèles des classes Inachidae et Porifera), et nous observons les résultats suivants :

- Inachidae : Cette classe présente les meilleures performances globales, avec des valeurs de MSE, RMSE, et d’erreur relative les plus faibles de l’ordre de 0,002 pour la MSE. Cela suggère que le modèle a réussi à bien capturer les caractéristiques distinctives de cette classe, ce qui peut être attribué à des structures d’image similaires ou à une plus grande homogénéité au sein de la classe, raison pour laquelle le modèle attribue l’étiquette de classe existante.
- Porifera : Cette classe montre des performances légèrement inférieures à celles d’Inachidae. Les valeurs de MSE et RMSE sont modérément élevées(0,04), ce qui pourrait indiquer une certaine dissimilarité avec cette classe ou une variabilité plus grande entre les exemples de la classe.

Évolution des métriques d'apprentissage continu

Les Figures 3.13, 3.14, 3.15 présentent l'évolution des principales métriques d'apprentissage continu, à savoir l'AIA, le FM, et la Plasticité, au fur et à mesure de l'ajout de nouvelles classes, basées sur les trois types d'erreurs de reconstruction (MSE, RMSE et erreur relative) utilisés pour le jeu de données des invertébrés marins.

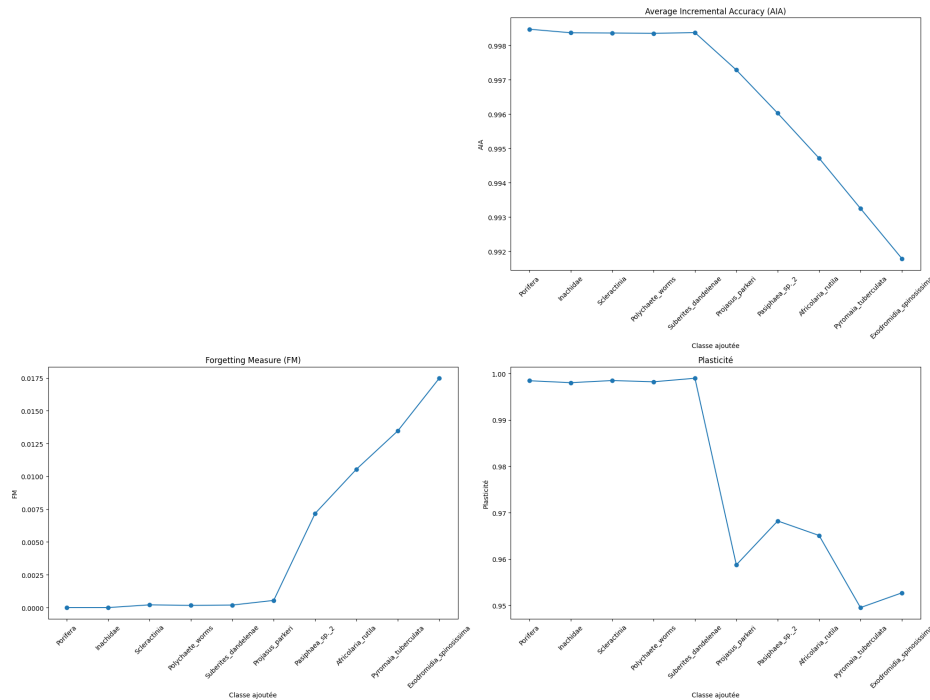


FIGURE 3.13 – Évolution des métriques d'apprentissage continu basée sur la MSE

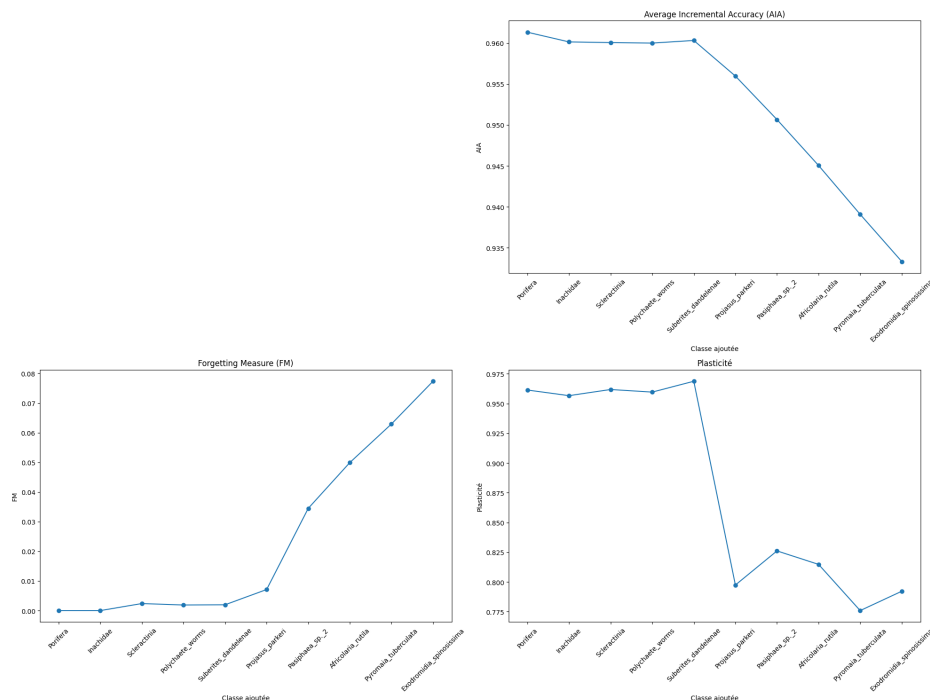


FIGURE 3.14 – Évolution des métriques d'apprentissage continu basée sur la RMSE

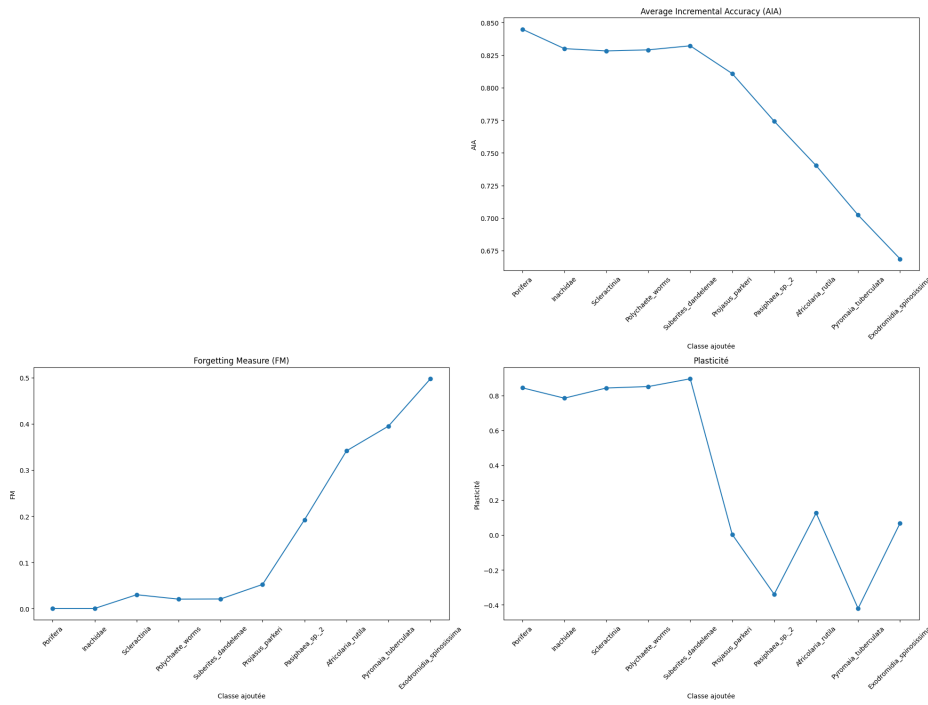


FIGURE 3.15 – Évolution des métriques d'apprentissage continu basée sur l'erreur relative

Les Figures 3.16, 3.17, 3.18 présentent l'évolution des principales métriques d'apprentissage continu, à savoir l'AIA, le FM, et la Plasticité, au fur et à mesure de l'ajout de nouvelles classes, basées sur les trois types d'erreurs de reconstruction (MSE, RMSE et erreur relative) utilisés pour le jeu de données d'oiseaux.

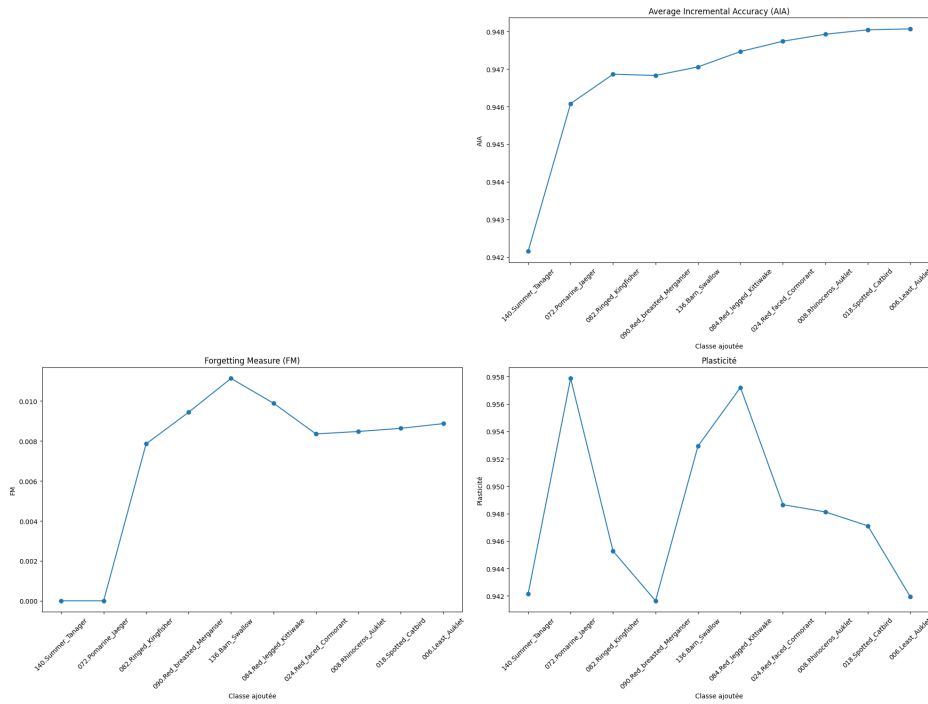


FIGURE 3.16 – Évolution des métriques d'apprentissage continu basée sur la MSE pour le jeu de données d'oiseaux

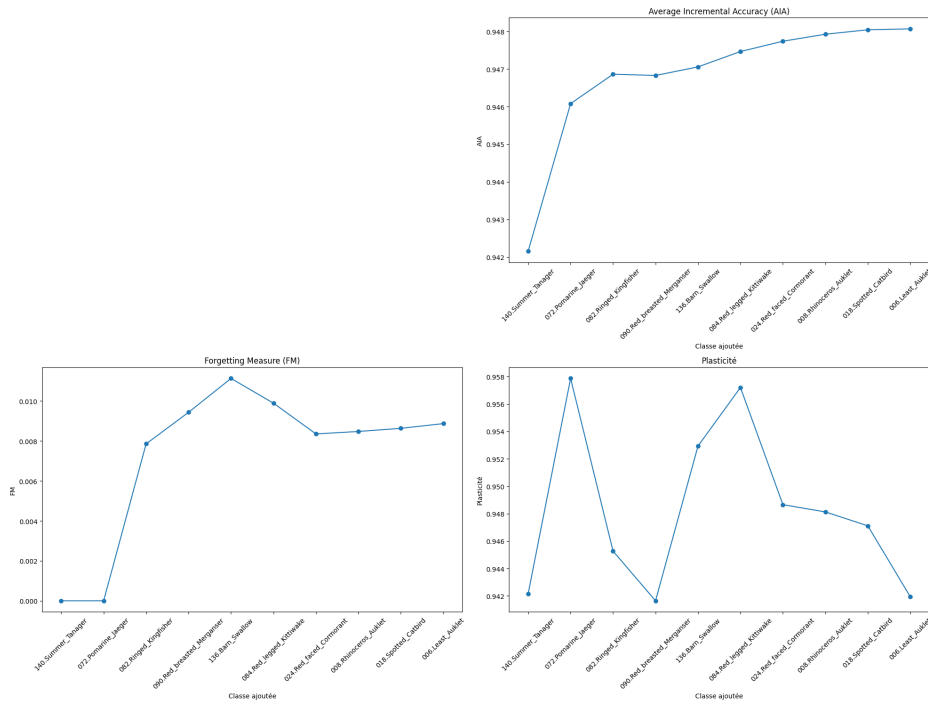


FIGURE 3.17 – Évolution des métriques d'apprentissage continu basée sur la RMSE pour le jeu de données d'oiseaux

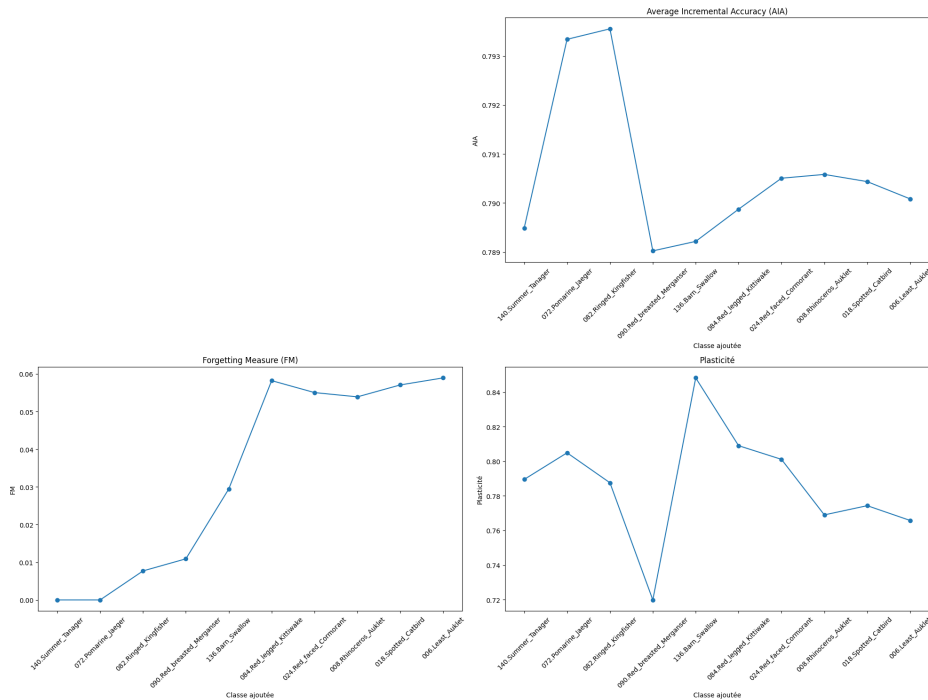


FIGURE 3.18 – Évolution des métriques d'apprentissage continu basée sur l'erreur relative pour le jeu de données d'oiseaux

Average Incremental Accuracy (AIA) La courbe de l'AIA montre une tendance globalement descendante à mesure que de nouvelles classes sont ajoutées pour tous les types d'erreur de reconstruction dans le jeu de données des invertébrés marins, tandis que pour le jeu de données

d’oiseaux, elle est croissante. Les cinq premières classes ajoutées, qui sont celles contenant le plus d’images dans le jeu de données, montrent une AIA proche de 90% de performance pour les erreurs de reconstruction MSE et RMSE, indiquant une interférence minimale entre les classes pour les deux jeux de données et proche de 80% pour l’erreur relative. Cependant, lorsque les cinq classes suivantes, contenant moins d’images, sont ajoutées, l’AIA commence à diminuer de manière significative. Cela suggère que l’introduction de classes avec moins d’exemples peut causer davantage d’interférence, probablement parce que le modèle doit ajuster ses représentations pour intégrer des classes moins bien représentées, ce qui perturbe davantage les classes précédemment apprises.

Il est aussi important de noter que cette dynamique d’interférence pourrait être influencée par l’ordre d’ajout des classes. Si l’ordre était modifié, l’allure de la courbe de l’AIA pourrait également changer, car les classes avec plus ou moins d’images auraient des impacts différents sur les représentations du modèle.

Forgetting Measure (FM) Le FM montre une tendance globalement ascendante à mesure que de nouvelles classes sont ajoutées pour tous les types d’erreur de reconstruction. Les cinq premières classes ajoutées, qui sont celles contenant le plus d’images dans les jeux de données, montrent une FM proche de 0,0008% de performance pour toutes les erreurs de reconstruction MSE et RMSE, indiquant que le modèle possède un faible taux d’oubli lors de l’ajout de ces classes, et proche de 0,5% pour l’erreur relative. Cependant, lorsque les cinq classes suivantes, contenant moins d’images, sont ajoutées, la FM commence à augmenter en performance, mais pas de manière significative. Cela suggère que l’introduction de classes avec moins d’exemples provoque davantage l’oubli des classes précédentes.

Ce résultat peut être attribué à la structure modulaire du modèle, qui est constituée de 10 sous-modules : 5 pour les classes avec le plus grand nombre d’images et 5 pour celles avec le moins d’images, permettant de préserver les connaissances spécifiques à chaque classe tout en continuant à apprendre de nouvelles classes.

Plasticité La courbe de plasticité montre des diminutions importantes pour tous les types d’erreurs de reconstruction, avec des points bas notables pour certaines classes spécifiques. Ces variations de plasticité suggèrent que certaines classes sont plus facilement apprises ou mieux intégrées dans le modèle que d’autres. Par exemple, un point bas de plasticité peut être associé à une classe partageant des caractéristiques intrinsèques similaires avec une classe déjà apprise, ce qui rend difficile son apprentissage.

On observe un point bas de plasticité particulièrement élevé pour la classe *Projasus parkeri*, suivi par un second point bas important pour la classe *Pyromaia tuberculata*. Cela pourrait s’expliquer par des similarités avec des classes déjà apprises, comme la classe *Inachidae* qui présente une légère diminution de performance de plasticité, ce qui pourrait indiquer une difficulté d’intégration due à des traits communs. Les figures 3.19 et 3.20 illustrent ces similarités visuelles entre les classes *Inachidae* et *Pyromaia tuberculata*.

En revanche, des augmentations de la performance de plasticité, comme celles observées pour les classes *Scleractinia* ou *Polychaete worms*, peuvent indiquer des facilités d’intégration pour ces classes, suggérant que leur apprentissage est aisé en raison de caractéristiques distinctes ou uniques par rapport aux classes précédemment apprises.



FIGURE 3.19 – Image de la classe *Inachidae*



FIGURE 3.20 – Image de la classe *Pyromaia tuberculata*

3.7 Bilan du chapitre

Dans ce chapitre, nous avons présenté le cadre expérimental de notre étude sur l'apprentissage continu appliqué à la classification. Après avoir décrit le jeu de données et expliqué les métriques d'évaluation choisies (Précision Moyenne Globale AIA, Oubli Moyen FM et Plasticité), nous avons détaillé notre protocole expérimental et analysé les résultats obtenus. Suite à l'intégration de 10 classes, le modèle a démontré des performances variées selon les jeux de données. Pour le jeu de données d'invertébrés marins, nous avons obtenu une Précision Moyenne Globale de 99,18%, un Oubli Moyen de 1,75% et une Plasticité de 95,27% basés sur l'erreur de reconstruction MSE ; une Précision Moyenne Globale de 93,33%, un Oubli Moyen de 7,73% et une Plasticité de 79,22% basés sur l'erreur de reconstruction RMSE ; et une Précision Moyenne Globale de 66,86%, un Oubli Moyen de 49,77% et une Plasticité de 6,8% basés sur l'erreur de reconstruction relative. Pour le jeu de données d'oiseaux, les résultats montrent une Précision Moyenne Globale de 99,71%, un Oubli Moyen de 0,08% et une Plasticité de 99,63% basés sur l'erreur de reconstruction MSE ; une Précision Moyenne Globale de 94,81%, un Oubli Moyen de 0,83% et une Plasticité de 94,19% basés sur l'erreur de reconstruction RMSE ; et une Précision Moyenne Globale de 79,01%, un Oubli Moyen de 5,8% et une Plasticité de 76,58% basés sur l'erreur de reconstruction relative. L'analyse des courbes d'AIA et de Plasticité a mis en lumière la capacité variable du modèle à intégrer de nouvelles connaissances.

CHAPITRE 4

CONCLUSION GÉNÉRALE

Dans ce travail, nous avons abordé le problème de la détection de nouvelles classes dans des environnements dynamiques, où les systèmes d'apprentissage automatique sont souvent confrontés à des éléments inconnus ou nouveaux. Pour répondre à cette problématique, nous avons proposé un modèle modulaire basé sur des autoencodeurs, capable de classifier les données tout en intégrant progressivement des classes non observées lors de l'entraînement initial. L'approche repose sur un modèle généraliste qui apprend les caractéristiques globales des classes connues, complété par des sous-modèles spécialisés, construits à partir d'autoencodeurs, chacun dédié à une classe spécifique. En exploitant les erreurs de reconstruction, le modèle identifie les nouvelles classes et, grâce à un mécanisme d'apprentissage continu, ajoute automatiquement des sous-modèles pour ces nouvelles catégories.

Les expérimentations menées sur des ensembles de données d'images d'espèces marines et d'oiseaux ont donné les résultats suivants : une Précision Moyenne Globale de 99,18 %, un Oubli Moyen de 1,75 % et une Plasticité de 95,27 % pour les invertébrés marins. Les résultats pour les oiseaux montrent également une Précision Moyenne Globale de 99,71 %, un Oubli Moyen de 0,08 % et une Plasticité de 99,63 %. Ces performances ouvrent la voie à de nouvelles applications dans des domaines nécessitant une adaptation continue, tels que la surveillance des écosystèmes marins ou d'autres environnements dynamiques.

Ce travail ouvre également des perspectives intéressantes pour l'amélioration future du modèle. Parmi les pistes de recherche envisagées, on peut citer l'amélioration de la plasticité du modèle, en particulier pour résoudre le problème du "fine-grained" classification. L'adaptation du modèle à des ensembles de données encore plus vastes et diversifiés permettrait également de tester sa robustesse face à une plus grande variété de cas "fine-grained". De plus, l'intégration d'autres techniques d'apprentissage continu, telles que la mémoire de rejeu, ainsi que l'application de cette architecture modulaire à d'autres domaines industriels ou scientifiques, constituent des pistes de recherche prometteuses.

BIBLIOGRAPHIE

- [1] I. Goodfellow, Y. Bengio, and A. Courville. *Deep Learning*. Adaptive Computation and Machine Learning series. MIT Press, 2016.
- [2] Michael Jordan and T.M. Mitchell. Machine learning : Trends, perspectives, and prospects. *Science (New York, N.Y.)*, 349 :255–60, 07 2015.
- [3] Tom M. Mitchell. *Machine Learning*. McGraw-Hill, 1997.
- [4] Arnav Varma, E. Arani, and Bahram Zonooz. Dynamically modular and sparse general continual learning. *ArXiv*, abs/2301.00620, 2023.
- [5] Berndt Brehmer and Dietrich Dörner. Experiments with computer-simulated microworlds : Escaping both the narrow straits of the laboratory and the deep blue sea of the field study. *Computers in Human Behavior*, 9(2) :171–184, 1993.
- [6] Jinwon An and Sungzoon Cho. Variational autoencoder based anomaly detection using reconstruction probability. *Special Lecture on IE*, 2(1), 2015.
- [7] Liyuan Wang, Xingxing Zhang, Hang Su, and Jun Zhu. A comprehensive survey of continual learning : Theory, method and application. *IEEE transactions on pattern analysis and machine intelligence*, PP, February 2024.
- [8] Matthias De Lange, Rahaf Aljundi, Marc Masana, Sarah Parisot, Xu Jia, Aleš Leonardis, Gregory Slabaugh, and Tinne Tuytelaars. A continual learning survey : Defying forgetting in classification tasks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 44(7) :3366–3385, 2022.
- [9] Saumya Sinha, Sophie Giffard-Roisin, Fatima Karbou, Michael Deschatres, Anna Karas, Nicolas Eckert, Cécile Coléou, and Claire Monteleoni. Variational autoencoder anomaly-detection of avalanche deposits in satellite sar imagery. In *Proceedings of the 10th International Conference on Climate Informatics*, CI2020, page 113–119, New York, NY, USA, 2021. Association for Computing Machinery.
- [10] Cyriana Roelofs, Marc-Alexander Lutz, S. Faulstich, and Stephan Vogt. Autoencoder-based anomaly root cause analysis for wind turbines. *Energy and AI*, 4 :100065, 03 2021.

- [11] Hongzuo Xu, Guansong Pang, Yijie Wang, and Yongjun Wang. Deep isolation forest for anomaly detection. *IEEE Transactions on Knowledge and Data Engineering*, 35(12) :12591–12604, 2023.
- [12] Keval Doshi and Yasin Yilmaz. Continual learning for anomaly detection in surveillance videos. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pages 1025–1034, 2020.
- [13] Laura Beggel, Michael Pfeiffer, and Bernd Bischl. Robust anomaly detection in images using adversarial autoencoders. In Ulf Brefeld, Elisa Fromont, Andreas Hotho, Arno Knobbe, Marloes Maathuis, and Céline Robardet, editors, *Machine Learning and Knowledge Discovery in Databases*, pages 206–222, Cham, 2020. Springer International Publishing.
- [14] David Lopez-Paz and Marc’Aurelio Ranzato. Gradient episodic memory for continual learning. In *Proceedings of the 31st International Conference on Neural Information Processing Systems, NIPS’17*, page 6470–6479, Red Hook, NY, USA, 2017. Curran Associates Inc.
- [15] Arslan Chaudhry, Marc’Aurelio Ranzato, Marcus Rohrbach, and Mohamed Elhoseiny. Efficient lifelong learning with a-gem. *ArXiv*, abs/1812.00420, 2018.
- [16] Rahaf Aljundi, Francesca Babiloni, Mohamed Elhoseiny, Marcus Rohrbach, and Tinne Tuytelaars. Memory aware synapses : Learning what (not) to forget. In *Computer Vision – ECCV 2018 : 15th European Conference, Munich, Germany, September 8–14, 2018, Proceedings, Part III*, page 144–161, Berlin, Heidelberg, 2018. Springer-Verlag.
- [17] Seyed Iman Mirzadeh, Arslan Chaudhry, Dong Yin, Timothy Nguyen, Razvan Pascanu, Dilan Gorur, and Mehrdad Farajtabar. Architecture matters in continual learning. *ArXiv*, abs/2202.00275, 2022.