



UNIVERSITÉ DE MONTPELLIER

Classification d'images de pièges-photos par apprentissage profond

CONFIDENTIEL

ANAS ZAKROUM

Encadrants:

M. Y. LE BRAS - *Biotope*
M. B. CHARLIER - *IMAG*

Jurys:

MME. E. BRUNEL-PICCININI - *IMAG*
M. G. DUCHARME - *IMAG*
M. G. DIDIER - *IMAG*

Rapport de stage soumis en vue
de satisfaire aux exigences
du diplôme de Master
Mention - Mathématiques

Année académique 2021–2022

STAGE EFFECTUÉ À BIOTOPE



Remerciements

Ce rapport n'aurait pu être écrit sans l'aide de nombreuses personnes.

Tout d'abord, j'aimerais adresser mes remerciements à ma famille pour leurs encouragements inconditionnels.

Mes remerciements sont adressés à M. Yves Le Bras pour son tutorat et pour la relecture de ce rapport et plus globalement aux gens de Biotope, pour la convivialité, les pauses café et les sorties Baie d'Ha.

J'aimerais adresser mes remerciements aux jurys pour la lecture de ce rapport et à M. Benjamin Charlier pour avoir accepté mon encadrement.

Merci à toi Kamal et à ta trottinette électrique. Solène, Thomas et Max, je vous remercie pour toutes les discussions (sérieuses) et pour votre chaleureuse bienveillance.

L'équipe de Bir Rami Est, à jamais dans mon cœur...

Gwenaëlle, je te remercie pour ta présence, pour toutes nos folies et fou-rires et de ta compagnie le long de cette aventure. Tu es juste trop cool @->-

Anas, Août 2022.

Resumé

Le recours au photo-piégeage pour l'échantillonnage de la faune est de plus en plus utilisé dans les études de terrain. La démocratisation des ressources de calcul et du partage public de jeux de données massifs à des fins scientifiques a d'autre part contribué au développement de méthodes de vision par ordinateur avancées. A l'interface de ces deux domaines, un attrait vers l'automatisation du traitement des données écologiques issues du photo-piégeage à l'aide de la vision par ordinateur est né. Les champs d'applications sont nombreux, que ce soit dans la reconnaissance de formes, l'étude du comportement animal, l'estimation de l'abondance la richesse des espèces, la documentation de nouvelles espèces, etc.

Le sujet de ce stage porte sur le développement des méthodes d'apprentissage profond pour la classification d'images d'espèces issues de photo-piégeage. Nous explorons cette problématique depuis la récolte des données jusqu'à la détection et la classification des espèces observées en passant par la préparation des données, la recherche et la construction d'un modèle adapté à la tâche de classification, son évaluation et finissons sur une conclusion et une ouverture sur de potentielles perspectives en lien avec la problématique.

Introduction

Les outils de photo-piégeage sont de plus en plus utilisés pour le suivi et les inventaires de la faune. L'avantage de l'utilisation des pièges-photos par rapport à d'autres méthodes d'échantillonnage telles que le radio-pistage, le piégeage ou l'observation directe réside dans la facilité logistique qu'elles procurent et dans la qualité des données collectées sans que l'animal soit marqué ni que les conducteurs de l'étude soient présents sur le terrain. Le recours aux pièges photographiques permet également de réduire le coût, la main-d'oeuvre et la logistique liée à l'étude.

Les pièges photographiques génèrent un nombre important d'images qui souvent doivent être inspectées visuellement avant de fournir des informations exploitable pour étudier la biodiversité. Ce processus manuel nécessitant un effort humain considérable se retrouve être un facteur limitant dans le recours à cette méthode d'échantillonnage. Bien que le photo-piégeage ait grandement amélioré l'échantillonnage de la faune, les avantages de la collecte continue des données sur le terrain peuvent rapidement se transformer en un fardeau lorsqu'il s'agit de transformer les données récoltées en de l'information exploitable sur la présence, l'abondance ou le comportement des animaux (plus de 7 millions d'images pour la seule initiative Snapshot Serengeti).

Le recours à la science participative¹ peut avoir lieu dans de tels cas mais se confronte à des limitations liées au temps long de traitement et de protection de la vie privée.

Quand les études ont lieu dans les espaces publics comme des réserves naturelles, les caméras peuvent être enclenchées par les passants. Les données doivent donc passer par un premier tri éliminant les images des passants avant d'être mise à disposition du public volontaire.

L'objectif de ce stage est la construction d'un modèle de classification des espèces présentes dans la zone d'étude, ce qui permettrait potentiellement de réduire le temps et l'effort consacrés à l'annotation des images. Le modèle construit peut être utilisé pour classifier automatiquement les images ou pour fournir des suggestions au personnel de l'entreprise responsable de l'annotation.

Des images annotées issues d'une étude réalisée par l'entreprise sont utilisées pour l'entraînement de réseaux de neurones convolutifs pour la détection et la classification des espèces observées dans la zone d'étude. Le nombre d'images nécessitant une inspection manuelle peut être considérablement réduit en entraînant un modèle pour la détection d'objets. Il servira à différencier les images vides des images non vides (animal, humain, véhicule). De plus un second modèle sera entraîné pour l'identification des espèces dans les images.

¹ Formes de production de connaissances scientifiques auxquelles des acteurs non-scientifiques-professionnels participent de façon active et délibérée. (Houllier, F., 2020)

Le **Chapitre 1** présente une introduction au piégeage photo dans un contexte de l'inventaire et le suivi de la faune. Les propriétés des caméras de piégeage sont expliquées et un aperçu non exhaustif des applications du photo-piégeage est donné. Le chapitre évoque également des exemples de projets de science participative de longue envergure concernant l'annotation de pièges photographiques et qui ont entre autre permis le développement de modèles de détections d'animaux performants.

Dans le **Chapitre 2**, nous commençons par une introduction générale sur la classification d'images et poursuivons sur un bref aperçu de la théorie mathématique de l'apprentissage. Nous présentons par la suite un aperçu global sur l'apprentissage profond et ses principes. Nous énonçons les propriétés des réseaux de neurones et en particulier les réseaux de neurones convolutifs utilisés dans le traitement d'images.

Le **Chapitre 3** fournit une description des données à disposition. La structure des images des pièges photographiques utilisées ainsi que leur annotations sont présentés. La taxonomie des espèces présentes sur le site de l'étude est explicitée et un comptage des espèces observées est fourni. Le chapitre aborde également à partir des données à disposition, les obstacles reliés à la classification d'images de pièges photographiques et traite les étapes suivies dans l'exploration et la pré-traitement des données.

Dans le **Chapitre 4** nous construisons une méthodologie pour répondre aux différentes problématiques soulevées durant le stage à savoir, la sélection d'un modèle qui permet d'extraire les caractéristiques contenues dans les images, la construction et l'entraînement d'un réseau de neurone adapté à la tâche de classification et de son calibrage.

Le **Chapitre 5** présente les résultats trouvés en suivant la méthodologie proposée. Une exposition des métriques nécessaires à l'évaluation des performances d'un modèle de classification est donnée et ensuite reportée sur le réseau de neurones utilisé.

Enfin, nous finirons sur une conclusion et discuterons de l'interprétation des métriques de classification dans un contexte de conservation des espèces.

Table des matières

Remerciements	i
Resumé	iii
Introduction	v
Liste des tableaux	ix
Table des figures	xi
1 Le photo-piégeage pour le suivi de la faune	1
1.1 Aux origines du photo-piégeage	1
1.2 Les détecteurs à infrarouge passif	1
1.3 Fonctionnement des pièges photographiques	2
1.4 Propriétés des pièges photographiques	3
1.5 Quelques applications du photo-piégeage	5
1.6 Considérations méthodologiques	7
1.7 Les sciences participatives au service du photo-piégeage	7
1.8 Considérations de vie privée et protection des espèces	8
2 L'apprentissage profond et la classification d'images	11
2.1 La classification d'images	11
2.2 Un formalisme de l'apprentissage	12
2.3 Principes de l'apprentissage profond	15
2.3.1 Quelques notes historiques	15
2.3.2 Les réseaux de neurones	16
2.3.3 L'apprentissage par gradient	20
2.4 Réseaux convolutifs, représentations latentes et classification	22
2.4.1 Les couches convolutives	22
2.4.2 Adaptation d'un réseau au problème de classification	25
2.4.3 Exemple d'architecture d'un réseau convolutif	26
3 Description des données et contexte	29
3.1 Zone de l'étude et méthodologie d'échantillonnage	29
Configuration du déploiement des pièges photographiques	30
3.2 Aperçu des données récoltées	31
3.3 Les défis de la classification d'images de photo-piégeage	32
3.3.1 Obstacles circonstanciels	32
3.3.2 Le déséquilibre des classes	34

3.4	Préparation des données pour la classification	36
3.4.1	Révision et nettoyage	36
3.4.2	Extraction des régions d'intérêt	37
4	Méthodologie	43
4.1	Répartition des données pour l'entraînement et l'évaluation du modèle	43
4.2	Analyse comparative pour la sélection de modèle	44
4.3	Construction du réseau de neurones et apprentissage par transfert	45
4.4	Calibrage du modèle	46
5	Résultats et discussion	47
5.1	Données d'apprentissage, augmentation des images, et données d'évaluation	48
5.2	Entraînement du modèle et choix des hyper-paramètres	48
5.3	Métriques d'évaluation	49
5.4	Résultats de sélection de modèles (benchmarking)	50
5.5	Évaluation du modèle sélectionné : EfficientNetV2S	50
5.5.1	Adaptation des paramètres (finetuning)	52
6	Livrables pour l'entreprise	53
	Conclusions	55

Liste des tableaux

3.1	Inventaire des espèces observées, toutes campagnes confondues.	31
3.2	Matrice de confusion des résultats du MegaDetector.	39
5.1	Résultats de l'analyse comparative sur les données de validation	50
5.2	Performances du classifieur en utilisant les représentations latentes inférées par EfficientNetV2S. Les résultats sont rapportés sur les données d'évaluation (testing set).	51
5.3	Performance du classifieur entraîné séparément sur les jeux de données original, augmenté et équilibré. (P) Précision. (R) Rappel.	51
6.1	Matrice de confusion - modèle entraîné sur les données originales	62
6.2	Matrice de confusion - modèle entraîné sur les données augmentées	63
6.3	Matrice de confusion - modèle entraîné sur les données équilibrées	63

Table des figures

1.1	Caméra Bushnell. Source de l'image www.bushnell.com . . .	2
1.2	Bandes de détections d'un piège photographique Reconyx. Image de	2
1.3	Gauche. Image en mode jour. Droite. Image en mode nuit.	3
1.4	Facteurs impactant les probabilités de détection. Image issue de [Bur+15]	4
1.5	Cliché d'un piège photographique montrant le Cricetomys emini avec des graines de Carapa grandiflora. Image issue de [Nyi+11].	5
1.7	Images de pièges photographiques de léopards mélaniques en (a) mode nuit et (b) en mode infrarouge. Image issue de [Hed+15].	6
1.6	Modes de locomotion de Gibbons de Hainan sur un pont de canopée. (a,b) En escaladant. (c) En marchant. (d) En brachiation. Image de [Cha+20]	6
2.1	Illustration du principe de minimisation du risque structurel. Les courbes de trois erreurs sont illustrées en fonction de la "mesure de capacité" i.e la taille de l'ensemble des hy- pothèses. Plus cette taille grandit, l'erreur d'entraînement décroît tandis que le terme de complexité grandit. Le principe de minimisation du risque structurel (en rouge) sélectionne l'hypothèse qui minimise la borne sur l'erreur de généralisation.	15
2.2	Illustration du perceptron. s^i est la i^{eme} coordonnée du vec- teur d'entrée s	17
2.3	Exemple de deux classes linéairement séparables.	18
2.4	Gauche : Illustration d'une architecture d'un perceptron multi-couches. Cette architecture contient $L = 4$ couches définissant un ensemble de fonctions $f(x; \theta)$ de dimension d'entrée $n_0 = 4$ et de dimension de sortie $n_4 = 1$. Les couches cachées possèdent chacune $n_1, n_2, n_3 = 5$ neurones. Droite : Structure détaillée de chaque neurone qui <i>(i)</i> agrège le biais au signaux pondérés pour produire la <i>pré-activation</i> , <i>(ii)</i> génère l'activation, et <i>(iii)</i> multiplie l'activation par les poids de la couche suivante. Image de [RYH22]	18
2.5	Illustration d'un perceptron multi-classes. \mathbf{x} représente le vecteur d'entrée. \mathbf{W} représente les poids du perceptron sous forme matricielle. Image de	19

2.6	Exemples de dynamiques de la descente du gradient (gauche) et de la descente du gradient stochastique (droite) pour une fonction à deux extrema. Image de [Ber+21].	21
2.8	Exemple illustratif d'une hiérarchie de caractéristiques dont la combinaison simultanée permet de former un concept particulier. Image de [Cho21].	23
2.7	Illustration d'un opérateur de convolution. L'image d'origine est convoluée par un filtre de convolution de taille 3×3 . Image de [FD17].	23
2.9	Résultats des opérateurs de sous-échantillonnage de (a) La donnée d'entrée, par (b) valeur maximale. (c) la moyenne. (d) la somme. Image de [FD17].	24
2.10	Illustration du champ de vision d'un réseau convolutif. La partie visible de l'image est de plus en plus grande. Image de [FD17].	25
2.11	Architecture du réseau VGG16.	26

29figure.caption.25

3.2	Configuration de la grille d'échantillonnage des pièges photographiques du	30
3.3	Quelques exemples d'espèces observées par les pièges photographiques.	32
(a)	Athérure	32
(b)	Elephant	32
(c)	Mandrills	32
(d)	Oiseau	32
(e)	Chimpanzés	32
(f)	Potamochère	32
(g)	Céphalophe	32
(h)	Pintade	32
3.4	Exemples de scénarios possibles de classification.	33
(a)	Mauvaise illumination	33
(b)	Occlusion due au contexte	33
(c)	Sous-exposition	33
(d)	Floutage	33
(e)	Camouflage	33
(f)	Animal trop proche	33
(g)	Limitation matérielle	33
(h)	Partie d'animal	33
(i)	Animal en mouvement	33
(j)	Idéal	33
(k)	Auto-occlusion	33
3.5	Exemple de sur-échantillonnage en appliquant une combinaison aléatoire de transformations élémentaires.	35
(a)	Image originale	35
(b)	Rotation et zoom avant	35
(c)	Translation, zoom arrière et rotation	35
(d)	Effet miroir et rotation	35

3.6	Schéma illustratif des étapes de la phase de pré-traitement des données. La spirale rouge indique la nécessité d'une inspection manuelle.	36
3.7	Schéma illustratif de la méthode utilisée pour l'annotation automatique des images noires.	37
3.8	Illustration des caractéristiques utilisées dans la discrimination entre un Husky et un loup. (a) Image d'un Husky classé comme loup. (b) Caractéristiques discriminatives de la décision.	38
3.9	Exemples de sorties du MegaDetector. La classe des objets est codée sous forme de couleur. (Rouge) Animal. (Bleu) Humain. (Vert) Véhicule.	38
	(a) caption1	38
	(b) caption2	38
	(a) Échec de détection (Céphalophe)	39
	(b) Une bonne détection et un faux positif (Oiseau, branche)	39
	(c) Un cas difficile (Athérure)	40
	(d) Cas difficile avec occlusion de l'animal (Athérure)	40
	(e) Deux catégories pour le même objet (Chimpanzé)	40
	(f) Difficulté à déterminer l'animal à cause du contexte (Hocheur)	40
	(g) Difficulté à déterminer l'animal (Céphalophe)	40
	(h) Deux détections pour le même animal (Civette)	40
3.8	Exemples de scénarios de détections du MegaDetector.	41
	(i) Une bonne détection et un faux positif (Chimpanzé, feuille)	41
	(j) Un cas difficile (Céphalophe)	41
3.9	Distribution des niveaux de confiances en fonction des trois catégories de détection pour la troisième campagne de photo-piégeage.	41
4.1	Schéma explicatif de la phase de répartition des données en bases d'entraînement, d'évaluation et de test	43
4.2	Schéma illustratif de la phase de sélection de modèle.	44
4.3	Construction de l'architecture du réseau de neurones adapté à notre jeu de données.	45
5.1	Fréquences des classes de la troisième campagne de photo-piégeage	47

CHAPITRE UN

Le photo-piégeage pour le suivi de la faune

Les pièges photographiques sont de plus en plus utilisés dans l'étude de la faune. Cela ne vient pas sans considérations techniques et méthodologiques. Nous expliquons dans ce chapitre les caractéristiques générales des pièges photos et abordons leur points forts ainsi que leurs limitations. Ensuite, nous donnons un aperçu non-exhaustif des applications du photo-piégeage. La science participative est ensuite abordée comme un outil permettant aux citoyens d'apporter leur assistance dans la conduite d'études de photo-piégeage de grande échelle.

1.1 Aux origines du photo-piégeage

George Shiras a confectionné en 1890, un outil mécanique permettant la capture de clichés via enclenchement passif à l'aide d'un fil-piège et un système flash [GSV16]. Les études de faune à l'aide du photo-piégeage sont présentes dans la littérature scientifique depuis les débuts du 20^{eme} siècle [RC08]. Cependant, à cause de limitations techniques et à ses coûts élevés, l'utilisation des pièges photos ne s'est popularisée qu'à la fin du même siècle lorsque le développement commercial des systèmes de caméras a substantiellement progressé [Swa+04]. De nos jours, des systèmes plus avancés sont disponibles utilisant par exemple des technologies infrarouge pour l'enclenchement des caméras. Les pièges photos modernes sont portables, résistant aux conditions météorologiques difficiles, de petites tailles et de coûts beaucoup plus abordables [GSV16]. Par conséquent, ce sont devenus des outils de référence pour le suivi et l'inventaire de la faune [RC08].

[GSV16] Gomez, Salazar et Vargas, « Towards automatic wild animal monitoring : Identification of animal species in camera-trap images using very deep convolutional neural networks »

[RC08] Rowcliffe et Carbone, « Surveys using camera traps : are we looking to a brighter future ? »

[Swa+04] Swann et al., « Infrared-triggered cameras for detecting wildlife : an evaluation and review »

1.2 Les détecteurs à infrarouge passif

Les détecteurs à infrarouge passifs (équipé de capteurs PIR pour "Passive Infra Red") sont devenus les capteurs les plus utilisés dans les pièges photographiques [SKP11]. Cette technologie permet de détecter des gradients de température entre l'arrière plan et la température à la surface des objets. Le déclenchement du piége a lieu lorsque un brusque changement de la température se produit dans le champ du capteur PIR [Wel+16].

[SKP11] Swann, Kawanishi et Palmer, « Evaluating types and features of camera traps in ecological studies : a guide for researchers »

[Wel+16] Welbourne et al., « How do passive infrared triggered camera traps operate and why does it matter ? Breaking down common misconceptions »

Ce type de technologie est sujet à deux types d'erreurs : les faux déclenchements (faux positifs) et l'échec de détection de l'animal (faux négatifs)



Figure 1.1 – Caméra Bushnell. Source de l'image www.bushnell.com

[Swa+04] Swann et al., « Infrared-triggered cameras for detecting wildlife : an evaluation and review »

[Swi+14] Swinnen et al., « A novel method to reduce time investment when processing videos from camera trap studies »

[Wel+16] Welbourne et al., « How do passive infrared triggered camera traps operate and why does it matter ? Breaking down common misconceptions »

[Swa+04]. Le rayonnement thermique pouvant provenir des objets non-cibles et du capteur lui même peuvent résulter en un bruitage de fond. Un seuil est alors mis en place afin de prévenir les occurrences de faux déclenchements [Swa+04]. Néanmoins, en pratique cela ne suffit pas à empêcher l'occurrence de faux déclenchements. Des causes possibles sont le vent pouvant faire bouger la végétation dans le champ de la caméra ou faire bouger le support de la caméra (généralement attachée à un arbre). En pratique, il est important d'optimiser le positionnement du piège sur le terrain pour limiter la fréquence des faux positifs, par exemple en choisissant un lieu où la végétation est dégagé devant le capteur, où la situation est ombragée et orientant le piège vers le nord.

D'autres causes peuvent être l'eau courante ou la chaleur émanant localement dans le champ de détection du capteur, ...[Swa+04].

Si un animal se trouve à l'intérieur du champ de détection mais en dehors de la portée de la caméra, ou, si la caméra ne se déclenche qu'après le passage d'un animal, cela provoque un faux enclenchement [Swi+14].

Par ailleurs, la configuration thermique de l'arrière plan est souvent hétérogène car l'arrière plan possède plusieurs objets de propriétés thermiques différentes comme de l'herbe, des arbres, ... Ces objets disposent de différentes températures à la surface et l'hétérogénéité thermique résultante peut être à l'origine de faux enclenchements [Wel+16].

D'autre part, les erreurs du deuxième type : l'échec de la capture de l'animal sont plus difficiles à connaître. Contrairement aux faux enclenchements qui peuvent être observés lors de l'inspection des images, il n'existe pas d'explications claires en ce qui concerne l'échec de capture de la caméra [Swa+04].

1.3 Fonctionnement des pièges photographiques

Les images qui seront utilisées pour la classification sont issues de pièges photographiques **Bushnell** équipés d'un capteur PIR. Le module infrarouge est aligné avec la lentille de la caméra de manière à détecter seulement les objets dans le champ de vision de la caméra.

Les modules de détection des pièges photographiques utilisent en général des bandes de détections horizontales. Les bandes de détections sont divisées en plusieurs zones. Deux choses devront se produire pour déclencher le piège.



Figure 1.2 : Bandes de détections d'un piège photographique Reconyx. Image de

1.4. PROPRIÉTÉS DES PIÈGES PHOTOGRAPHIQUES

Premièrement, un objet dont la température est différente de la température de l'arrière plan doit se trouver dans une des bandes de détections. Deuxièmement, l'objet doit entrer ou sortir d'une des zones de détection. Ainsi, dans la Figure 1.2 les cerfs placés à droite et au milieu pourront enclencher la caméra, mais pas le cerf tout à gauche car il se trouve dans aucune zone de détection.

Les pièges photographiques fonctionnent également la nuit. Ils sont équipés d'une technologie de vision nocturne et le passage entre les deux modes se fait de manière automatique. Les images prises durant la journée peuvent être en couleur ou en noir et blanc quand l'exposition à la lumière est faible. Les clichés pris pendant la nuit sont monochromes.



Figure 1.3 – Gauche. Image en mode jour. Droite. Image en mode nuit.

1.4 Propriétés des pièges photographiques

Avantages

L'avantage que présentent les pièges photographiques par rapport aux autres méthodes d'échantillonnage de la faune comme l'observation directe, le piégeage et le pistage réside dans la passivité et la précision des données collectées sans avoir recours au marquage de l'animal et sans que la présence du conducteur d'études soit nécessaire. Seulement des passages sont organisés pour récupérer les composants de mémoire ou pour changer les piles. Les pièges photographiques peuvent également permettre la collecte d'informations sur des espèces très cryptiques et qui se trouvent sur des terrains difficiles [Row+08]. Un autre avantage est que les données brutes collectées peuvent être partagées avec d'autres chercheurs contrairement aux données de piégeage classique ou d'observations directes [SKP11].

[Row+08] Rowcliffe et al., « Estimating animal density using camera traps without the need for individual recognition »

[SKP11] Swann, Kawanishi et Palmer, « Evaluating types and features of camera traps in ecological studies : a guide for researchers »

[RC08] Rowcliffe et Carbone, « Surveys using camera traps : are we looking to a brighter future ? »

[Bur+15] Burton et al., « Wildlife camera trapping : a review and recommendations for linking surveys to ecological processes »

Il est également possible que les clichés obtenus lors d'une étude peuvent être réutilisés dans une autre étude bien que les deux études ne ciblent les mêmes espèces [RC08]. Cependant, comme l'indiquent de nombreux auteurs (par exemple [RC08]), la croissance considérable des études par photo-piégeage est peu coordonnée et ne bénéficie pas d'une intégration protocolaire commune. Par conséquent, pour exploiter pleinement leur potentiel, il est nécessaire de se diriger vers un consensus commun. [Bur+15], parmi d'autres, appelle à une communication plus approfondie sur des précisions protocolaires et méthodologiques.

Inconvénients

[SKP11] Swann, Kawanishi et Palmer, « Evaluating types and features of camera traps in ecological studies : a guide for researchers »

[FH12] Foster et Harmsen, « A critique of density estimation from camera-trap data »

[Sil+04] Silver et al., « The use of camera traps for estimating jaguar *Panthera onca* abundance and density using capture/recapture analysis »

Contrairement aux avantages du photo-piégeage qui bénéficient d'une bonne couverture dans la littérature existante, les inconvénients ont reçu moins d'attention de la part de la communauté scientifique [SKP11]. D'abord, nous avons mentionné auparavant les faux déclenchements ou l'échec de la capture des animaux. Ceux-ci peuvent être plus ou moins fréquents selon les endroits où sont montés les pièges photographiques comme nous avons pu l'observer dans l'inspection de nos propres données.

La majorité des difficultés rencontrées apparaissent sur le terrain. Un des problèmes les plus récurrents sont les pannes d'ordre mécanique ou numérique des équipements. Ceci est davantage problématique lorsque les caméras sont installées dans des zones éloignées où la panne n'est observée que quelques mois après son occurrence [SKP11].

D'autre part, les pièges-photos positionnés dans des espaces publics peuvent faire l'objet de vol ou de vandalisme [FH12 ; Sil+04]. Des protections métalliques peuvent éventuellement être utilisées afin de prévenir des dommages causés par les animaux (éléphants, chimpanzés, ...).

Comme pour de nombreuses méthodes d'échantillonnage, le photo-piégeage fait aussi face aux erreurs d'échantillonnage comme les détections imparfaites où l'individu ou l'espèce connu(e) pour être présent dans la zone d'étude n'est pas détecté [Bur+15]. Les pièges photographiques ciblant des objets en mouvement, ils doivent faire face à ces obstacles de nature spatiale : les animaux qui passent par les zones de détection de la caméra peuvent ne pas être détectés et les animaux occupant une petite partie du champ de vision de la caméra peuvent ne jamais passer par les zones de détection. [Bur+15]. La probabilité de détection est impactée par plusieurs facteurs opérants à différentes échelles comme illustré dans [Bur+15] (voir Figure 1.4). Dans [Bur+15], les auteurs énoncent que la probabilité de détection peut être impactée par plusieurs facteurs opérants à différentes échelles comme illustré dans la Figure 1.4.

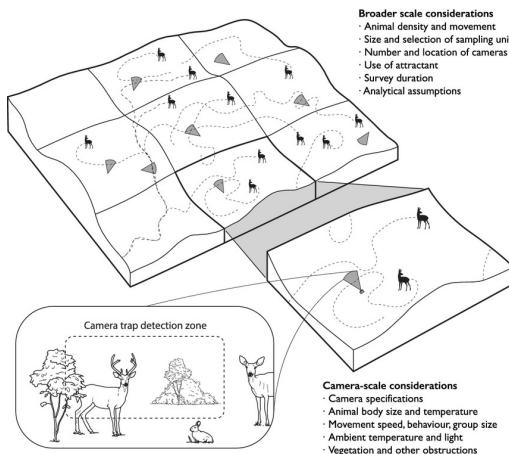


Figure 1.4 – Facteurs impactant les probabilités de détection. Image issue de [Bur+15]

1.5 Quelques applications du photo-piégeage

Le photo-piégeage possède un large éventail d'applications. L'une des applications les plus simples consiste à confirmer la présence d'une certaine espèce animale. L'absence d'un animal ne peut pas être confirmée à l'aide de pièges photographiques. Quand un animal n'a pas été observé sur les images de l'étude, cela suggère seulement la faible vraisemblance de sa présence mais ne signifie pas nécessairement que l'animal est absent de la zone de l'étude.

Dans, [Hen+10], les pièges photographiques sont utilisés pour rechercher des lions. L'espèce de lion qui fait l'objet de l'étude (*Panthera leo*) est classée dans la liste rouge de l'IUCN des espèces menacées au statut de vulnérable. La présence de l'espèce a été confirmée dans deux sites de conservation des lions en Afrique de l'Ouest seulement, contrairement à la zone d'étude qui incluait également l'Afrique Centrale. Les auteurs concluent alors que leurs résultats suggèrent la possibilité de non existence de l'espèce dans les sites de l'Afrique Centrale.

D'autres applications du photo-piégeage comptent l'inventaire, l'estimation de l'abondance, la découverte de nouvelles espèces, l'étude des dynamiques des populations [Hen+10 ; Tob+15] ou encore les interactions plantes-animal. Les pièges photographiques sont révolutionnaires quant à ce dernier, car ils permettent l'identification d'espèces nocturnes et timides qui sont difficilement observables. Ceci est illustré dans [Nyi+11].



Figure 1.5 – Cliché d'un piège photographique montrant le Cricetomys emini avec des graines de *Carapa grandiflora*. Image issue de [Nyi+11].

Les auteurs ont découvert qu'une espèce de rongeur, le Cricetomys emini (rat géant de forêt), était responsable de la dispersion secondaire des grosses graines dans une forêt afro-tropicale. Voir Figure 1.5.

Tobler et al. [Tob+15] ont développé un modèle "d'occupancy" (occupancy pouvant se traduire par probabilité de présence) multi-espèce et multi-saison afin d'améliorer les estimations de la richesse d'espèces et d'occupation basée sur le nombre total de détections par espèce. Le modèle peut être utilisé pour le suivi de groupes de mammifères au cours du temps et étudier les évolutions de la distribution et de la composition des communautés en prenant compte différents facteurs (anthropiques ou naturels). Le photo-piégeage a également été utilisé pour la découverte de nouvelles espèces. En 2005, une nouvelle espèce, le Musaraigne à trompe (un petit mammifère se nourrissant d'insectes) a été décrit [Rov+08]. Sa découverte a été située du côté nord

[Hen+10] Henschel et al., « Lion status updates from five range countries in West and Central Africa »

[Tob+15] Tobler et al., « Spatiotemporal hierarchical modelling of species richness and occupancy using camera trap data »

[Nyi+11] Nyiramana et al., « Evidence for seed dispersal by rodents in tropical montane forest in Africa »

[Rov+08] Rovero et al., « A new species of giant sengi or elephant-shrew (genus *Rhynchocyon*) highlights the exceptional biodiversity of the Udzungwa Mountains of Tanzania »

CHAPITRE 1. LE PHOTO-PIÉGEAGE POUR LE SUIVI DE LA FAUNE

[LZF15] Li, Zhao et Fan, « White-cheeked macaque (*Macaca leucogenys*) : A new macaque species from Medog, southeastern Tibet »

[Swi+14] Swinnen et al., « A novel method to reduce time investment when processing videos from camera trap studies »

[Rov+08] Rovero et al., « A new species of giant sengi or elephant-shrew (genus *Rhynchocyon*) highlights the exceptional biodiversity of the Udzungwa Mountains of Tanzania »

[Cha+20] Chan et al., « First use of artificial canopy bridge by the world's most critically endangered primate the Hainan gibbon *Nomascus hainanus* »

des montagnes d'Udzungwa en Tanzanie, basée sur les images de pièges photographiques. Les auteurs de [LZF15] décrivent une nouvelle espèce de macaques dans le Sud du Tibet découverte à l'aide du photo-piégeage. Au sud-est du Tibet, les macaques vivent dans des forêts tropicales ou sous-tropicales denses dans des terrains montagneux. Comme il était difficile de suivre ces groupes non habitués à l'homme, des pièges-photos ont été alors utilisés pour étudier leurs caractéristiques ou leurs comportement.

[Swi+14] énonce l'étude de l'exclusion compétitive, des structures des populations, de la différenciation de niche et du comportement de préation comme des applications possibles. Le suivi des populations sur le temps, l'analyse des associations d'habitats, l'estimation de survie, la reproduction, le régime et le mode alimentaire sont également mentionnés dans [Rov+08]. D'autres études ont démontré l'efficacité des solutions de connectivité artificielles par la faune. Les pièges photos sont installés de sorte à identifier les animaux utilisant ces aménagements artificiels. Un exemple est l'observation dans [Cha+20] de l'un des primates les plus menacés parcourant un pont artificiel en canopée à l'aide de pièges photographiques . La Figure 1.6 ci-dessus illustre des Gibbons de Hainan entrain d'utiliser le pont artificiel.



Figure 1.6 – Modes de locomotion de Gibbons de Hainan sur un pont de canopée. (a,b)
En escaladant. (c) En marchant. (d) En brachiation. Image de [Cha+20]



Figure 1.7 : Images de pièges photographiques de léopards mélaniques en (a) mode nuit et (b) en mode infrarouge. Image issue de [Hed+15].

L'usage du photo-piégeage est également utilisé dans la documentation d'espèces rares ou d'animaux mélaniques. Par exemple, dans [Hed+15], les auteurs ont proposé le forçage du mode nuit de la caméra de manière à pouvoir observer les taches de léopards mélaniques en Malaisie.

Comme la quasi totalité des léopards en Malaisie sont mélaniques, cette méthode ouvre la possibilité d'une identification individuelle en utilisant des pièges photographiques standards ce qui peut être utilisé pour l'estimation de densité ou pour une modélisation par capture-marquage-recapture.

1.6 Considérations méthodologiques

Les champs d'application du photo-piégeage sont donc nombreux. Cependant, le photo-piégeage fait l'objet de considérations méthodologiques. Comme énoncé dans [Swa+04], il existe un lien entre le déclenchement des pièges photographiques et la masse corporelle de l'espèce. Les auteurs concluent que la fréquence de piégeage ne peut pas être un indicateur robuste pour la comparaison de l'abondance relative des espèces.

D'autres facteurs peuvent également impacter la fréquence de piégeage comme la vitesse et le taux de déplacement ou l'abondance [RC08].

D'autre part, les modèles de capture-marquage-recapture peuvent être utilisés sur les animaux identifiables à leur pelage comme certains grands félins, ou tout animal reconnaissable individuellement (marque, cicatrice, ...). Néanmoins, cela n'est pas toujours possible au vu des modes de vie de certaines espèces [Car+17].

Le nombre de facteurs impactants les estimations de densité ou d'abondance des espèces étant considérable, les chercheurs ont souvent recours à des méthodes où les quantités d'intérêt peuvent être extraites à partir de modélisations indirectes.

Dans [Row+08], les auteurs proposent un modèle statistique permettant d'estimer par vraisemblance la densité à partir des fréquences d'apparition dans les pièges photographiques. Le modèle porte le nom de Random Encounter Model. [Row+08] Des extensions du modèle d'origine ont été développées depuis [NFS18 ; NHA20].

D'autres modèles existent comme les modèles d'occupation [Mac+02]. Les auteurs définissent l'occupation comme la proportion du site occupée par l'espèce. En subdivisant le site de l'étude en plusieurs zones, ce modèle prend en compte la probabilité de détection relativement à des observations d'absence-présence de l'espèce dans les zones concernées.

1.7 Les sciences participatives au service du photo-piégeage

Afin d'examiner les grandes quantités d'images issues de campagnes de photo-piégeage, il est possible d'avoir recours au bénévolat. Les amateurs de la faune sauvage se proposent sur la base du volontariat pour aider à identifier les animaux des images de pièges-photos. Mais cela ne s'arrête pas là. Les participants peuvent aussi extraire d'autres informations s'avérant être utiles à l'étude comme les caractéristiques des animaux ou encore leur comportement. Les contributions des bénévoles sont souvent essentielles à la réussite de projets scientifiques [Kyb+13 ; Sil09] et à l'élargissement des connaissances. Par exemple, dans [Ree+13], les auteurs investiguent les motivations des volontaires. Les résultats des analyses factorielles suggèrent la recherche d'engagement social, de l'envie d'aider parmi d'autres constituent une grande partie des motivations des volontaires à participer dans de tels projets.

Dans ses débuts, la science participative prenait une forme où les contributions

[Swa+04] Swann et al., « Infrared-triggered cameras for detecting wildlife : an evaluation and review »

[RC08] Rowcliffe et Carbone, « Surveys using camera traps : are we looking to a brighter future ? »

[Car+17] Caravaggi et al., « A review of camera trapping for conservation behaviour research »

[Row+08] Rowcliffe et al., « Estimating animal density using camera traps without the need for individual recognition »

[NFS18] Nakashima, Fukasawa et Samejima, « Estimating animal density without individual recognition using information derivable exclusively from camera traps »

[NHA20] Nakashima, Hongo et Akomo-Okoue, « Landscape-scale estimation of forest ungulate density and biomass using camera traps : Applying the REST model »

[Mac+02] MacKenzie et al., « Estimating site occupancy rates when detection probabilities are less than one »

[Kyb+13] Kyba et al., « Citizen science provides valuable data for monitoring global night sky luminance »

[Sil09] Silvertown, « A new dawn for citizen science »

[Ree+13] Reed et al., « An exploratory factor analysis of motivations for participating in Zooniverse, a collection of virtual citizen science projects »

CHAPITRE 1. LE PHOTO-PIÉGEAGE POUR LE SUIVI DE LA FAUNE

[Coo+07] Cooper et al., « Citizen science as a tool for conservation in residential ecosystems »

[DZB10] Dickinson, Zuckerberg et Bonter, « Citizen science as an ecological research tool : challenges and benefits »

[Swa+15] Swanson et al., « Snapshot Serengeti, high-frequency annotated camera trap images of 40 mammalian species in an African savanna »

se faisaient à échelle individuelle par le biais de collecte d'information [Coo+07], ou en assistant les chercheurs sur le terrain [DZB10]. Grâce au numérique, la science participative peut être pratiquée à travers le monde entier et par le biais d'intérêt. Son accès y est devenu plus facile car il suffit d'interagir avec la plateforme web du projet. Les données collectées par le biais des sciences participatives peuvent également être utilisées dans l'automatisation de certains processus.

Un exemple de projet de science participative virtuelle est l'initiative Snapshot Serengeti. Ce projet a donné lieu à un énorme jeu d'images d'animaux étiquetées. Près de deux cents pièges photographiques ont été déployés dans le Parc National Serengeti en Tanzanie. Près de 7.1 millions d'images ont été produites au total et 1.2 millions d'images ont été classifiées par des volontaires en 2013 résultant en 10.8 millions de classification [Swa+15]. Afin d'améliorer la précision et la robustesse des contributions des utilisateurs, chaque image a été distribuée à différents utilisateurs et un algorithme a été utilisé pour agréger les différentes classifications en un consensus commun. Les classifications retenues ont été ensuite comparées à celles d'experts. Le taux de précision chez les volontaires a été estimé à 96% [Swa+15].

Une utilisation croissante du photo-piégeage se trouve dans l'éducation et la sensibilisation aux problématiques environnementales. Les sciences participatives sont un excellent outil pour impliquer le grand public dans la recherche. La plateforme de science participative *eMammal*² offre l'opportunité aux écoles de déployer des pièges photographiques et offre également un support technique pour l'élaboration de projets. De cette manière les élèves développent une connaissance des animaux aux alentours de leurs lieux de résidence [Sch+19]. D'autre part, une initiative de science participative de photo-piégeage a été lancée en France par le CREA³ Mont-Blanc sur la célèbre plateforme Zooniverse⁴. Les images sont un outil simple et efficace pour promouvoir une sensibilité aux problématiques de conservation de la biodiversité en donnant un aperçu de notre entourage bien souvent inaccessible [DZB10].

1.8 Considérations de vie privée et protection des espèces

Dans une société se dirigeant vers l'ouverture, l'accès et le partage des données avec le grand public, des considérations de protection de vie privée se révèlent être nécessaires. Les pièges photographiques pouvant être déployés dans les espaces publics, les humains passant à côté peuvent enclencher le mécanisme de prise de photos et par conséquent ces images doivent faire l'objet d'un tri au préalable avant d'être mises à disposition du public. Ce processus peut être automatisé en entraînant un réseau de neurones profond [Nor+21]. De plus en plus des pays prennent en considération les enjeux de respect de vie privée en lien avec le photo-piégeage en implantant un cadre légal à son utilisation comme c'est le cas en Suisse et en Australie [Rov+13]. En France, seul le régime général relatif au respect de la vie privée est appliqué au vu de l'Article 9 du code civil. A notre connaissance il n'existe pas encore

[Sch+19] Schuttler et al., « Citizen science in schools : students collect valuable mammal data for science, conservation, and community engagement »

³ Centre de Recherches sur les Ecosystèmes d'Altitude

⁴ www.zooniverse.org

[Nor+21] Norouzzadeh et al., « A deep active learning system for species identification and counting in camera trap images »

[Rov+13] Rovero et al., « " Which camera trap type and how many do I need ?" A review of camera features and study designs for a range of wildlife research applications. »

1.8. CONSIDÉRATIONS DE VIE PRIVÉE ET PROTECTION DES ESPÈCES

de restrictions quant à l'exploitation des images de pièges photographiques concernant les espèces menacées en France.

D'autre part, des considérations importantes sont à prendre en compte lorsqu'il s'agit de révéler la localisation d'espèces menacées en rendant les données publiques.

Les activités de braconnage font aussi l'objet de considérations sérieuses. Cependant l'automatisation de l'étiquetage des images issues des pièges photographiques pourrait aussi aider dans le repérage d'activités de braconnage [Fer+18 ; BB21].

[Fer+18] Ferreguetti et al., « One step ahead to predict potential poaching hotspots : Modeling occupancy and detectability of poachers in a neotropical rainforest »

[BB21] Beery et Bondi, « Can poachers find animals from public camera trap images ? »

CHAPITRE DEUX

L'apprentissage profond et la classification d'images

Afin d'identifier les espèces d'animaux capturés par les pièges photographiques, un classifieur d'images doit être construit. Le rôle du classifieur est d'associer à chaque image un label parmi un ensemble de labels fini. Les images de pièges photographiques se caractérisent par une forte hétérogénéité dans les clichés capturés (voir Section 1.4). Les conditions environnementales, particulièrement dans notre cas, y contribuent fortement car les caméras sont déployées dans des endroits denses en végétation ; par exemple, les animaux peuvent se trouver derrière la végétation. D'autres sources d'hétérogénéité sont les limitations du matériel, le comportement des animaux ou encore l'abondance des espèces. Les individus d'espèces abondantes seront capturés de manière plus fréquente, par exemple. Par conséquent, le jeu de données présente un déséquilibre entre les classes, ce qui pourrait potentiellement biaiser le classifieur en faveur des classes les plus fréquentes. Dans ce qui suit, nous exposons le problème de classification d'images et nous évoquons l'état de l'art pour répondre à cette problématique : l'apprentissage automatique et profond.

2.1 La classification d'images

La classification d'images peut être résumée de la manière suivante. C'est la tâche d'attribution d'un label à une image donnée, parmi un ensemble de labels fini. Ainsi, une image est "*transformée*" en un label. Dans le cas des réseaux de neurones convolutifs, le modèle attribue en sortie des probabilités à toutes les classes, puis la classe la plus probable est sélectionnée [GBC16]. Concernant le jeu de données à disposition, les classes possibles sont celles qui ont été *observées* sur les pièges photographiques. Une nouvelle espèce qui n'a jamais été observée par les pièges photographiques (non identifiée par les classes existantes) sera alors classifiée incorrectement.

La classification d'images peut se faire en plusieurs étapes : le pré-traitement des données, la sélection de l'échantillon d'entraînement, l'extraction des caractéristiques, la sélection d'une approche de classification convenable et l'évaluation de la performance du modèle [LW07].

Il existe deux types de classification : la classification supervisée et la classification non-supervisée. La classification supervisée requiert l'utilisation d'un jeu de données d'entraînement pour que le classifieur apprenne à discriminer entre les classes. Contrairement à la classification supervisée, la

[GBC16] Goodfellow, Bengio et Courville, *Deep learning*

[LW07] Lu et Weng, « A survey of image classification methods and techniques for improving classification performance »

CHAPITRE 2. L'APPRENTISSAGE PROFOND ET LA CLASSIFICATION D'IMAGES

- [CH67] Cover et Hart, « Nearest neighbor pattern classification »
- [CV95] Cortes et Vapnik, « Support-vector networks »
- [Bre01] Breiman, « Random forests »
- [Bia12] Biau, « Analysis of a random forests model »
- [Has+09] Hastie et al., *The elements of statistical learning : data mining, inference, and prediction*
- [KSH12] Krizhevsky, Sutskever et Hinton, « Imagenet classification with deep convolutional neural networks »
- [Den+09] Deng et al., « Imagenet : A large-scale hierarchical image database »

[Cas+15] Castelluccio et al., « Land use classification in remote sensing images by convolutional neural networks »

⁵ Capacité du modèle de généraliser son apprentissage à des données jamais vues.

[Yos+14] Yosinski et al., « How transferable are features in deep neural networks ? »

classification non-supervisée explore les structures sous-jacentes contenues dans les données et procède à un partitionnement automatique basé sur ces structures. Un exemple de classification non-supervisée est l'algorithme des K-plus proches voisins où une nouvelle entrée est étiquetée selon sa ressemblance avec l'entrée la plus proche au sens de la minimisation d'une certaine métrique [CH67]. Des exemples de classification supervisée sont les machines à vecteurs de support (SVM) [CV95] et les forêts aléatoires (RF) [Bre01 ; Bia12]. Un exposé compréhensif des méthodes de classification peut être trouvé dans [Has+09].

Au cours des dernières années, les problèmes de classification d'images ont été largement dominés par les réseaux de neurones convolutifs qui dépassent de manière significative les performances des autres modèles existants [KSH12]. Les performances exceptionnelles des réseaux de neurones convolutifs sont en grande partie dues au fait que des architectures complexes sont pré-entraînées sur des bases de données gigantesques (par exemple ImageNet [Den+09]) en utilisant des clusters de machines de calculs à haute performance.

Globalement, trois approches sont utilisées sur des architectures déjà existantes : (1) procéder à un entraînement complet du modèle sur un nouveau jeu de données, (2) calibrer (fine-tuning) un modèle pré-entraîné (préalablement sur un jeu de données) sur le nouveau jeu de données, (3) utiliser le modèle comme outil d'extraction de caractéristiques et le coupler avec un classifieur. L'approche permettant l'obtention du meilleur résultat dépend des caractéristiques du jeu de données à disposition et du jeu de données sur lequel le modèle a été pré-entraîné.

De manière générale, lorsque le jeu de données à disposition est petit et similaire au jeu de données sur lequel le modèle est pré-entraîné, le calibrage est le choix le plus approprié. Lorsque le jeu de données à disposition est grand et différent du jeu de données de pré-entraînement du modèle (ne provient pas d'une distribution équivalente), il est préférable de procéder à un entraînement complet [Cas+15]. Cependant, la combinaison de (2) et (3) est préférable car elle permet une meilleure généralisation ⁵ [Yos+14].

Dans ce qui suit, nous abordons quelques éléments de la théorie formelle de l'apprentissage et nous exposons plus en détail les principes des réseaux de neurones convolutifs et de l'apprentissage profond.

2.2 Un formalisme de l'apprentissage

Le scénario d'apprentissage consiste à apprendre, par un *apprenant*, un *concept* à travers un *algorithme d'apprentissage*. Un pilier fondamental dans le scénario d'apprentissage est la capacité de l'apprenant à "reconnaître" le concept dans des situations qu'il n'a jamais rencontré auparavant. Dans le scénario d'apprentissage, cet aspect porte le nom de "*généralisation*".

Notons par \mathcal{X} l'ensemble de tous les exemples et par \mathcal{Y} , l'ensemble de tous les labels associés à ces exemples.

Pour des raisons de simplicité, supposons que nous disposons de deux labels, i.e $\mathcal{Y} = \{0, 1\}$ (e.g écureuil, non-écureuil).

Un **concept** $c : \mathcal{X} \rightarrow \mathcal{Y}$ est une fonction qui associe un label à un exemple.

2.2. UN FORMALISME DE L'APPRENTISSAGE

Une **classe de concept** \mathcal{C} est un ensemble de concepts que l'on aimera apprendre.

Notons par $h : \mathcal{X} \rightarrow \mathcal{Y}$, une **hypothèse** parmi toutes les hypothèses possibles \mathcal{H} . Il y a une distinction entre h et c . Dans ce contexte, l'apprenant ne connaît pas c , la vraie hypothèse qui associe le bon label à une entrée, par contre, il connaît l'hypothèse h . L'objectif est de choisir une hypothèse h qui approche c . Supposons que les exemples sont tirés indépendamment selon une distribution donnée fixe \mathcal{D} mais inconnue. Le problème de l'apprentissage est formulé comme suit.

L'algorithme d'apprentissage considère un ensemble de concepts *possibles* i.e d'hypothèses \mathcal{H} . Il prend en entrée un échantillon $\mathcal{S} = \{x^1, x^2, \dots, x^N\}$ de taille N tirés selon \mathcal{D} et leur labels $\{c(x^1), c(x^2), \dots, c(x^N)\}$ où c est le concept à apprendre.

L'ensemble formé par les pairs $\{(x^i, c(x^i))\}$ pour $i = 1, \dots, N$ est l'ensemble d'entraînement.

Afin de choisir la fonction (l'hypothèse) qui approche le mieux le concept à apprendre, nous avons besoin d'une mesure d'erreur. Dans un contexte de classification, nous pourrions choisir une *fonction de perte* qui indique si h se trompe dans le label :

$$1_{h(x) \neq c(x)} = \begin{cases} 1 & h(x) \neq c(x) \\ 0 & h(x) = c(x) \end{cases}.$$

Ainsi la fonction de perte définit la mesure d'erreur de l'hypothèse h .

L'erreur de généralisation de l'hypothèse h est l'erreur de h sur l'ensemble de tous les exemples et est définie comme suit.

Definition 2.1 (Erreur de généralisation). Soit $h \in \mathcal{H}$ une hypothèse, c un concept de \mathcal{C} et \mathcal{D} la distribution sous-jacente des éléments de \mathcal{X} . L'erreur de généralisation ou le risque de l'hypothèse h est défini par

$$R(h) = \mathbb{P}_{x \sim \mathcal{D}}(h(x) \neq c(x)) = \mathbb{E}_{x \sim \mathcal{D}}[1_{h(x) \neq c(x)}].$$

L'objectif de l'algorithme d'apprentissage est de choisir, en se basant sur l'échantillon à disposition, une hypothèse qui possède une petite erreur de généralisation. Cependant, cette quantité est inaccessible pour l'algorithme d'apprentissage car le concept c et la distribution des $x \in \mathcal{X}$ sont inconnus. Seul l'échantillon $\mathcal{S} = \{(x^i, c(x^i))\}_{i=1}^N$ est accessible. Nous parlons alors **d'erreur empirique** ou *d'erreur d'entraînement* qui représente l'erreur de l'hypothèse h sur l'échantillon \mathcal{S} . Elle est définie comme suit.

Definition 2.2 (Erreur empirique). Soit $h \in \mathcal{H}$ une hypothèse, c un concept et un échantillon $\{(x^i, c(x^i))\}_{i=1}^N$.

L'erreur empirique de h aussi appelée le risque empirique est défini par

$$\hat{R}(h) = \frac{1}{N} \sum_{i=1}^N \mathbb{1}_{h(x^i) \neq c(x^i)}.$$

Ainsi l'erreur de généralisation est l'espérance de la perte par rapport à la distribution \mathcal{D} et l'erreur empirique est la moyenne de la perte basée sur l'échantillon. La recherche d'une hypothèse h qui minimise le risque empirique est ce qui est appelé dans la littérature *le principe Minimisation du Risque Empirique* [SB14 ; MRT18].

D'autre part, l'aléa contenu dans le tirage de l'échantillon \mathcal{S} se reflète dans le choix de l'hypothèse h par l'algorithme et par conséquent, il est transmis à l'erreur. Par exemple l'échantillon \mathcal{S} pourrait ne pas être représentatif de la distribution \mathcal{D} . Il est alors possible que l'hypothèse h n'approche pas suffisamment bien le concept à apprendre.

La théorie classique de l'apprentissage statistique établit des garanties sur la performance d'un estimateur (hypothèse) en fournissant des limites supérieures pour l'erreur de généralisation en fonction de l'erreur empirique (quantité observable) et du terme de complexité computationnelle, selon le paradigme d'apprentissage dans lequel on peut se trouver. Par exemple, pour l'apprentissage PAC [Val84] la complexité computationnelle des éléments de \mathcal{X} et du concept à apprendre, la cardinalité de l'ensemble des hypothèses jouent un rôle clef dans la garantie de l'apprentissage [SB14 ; MRT18]. Dans le cas d'un ensemble infini d'hypothèses possibles, d'autres cadres sont nécessaires afin d'obtenir une garantie d'apprentissage comme par exemple le cadre proposé par Vapnick et Chervonenkis utilisant la notion de la dimension-VC [VC15 ; MRT18].

En réalité, le principe de minimisation du risque empirique à lui seul, ne suffit pas à garantir la généralisation de l'algorithme d'apprentissage. L'algorithme pourrait avoir un très bon ajustement sur les données d'entraînement, mais n'arrive pas à se généraliser à de nouveaux exemples. Ce phénomène est appelé **sur-ajustement**, abordé dans [Mit80 ; SB14]. Dans [VC74], le principe de minimisation du risque empirique est destiné aux échantillons de grande taille. Quand l'échantillon à disposition est petit par exemple, une petite erreur empirique \hat{R} petit ne suffit pas à garantir une petite erreur de généralisation R [Vap99]. Afin de pallier à ce problème, les auteurs dans [VC74] introduisent le principe de minimisation du risque structurel qui sert à définir un compromis entre la qualité de l'approximation et la complexité de l'hypothèse (le nombre de ses paramètres). D'une part, une hypothèse de faible complexité est en faveur de la généralisation. Cependant, une telle hypothèse pourrait échouer à réduire l'erreur empirique (expliquer les données). D'autre part, une hypothèse de complexité riche pourrait avoir une faible erreur empirique mais fait grandir la borne supérieure de l'erreur de généralisation, ce qui conduit au sur-ajustement.

[SB14] Shalev-Shwartz et Ben-David, *Understanding machine learning : From theory to algorithms*
 [MRT18] Mohri, Rostamizadeh et Talwalkar, *Foundations of machine learning*

[Val84] Valiant, « A theory of the learnable »

[VC15] Vapnik et Chervonenkis, « On the uniform convergence of relative frequencies of events to their probabilities »

[Mit80] Mitchell, *The need for biases in learning generalizations*

[VC74] Vapnik et Chervonenkis, *Theory of pattern recognition*

[Vap99] Vapnik, *The nature of statistical learning theory*

2.3. PRINCIPES DE L'APPRENTISSAGE PROFOND

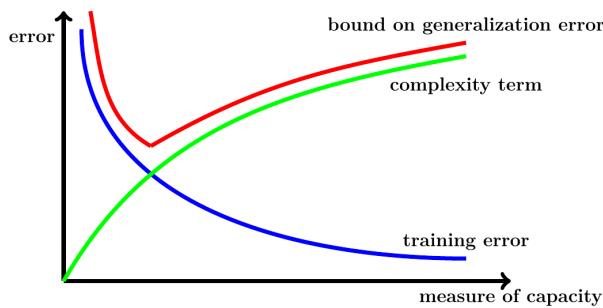


Figure 2.1 – Illustration du principe de minimisation du risque structurel. Les courbes de trois erreurs sont illustrées en fonction de la "mesure de capacité" i.e la taille de l'ensemble des hypothèses. Plus cette taille grandit, l'erreur d'entraînement décroît tandis que le terme de complexité grandit. Le principe de minimisation du risque structurel (en rouge) sélectionne l'hypothèse qui minimise la borne sur l'erreur de généralisation.

Une solution pour éviter le sur-ajustement est d'appliquer le principe de minimisation de risque sur un espace d'hypothèses restreint [SB14]. Plus précisément, une classe particulière d'hypothèses pourrait être pré-définie idéalement basée sur la connaissance a priori du problème à disposition.

La théorie de l'apprentissage est une discipline bien développée de nos jours et des recherches approfondies sont toujours en cours. Un exposé compréhensif sur cette discipline est présenté dans [SB14 ; MRT18].

2.3 Principes de l'apprentissage profond

2.3.1 Quelques notes historiques

Le psychologue Donald O. Hebb a proposé en 1949 un travail fondamental sur le processus d'apprentissage [Heb05]. Il suggère que l'apprentissage dans le cerveau se fait par la formation et le changement des synapses entre les neurones.

Dix ans plus tard, le psychologue Frank Rosenblatt publie des travaux sur le modèle du Perceptron [Ros58 ; Ros61] en se basant sur le travail d'autres scientifiques dont Donald O. Hebb.

En 1969, les chercheurs Marvin Minsky et Seymour Papert publient un livre où ils exposent des limitations fondamentales du Perceptron [MP69]. Par exemple, un seul Perceptron ne pouvait pas résoudre un problème XOR. De plus, ils conjecturent à tort que de telles limitations étaient également valables pour les perceptrons multi-couches. Le livre a provoqué une baisse significative dans l'intérêt et le financement des recherches autour des réseaux de neurones.

Trois années plus tard, Stephen Grossberg publie une série de travaux sur la modélisation du problème XOR avec des réseaux de neurones [Gro82]. En 1987, Minsky et Papert republient une nouvelle édition de leur livre contenant des corrections des erreurs du livre original. Les auteurs incluent également dans cette édition une dédicace à titre posthume écrite à la main pour Frank Rosenblatt⁶. Plus de détails sur cette controverse peuvent être trouvés dans [Ola96].

[Heb05] Hebb, *The organization of behavior : A neuropsychological theory*

[Ros58] Rosenblatt, « The perceptron : a probabilistic model for information storage and organization in the brain. »

[Ros61] Rosenblatt, *Principles of neurodynamics. perceptrons and the theory of brain mechanisms*

[MP69] Minsky et Papert, « An introduction to computational geometry »

[Gro82] Grossberg, « Contour enhancement, short term memory, and constancies in reverberating neural networks »

⁶ A titre anecdotique, Rosenblatt et Minsky étaient des amis depuis leur adolescence. Ils ont étudié dans le même lycée avec un an d'écart. Plus tard, ils ont suivis deux écoles différentes dans la recherche en intelligence artificielle. Minsky était un advocateur du symbolisme tandis que Rosenblatt s'est intéressé au connectionnisme et à l'apprentissage.

[Ola96] Olazaran, « A sociological study of the official history of the perceptrons controversy »

Le domaine des réseaux de neurones a continué de rencontrer des difficultés à cause de limitations dans la disponibilité des données et de capacités de calculs entre autre. Ce n'est qu'en 2006 que l'intérêt envers les réseaux de neurones a été ranimé après l'introduction des réseaux profonds ce qui conduira petit à petit au succès qu'ils connaissent aujourd'hui.

2.3.2 Les réseaux de neurones

Nous avons abordé dans la Section 2.2 qu'un problème d'apprentissage consiste en la recherche d'une hypothèse (une fonction) qui s'approche du concept (une autre fonction) à apprendre. Par exemple, dans un contexte de régression linéaire, le concept à apprendre est une droite qui s'ajuste aux données à travers une hypothèse linéaire. Pour faire référence à la section précédente, la recherche d'une hypothèse linéaire signifie la restriction de l'espace des hypothèses aux fonctions affines.

Les réseaux de neurones prennent leur sens quand la fonction à approcher est complexe et hautement non-linéaire. Par exemple, la fonction qui, à une image, associe un label (écureuil, non-écureuil). Le composant le plus élémentaire d'un réseau de neurones est le neurone. Il consiste en deux opérations :

- La *pré-activation* z_i d'un neurone est une agrégation linéaire de signaux s_j pondérés par un poids W_{ij} et un biais b_i ,

$$z_i(s) = b_i + \sum_{j=1}^{n_{in}} W_{ij} s_j \quad \text{pour } i = 1, \dots, n_{out}. \quad (2.1)$$

- L'*activation*. Chaque neurone émet un signal selon la valeur de z_i et produit une activation

$$\sigma_i = \sigma(z_i), \quad (2.2)$$

La fonction σ est appelée la **fonction d'activation**.

où n_{out} représente le nombre de neurones d'une couche⁷, qui prend un vecteur n_{in} -dimensionnel de signaux s_j et renvoie en sortie un vecteur n_{out} -dimensionnel d'activations σ_i .

Les fonctions d'activation sont souvent non linéaires de manière à ce que l'empilement successif de couches de neurones accumule de la non-linéarité afin de permettre au réseau de neurones d'exprimer des fonctions hautement complexes.

Une fonction d'activation d'importance historique est la fonction **Sigmoïde**

$$\sigma : z \mapsto \frac{1}{1 + \exp^{-z}} = \frac{1}{2} + \frac{1}{2} \tanh\left(\frac{z}{2}\right). \quad (2.3)$$

Son utilité réside dans le fait qu'elle préserve la magnitude de la pré-activation, particulièrement au voisinage de $z = 0$ où la fonction est quasi-linéaire et, quand l'entrée s'éloigne de zéro, la fonction *sature*. La dérivabilité de la Sigmoïde a été essentielle dans le développement de l'algorithme d'apprentissage des réseaux de neurones [RHW86]. Cependant, cette fonction d'activation possède des limitations du fait qu'elle ne passe pas par l'origine [RYH22].

⁷ Une couche est formée par les n_{out} neurones.

[RHW86] Rumelhart, Hinton et Williams, « Learning representations by back-propagating errors »

[RYH22] Roberts, Yaida et Hanin, *The Principles of Deep Learning Theory : An Effective Theory Approach to Understanding Neural Networks*

2.3. PRINCIPES DE L'APPRENTISSAGE PROFOND

Une fonction qui contourne certaines limitations de la sigmoïde est la **Tangente hyperbolique**. C'est une remise à échelle translatée de la sigmoïde,

$$\sigma : z \mapsto \tanh(z) = \frac{\exp^z - \exp^{-z}}{\exp^z + \exp^{-z}}. \quad (2.4)$$

La tangente hyperbolique passe par l'origine. Cette propriété est d'importance particulière car elle permet un entraînement efficace du réseau [LeC+12]. Les fonctions sigmoïdes soulèvent un problème [HM95 ; Hoc98] pour les méthodes d'apprentissage par gradient (voir Section 2.3.3). Elles ne sont sensibles à z qu'autour de 0. Toutefois, elles possèdent la propriété gênante de basculer dans un régime de saturation une fois que la magnitude de z est très élevée. Cette saturation amène le gradient (Section 2.3.3) à être très proche de 0, ce qui soit arrête le processus d'apprentissage, soit le rend très lent.

La conception de nouvelles fonctions d'activation est un domaine actif. Récemment, de nouvelles fonctions ont été proposées avec beaucoup d'attention portée à la question de la saturation. Par exemple, la ReLU⁸ [GBB11 ; NH10 ; Jar+09]

$$\sigma : z \mapsto \text{relu}(z) = \max(z, 0) \quad (2.5)$$

ne possède pas le problème de saturation. De plus, elle permet un entraînement plus rapide [KSH12]. Plus tard, des améliorations ont été proposées comme la Leaky ReLU [MHN+13]. Plusieurs autres types de fonctions d'activations sont possibles mais leur utilisation reste moins commune que celles citées ci-dessus. Le lecteur peut se référer à [GBC16] pour plus de détails sur les fonctions d'activation.

Le perceptron et le perceptron multi-couche

Le perceptron introduit par Rosenblatt (Figure 2.2) est simplement un seul neurone.

C'est une fonction linéaire en son entrée,

$$\hat{y} = f(s) = \sigma(\mathbf{s} \cdot \mathbf{w}) + b = \sigma\left(\sum_{i=1}^{n_{in}} s_i w_i + b\right), \quad (2.6)$$

où la fonction d'activation est $z \mapsto 1_{\{z \in \mathbb{R}_+\}}$. Le neurone *s'active* ou pas selon le signe de l'entrée.

Son apprentissage consiste en la recherche des poids et du biais qui rendent \hat{y} proche de l'objectif y . L'algorithme d'apprentissage proposé par Rosenblatt consiste en la mise à jour des poids en augmentant \mathbf{w} si la sortie \hat{y} est plus petite que y ou dans le cas contraire, en réduisant \mathbf{w} de la manière suivante

$$\mathbf{w} \leftarrow \mathbf{w} - (\hat{y} - y)\mathbf{s}, \quad (2.7)$$

où $y \in \{0, 1\}$.

L'algorithme continue l'apprentissage tant que le modèle commet une erreur dans la classification.

[LeC+12] LeCun et al., « Efficient backprop »

[HM95] Han et Moraga, « The influence of the sigmoid function parameters on the speed of back-propagation learning »

[Hoc98] Hochreiter, « The vanishing gradient problem during learning recurrent neural nets and problem solutions »

⁸ Rectified Linear Unit

[GBB11] Glorot, Bordes et Bengio, « Deep sparse rectifier neural networks »

[NH10] Nair et Hinton, « Rectified linear units improve restricted boltzmann machines »

[Jar+09] Jarrett et al., « What is the best multi-stage architecture for object recognition ? »

[KSH12] Krizhevsky, Sutskever et Hinton, « Imagenet classification with deep convolutional neural networks »

[MHN+13] Maas, Hannun, Ng et al., « Rectifier nonlinearities improve neural network acoustic models »

[GBC16] Goodfellow, Bengio et Courville, *Deep learning*

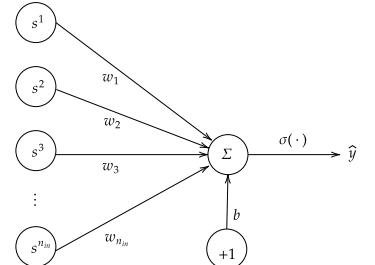


Figure 2.2 : Illustration du perceptron. s^i est la i^{eme} coordonnée du vecteur d'entrée s .

[Nov63] Novikoff, *On convergence proofs for perceptrons*

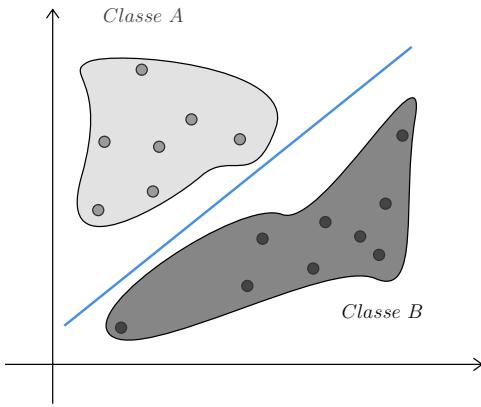


Figure 2.3 : Exemple de deux classes linéairement séparables.

⁹ i.e les sorties d'une couche servent d'entrée à la couche suivante

La convergence de cet algorithme a été démontrée dans le cas de deux classes séparables linéairement [Nov63]. (voir Figure 2.3). Toutefois, dans le cas où les classes ne sont pas linéairement séparables, l'algorithme ne converge jamais.

Le nombre "optimal" de couches et le nombre de neurones qu'elles contiennent dépend fortement du problème à résoudre et doit souvent être déterminé par essais.

En organisant plusieurs neurones dans une couche puis en les empilant de manière itérative ⁹, nous construisons une **architecture** réseau de neurones.

Cette structure est ce qui est appelée le **Perceptron-multicouches** (Voir Figure 2.4). La profondeur du réseau de neurones fait référence au nombre de couches et **l'apprentissage profond** désigne en ce sens des réseaux de neurones contenant beaucoup de couches. La fonction d'activation est souvent non linéaire de manière à ce que leur empilement successif à travers les couches accumule de la non-linéarité et permette au réseau de neurones d'exprimer des fonctions complexes en sortie.

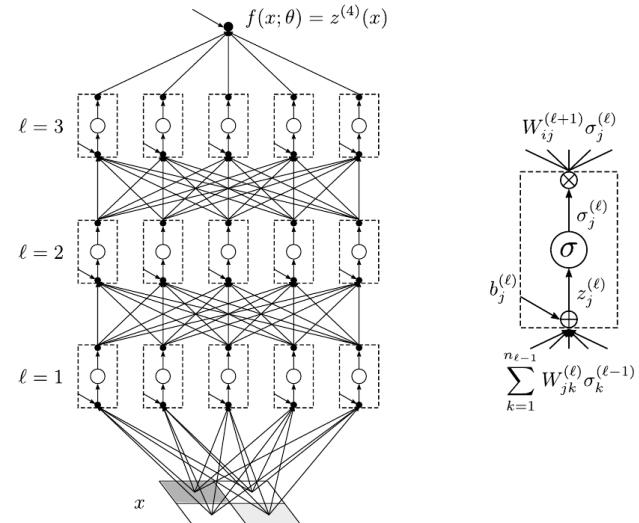


Figure 2.4 – Gauche : Illustration d'une architecture d'un perceptron multi-couches. Cette architecture contient $L = 4$ couches définissant un ensemble de fonctions $f(x; \theta)$ de dimension d'entrée $n_0 = 4$ et de dimension de sortie $n_4 = 1$. Les couches cachées possèdent chacune $n_1, n_2, n_3 = 5$ neurones. **Droite :** Structure détaillée de chaque neurone qui (i) agrège le biais au signaux pondérés pour produire la pré-activation, (ii) génère l'activation, et (iii) multiplie l'activation par les poids de la couche suivante. Image de [RYH22]

[RYH22] Roberts, Yaida et Hanin, *The Principles of Deep Learning Theory : An Effective Theory Approach to Understanding Neural Networks*

Supposons qu'on dispose d'un échantillon $\mathcal{D} = \{x_{i;\alpha}\}$ tel que $\alpha = 1, \dots, N_D$ représente la α^{eme} entrée et $i = 1, \dots, n_0$ représente la i^{eme} coordonnée du vecteur x . Le perceptron multi-couche défini par [RYH22]

$$z_i^{(1)}(x_\alpha) \equiv b_i^{(1)} + \sum_{j=1}^{n_0} W_{ij}^{(1)} x_{j;\alpha}, \quad \text{pour } i = 1, \dots, n_1, \quad (2.8)$$

2.3. PRINCIPES DE L'APPRENTISSAGE PROFOND

et

$$z_i^{(\ell+1)}(x_\alpha) \equiv b_i^{(\ell+1)} + \sum_{j=1}^{n_\ell} W_{ij}^{(\ell+1)} \sigma(z_j^{(\ell)}(x_\alpha)), \quad (2.9)$$

pour $i = 1, \dots, n_{\ell+1}; \ell = 1, \dots, L-1$ décrit un réseau à L couches de neurones où la couche ℓ est composée de n_ℓ neurones. Sa sortie

$$f(x; \theta) = z^{(L)}(x), \quad (2.10)$$

sert de fonction d'approximation du concept à apprendre où θ est l'ensemble des paramètres du modèle, i.e les poids W_{ij} et les biais b_i de toutes les couches précédentes.

Le perceptron multi-classe

D'autre part, il est possible d'étendre le perceptron à plus de deux classes en connectant plusieurs perceptrons à la même entrée, chaque perceptron possédant ses propres poids et biais. La sortie est alors un vecteur dont la dimension est le nombre de classe

$$\hat{y}_j = f_j(\mathbf{x}) \quad \forall j = 1, \dots, K.$$

Avec K le nombre de classe.

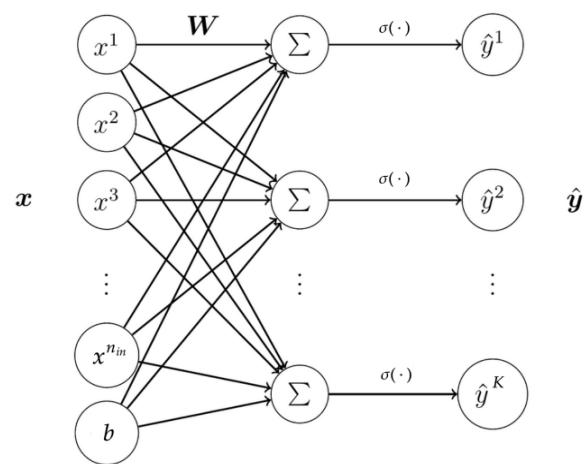


Figure 2.5 – Illustration d'un perceptron multi-classes. \mathbf{x} représente le vecteur d'entrée. W représente les poids du perceptron sous forme matricielle. Image de

La classe k prédite est alors, la classe dont la sortie y_k est la plus grande. L'entraînement du perceptron multi-classe se fait en appliquant la même règle d'apprentissage du perceptron comme décrit précédemment. En utilisant une notation matricielle, notons $\mathbf{W} \in \mathbb{R}^{(n_{in}+1) \times K}$ les poids du perceptron multi-classes et \mathbf{s} un vecteur de dimension $1 \times n_{in} + 1$,

$$\mathbf{W} \leftarrow \mathbf{W} - \mathbf{x}^T \cdot (\hat{\mathbf{y}} - \mathbf{y}) \quad (2.11)$$

2.3.3 L'apprentissage par gradient

Les paramètres du réseau de neurones (i.e les poids et biais) sont tirés selon une distribution a priori en guise d'initialisation. Ils doivent ensuite être ajustés afin de fournir de bonnes prédictions. Pour une entrée \mathbf{x}_α associée à un label y_α tirés selon une distribution $p(x, y)$, le modèle a pour objectif d'émettre une sortie $f(\mathbf{x}_\alpha, \theta)$ qui soit *en moyenne* la plus proche possible du label y_α .

Pour mesurer la proximité, une fonction de coût doit être définie

$$\mathcal{L}(f(\mathbf{x}_\alpha, \theta), y_\alpha), \quad (2.12)$$

¹⁰ La fonction de coût peut être différente de l'indicatrice de l'erreur de classification. Un exemple intuitif d'une fonction de coût est la moyenne des carrés des écarts

$$\mathcal{L}_{MSE}\left(f(\mathbf{x}_\alpha, \theta), y_\alpha\right) = \frac{1}{2} \left[f(\mathbf{x}_\alpha, \theta) - y_\alpha \right]^2.$$

de sorte à ce qu'elle possède la propriété que plus la sortie du modèle $f(\mathbf{x}_\alpha, \theta)$ est proche du vrai label, plus sa valeur est petite ¹⁰.

La configuration adéquate des paramètres du modèle est formulée comme un problème d'optimisation où la fonction objectif à minimiser est l'erreur sur l'échantillon

$$\theta^* = \arg \min_{\theta} \mathbb{E} [\mathcal{L}(\theta)] \quad (2.13)$$

La vraie distribution étant inconnue, on se réfère à une construction empirique de cette espérance sur tous les exemples. Bien qu'il existe plusieurs méthodes pour résoudre ce problème d'optimisation, les méthodes les plus communes sont des méthodes à gradient.

La descente du gradient

La descente du gradient permet de minimiser des fonctions non triviales comme la fonction de perte. L'algorithme consiste en le calcul du gradient de la perte et l'ajustement des paramètres du modèle dans la direction (négative) du gradient

$$\theta(t+1) = \theta(t) - \eta \frac{d\mathcal{L}}{d\theta}|_{\theta=\theta(t)}, \quad (2.14)$$

où t est un compteur des étapes du processus itératif d'entraînement, avec la convention $t = 0$ étant le point d'initialisation. Le paramètre η est un **hyperparamètre** appelé le **taux d'apprentissage** et contrôle la largeur du saut entre les itérations. La principale raison de leur popularité est que le calcul des dérivées de $f(\mathbf{x}, \theta)$ se fait de manière précise et efficace grâce à des dérivations automatiques dans le cas de réseaux de neurones entièrement connectés [Ber+21]. Ce principe est appelé *L'algorithme de rétro propagation* [RHW86 ; GW08].

Une variante de la descente du gradient fréquemment utilisée est la **descente du gradient stochastique**. L'ajustement des poids se fait de la manière suivante

$$\theta(t+1) = \theta(t) - \eta \frac{d\mathcal{L}_{\mathcal{S}_t}}{d\theta}|_{\theta=\theta(t)}, \quad (2.15)$$

où \mathcal{S}_t est un sous-ensemble de l'ensemble d'entraînement. Chaque sous ensemble \mathcal{S}_t est appelé un **lot** ou **mini-batch** en anglais. L'entraînement se

[Ber+21] Berner et al., « The modern mathematics of deep learning »

[RHW86] Rumelhart, Hinton et Williams, « Learning representations by back-propagating errors »

[GW08] Griewank et Walther, *Evaluating derivatives : principles and techniques of algorithmic differentiation*

2.3. PRINCIPES DE L'APPRENTISSAGE PROFOND

fait alors par **époques**. Une époque est un passage complet sur l'ensemble d'entraînement. Pour chaque époque, l'ensemble d'entraînement est aléatoirement partitionné en sous ensembles de même taille, puis ils sont utilisés pour estimer le gradient. La stochasticité induite par cet aléa agit sur la direction de la descente du gradient et impose une contrainte moins forte sur l'algorithme : la direction de la descente du SGD (Stochastic Gradient Descent) doit être en moyenne égale à celle du gradient déterministe [SB14].

[SB14] Shalev-Shwartz et Ben-David, *Understanding machine learning : From theory to algorithms*

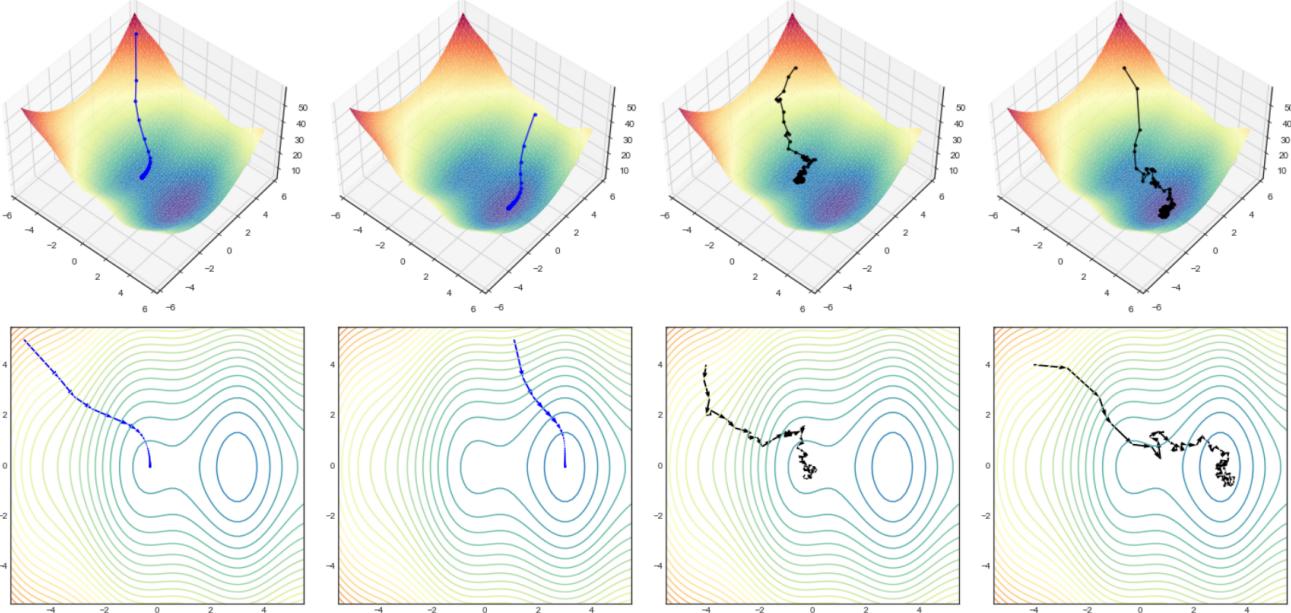


Figure 2.6 : Exemples de dynamiques de la descente du gradient (**gauche**) et de la descente du gradient stochastique (**droite**) pour une fonction à deux extrémas. Image de [Ber+21].

Comme illustré dans la Figure 2.6, la condition initiale et les fluctuations stochastiques du SGD peuvent mener l'algorithme soit à l'échec, soit à la réussite de la recherche du minimum local. Toutefois, il existe des régimes selon lesquels la convergence du SGD est garantie, du moins, en moyenne [Sha+09 ; NY83 ; Nem+09].

D'autre part, la descente du gradient stochastique suit une procédure pré-déterminée qui ne prend pas en compte les caractéristiques des données observées, et par conséquent, peut résulter en un entraînement instable ou sous optimal selon la valeur fixée du taux d'apprentissage η . Dans [DHS11], les auteurs introduisent Adagrad, un algorithme basé sur le SGD qui intègre l'information contenue des gradients des itérations précédentes dans l'itération actuelle, en normalisant chaque coordonnée par la somme des carrés des gradients passés. Toutefois, il a été observé que cet algorithme possède une moins bonne performance lorsqu'il était appliqué à des réseaux de neurones profonds [Wil+17]. Les auteurs dans [TH12] proposent une moyenne mobile exponentielle à la place de la somme comme solution à la limitation d'Adagrad. Une amélioration devenue très populaire est l'algorithme Adam [KB14] combinant les avantages des deux derniers.

[Sha+09] Shalev-Shwartz et al., « Stochastic Convex Optimization. »

[NY83] Nemirovskij et Yudin, « Problem complexity and method efficiency in optimization »

[Nem+09] Nemirovski et al., « Robust stochastic approximation approach to stochastic programming »

[DHS11] Duchi, Hazan et Singer, « Adaptive sub-gradient methods for online learning and stochastic optimization. »

[Wil+17] Wilson et al., « The marginal value of adaptive gradient methods in machine learning »

[TH12] Tieleman et Hinton, « Rmsprop : Divide the gradient by a running average of its recent magnitude. coursera : Neural networks for machine learning »

[KB14] Kingma et Ba, « Adam : A method for stochastic optimization »

2.4 Réseaux convolutifs, représentations latentes et classification

[DY+14] Deng, Yu et al., « Deep learning : methods and applications »

¹¹ i.e Couches convolutives. Présentées dans la Section 2.4.1.

[BCV13] Bengio, Courville et Vincent, « Representation learning : A review and new perspectives »

[Fuk79] Fukushima, « Neural network model for a mechanism of pattern recognition unaffected by shift in position-Neocognitron »

¹² Histogram of Oriented Gradients

¹³ Scale-Invariant Feature Transform

[DT05] Dalal et Triggs, « Histograms of oriented gradients for human detection »

[Low99] Lowe, « Object recognition from local scale-invariant features »

L'apprentissage profond fait partie d'une grande famille d'algorithmes d'apprentissage automatique. Dans [DY+14], les auteurs définissent l'apprentissage profond par une classe d'algorithmes qui (*i*) utilisent une cascade de couches d'opérations non linéaires pour la transformation et l'extraction de caractéristiques (représentations). (*ii*) l'apprentissage de ces couches peut être fait de manière supervisée ou non supervisé. (*iii*) chaque couche est perçue comme un niveau d'abstraction de représentations.

L'empilement des couches forme une hiérarchie de concepts allant des plus basiques aux plus abstraits et il a été observé que l'empilement des couches d'extractions de caractéristiques ¹¹ donnait souvent lieu à de meilleures représentations latentes des données [BCV13]. La variation du nombre de couches et de leurs tailles peut produire différents niveaux d'abstraction [BCV13].

Toutefois, dans le contexte de vision par ordinateur, l'aspect spatial contenu dans les images est un facteur clé dans l'extraction des caractéristiques. Le perceptron multi-couches est insouciant envers l'information spatiale contenue dans les images car tous les pixels sont connectés à tous les neurones de toutes les couches.

En 1979, Kunihiro Fukushima introduit Le *Neocognitron*[Fuk79], le premier réseau de neurones à avoir incorporé des connaissances neuro-physiologiques sur le cortex visuel dans un contexte d'apprentissage. Son modèle introduit les couches convolutives qui consistent en un ensemble d'opérateurs de convolutions paramétrés avec des poids, sous formes de matrices et parcourant localement une entrée bi-dimensionnelle (une image). En ce sens les **réseaux de neurones convolutifs** sont différents des réseaux de neurones conventionnels.

2.4.1 Les couches convolutives

Dans la reconnaissance de formes appliquée aux images, nous pouvons supposer qu'un motif élémentaire tel qu'un coin ou un bord peut apparaître plusieurs fois dans une image. Il est donc logique d'utiliser le même capteur sur toute l'image afin de détecter le motif, au lieu d'utiliser différents capteurs à différents endroits. Cet aspect implique la nécessité de prendre en compte deux aspects : (*i*) les voisinages des pixels et (*ii*) d'être invariant aux translations et aux rotations. Des descripteurs utilisés dans l'analyse (locale) d'images possédant ces caractéristiques sont le HOG¹² et le SIFT¹³ [DT05 ; Low99]. Ces opérateurs peuvent être généralisés par ce qui est appelé, **des opérateurs de convolution**.

Opérateurs de convolution

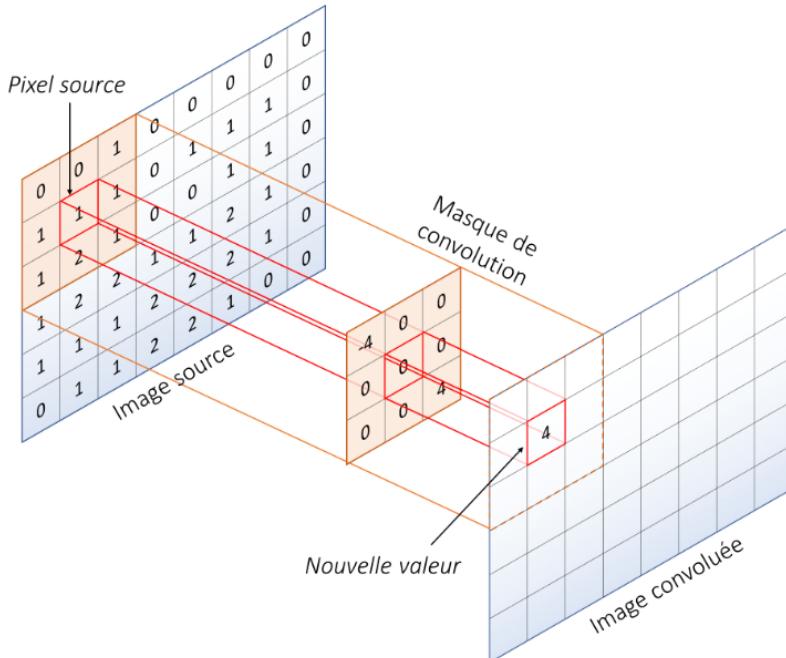
Les convolutions utilisées dans les réseaux de neurones sont des filtres carrés qui passent pas à pas sur une entrée bi-dimensionnelle (une image). Le produit de convolution d'une image est défini comme suit [DV16]

2.4. RÉSEAUX CONVOLUTIFS, REPRÉSENTATIONS LATENTES ET CLASSIFICATION

Definition 2.3 (Produit de convolution). Le produit de convolution de l'image f par le filtre de convolution g de taille $L \times L$, avec $K = \frac{L-1}{2}$ est donné par

$$(f * g)(n, m) = \sum_{i=-K}^K \sum_{j=-K}^K f(n-i, m-j) \times g(i, j).$$

Les entrées des opérateurs de convolutions sont les canaux de couleur des images et leur sortie est appelée une **carte de caractéristique** ou **feature map**. Ce procédé est réitéré tout au long du réseau de neurones et produit plusieurs cartes de caractéristiques.



Le centre du masque de convolution est placée au dessus du pixel source. Sa nouvelle valeur est la somme pondérée des poids du masque par le pixel source et ses voisins.

$$\begin{aligned} & (-4 \times 0) + (0 \times 0) + (0 \times 1) \\ & +(0 \times 1) + (0 \times 1) + (0 \times 1) \\ & +(0 \times 1) + (0 \times 2) + (4 \times 1) = 4 \end{aligned}$$

Par conséquent, une couche de convolution tardive est une combinaison des convolutions. Cette itération permet l'extraction de caractéristiques de plus en plus complexes. Informellement, les couches situées aux débuts serviront à l'extraction de motifs élémentaires comme un arrondi ou un bord, et les couches suivantes, une combinaison de ces motifs afin de détecter un trait particulier comme un museau, un œil, etc. Les couches tardives du réseau peuvent être utilisées pour déceler simultanément la présence de l'œil et du museau et reconnaître la tête d'un animal comme illustré dans la Figure 2.8.

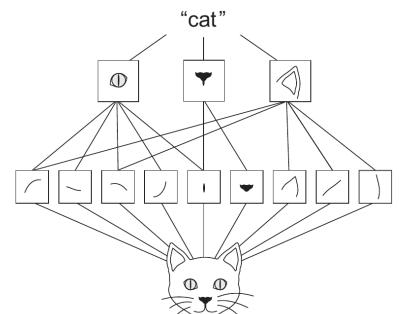


Figure 2.8 : Exemple illustratif d'une hiérarchie de caractéristiques dont la combinaison simultanée permet de former un concept particulier. Image de [Cho21].

Opérateurs de sous-échantillonnage

Les couches de convolutions sont alternées avec des couches de sous-échantillonnage. Elles ont pour vocation de réduire la taille des cartes des caractéristiques et par conséquent le nombre de paramètres du réseau tout en gardant l'information extraite par la convolution.

Ces opérateurs agissent comme dans le cas des convolutions, en passant une fenêtre glissante, de sorte à résumer l'information extraite par les cartes de caractéristiques. Des stratégies différentes peuvent être utilisées. Dans la Figure 2.9, nous présentons trois exemples de référence :

- Le sous-échantillonnage par valeur maximale (maximum pooling) qui consiste à récupérer la valeur maximale dans la fenêtre glissante.
- Le sous-échantillonnage par la somme (sum pooling) qui consiste à récupérer la somme des valeurs de la fenêtre glissante.
- Le sous-échantillonnage par la moyenne où la valeur renvoyée est la moyenne des valeurs de la fenêtre glissante

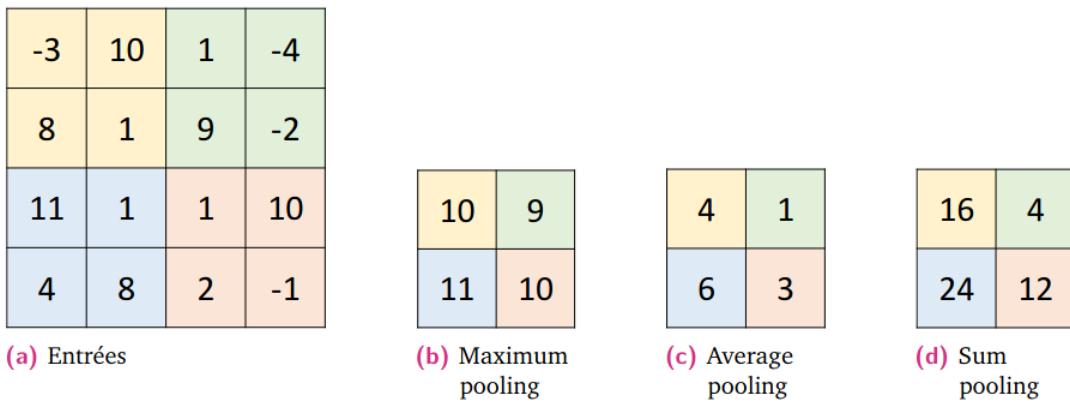


Figure 2.9 : Résultats des opérateurs de sous-échantillonnage de (a) La donnée d'entrée, par (b) valeur maximale. (c) la moyenne. (d) la somme. Image de [FD17].

Ainsi, une image (ou une carte de représentation si l'opérateur intervient à l'intérieur du réseau) est divisée en plusieurs petits blocs puis l'opérateur donne une seule sortie pour chaque bloc. Il est important de noter que les couches de sous-échantillonnage ne possèdent pas de poids et par conséquent n'augmentent pas la complexité du modèle.

L'empilement des couches convolutives et des couches de sous-échantillonnage successivement permet de couvrir une certaine partie de l'image d'origine. La partie de l'image couverte par le réseau est appelée le **champ réceptif** du réseau ou le champ de vision du réseau.

2.4. RÉSEAUX CONVOLUTIFS, REPRÉSENTATIONS LATENTES ET CLASSIFICATION

Chaque couche (de convolution ou de sous-échantillonnage) à l'intérieur du réseau dépend des caractéristiques extraites par la couche précédente et ainsi de suite. De cette manière, l'information est de plus en plus résumée et le

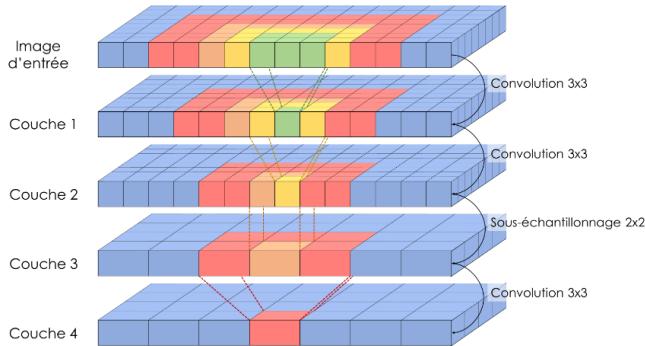


Figure 2.10 – Illustration du champ de vision d'un réseau convolutif. La partie visible de l'image est de plus en plus grande. Image de [FD17].

champ réceptif du réseau¹⁴ s'élargit de plus en plus et permet l'extraction des caractéristiques dominantes à travers l'image d'origine qui sont à la fois invariantes en rotation et en position. A la fin d'un réseau convolutif, les couches de convolutions généralement suivies d'une ou de plusieurs couches de neurones entièrement connectées dans le but d'apprendre une combinaison non linéaire des caractéristiques extraites pour effectuer une tache de classification ou de détection.

¹⁴ i.e la partie visible de l'image par le réseau

2.4.2 Adaptation d'un réseau au problème de classification

Dans le cas des problèmes de classification, il est souhaitable que les prédictions \hat{y} donnent en sortie des probabilités d'appartenance à une classe particulière de manière à satisfaire les critères

$$\hat{y}^i \geq 0 \quad \text{et} \quad \sum_{i=1}^K \hat{y}^i = 1 \quad \text{où } K \text{ est le nombre de classes.}$$

La fonction sigmoïde, par exemple, satisfait uniquement la première contrainte. Il est alors convenable d'utiliser la fonction d'activation **softmax**

$$\sigma^i(z) = \frac{\exp^{z^i}}{\sum_{i=j}^K \exp^{z^i}} \quad (2.16)$$

qui satisfait simultanément les deux critères.

Un thème récurrent dans la conception des réseaux neuronaux est que le gradient de la fonction de coût doit être suffisamment grand et prévisible pour guider l'algorithme d'apprentissage. Les fonctions d'activation qui saturent nuisent à cet objectif car elles rendent le gradient très petit [GBC16]. D'autre part, l'utilisation du carré des écarts comme fonction de perte n'est pas adapté à cause de sa non convexité par rapport aux poids [GBC16] rendant la recherche du minimum global par le SGD plus difficile dans le cas des réseaux de neurones profonds. Dans [ZS18], les auteurs proposent l'utilisation de

[GBC16] Goodfellow, Bengio et Courville, *Deep learning*

[ZS18] Zhang et Sabuncu, « Generalized cross entropy loss for training deep neural networks with noisy labels »

l'entropie croisée

$$I(y, f(x)) = - \sum_{i=1}^K y^i \log(f^i(x)) \quad (2.17)$$

qui non seulement, possède la propriété de convexité par rapport aux poids mais dont la combinaison de la fonction d'activation softmax possède des propriétés de simplicité computationnelle lors de la phase de rétro-propagation [GBC16]. Ainsi cette combinaison requiert alors que la dernière couche soit au nombre de classes entre lesquelles on souhaite discriminer.

2.4.3 Exemple d'architecture d'un réseau convolutif

Les premiers succès des réseaux de neurones convolutifs reviennent à l'utilisation de la base de données gigantesque ImageNet [Den+09] contenant plus de 14 millions d'images annotées par 1000 classes. Cette base de données est devenue une référence pour comparer les meilleurs algorithmes de classification d'images. Les performances sur cette base de données sont évaluées conventionnellement sur la métrique d'exactitude du top 1 et du top 5. Nous illustrons ci-dessous l'architecture du réseau VGG16 qui était l'un des premiers à utiliser des filtres de petites tailles ce qui lui a permis de gagner en profondeur.

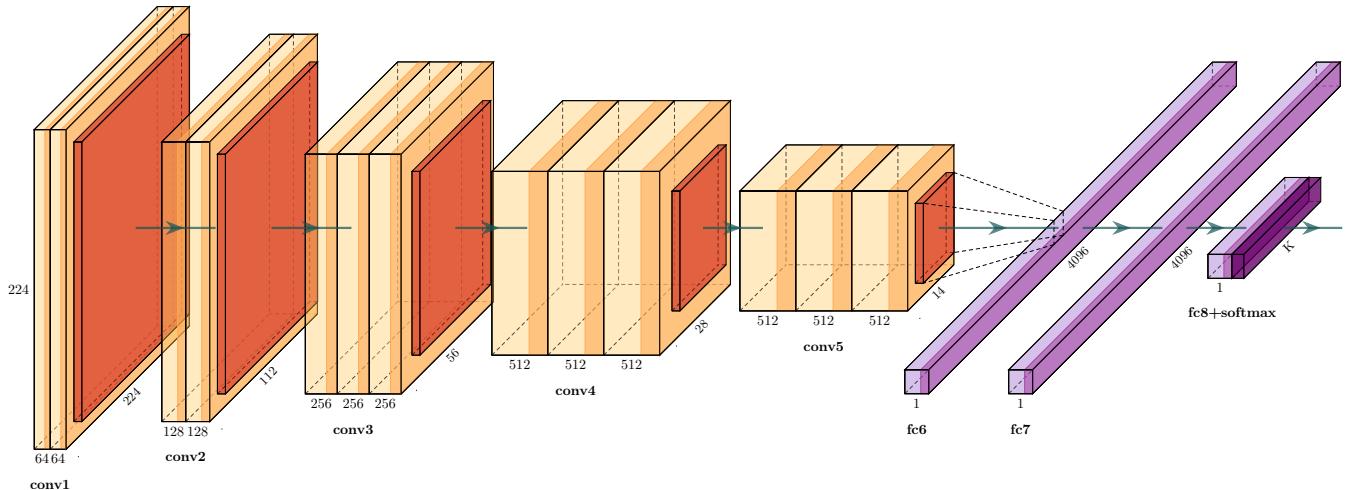


Figure 2.11 : Architecture du réseau VGG16.

¹⁵ Les réseaux existant comme AlexNet utilisaient des filtres de taille 11x11

[SZ14] Simonyan et Zisserman, « Very deep convolutional networks for large-scale image recognition »

Les concepteurs du VGG16 ont augmenté la profondeur en utilisant une architecture avec de très petits (3×3) filtres de convolution ¹⁵, ce qui a permis une amélioration significative des performances du modèles par rapport à l'état de l'art existant. La profondeur était de 16 couches munies de poids, ce qui donne environ 138 millions de paramètres entraînables. Il contient 13 couches convolutives, 5 couches de sous-échantillonnage par maximum et trois couches denses (entièrement connectées), soit un total de 21 couches, mais il n'y a que 16 couches de poids, c'est-à-dire des couches de paramètres apprenables [SZ14]. Sa performance a été évaluée sur un sous-ensemble de la base de donnée ImageNet de 1.2 millions d'images avec un score de 74.4% et de 91.9% en exactitude du top 1 et top 5 respectivement.

CHAPITRE TROIS

Description des données et contexte

Pour développer un classificateur, des données d'entraînement sont nécessaires. Dans le cadre de ce travail, les données sont les images issues du photo-piègeage ainsi que leurs étiquettes (alias "labels") associés. Nous abordons dans ce chapitre le contexte dans lequel l'étude a été réalisée et donnons un descriptif des données qui serviront à la réalisation de la tâche de classification.

3.1 Zone de l'étude et méthodologie d'échantillonnage

Le parc national des [REDACTED]

Les images utilisées pour l'entraînement du réseau de neurones proviennent de trois campagnes de photo-piègeage réalisées par Biotope dans le cadre d'une expertise de terrain. Les pièges photographiques ont été déployés dans [REDACTED]. Il est constitué de [REDACTED] (voir Figure 3.1).

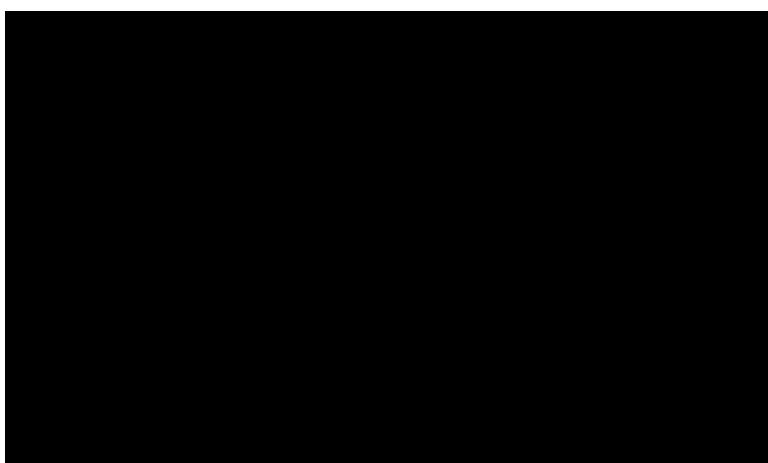


Figure 3.1 – Localisation du parc [REDACTED]. Image issue de Wikipédia.

Cette réserve naturelle couvrant partiellement la province [REDACTED], s'étend sur une superficie [REDACTED]. Elle fait partie d'une chaîne montagneuse allant du [REDACTED] dans le

CHAPITRE 3. DESCRIPTION DES DONNÉES ET CONTEXTE

[PTC02] Pauwels, Toham et Chimsunchart, « Recherches sur l'herpétofaune des Monts de Cristal, Gabon »

[Wil90] Wilks, *La conservation des écosystèmes forestiers du Gabon*

[SWI04] Sunderland, Walters et Issembe, *ETUDE PRELIMINAIRE DE LA VEGETATION DU PARC NATIONAL DE MBE, MONTS DE CRISTAL, GABON*

[Hed+18] Hedwig et al., « A camera trap assessment of the forest mammal community within the transitional savannah-forest mosaic of the Batéké Plateau National Park, Gabon »

[REDACTED]. Bien que le [REDACTED] soit situé sur [REDACTED], le climat présente [REDACTED] [PTC02]. Le [REDACTED] est considéré comme l'une des zones de [REDACTED] [Wil90]. Il contient le [REDACTED]. La [REDACTED] est une source [REDACTED] [SWI04]. Les campagnes de photo-piégeage ont été conduites dans le cadre d'une [REDACTED]

Configuration du déploiement des pièges photographiques

Trois campagnes de photo-piégeage ont été réalisées pendant la saison sèche et pluvieuse de l'année 2021. La configuration de l'emplacement des pièges photographiques a été faite en s'appuyant sur la méthodologie proposée dans [Hed+18] Il s'agit d'un plan d'échantillonnage systématique où les pièges photographiques [REDACTED]

[REDACTED] Il y a un total de [REDACTED] pièges-photographiques dans la seule zone [REDACTED]. [REDACTED] pièges supplémentaires ont été déposés ultérieurement dans la zone de [REDACTED].

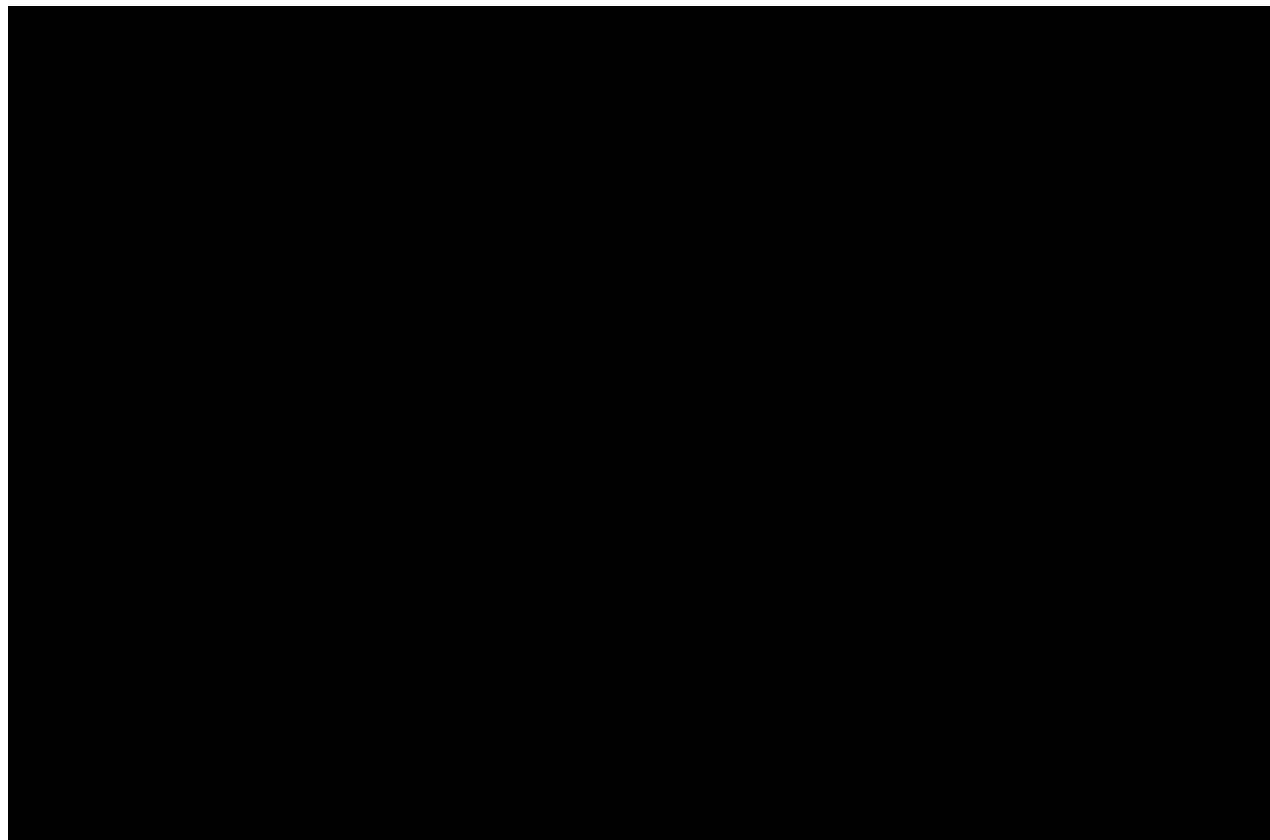


Figure 3.2 : Configuration de la grille d'échantillonnage des pièges photographiques
du [REDACTED]

Une fois que la grille est déterminée, les pièges sont montés à côté d'un chemin de passage naturel en suivant un schéma protocolaire déterminant la hauteur de pose, l'orientation, etc... Des visites ont été organisées pour le changement des cartes mémoires et des batteries ainsi que pour l'ajustement

3.2. APERÇU DES DONNÉES RÉCOLTÉES

de l'emplacement et de l'orientation des caméras et le remplacement de l'équipement endommagé. Au total, █ pièmes photographiques ont été déployés █. Quand un pième photographique est déclenché, plusieurs photos sont prises consécutivement. Les pièmes photographiques ont été configurés de manière à prendre deux photos et une vidéo de dix secondes à chaque déclenchement puis une pause de quelques secondes. Si la caméra détecte encore du mouvement, une deuxième série d'images est prise, etc... .

3.2 Aperçu des données récoltées

À l'issue des campagnes de photo-piègeage, près de █ images¹⁶ ont été récoltés indiquant la présence de mammifères, oiseaux et humains. Les vidéos n'ont pas fait l'objet de traitement et n'ont pas été utilisé dans l'entraînement du classificateur. La Table 3.1 donne une vue d'ensemble des espèces observées.

¹⁶ Pour chaque deux images, une vidéo de dix secondes

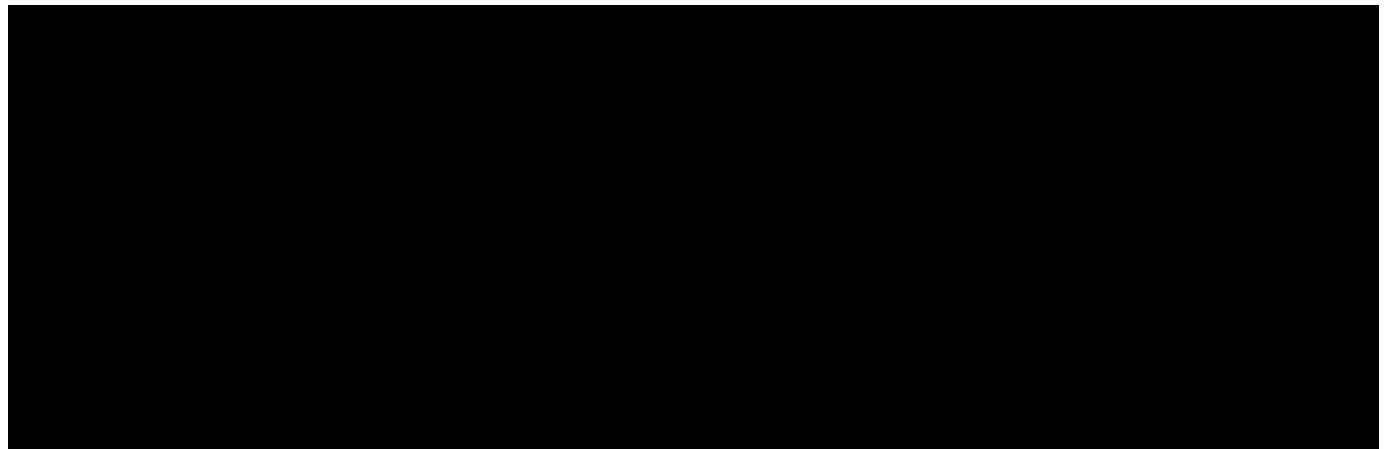


Table 3.1 : Inventaire des espèces observées, toutes campagnes confondues.

L'annotation des images a été effectuée manuellement par un salarié de Biotope : le primatologue en charge de la collecte des données pour cette étude. Ne sont pas inclus dans la table les images vides (au nombre de █) et les images contenant des humains (au nombre de █). Au total, █ espèces ont été observées. █ espèces d'oiseaux ont été identifiées parmi toutes les images contenant des oiseaux. Les oiseaux non-identifiés ont été catégorisés comme appartenant à la classe Oiseau. Le reste des classes étant essentiellement des mammifères. Toutefois, la détermination jusqu'à l'espèce n'est pas toujours possible en raison de la qualité très variable des photographies (Voir Section 3.3.1. Par conséquent, dans une optique de diminution du nombre de classes et de gestion des identifications incomplètes, les espèces similaires en termes de morphologie et/ou taxonomie ont été regroupées en une classe (eg. Céphalophe bleu, Céphalophe à bande dorsale noire, etc.) en faisant attention aux enjeux de conservations propres à chaque espèce¹⁷. Cela permet de simplifier le problème d'apprentissage et la question des espèces indéterminées sans causer une perte importante d'information vis-à-vis des objectifs de ce type d'études.

¹⁷ Deux espèces morphologiquement proches ne seront pas rassemblées en une classe.

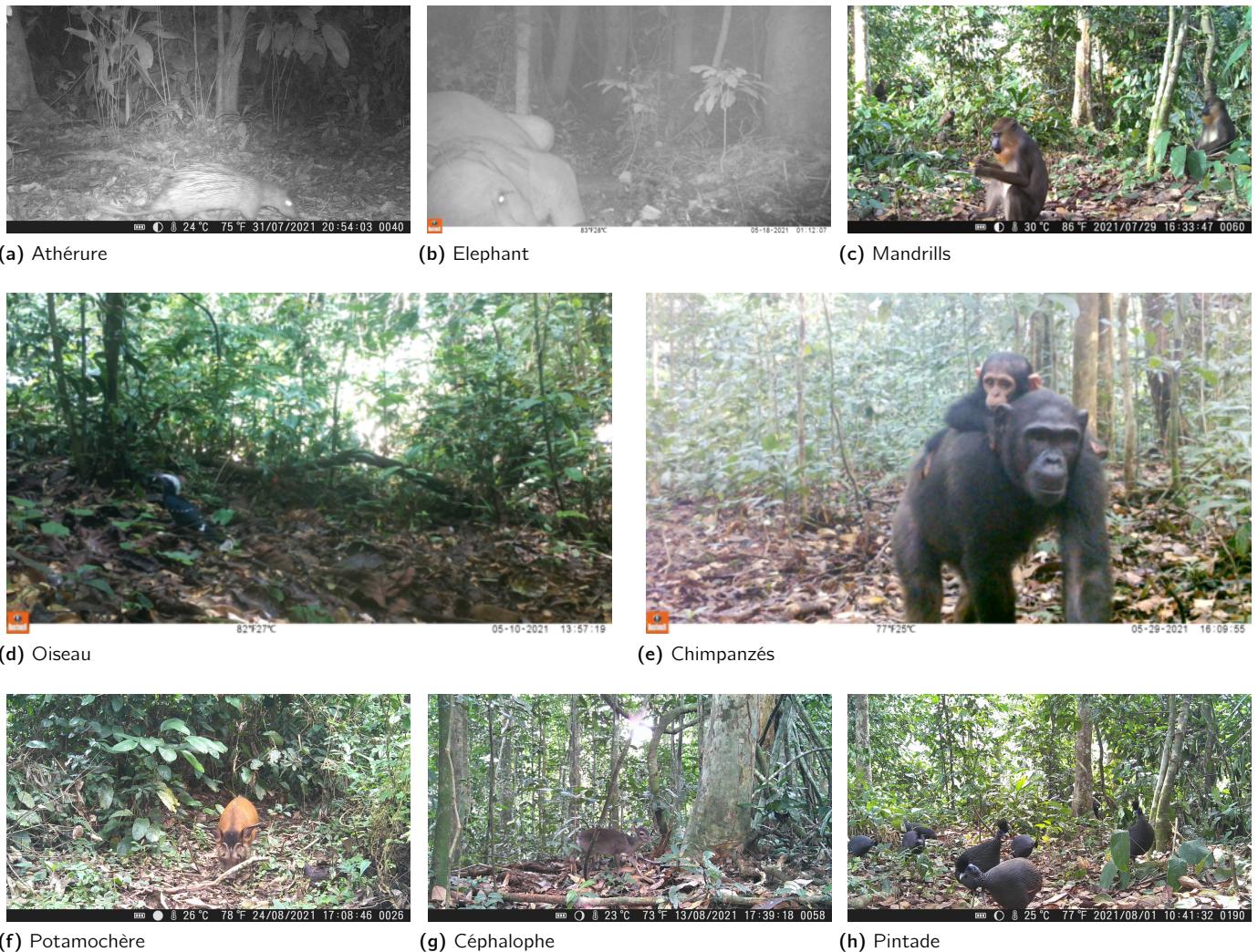


Figure 3.3 : Quelques exemples d'espèces observées par les pièges photographiques.

3.3 Les défis de la classification d'images de photo-piégeage

La classification automatique d'images de photo-piégeage vient avec quelques obstacles que nous abordons ci-dessous.

3.3.1 Obstacles circonstanciels

La qualité de l'image influence directement le résultat du processus de classification, que ce soit par des humains ou par des machines. De nombreux facteurs et circonstances ont un impact direct sur la qualité de l'image. Ils peuvent être divisés trois groupes, à savoir (i) les conditions environnementales, (ii) le comportement des animaux et (iii) les limitations matérielles [GSV16]. La Figure 3.4 illustre quelques exemples.

Les conditions environnementales font référence aux conditions dans lesquelles

[GSV16] Gomez, Salazar et Vargas, « Towards automatic wild animal monitoring : Identification of animal species in camera-trap images using very deep convolutional neural networks »

3.3. LES DÉFIS DE LA CLASSIFICATION D'IMAGES DE PHOTO-PIÉGEAGE

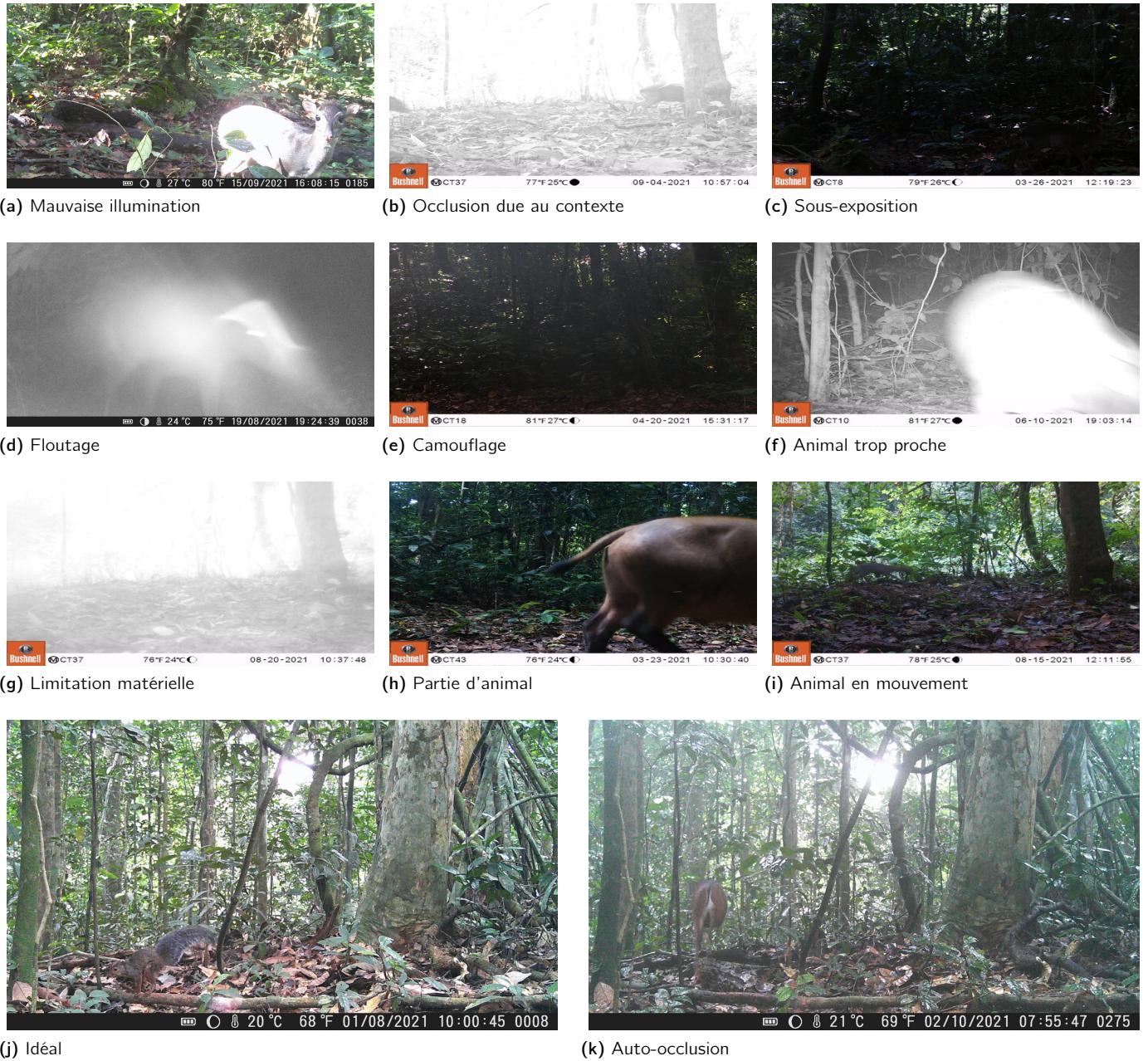


Figure 3.4 : Exemples de scénarios possibles de classification.

CHAPITRE 3. DESCRIPTION DES DONNÉES ET CONTEXTE

[GSV16] Gomez, Salazar et Vargas, « Towards automatic wild animal monitoring : Identification of animal species in camera-trap images using very deep convolutional neural networks »

les pièges photographiques sont déployés. Comme ils sont déployés dans des environnements naturels, plusieurs objets (troncs, lianes, branches etc.) peuvent occulter l'animal que l'on souhaite échantillonner [GSV16]. De plus, l'environnement n'est pas statique, un endroit qui n'est pas occulté au départ peut le devenir (branches déplacées sous l'effet d'un coup de vent ou d'un animal, etc.). Un autre défi est celui des conditions d'éclairage changeantes, la couleur d'un objet peut changer au cours de la journée et entre différents jours selon l'illumination naturelle. Le jour et la nuit possèdent des conditions d'illumination très différentes et spécifiquement, la transition entre le jour et la nuit peut être à l'origine d'images sur-exposées.

Une quantité importante des images contient uniquement une partie de l'animal à cause des occlusions relatives au contexte ou quand l'animal est trop proche de la caméra. Un autre défi est qu'un animal en mouvement peut paraître flou ou rallongé surtout la nuit en raison du temps d'exposition plus long de la caméra. Par conséquent, la même espèce peut prendre des formes différentes pour l'algorithme. Les animaux vivant en communauté comme les éléphants se déplacent en groupe. Il est alors possible d'avoir plusieurs individus d'une même espèce et la détection des bords devient plus difficile.

Les obstacles liés aux limitations techniques et aux réglages de la caméra se manifestent par des images floues ou des animaux sur-exposés. Les formes et les éventuels motifs sur les fourrures deviennent alors indiscernables, ce qui augmente la difficulté de la tâche de classification pour les individus d'espèces proches.

3.3.2 Le déséquilibre des classes

Le déséquilibre entre la fréquence des différentes classes est un problème courant dans le cas des tâches de détection et de classification d'objets du monde réel. Selon l'abondance des espèces, leur détectabilité et leur comportement vis-à-vis des pièges les images sont bien plus abondantes pour certaines que pour d'autres.

Le jeu de données à disposition possède une grande hétérogénéité au niveau des fréquences des espèces. Une différence importante dans le nombre d'images des classes est reportée dans la Table 3.1 entraînant un **déséquilibre de classes** [Lop+13]. L'inconvénient d'un tel déséquilibre est que le modèle apprendra en priorité les caractéristiques des classes les plus représentées, et les classes sous représentées, ne disposant autant de richesse contextuelle, ne seront pas assez observées et le modèle échouera à apprendre leurs caractéristiques discriminantes. Dans un cas extrême de déséquilibre le modèle pourrait se contenter de toujours prédire la classe la plus représentée, il aurait ainsi un bon score tout en étant totalement inutile sur le plan opérationnel. Dans le contexte du suivi de la biodiversité c'est un problème particulièrement important puisque les espèces les plus rares sont généralement celles que l'on cherche à protéger en priorité et donc à identifier de façon fiable [Nor+]. Dans le cas du ██████████, ce sont les grands singes, particulièrement les gorilles et les chimpanzés sur lesquels l'attention est focalisée de par leur statut de conservation. Toutefois, plusieurs solutions ont été développées pour remédier à ce problème [BMM18]. Deux méthodes sont communément utilisées, la première intervenant au niveau de l'échantillonnage, et la seconde,

[Lop+13] López et al., « An insight into classification with imbalanced data : Empirical results and current trends on using data intrinsic characteristics »

[Nor+] Norouzzadeh et al., « Automatically identifying wild animals in camera trap images with deep learning »

[BMM18] Buda, Maki et Mazurowski, « A systematic study of the class imbalance problem in convolutional neural networks »

3.3. LES DÉFIS DE LA CLASSIFICATION D'IMAGES DE PHOTO-PIÉGEAGE

au niveau de la définition de la métrique de coût [Lóp+13 ; BMM18].

L'allégement par échantillonnage

L'approche de l'échantillonnage intervient au niveau du pré-traitement des données et consiste en la modification du jeu de données original de sorte à mitiger le déséquilibre entre les classes. Les images peuvent être (*i*) sur-échantillonnées, (*ii*) sous-échantillonnées ou (*iii*) un mélange des deux. Le sous-échantillonnage consiste à abaisser la probabilité d'échantillonner les images des classes les plus représentées. À l'inverse, le sur-échantillonnage consiste à augmenter la probabilité d'échantillonner les images des classes les moins représentées. Une pratique commune¹⁸ dans les problèmes de vision par ordinateur quand on est dans l'impossibilité d'effectuer du sur-échantillonnage est de recourir à la duplication des images des classes sous-représentées en appliquant des transformations élémentaires (rotation, translation, etc.) de sorte à introduire de la variabilité dans le jeu de données

¹⁸ Et sans doutes, avec des défauts

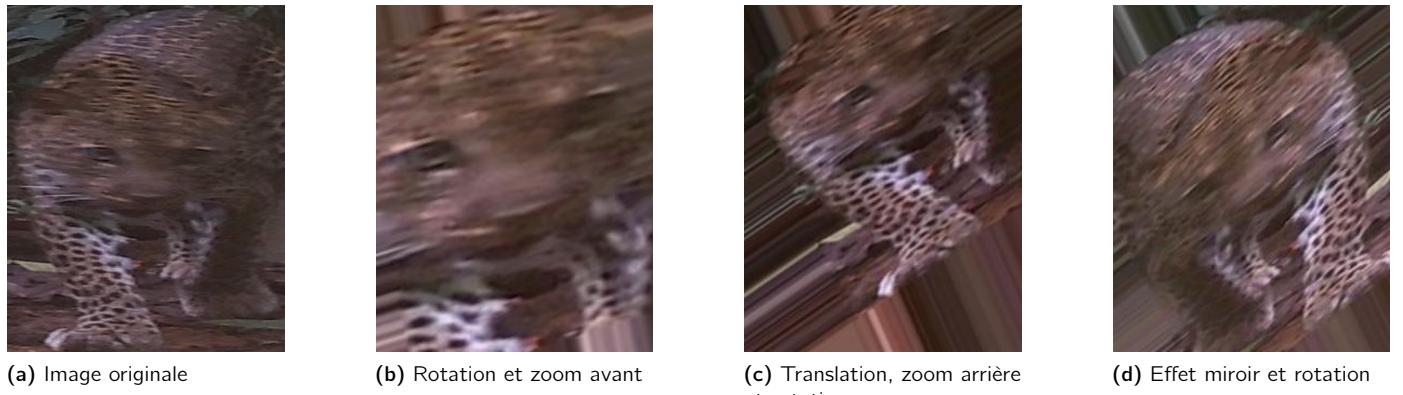


Figure 3.5 : Exemple de sur-échantillonnage en appliquant une combinaison aléatoire de transformations élémentaires.

Les méthodes de ré-échantillonnage présentent des inconvénients. Dans le cas du sous-échantillonnage, des données potentiellement utiles se retrouvent écartées, tandis que dans le cas du sur-échantillonnage, le modèle risque de sur-ajuster sur les exemples des classes les moins représentées et par conséquent échouera dans la généralisation de ces classes [Lóp+13].

L'allégement par la pondération des poids des classes

Pour les tâches de classification, la performance d'un modèle est souvent définie par la proportion d'exemples correctement classifiés [Nor+]. Cette méthode intervient tant au niveau des données qu'au niveau algorithmique de manière à donner plus d'importance aux classes les moins représentées en associant pour chaque classe, un poids proportionnel à son nombre d'images. Par conséquent, pendant l'apprentissage, une pénalité plus importante est appliquée lorsque le modèle se trompe dans la classification des classes les moins représentées.

Notons par N le nombre d'images totales et par N_i le nombre d'images de classe i . L'importance de la classe i est définie par

$$f_i = \frac{N}{n_i}. \quad (3.1)$$

Le poids de la classe i est alors

$$w_i = \frac{f_i}{\sum_{j=1}^N f_j} \quad (3.2)$$

[Nor+] Norouzzadeh et al., « Automatically identifying wild animals in camera trap images with deep learning »

Toutefois, cette approche peut résulter en des gradients extrêmement petits ou extrêmement grands. Par conséquent le processus d'apprentissage peut en être négativement impacté. Il est possible de limiter cet effet en bornant les gradients dans un certain intervalle [Nor+]. Dans ce papier, les auteurs ont observé que cette méthode peut améliorer les scores de prédiction pour les classes rares sans pour autant baisser les scores des autres classes.

3.4 Préparation des données pour la classification

Dans un objectif d'amélioration des performances du modèle de classification, le pré-traitement des données est une étape nécessaire. La stratégie adoptée par Biotope consiste en une investigation manuelle minutieuse des images de l'ensemble des campagnes de photo-piègeage. Ce processus bien que laborieux et chronophage, puise sa légitimité dans la nécessité d'alimenter le modèle d'un jeu de données contenant le moins d'erreurs possibles. La Figure 3.6 ci-dessous décrit les étapes suivies pour le pré-traitement des données.

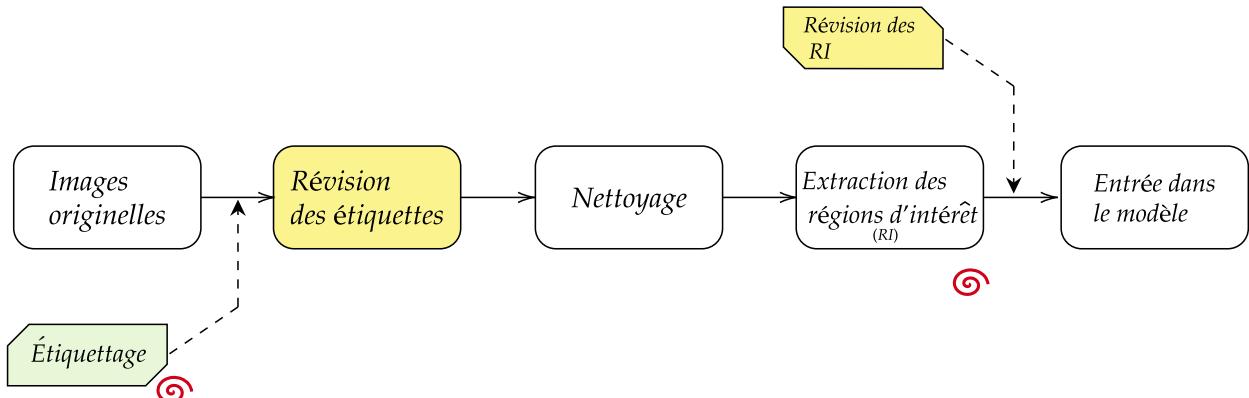


Figure 3.6 : Schéma illustratif des étapes de la phase de pré-traitement des données. La spirale rouge indique la nécessité d'une inspection manuelle.

3.4.1 Révision et nettoyage

Les étiquettes fournies avec le jeu de données ont fait l'objet d'une révision manuelle afin de corriger les éventuelles erreurs d'étiquetage et éliminer les images corrompues.

D'autre part, certaines caméras présentaient de problèmes matériels résultant

3.4. PRÉPARATION DES DONNÉES POUR LA CLASSIFICATION

en des images complètement noires. Le problème des images noires était présent dans différents pièges pour deux des trois campagnes. Les images non noires possèdent un large panel de couleurs et par conséquent, une variation marquée dans les valeurs des pixels. Les images noires en contrepartie possèdent une petite variation dans les valeurs des pixels.

Une heuristique a été développée sur la base de ce constat afin d'automatiser l'annotation des images noires :

Les images, après conversion en niveaux de gris, ont été partitionnées en n blocs et leur variance¹⁹ a été calculée. Les variances résultantes ont été considérées comme un seul vecteur pour lequel l'écart-type a servi de discrimination en introduisant un seuil qui sépare les images noires des images non-noires. La Figure 3.7 présente un schéma illustratif.

¹⁹ Les blocs ont été considérés comme un vecteur.

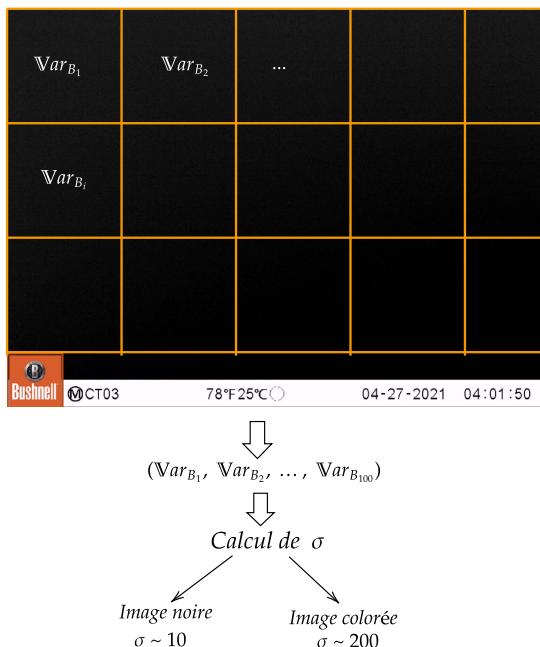


Figure 3.7 – Schéma illustratif de la méthode utilisée pour l'annotation automatique des images noires.

Le seuil a été déterminé grâce à une évaluation de 700 images dont 450 noires et 250 non-noires. Parmi les images classées comme noires, un taux d'erreur de 3,8% a été relevé. Une inspection a révélé que les images mal classées étaient très sous-exposées ou complètement blanches. Parmi les images classées comme non-noires, aucune n'était noire.

3.4.2 Extraction des régions d'intérêt

Le photo-piégeage est une méthode non-invasive destinée à capturer les individus dans leur milieu naturel. Par conséquent, les images récoltées capturent également la végétation autour de l'animal ainsi que son arrière-plan. Les images telles que récoltées peuvent servir d'entrée au modèle de classification. Toutefois, les images brutes capturant également la végétation autour de l'animal ainsi que son arrière-plan, contiennent par conséquent

CHAPITRE 3. DESCRIPTION DES DONNÉES ET CONTEXTE

[RSG16] Ribeiro, Singh et Guestrin, « " Why should i trust you ?" Explaining the predictions of any classifier »

[GSV16] Gomez, Salazar et Vargas, « Towards automatic wild animal monitoring : Identification of animal species in camera-trap images using very deep convolutional neural networks »

20 i.e Les régions d'intérêt où se trouve l'animal.

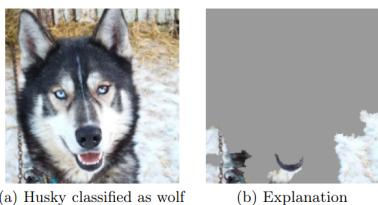


Figure 3.8 : Illustration des caractéristiques utilisées dans la discrimination entre un Husky et un loup. (a) Image d'un Husky classé comme loup. (b) Caractéristiques discriminatives de la décision.

[Sha18] Shane, « Do neural nets dream of electric sheep »

21 i.e détection des animaux

[Bou14] Bouwmans, « Traditional and recent approaches in background modeling for foreground detection : An overview »

[SBB14] St-Charles, Bilodeau et Bergevin, « SubSENSE : A universal change detection method with local adaptive sensitivity »

[BMY19] Beery, Morris et Yang, « Efficient Pipeline for Camera Trap Image Review »

22 Faster Recurrent Convolutional Neural Network



(a) caption1

Figure 3.9 : Exemples de sorties du MegaDetector. La classe des objets est codée sous forme de couleur. (Rouge) Animal. (Bleu) Humain. (Vert) Véhicule.

de l'information non-discriminative entre les espèces. Il a été conjecturé et observé empiriquement que l'alimentation de modèles de classification par des images contenant uniquement de l'information discriminative permettait de réduire significativement les erreurs de classification [RSG16]. Les auteurs dans [GSV16] ont comparé dans un contexte de photo-piégeage, les performances de plusieurs réseaux de neurones convolutifs pour la tâche de classification entre un jeu de données brut et un jeu de données constitué d'images segmentées²⁰ manuellement. Sur toutes les architectures testées, la performance était meilleure sur le jeu de données des images segmentées. Dans [RSG16], les auteurs ont constaté que leur classifieur pouvait être induit en erreur par la présence de neige dans l'image lors de la discrimination entre les loups et les huskys comme illustré dans la Figure 3.8.

D'autres exemples de tels scénarios peuvent être trouvés dans [Sha18] où l'arrière-plan et le voisinage de l'animal contribuent à tort dans la discrimination entre les classes.

Ainsi il est recommandé d'extraire les régions d'intérêt des images brutes dans le but de renforcer le pouvoir discriminatif du modèle. Comme les pièges photographiques sont statiques, l'extraction des régions d'intérêt²¹ pourrait être considérée comme un problème de détection de changement ou de détection d'avant-plan. La détection de changements et/ou d'avant-plan par rapport à l'arrière-plan est un problème bien étudié [Bou14 ; SBB14].

Afin de détecter les animaux dans les pièges photographiques, nous avons utilisé la version 4 du MegaDetector développé par Microsoft [BMY19], un modèle de détection capable de reconnaître les animaux des humains et des véhicules. Le MegaDetector est un réseau de neurones convolutif profond basé sur une architecture Faster-RCNN²² [Ren+15]. Il prend en entrée les images brutes et associe à chaque image, un rectangle délimitant le ou les objets détectés, un niveau de confiance compris entre 0 et 1 et une classe parmi [Animal, Humain, Véhicule] comme illustré dans la Figure 3.9.



(b) caption2

3.4. PRÉPARATION DES DONNÉES POUR LA CLASSIFICATION

Un premier test a été réalisé sur 3999 images²³ issues de la première campagne de photo-piégeage dans une optique d'évaluation des performances du modèle. Nous illustrons dans la Table 3.2 ci-dessous la matrice de confusion croisant les prédictions du modèle contre les étiquettes du jeu de données. Les résultats de la classification sont résumés sous la forme d'une matrice de confusion des vrais positifs (VP), vrai négatif (VN), faux positifs (FP), faux négatifs (FN).

Table 3.2 – Matrice de confusion des résultats du MegaDetector.

		Détections du modèle (Animal, humain)	
Total 3999		Faux	Vrai
Etiquettes véritables (Animal, humain)	Faux	1403	89
	Vrai	15	2492

Le modèle a été également évalué grâce aux métriques définies dans la Section ?? avec une précision de 97% et un rappel de 99%. Toutefois, l'évaluation du modèle a été basée sur sa capacité à détecter au moins un objet dans une image pour laquelle la véritable étiquette correspond à un humain ou un animal. Ainsi une image contenant plus d'un animal hérite de l'étiquette de l'image originale. Un autre cas de figure est que le MegaDetector détecte plusieurs objets (une feuille et une panthère, par exemple) dans une image qui contient réellement qu'un seul animal (une panthère). Dans ce cas, les deux détections héritent de l'étiquette de l'image originale. La Figure 3.8 ci-dessous illustre différents scénarios de détections.



(a) Échec de détection (Céphalope)



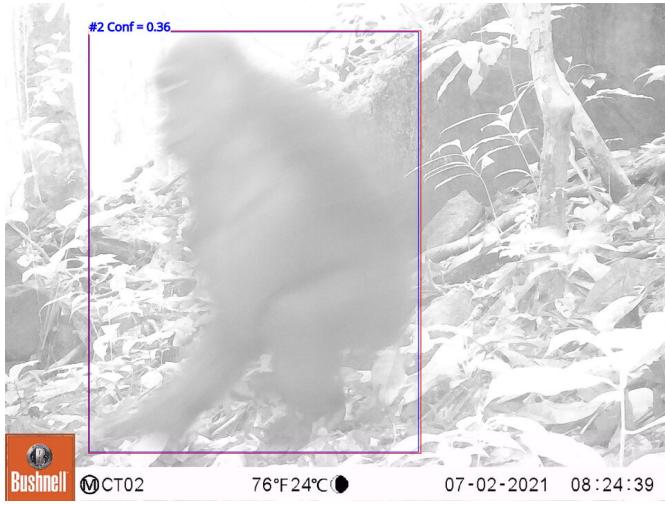
(b) Une bonne détection et un faux positif (Oiseau, branche)

²³ La totalité des images n'était pas encore livré au moment de la réalisation du test.

CHAPITRE 3. DESCRIPTION DES DONNÉES ET CONTEXTE



(c) Un cas difficile (Athérure)



(e) Deux catégories pour le même objet (Chimpanzé)



(g) Difficulté à déterminer l'animal (Céphalophe)



(d) Cas difficile avec occlusion de l'animal (Athérure)



(f) Difficulté à déterminer l'animal à cause du contexte (Hocheur)



(h) Deux détections pour le même animal (Civette)

3.4. PRÉPARATION DES DONNÉES POUR LA CLASSIFICATION

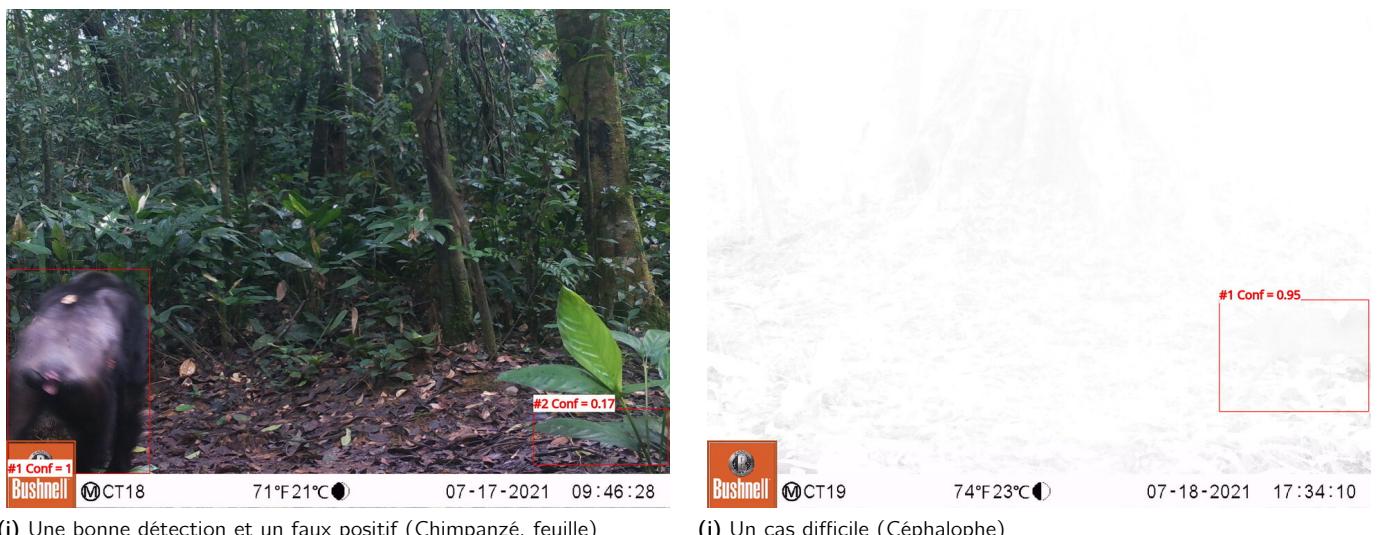


Figure 3.8 : Exemples de scénarios de détections du MegaDetector.

Ainsi, les scores du MegaDetector citées ci-dessus sont à nuancer. Elles évaluent le modèle seulement au niveau de l'image complète et non au niveau des détections. Nous avons alors procédé à une révision manuelle des détections de la troisième campagne de photo-piégeage et avons divisé les détections en trois catégories : (i) Les bonnes détections, des détections où l'animal/partie de l'animal est repérable à l'oeil nu, (ii) Les mauvaises détections où l'animal n'est pas clairement reconnaissable²⁴, (iii) Les faux positifs regroupant toutes les détections ne correspondant pas à un animal. Ci-dessous nous illustrons les distributions des niveaux de confiance des détections de ces trois catégories.

²⁴ Cette catégorie est basée sur un point de vue subjectif de ce qu'est une mauvaise détection.

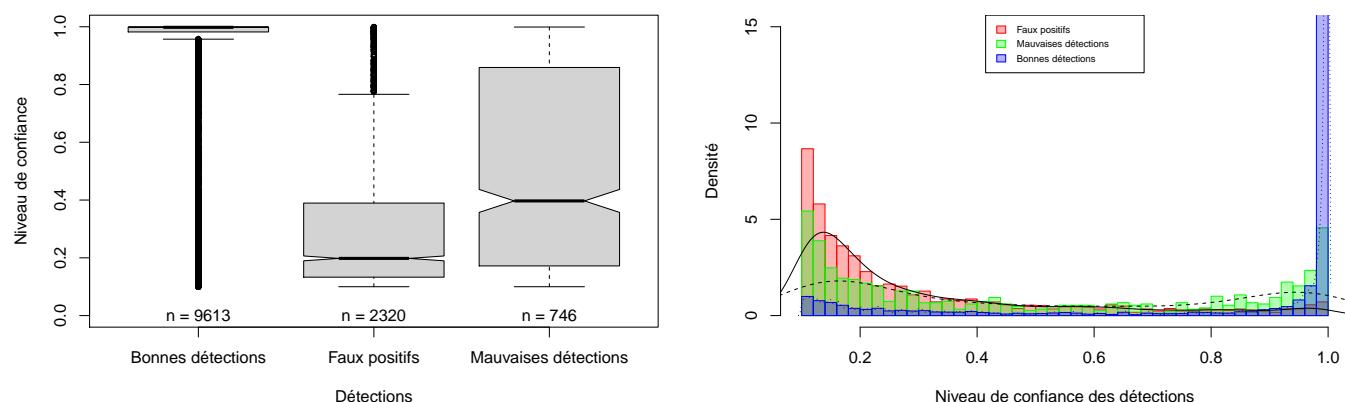


Figure 3.9 : Distribution des niveaux de confiances en fonction des trois catégories de détection pour la troisième campagne de photo-piégeage.

La distribution des niveaux de confiance du détecteur pour les bonnes détections est très étroite. La majorité se caractérisant par un niveau de confiance

CHAPITRE 3. DESCRIPTION DES DONNÉES ET CONTEXTE

élevé. A l'inverse les faux positifs, se caractérisent par un niveau de confiance de détection plutôt bas. Toutefois, la discrimination entre les faux positifs et les bonnes détections basée uniquement sur le niveau de confiance ne peut pas être fiable à partir du moment où l'on considère la distribution des niveaux de confiance des mauvaises détections. La considération d'autres covariables telle que l'espèce, la surface de la détection, ainsi que l'emplacement des pièges pourrait apporter un éclairage sur les disparités entre ces trois catégories.

CHAPITRE QUATRE

Méthodologie

Nous décrivons dans ce chapitre les différentes étapes suivies pour effectuer la tâche de classification. Nous discutons de la phase préparatoire finale des données, de la procédure suivie pour le choix du modèle qui sera utilisé pour la classification et enfin de l'entraînement et le calibrage du modèle sélectionné.

4.1 Répartition des données pour l'entraînement et l'évaluation du modèle

Après la phase de pré-traitement des données présentée dans la Section 3.4, les données doivent être réparties en une base d'entraînement nécessaire à l'apprentissage et l'entraînement du modèle, et une base de test qui servira uniquement à l'évaluation du modèle. Une partie de la base d'entraînement (environ 10%) appelée données de validation est réservée à la supervision (i.e. au contrôle) de l'apprentissage à travers le suivi de l'évolution de la fonction de coût et par conséquent la détection de la divergence du modèle ou de phénomènes comme le sur-apprentissage ou le sous-apprentissage. Elle sera également utilisée pour la phase de sélection de modèle présentée dans la section suivante.

La phase de découpage des données est illustrée par la Figure 4.1 et sera discutée plus en détail dans le chapitre suivant.

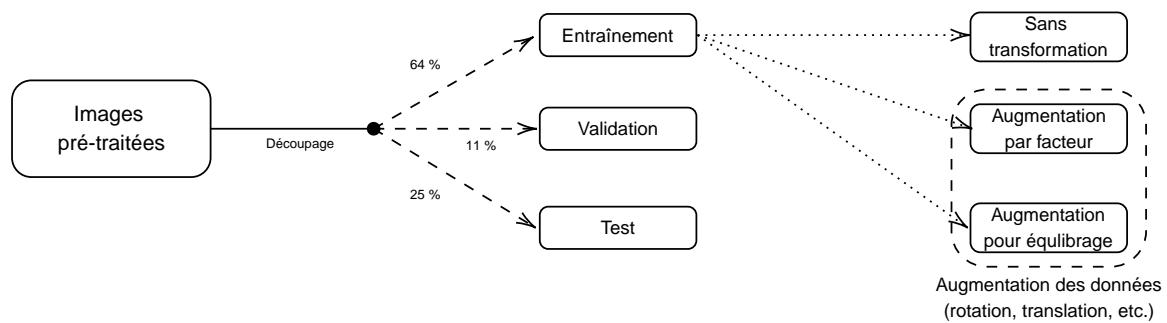
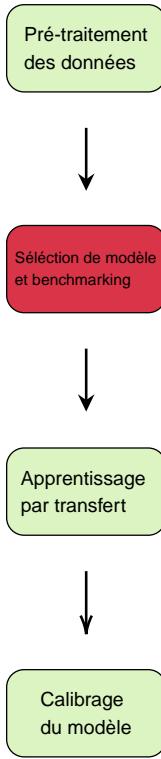


Figure 4.1 : Schéma explicatif de la phase de répartition des données en bases d'entraînement, d'évaluation et de test

4.2 Analyse comparative pour la sélection de modèle



Il existe une multitude de réseaux de neurones pré-entraînés sur de grandes bases de données de référence. Le principal aspect des couches inférieures (se trouvant au début du réseau) réside dans l'extraction des cartes de caractéristiques (feature maps) servant à la construction de représentations latentes des images d'entrées.

Afin de choisir un modèle adapté à notre jeu de données, nous proposons de procéder à une analyse comparative sur les **données de validation** de plusieurs réseaux de neurones d'architectures différentes. Cette analyse vise à reconnaître parmi un ensemble de modèles, celui qui permet l'extraction des meilleures représentations latentes qui maximisent la probabilité d'une reconnaissance correcte d'animaux dans les images. Les modèles considérés dans cette analyse sont pré-entraînés sur les 1000 classes (différentes des nôtres) de la base de données ImageNet regroupant tout type d'objets dont une multitude d'animaux.

Par conséquent, nous avons construit un dictionnaire qui associe à chaque classe de notre jeu de données, une ou plusieurs classes du modèle pré-entraîné sur ImageNet sur la base d'un ensemble de critères : (i) similarités taxonomiques, (ii) similarités morphologique, (iii) enjeux de conservation. Par exemple, un athérure sera associé à des espèces de rongeur taxonomiquement proches comme le porc-épic. Un céphalophe sera associé à quelques classes de cervidés. Les classes communes à notre jeu de données et à celui d'ImageNet seront en correspondance directe. Pour les espèces présentant un enjeu de conservation important et communs aux classes d'ImageNet, la correspondance est alors unique.

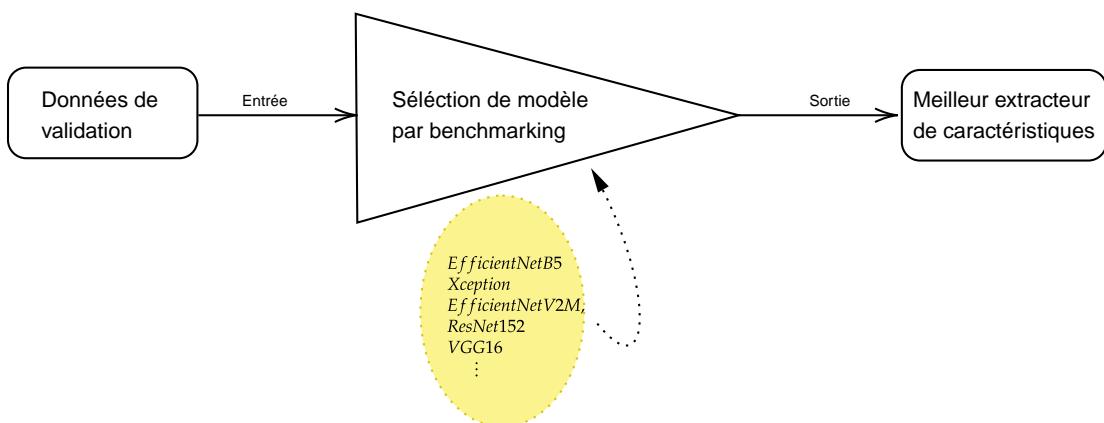


Figure 4.2 : Schéma illustratif de la phase de sélection de modèle.

A l'aide du dictionnaire établi, différents modèles sont évalués sur la base des métriques de précision et de rappel (voir Section 5.3), et le modèle présentant le meilleur score, sans entraînement préalable sur notre jeu de données, servira à la construction de la partie "d'inférence de représentations latentes" de notre propre réseau de neurones. Toutefois, cette approche n'est pas sans défauts car l'établissement de telles correspondances implique nécessairement l'introduction d'un biais de subjectivité dans l'appréciation

4.3. CONSTRUCTION DU RÉSEAU DE NEURONES ET APPRENTISSAGE PAR TRANSFERT

des similarités entre les classes des deux jeux de données.

4.3 Construction du réseau de neurones et apprentissage par transfert

Le modèle sélectionné servira à la construction de notre propre modèle qui sera adapté aux nombre de classes dont nous disposons. Il jouera le rôle d'extracteur de caractéristiques. Le modèle que nous construisons sera alors une jonction de la partie d'extraction de caractéristiques (modèle sélectionné) avec une partie de couches de neurones entièrement connectées (appelées couches denses.) qui servira de classifieur.

L'architecture que nous proposons est comme suit :

1. Partie du modèle sélectionné duquel nous ôtons toutes les couches supérieures servant à la classification, et dont les poids seront figés (i.e ne changeront pas durant l'entraînement).
2. Une couche de sous-échantillonnage global 2D (c.f Section 2.4.1) qui est une opération qui prend en entrée une carte de caractéristiques et donne en sortie, sa moyenne globale (moyenne sur les lignes, puis sur les colonnes). Cette couche possède alors autant de neurones que de cartes de caractéristiques.
3. Une couche de neurones dense plus petite, entièrement connectée à la couche issue de l'opération de sous-échantillonnage global, et qui sert d'agrégation de caractéristiques.
4. La couche finale sera alors une couche dense possédant autant de neurones que de nombre de classes, suivie de la fonction d'activation **softmax** (comme expliqué dans la Section 2.4.2. Eq 2.16)

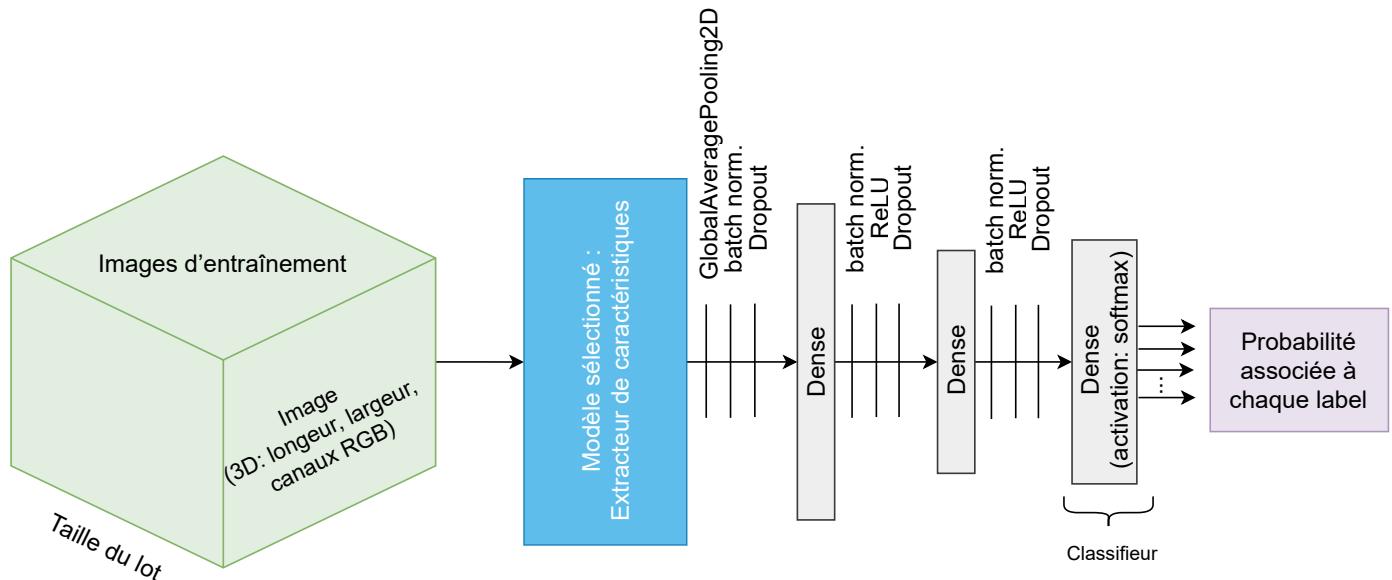
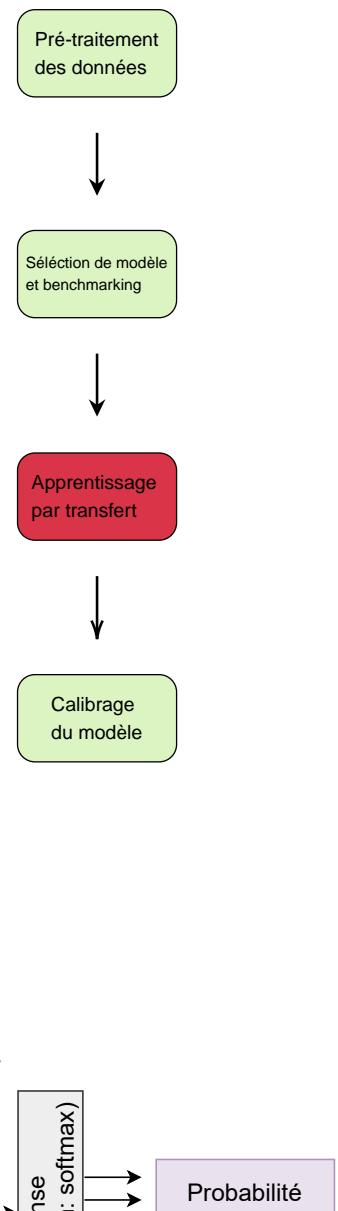


Figure 4.3 : Construction de l'architecture du réseau de neurones adapté à notre jeu de données.

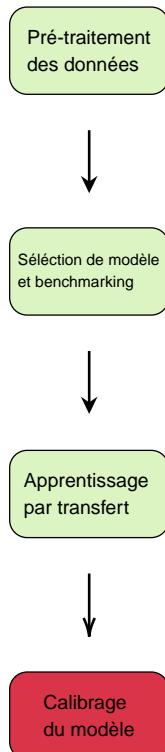
CHAPITRE 4. MÉTHODOLOGIE

- [IS15] Ioffe et Szegedy, « Batch normalization : Accelerating deep network training by reducing internal covariate shift »
 [Shi00] Shimodaira, « Improving predictive inference under covariate shift by weighting the log-likelihood function »
 [Sri+14] Srivastava et al., « Dropout : a simple way to prevent neural networks from overfitting »

Les blocs (2) et (3) commencent par une couche de normalisation par lot afin d'atténuer le problème du décalage interne des covariables (internal covariate shift) et de permettre l'utilisation de taux d'apprentissage plus élevés pour accélérer l'apprentissage [IS15 ; Shi00]. Ils sont suivis d'une fonction d'activation ReLU (Eq 2.5). Cette dernière passe par une couche de **régularisation par abandon** (dropout) qui fixe aléatoirement une fraction des neurones à zéro lors de la phase d'entraînement afin d'éviter le sur-ajustement en simulant différentes architectures du réseau [Sri+14]. La Figure 4.3 illustre l'architecture du réseau de neurones que nous proposons.

Entraînement du modèle et l'apprentissage par transfert L'entraînement du réseau de neurones se fait à l'aide de l'apprentissage par transfert, qui consiste à entraîner la seconde partie du modèle en figeant les poids de la partie d'inférence de caractéristiques du modèle. L'apprentissage par transfert est un paradigme d'apprentissage qui s'applique à différents modèles d'apprentissage automatique. Son utilisation tire son avantage du fait que les architectures existantes sont entraînées sur des bases de données gigantesques et par conséquent, leur partie d'inférence de caractéristiques est un excellent moyen de reconnaissance de formes et d'extraction de représentations latentes pertinentes qui conviennent à la tâche de classification.
 Enfin, les hyperparamètres liés à l'entraînement du modèle sont discutés dans le chapitre suivant.

4.4 Calibrage du modèle



Une fois la seconde partie du modèle est entraînée pour la tâche de classification, il est possible de calibrer les couches de la partie d'inférence de caractéristiques du modèle sélectionné dans une tentative d'obtenir un ajustement des couches d'extraction de caractéristiques plus adapté à notre jeu de données. A contrario, l'entraînement du modèle en entier peut négativement impacter les couches d'extraction de caractéristique en propageant les erreurs de prédictions trop grandes, car ces couches ne sont pas encore optimisées [GBC16]. Le calibrage du modèle se fait donc en fixant un taux d'apprentissage et un nombre d'époques très petits afin d'éviter l'occurrence d'une trop grande perturbation des paramètres de la base d'inférence de caractéristiques.

[GBC16] Goodfellow, Bengio et Courville, *Deep learning*

CHAPITRE CINQ

Résultats et discussion

Dans cette section, nous présentons les résultats de l'analyse comparative pour la sélection de modèles ainsi que les scores de l'apprentissage par transfert (Transfer Learning) et la mise au point ²⁵ du meilleur modèle (Finetuning). Pour des raisons de limitations matérielles, nous avons décidé de reporter nos résultats sur une sous partie de l'ensemble des images, à savoir, la troisième campagne de photo-piégeage, qui, à notre sens a fait l'objet de l'inspection la plus rigoureuse des trois campagnes. Le nombre d'espèces totales pour cette campagne étant au nombre de 31 et dont la répartition des fréquences est très déséquilibrée, nous avons procédé à un recodage de certaine sespèces en groupes taxonomiques proches, et avons éliminé les classes contenant moins de 30 images de la phase d'entraînement. La Figure 5.1 résume les fréquences des détections pour la troisième campagne de photo-piégeage.

²⁵ i.e calibrage.

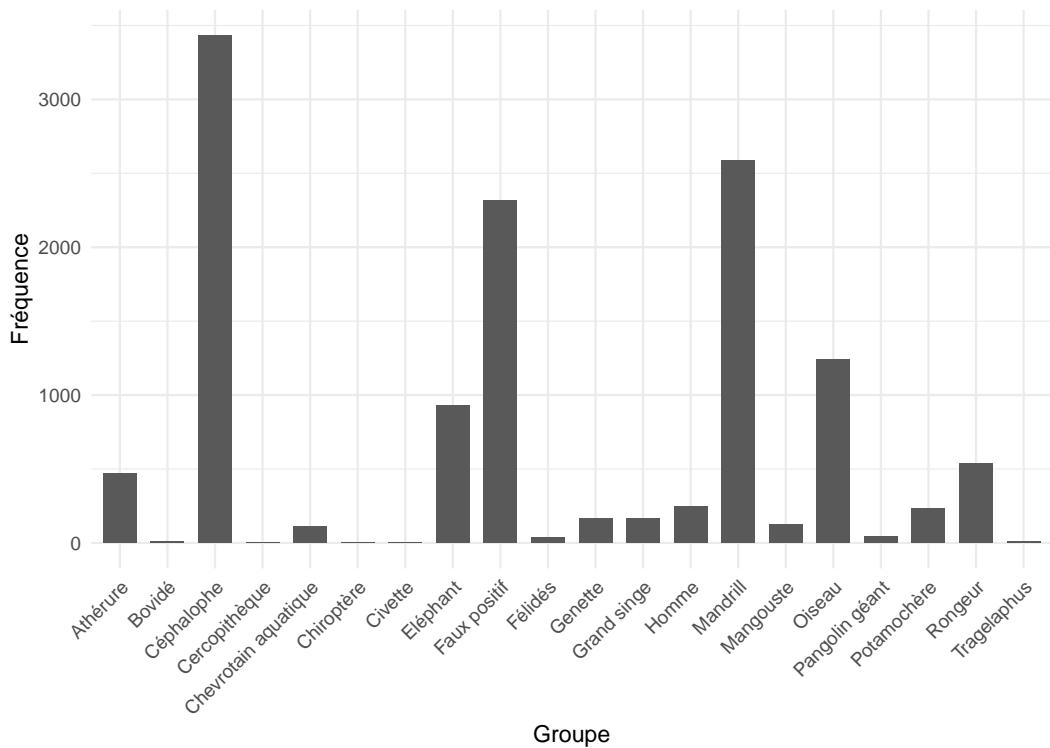


Figure 5.1 : Fréquences des classes de la troisième campagne de photo-piégeage

5.1 Données d'apprentissage, augmentation des images, et données d'évaluation

Les données sont divisées en deux sous-ensembles : 75% des données sont réservées pour l'apprentissage (64%) et la validation (11%), et 25% des images sont réservées pour l'évaluation des modèles entraînés. Dans les deux sous-ensembles, les images sont sélectionnées aléatoirement et uniformément à partir de chaque classe. Les données de test sont réservées exclusivement à l'évaluation ; elles ne sont jamais utilisées ni au cours de l'apprentissage pour adapter les paramètres du modèle (rétro-propagation), ni pour le suivi et la validation de l'apprentissage.

Pour étudier l'effet de l'augmentation des données sur la performance du modèle, nous avons construit à partir de l'ensemble des données d'entraînement deux nouveaux ensembles d'entraînement en utilisant le sur-échantillonnage :

- Le premier consiste en l'augmentation des images de chaque classe en un facteur de 16. Ainsi, il est construit, à partir de chaque image, 16 images augmentées de façon aléatoires selon le processus expliquée dans la Section 3.3.2. Dans ce qui suit, nous appelons cet ensemble d'apprentissage : “**données augmentées**”.
- Le second consiste en l'augmentation des images de telle sorte à équilibrer le nombre d'image par classe. Ainsi, pour chaque classe, nous générions des images d'augmentation de façon aléatoires (cf. Section 3.3.2 jusqu'à ce que le nombre d'image de la classe atteint le nombre d'image de la classe la plus représentées. Il est à noter que la classe la plus représentée n'est pas augmentée. Dans ce qui suit, nous appelons cet ensemble d'apprentissage : “**données équilibrées**”.

De ce fait, chaque modèle est entraîné séparément sur les trois ensembles d'apprentissage : données originales, données augmentées et données équilibrées, et les scores sont rapportés pour les trois ensembles. Quand le modèle est entraîné sur les données déséquilibrées (c'est-à-dire originales et augmentées), nous effectuons pendant l'entraînement un allégement du déséquilibre des classes par pondération tel que décrit dans la Section 3.3.2.

Il est à noter qu'une deuxième méthode d'augmentation des données consiste à intégrer des couches de transformations aléatoires d'images au modèle, et d'entraîner le modèle sur un nombre important d'époques. Cependant, cette méthode possède deux inconvénients : (i) les données d'entraînement restent déséquilibrés, ce qui pourrait biaiser le modèle en faveur des classes les mieux représentées, (ii) l'apprentissage devient long et coûteux car le nombre d'époques d'apprentissage est élevé.

5.2 Entraînement du modèle et choix des hyperparamètres

Pour entraîner le modèle, nous avons utilisé l'algorithme d'apprentissage Adam [KB14] avec un taux d'apprentissage initial de 0.01. L'entraînement s'est effectué sur 300 époques avec une stratégie d'arrêt prématurée (early

[KB14] Kingma et Ba, « Adam : A method for stochastic optimization »

5.3. MÉTRIQUES D'ÉVALUATION

stopping). Il s'agit d'une méthode de régularisation qui interrompt le processus d'entraînement du modèle lorsque la fonction coût calculée sur les données de validation ne change plus ou commence à diverger de celle calculée sur les données d'entraînement. La taille du lot d'apprentissage (mini-batches) utilisée est de 32 observations. Afin de régulariser le réseau de neurones, le taux de dropout utilisé est de 25%. Nous avons constaté qu'un taux plus élevé nécessite un apprentissage sur un nombre d'époques plus important. Une meilleure méthode de sélection d'hyper-paramètres consiste en l'utilisation d'une grille de recherche (grid search) avec une validation croisée sur plusieurs sous-ensembles (par exemple 10 sous-ensembles de validation croisée). Cependant, l'implémentation de cette méthode sur un problème de classification d'images, ou plus généralement sur des données de grande tailles, bien qu'étant simple, nécessite des ressources de calcul très avancées.

Concernant les technologies utilisées, nous avons opté pour le paquetage Keras de TensorFlow et nous avons utilisé son API Python en vertue de sa simplicité d'utilisation et sa documentation avancée et intuitive.

Pour contourner le chargement des données d'apprentissage en mémoire, qui est impossible à cause des limitations de ressources, nous avons utilisé l'API tensorflow.data.Dataset, qui permet de charger les images par lots (mini-batch) et d'effectuer les transformations nécessaires à la volée pendant l'entraînement.

5.3 Métriques d'évaluation

Pour évaluer les performances des modèles entraînés, nous utilisons les métriques de classifications classiques, à savoir la précision, le rappel et le score F1. Ces scores sont rapportés pour chaque classe i , selon les formules suivantes :

$$\text{précision}_i = \frac{\text{nombre d'images correctement attribuées à la classe } i}{\text{nombre de prédictions de la classe } i} \quad (5.1)$$

$$\text{rappel}_i = \frac{\text{nombre d'images correctement attribuées à la classe } i}{\text{nombre d'images appartenant à la classe } i} \quad (5.2)$$

$$F1_i = \frac{2 \times \text{précision} \times \text{rappel}}{\text{précision} + \text{rappel}} \quad (5.3)$$

Ensuite, la performance du modèle est évaluée en rapportant les moyennes globales (pondérées par le nombre d'images par classe) de la précision, du rappel et du score F1 sur l'ensemble des 15 classes.

$$\text{précision} = \sum_{i=1}^n \frac{n_i \times \text{précision}_i}{n} \quad (5.4)$$

$$\text{rappel} = \sum_{i=1}^n \frac{n_i \times \text{rappel}_i}{n} \quad (5.5)$$

CHAPITRE 5. RÉSULTATS ET DISCUSSION

$$F1 = \sum_{i=1}^n \frac{n_i \times F1_i}{n} \quad (5.6)$$

où n_i est le nombre d'images appartenant à la classe i et n est le nombre total d'images.

Enfin, nous rapportons l'exactitude du modèle :

$$\text{exactitude} = \frac{\text{nombre d'images correctement classifiées}}{\text{nombre totale d'images}} \quad (5.7)$$

5.4 Résultats de sélection de modèles (benchmarking)

Comme discuté dans la Section 4.2, nous évaluons un ensemble de modèles pré-entraînés sur la base de données "ImageNet". Nous utilisons ces modèles dans leurs formats bruts, en utilisant les poids inférés sur "ImageNet", sans entraînement préalable, et ce pour sélectionner le modèle capable d'extraire les meilleures représentations latentes des images d'entrées. Le Tableau 5.1 contient les scores de classification de chaque modèle utilisé dans l'analyse comparative.

Modèle	Xception	EffNetB3	EffNetB5	EffNetB7	EffNetV2L	EffNetV2M	EffNetV2S	InceptionResNetV2	ResNet152	VGG16	VGG19
Précision	0.95	0.94	0.92	0.92	0.92	0.94	0.92	0.95	0.90	0.91	0.90
Rappel	0.32	0.27	0.28	0.23	0.41	0.32	0.29	0.31	0.25	0.17	0.18
F1	0.47	0.41	0.42	0.36	0.56	0.47	0.42	0.47	0.39	0.29	0.29

Table 5.1 : Résultats de l'analyse comparative sur les données de validation

Ainsi, le modèle montrant le meilleur score F1 est "EfficientNetV2L". Cependant, ce modèle comporte 119 millions de paramètres, ce qui rend les tâches d'extraction des représentations latentes d'images et d'adaptation du modèle (fine-tuning) très coûteuse en termes de calcul. En constatant que la différence de performance entre "EfficientNetV2L" et ses variantes moins volumineuses "EfficientNetV2M" et "EfficientNetV2S" est minime, et tenant compte de la complexité des modèles "Xception" et "InceptionResNetV2" en termes de nombre de paramètres et de vitesse de convergence, nous avons opté pour l'utilisation du modèle "EfficientNetV2S" qui comporte 21 millions paramètres entrainables. Ce dernier est 81% plus petit que "EfficientNetV2L", ce qui permet un gain considérable en termes de ressources de calcul contre une différence légère de performance.

5.5 Évaluation du modèle sélectionné : EfficientNetV2S

Dans ce qui suit, nous rapportons les résultats d'évaluation du modèle "EfficientNetV2S", sur les trois jeux de données, à savoir les données originales, les données augmentées et les données équilibrées.

5.5. ÉVALUATION DU MODÈLE SÉLECTIONNÉ : EFFICIENTNETV2S

Le Tableau 5.2 contient une vue globale sur les scores du modèle EfficientNetV2S.

	Données originales	Données augmentées	Données équilibrées
Précision	0.81	0.82	0.84
Rappel	0.81	0.81	0.84
F1	0.81	0.82	0.84
Exactitude	0.81	0.82	0.85

Table 5.2 – Performances du classifieur en utilisant les représentations latentes inférées par EfficientNetV2S. Les résultats sont rapportés sur les données d'évaluation (testing set).

Nous constatons que le classifieur le plus performant est celui entraîné sur un jeu de données équilibré, et dont l'exactitude atteint 85%. Ceci permet de déduire que l'équilibrage des données joue un rôle plus important que le simple fait d'augmenter les images par un certain facteur. Pour avoir une vue plus détaillée de l'évaluation, nous avons calculé les scores précision, rappel et F1 pour chacune des classes. Le Tableau 5.3 rapporte les résultats du modèle entraîné sur les données originales, augmentées et équilibrées. Il est à noter que la colonne "Support" correspond au nombre d'images par classe dans le jeu de données d'évaluation.

	Données originales			Données augmentées			Données équilibrées			Support
	P	R	F1	P	R	F1	P	R	F1	
Athérure	0.66	0.79	0.72	0.69	0.81	0.74	0.77	0.82	0.79	154
Chevrotain aquatique	0.54	0.73	0.62	0.52	0.80	0.63	0.70	0.63	0.67	30
Céphalophe	0.83	0.84	0.84	0.90	0.77	0.83	0.85	0.86	0.86	1000
Eléphant	0.88	0.93	0.90	0.93	0.91	0.92	0.91	0.92	0.91	245
Félidés	0.38	0.82	0.51	0.89	0.73	0.80	1.00	0.64	0.78	11
Genette	0.52	0.53	0.52	0.53	0.55	0.54	0.69	0.67	0.68	51
Grand singe	0.61	0.62	0.62	0.60	0.60	0.60	0.80	0.53	0.64	45
Homme	0.91	0.75	0.82	0.82	0.89	0.85	0.89	0.78	0.84	65
Mandrill	0.90	0.87	0.89	0.86	0.91	0.89	0.85	0.93	0.89	701
Mangouste	0.58	0.60	0.59	0.59	0.57	0.58	0.76	0.46	0.57	35
Oiseau	0.84	0.81	0.82	0.85	0.83	0.84	0.87	0.80	0.84	333
Pangolin géant	0.40	0.17	0.24	0.60	0.25	0.35	0.83	0.42	0.56	12
Potamochère	0.78	0.74	0.76	0.66	0.84	0.74	0.88	0.74	0.80	68
Rongeur	0.66	0.59	0.62	0.63	0.70	0.66	0.70	0.64	0.67	157
Faux positif	0.79	0.76	0.77	0.69	0.82	0.75	0.76	0.82	0.79	245

Table 5.3 : Performance du classifieur entraîné séparément sur les jeux de données original, augmenté et équilibré. (P) Précision. (R) Rappel.

Nous constatons que plus la classe est représentée, plus sa reconnaissance est probable. Cela est vérifié pour les classes Céphalophe, Mandrill ou encore Oiseau. Aussi, la classe "Eléphant", bien que moyennement représentée, le modèle la prédit avec les scores précision, rappel et F1 les plus élevés. Une inspection visuelle du jeux de données d'entraînement nous a permis de constater que les images d'éléphants sont davantage nettes que celles des autres classes du fait de leur imposante taille et leurs traits facilement reconnaissables ce qui a rendu l'étiquetage manuel effectué au préalable plus

CHAPITRE 5. RÉSULTATS ET DISCUSSION

exacte. Cependant, le modèle échoue dans la prédiction des classes les moins représentées, telles que "Félidés" et "Pangolin géant", et ce lorsque le modèle est entraîné sans augmentation d'images.

Il est donc important de noter que la collecte d'un grand nombre d'images et un étiquetage de bonne qualité sont deux conditions primordiales pour l'obtention d'un modèle de classification performant.

Enfin, afin d'avoir une idée encore plus détaillée sur les résultats de classification, nous mettons en annexe les matrices de confusions.

5.5.1 Adaptation des paramètres (finetuning)

Pour améliorer la qualité du modèle tel que préconisé dans la littérature, nous avons sélectionné le meilleure modèle, à savoir "EfficientNetV2S + classifieur" entraîné sur le jeux de données équilibré, et nous l'avons ré-entraîné sur ce même jeux de données, mais en permettant l'adaptation des paramètres de l'extracteur de caractéristiques "EfficientNetV2S", jusqu'ici ayant les paramètres figés. Le ré-entraînement a été effectué en utilisant l'algorithme d'apprentissage SGD (Stochastic Gradient Descent) avec un taux d'apprentissage (learning rate) très bas de 10^{-7} et un nombre d'époque aussi très bas (10 époques). Cependant, bien que les paramètres du modèle changent très légèrement, la valeur de la fonction de coût calculé sur les données de validations (validation loss ; categorical crossentropy) chute dès la première époque, et ne remonte plus à cause du taux d'apprentissage très bas (sous-apprentissage).

Des tentatives de résolution de ce problème sont en cours d'évaluation.

CHAPITRE SIX

Livrables pour l'entreprise

En livrable pour l'entreprise, nous avons construit deux modules avec lesquels l'utilisateur peut interagir à travers un terminal Linux. Chaque module possède un argument `--help` qui permet d'accéder à un utilitaire de documentation détaillé.

- Un module dédié au sur-échantillonnage des données à travers leur augmentation *(i)* soit par facteur, *(ii)* soit au niveau de la classe la plus représentée. Ce module est accessible à l'aide du fichier `augmenter.py`
- Un module dédié à la construction de l'architecture proposée à l'aide d'une base de modèles existant , à l'entraînement du modèle résultant, son évaluation, son calibrage. Il est accessible à l'aide du fichier `transfer.py`. Les options supportées sont `train`, `evaluate`, `fintune`, `predict`. L'option `predict` est dédiée à effectuer la tâche de classification sur des éventuelles images non étiquetées
- Un script bash `run.sh` à l'intérieur duquel l'utilisateur peut renseigner les paramètres souhaités pour entraîner un modèle (le nom du modèle dont est issue la base convolutive, le nombre d'époques, etc.). Le script s'occupe alors de lancer le module `transfer.py` avec les arguments renseignés
- Un script bash `train_test_split.sh` qui procède au partitionnement du jeu de données en bases d'entraînement et de test.

La documentation complète de tous les modules et options associés est fournie en annexe.

Conclusions

Le réseau de neurones , bien qu'entraîné sur un petit jeu de données, possède un score global F1 de 84% et d'exactitude de 85%. L'entraînement d'un modèle sur un nombre important d'images est connu pour améliorer ses performances pour peu que l'étiquetage soit correctement fourni. Toutefois, les scores sont à orienter en fonction de l'usage que l'on souhaite faire du modèle, à savoir un outil d'annotation en complète autonomie ou un outil d'aide à l'annotation.

Discussion sur les métriques de précision et de rappel Il est certain que la combinaison du MegaDetector avec un modèle de classification entraîné sur un grand jeu de donnée et correctement étiquetté pourrait apporter un soutien considérable dans la tâche d'annotation, pour peu que l'on dispose d'assez de ressources matérielles. De plus, les scores de précision et de rappel pourraient être utilisés de manière efficace et être orientés selon les enjeux de conservation propres aux espèces. Le score de précision, par exemple, nous renseigne sur la pertinence du modèle à ne pas se tromper lors de sa prédiction d'une espèce particulière, tandis que le rappel nous renseigne sur le rapport entre les prédictions associées à une espèce, et le nombre d'images de cette même espèce. Ce sont deux notions très différentes, et bien que l'on aimeraient avoir sous les mains un modèle performant dans les deux, elles prennent leur sens lorsque la dimension des enjeux de conservations est prise en considération ;

Face à une espèce d'enjeux de conservation importants, souhaiterions nous disposer d'un modèle qui sache la reconnaître quand elle lui est présentée ²⁶, ou c'est justement pour de telles espèces qu'une attention particulière est portée lors de l'investigation visuelle des données ?

A l'inverse, un modèle avec un bon score de précision sur une espèce abondante (et donc, d'enjeux de conservations moins conséquent) serait à notre avis d'une utilité certaine dans l'accompagnement de l'annotation. Il permettrait une réduction considérable du temps consacré à révision des annotations et permettrait aux conducteurs de l'étude d'orienter leurs efforts dans d'autres tâches. Sous ce point de vue là, la métrique de précision semble posséder un grand potentiel. Toutefois, cette métrique à elle seule est loin d'être suffisante, et ce dans beaucoup de cas. Par exemple, un score de précision élevé et un score de rappel très bas signifierai que, malgré la tendance du modèle à ne pas se tromper lorsqu'il prédit une certaine espèce, son taux de réussite total reste tout de même bas en raison de son incapacité à reconnaître l'espèce dans la grande partie des cas.

²⁶ i.e avec un bon score de rappel

Ainsi, à notre sens, la complémentarité de ces deux métriques est frappante, et la valorisation de l'une en faveur de l'autre se doit d'être réfléchie tout en considérant le contexte dans lequel on souhaite les utiliser et également, de leur potentielles conséquences.

Limitations de l'approche suivie

- Dans la phase de sélection de modèle, l'analyse comparative effectuée repose sur l'élaboration d'un dictionnaire qui met en correspondance les classes de notre jeu de données avec celle de la base de donnée d'Imagenet. Une telle construction introduit un biais humain marqué par une appréciation subjective de la ressemblance entre deux espèces, et par conséquent c'est une procédure non-reproductible.
- D'autre part, l'utilisation d'un jeu de données plus grand permettrait probablement d'atteindre de meilleures performances, potentiellement sur les classes les moins représentées. Toutefois, bien qu'un temps considérable a été consacré à la phase de révision manuelle des étiquettes, le jeu de données contient quand même des erreurs.
Une telle tâche nécessite à notre sens, la mobilisation d'une équipe consacrée à sa réalisation afin de (*i*) assurer un étiquetage robuste à l'aide d'un système de révision par les pairs, (*ii*) réduire considérablement le temps de révision.
- La sélection des hyper-paramètres du classifieur a été faite sur la base d'essais-erreurs. Des méthodes plus optimisées de sélection d'hyper-paramètres par validation croisée existent. Cependant, ce processus est gourmand en temps de calcul et en raison de limitations matérielles et a été laissé pour une opportunité future.

Ouverture et perspectives Un autre volet de l'automatisation des processus d'annotation est l'incertitude induite lors de l'inférence des estimations des modèles de population quand la sortie du modèle est traitée telle quelle. L'extraction rapide de l'information grâce aux méthodes automatiques d'échantillonnage et de classification viennent sans doutes induire des biais de sous-estimations ou de sur-estimations, un volet que nous avons malheureusement pas pu parcourir en raison de du grand volume de données que nous devions traiter. Une piste potentielle serait d'intervenir au niveaux de modèles écologiques existant de sorte à prendre en compte les sorties d'un modèle de classification. Comme les probabilités des faux positifs et des faux négatifs sont renseignées par le modèle de classification. Cette information pourrait donc être exploitée, potentiellement dans un cadre bayésien de sorte à permettre l'inférence de paramètres écologiques conditionnellement à la présence des faux positifs et de faux négatifs.

Enfin, l'automatisation dans un contexte de photo-piégeage est un champ encore juvénile et la démocratisation des ressources computationnelles et des algorithmes d'automatisation est sans doutes une opportunité à fort potentiel que ce soit dans la recherche académique, citoyenne ou la recherche à caractère privé.

Annexe

augmenter.py

```
Listing 6.1 – augpenter.py
usage: augmenter.py [-h] [--flip | --no-flip] [--zoom | --no-zoom]
                   [--zoom-factor ZOOM_FACTOR] [--translate | --no-translate]
                   [--translation-factor TRANSLATION_FACTOR]
                   [--rotate | --no-rotate]
                   [--rotation-factor ROTATION_FACTOR]
                   [--fill-mode FILL_MODE] [--size SIZE]
                   [--fix-imbalance | --no-fix-imbalance]
                   DATASET_DIRECTORY

Camera Trap Image Dataset Augmentation Utility

positional arguments:
  DATASET_DIRECTORY      Specify the path of the image dataset to augment. !!!
                        Note: the output images are saved in DATASET_DIRECTORY
                        !!!
optional arguments:
  -h, --help            show this help message and exit
  --flip, --no-flip     Apply random horizontal (left-right) flips to images.
                        (default: False)
  --zoom, --no-zoom    Apply random zoom to the input images. (default:
                        False)
  --zoom-factor ZOOM_FACTOR
                        Specify the zoom factor. Default is 0.5. Example: if
                        the factor is 0.2, then random zoom of input image is
                        generated in interval [-20%, 20%].
  --translate, --no-translate
                        Apply random vertical and horizontal shifting to
                        images. (default: False)
  --translation-factor TRANSLATION_FACTOR
                        Specify the translation factor. Default is 0.25.
                        Example: if the factor is 0.2, then random horizontal
                        and vertical translation are generated in the range
                        [-20%, 20%].
  --rotate, --no-rotate
                        Apply random rotations to images. (default: False)
  --rotation-factor ROTATION_FACTOR
                        Specify the rotation factor as a fraction of 2 Pi.
                        Default is 0.1. Example: if the factor is 0.2, then
                        random rotation angles will be generated in the
                        interval [-20% * 2 Pi, 20% * 2 Pi].
  --fill-mode FILL_MODE
                        Points outside the boundaries of the input are filled
                        according to the given mode. Accepted values are:
                        "constant", "reflect", "wrap" and "nearest". See Keras
                        API documentation for more information. https://keras.io/api/layers/preprocessing\_layers/image\_augmentation/
  --size SIZE          Specify the number of augmentation images to generate.
                        Default is 16. If --fix-imbalance is provided, then
                        SIZE images are generated first, then, classes are
                        leveled up to the same number of images (per class).
  --fix-imbalance, --no-fix-imbalance
                        Fix class imbalance by leveling up underrepresented
                        classes with the most represented class. As a result,
                        all classes will have the same number of images.
                        (default: False)
```

transfer.py

Listing 6.2 – transfer.py

```
usage: transfer.py [-h] {train,finetune,evaluate,predict} ...

Utility for the classification of animal images.

optional arguments:
-h, --help            show this help message and exit

Commands:
{train,finetune,evaluate,predict}
    train           Train a new model
    finetune        Finetune a pretrained model
    evaluate        Evaluate a model
    predict         Predict using a pretrained model
```

transfer.py | train

Listing 6.3 – transfer.py | train

```
usage: transfer.py train [-h] --base-model BASE_MODEL --train-data-path
                        TRAIN_DATA_PATH --validation-data-path
                        VALIDATION_DATA_PATH [--batch-size BATCH_SIZE]
                        [--epochs EPOCHS] [--learning-rate LEARNING_RATE]
                        [--finetune | --no-finetune]
                        [--finetune-epochs FINETUNE_EPOCHS]
                        [--finetune-learning-rate FINETUNE_LEARNING_RATE]
                        [--fix-imbalance | --no-fix-imbalance] --output-path
                        OUTPUT_PATH

optional arguments:
  -h, --help            show this help message and exit
  --base-model BASE_MODEL
                        Specify the base model (cf.
                        https://keras.io/api/applications/).
  --train-data-path TRAIN_DATA_PATH
                        Specify the directory of the training dataset. The
                        directory is expected to contain folders of images.
                        The names of these folders are then considered as
                        labels.
  --validation-data-path VALIDATION_DATA_PATH
                        Specify the directory of the validation dataset. The
                        directory is expected to contain folders of images.
                        The names of these folders are then considered as
                        labels.
  --batch-size BATCH_SIZE
                        Specify training batch size. (default: 32)
  --epochs EPOCHS        Specify the number of training epochs. (default: 300)
  --learning-rate LEARNING_RATE
                        Specify the optimizer learning rate parameter. The
                        used optimizer is Adam. (default: 0.01)
  --finetune, --no-finetune
                        Finetune the parameters of the model. (default: False)
  --finetune-epochs FINETUNE_EPOCHS
                        Specify the number of training epochs when finetuning.
                        (default: 30)
  --finetune-learning-rate FINETUNE_LEARNING_RATE
                        Specify the optimizer learning rate parameter when
                        finetuning. The used optimizer is Adam. (default:
                        1e-05)
  --fix-imbalance, --no-fix-imbalance
                        Fix imbalance by calculating class weights during
                        training. (default: False)
  --output-path OUTPUT_PATH
                        Path of directory where to save the trained model and
                        the training history.
```

CHAPITRE 6. LIVRABLES POUR L'ENTREPRISE

transfer.py | evaluate

```
Listing 6.4 – transfer.py | evaluate
usage: transfer.py evaluate [-h] --model-path MODEL_PATH --test-data TEST_DATA
                            --output-path OUTPUT_PATH

optional arguments:
  -h, --help            show this help message and exit
  --model-path MODEL_PATH
                        Specify the directory of the model to evaluate.
  --test-data TEST_DATA
                        Specify the directory of the testing dataset. The
                        directory is expected to contain folders of images.
                        The names of these folders are then considered as
                        labels.
  --output-path OUTPUT_PATH
                        Path of directory where to save the evaluation
                        results.
```

transfer.py | predict

```
Listing 6.5 – transfer.py | predict
usage: transfer.py predict [-h] --model-path MODEL_PATH --images-path
                           IMAGES_PATH --output-path OUTPUT_PATH

optional arguments:
  -h, --help            show this help message and exit
  --model-path MODEL_PATH
                        Specify the directory of the model to use for
                        prediction.
  --images-path IMAGES_PATH
                        Specify the directory of images for which to perform
                        predictions.
  --output-path OUTPUT_PATH
                        Path of file where to save the prediction results.
```

transfer.py | finetune

```
Listing 6.6 – transfer.py | finetune
usage: transfer.py finetune [-h] --model-path MODEL_PATH --train-data-path
                           TRAIN_DATA_PATH --validation-data-path
                           VALIDATION_DATA_PATH [--batch-size BATCH_SIZE]
                           [--finetune-epochs FINETUNE_EPOCHS]
                           [--finetune-learning-rate FINETUNE_LEARNING_RATE]
                           [--fix-imbalance | --no-fix-imbalance]
                           --output-path OUTPUT_PATH

optional arguments:
  -h, --help            show this help message and exit
  --model-path MODEL_PATH
                        Specify the directory of the model to finetune.
  --train-data-path TRAIN_DATA_PATH
                        Specify the directory of the training dataset. The
                        directory is expected to contain folders of images.
                        The names of these folders are then considered as
                        labels.
  --validation-data-path VALIDATION_DATA_PATH
                        Specify the directory of the validation dataset. The
                        directory is expected to contain folders of images.
                        The names of these folders are then considered as
                        labels.
  --batch-size BATCH_SIZE
                        Specify training batch size. (default: 32)
  --finetune-epochs FINETUNE_EPOCHS
                        Specify the number of training epochs when finetuning.
                        (default: 30)
  --finetune-learning-rate FINETUNE_LEARNING_RATE
                        Specify the optimizer learning rate to use during
                        finetuning. The used optimizer is Adam. (default:
                        1e-05)
  --fix-imbalance, --no-fix-imbalance
                        Fix imbalance by calculating class weights during
                        training. (default: False)
  --output-path OUTPUT_PATH
                        Path of file where to save the finetuned model.
```

CHAPITRE 6. LIVRABLES POUR L'ENTREPRISE

Matrices de Confusion

	Rongeur	Potamochère	Pangolin géant	Oiseau	Mangouste	Mandrill	Homme	Grandsinge	Félidés	Genette	Fau positif	Eléphant	Céphalophe	Chevrotain aquatique	Athérure
Athérure	121	3	10	3	5	1	13	0	0	0	1	3	0	[I]Chevrotain aquatique	
Céphalophe	2	3	840	1	19	0	3	5	5	56	3	44	0	6	23
Eléphant	2	0	8	228	8	0	0	1	5	4	0	3	0	0	0
Faux positif	5	0	24	2	187	0	2	1	1	3	0	5	1	0	7
Félidés	0	0	10	0	1	9	0	0	1	1	0	1	0	0	1
Genette	7	0	4	1	2	0	27	0	0	0	2	1	1	1	6
Grand singe	0	0	3	2	1	0	0	28	0	11	0	0	0	0	1
Homme	0	0	1	0	1	1	0	0	49	1	0	0	1	0	0
Mandrill	0	0	39	3	4	0	0	8	3	612	1	3	0	2	2
Mangouste	2	0	4	1	1	0	1	0	0	1	21	1	1	0	3
Oiseau	0	2	26	2	6	0	0	2	1	5	1	269	1	2	5
Pangolin géant	1	0	0	0	0	0	0	0	0	0	0	0	2	2	0
Potamochère	2	0	8	2	0	0	0	0	0	0	2	0	0	0	50
Rongeur	9	0	14	0	9	0	3	0	0	4	4	3	1	1	92

Table 6.1 : Matrice de confusion - modèle entraîné sur les données originales

Table 6.2 : Matrice de confusion - modèle entraîné sur les données augmentées

Table 6.3 : Matrice de confusion - modèle entraîné sur les données équilibrées

Bibliographie

- [BB21] Sara Beery et Elizabeth Bondi. « Can poachers find animals from public camera trap images ? ». In : *arXiv preprint arXiv :2106.11236* (2021).
- [BCV13] Yoshua Bengio, Aaron Courville et Pascal Vincent. « Representation learning : A review and new perspectives ». In : *IEEE transactions on pattern analysis and machine intelligence* 35.8 (2013), p. 1798-1828.
- [Ber+21] Julius Berner et al. « The modern mathematics of deep learning ». In : *arXiv preprint arXiv :2105.04026* (2021).
- [Bia12] Gérard Biau. « Analysis of a random forests model ». In : *The Journal of Machine Learning Research* 13.1 (2012), p. 1063-1095.
- [BMM18] Mateusz Buda, Atsuto Maki et Maciej A Mazurowski. « A systematic study of the class imbalance problem in convolutional neural networks ». In : *Neural networks* 106 (2018), p. 249-259.
- [BMY19] Sara Beery, Dan Morris et Siyu Yang. « Efficient Pipeline for Camera Trap Image Review ». In : *arXiv preprint arXiv :1907.06772* (2019).
- [Bou14] Thierry Bouwmans. « Traditional and recent approaches in background modeling for foreground detection : An overview ». In : *Computer science review* 11 (2014), p. 31-66.
- [Bre01] Leo Breiman. « Random forests ». In : *Machine learning* 45.1 (2001), p. 5-32.
- [Bur+15] A Cole Burton et al. « Wildlife camera trapping : a review and recommendations for linking surveys to ecological processes ». In : *Journal of Applied Ecology* 52.3 (2015), p. 675-685.
- [Car+17] Anthony Caravaggi et al. « A review of camera trapping for conservation behaviour research ». In : *Remote Sensing in Ecology and Conservation* 3.3 (2017), p. 109-122.
- [Cas+15] Marco Castelluccio et al. « Land use classification in remote sensing images by convolutional neural networks ». In : *arXiv preprint arXiv :1508.00092* (2015).
- [CH67] Thomas Cover et Peter Hart. « Nearest neighbor pattern classification ». In : *IEEE transactions on information theory* 13.1 (1967), p. 21-27.

BIBLIOGRAPHIE

- [Cha+20] Bosco Pui Lok Chan et al. « First use of artificial canopy bridge by the world's most critically endangered primate the Hainan gibbon *Nomascus hainanus* ». In : *Scientific reports* 10.1 (2020), p. 1-9.
- [Coo+07] Caren B Cooper et al. « Citizen science as a tool for conservation in residential ecosystems ». In : *Ecology and society* 12.2 (2007).
- [CV95] Corinna Cortes et Vladimir Vapnik. « Support-vector networks ». In : *Machine learning* 20.3 (1995), p. 273-297.
- [Den+09] Jia Deng et al. « Imagenet : A large-scale hierarchical image database ». In : *2009 IEEE conference on computer vision and pattern recognition*. Ieee. 2009, p. 248-255.
- [DHS11] John Duchi, Elad Hazan et Yoram Singer. « Adaptive subgradient methods for online learning and stochastic optimization. » In : *Journal of machine learning research* 12.7 (2011).
- [DT05] Navneet Dalal et Bill Triggs. « Histograms of oriented gradients for human detection ». In : *2005 IEEE computer society conference on computer vision and pattern recognition (CVPR'05)*. T. 1. Ieee. 2005, p. 886-893.
- [DV16] Vincent Dumoulin et Francesco Visin. « A guide to convolution arithmetic for deep learning ». In : *arXiv preprint arXiv:1603.07285* (2016).
- [DY+14] Li Deng, Dong Yu et al. « Deep learning : methods and applications ». In : *Foundations and trends® in signal processing* 7.3–4 (2014), p. 197-387.
- [DZB10] Janis L Dickinson, Benjamin Zuckerberg et David N Bonter. « Citizen science as an ecological research tool : challenges and benefits ». In : *Annual review of ecology, evolution, and systematics* (2010), p. 149-172.
- [Fer+18] Atilla Colombo Ferrengutti et al. « One step ahead to predict potential poaching hotspots : Modeling occupancy and detectability of poachers in a neotropical rainforest ». In : *Biological Conservation* 227 (2018), p. 133-140.
- [FH12] Rebecca J Foster et Bart J Harmsen. « A critique of density estimation from camera-trap data ». In : *The Journal of Wildlife Management* 76.2 (2012), p. 224-236.
- [Fuk79] Kunihiko Fukushima. « Neural network model for a mechanism of pattern recognition unaffected by shift in position-Neocognitron ». In : *IEICE Technical Report, A* 62.10 (1979), p. 658-665.
- [GBB11] Xavier Glorot, Antoine Bordes et Yoshua Bengio. « Deep sparse rectifier neural networks ». In : *Proceedings of the fourteenth international conference on artificial intelligence and statistics*. JMLR Workshop et Conference Proceedings. 2011, p. 315-323.
- [GBC16] Ian Goodfellow, Yoshua Bengio et Aaron Courville. *Deep learning*. MIT press, 2016.

BIBLIOGRAPHIE

- [Gro82] Stephen Grossberg. « Contour enhancement, short term memory, and constancies in reverberating neural networks ». In : *Studies of mind and brain*. Springer, 1982, p. 332-378.
- [GSV16] Alexander Gomez, Augusto Salazar et Francisco Vargas. « Towards automatic wild animal monitoring : Identification of animal species in camera-trap images using very deep convolutional neural networks ». In : *arXiv preprint arXiv :1603.06169* (2016).
- [GW08] Andreas Griewank et Andrea Walther. *Evaluating derivatives : principles and techniques of algorithmic differentiation*. SIAM, 2008.
- [Has+09] Trevor Hastie et al. *The elements of statistical learning : data mining, inference, and prediction*. T. 2. Springer, 2009.
- [Heb05] Donald Olding Hebb. *The organization of behavior : A neuropsychological theory*. Psychology Press, 2005.
- [Hed+15] Laurie Hedges et al. « Melanistic leopards reveal their spots : infrared camera traps provide a population density estimate of leopards in Malaysia ». In : *The Journal of Wildlife Management* 79.5 (2015), p. 846-853.
- [Hed+18] Daniela Hedwig et al. « A camera trap assessment of the forest mammal community within the transitional savannah-forest mosaic of the Batéké Plateau National Park, Gabon ». In : *African Journal of Ecology* 56.4 (2018), p. 777-790.
- [Hen+10] PHILIPP Henschel et al. « Lion status updates from five range countries in West and Central Africa ». In : *Cat News* 52.Spring (2010), p. 34-39.
- [HM95] Jun Han et Claudio Moraga. « The influence of the sigmoid function parameters on the speed of backpropagation learning ». In : *International workshop on artificial neural networks*. Springer. 1995, p. 195-201.
- [Hoc98] Sepp Hochreiter. « The vanishing gradient problem during learning recurrent neural nets and problem solutions ». In : *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems* 6.02 (1998), p. 107-116.
- [IS15] Sergey Ioffe et Christian Szegedy. « Batch normalization : Accelerating deep network training by reducing internal covariate shift ». In : *International conference on machine learning*. PMLR. 2015, p. 448-456.
- [Jar+09] Kevin Jarrett et al. « What is the best multi-stage architecture for object recognition ? ». In : *2009 IEEE 12th international conference on computer vision*. IEEE. 2009, p. 2146-2153.
- [KB14] Diederik P Kingma et Jimmy Ba. « Adam : A method for stochastic optimization ». In : *arXiv preprint arXiv :1412.6980* (2014).
- [KSH12] Alex Krizhevsky, Ilya Sutskever et Geoffrey E Hinton. « Imagenet classification with deep convolutional neural networks ». In : *Advances in neural information processing systems* 25 (2012).

BIBLIOGRAPHIE

- [Kyb+13] Christopher Kyba et al. « Citizen science provides valuable data for monitoring global night sky luminance ». In : *Scientific reports* 3.1 (2013), p. 1-6.
- [LeC+12] Yann A LeCun et al. « Efficient backprop ». In : *Neural networks : Tricks of the trade*. Springer, 2012, p. 9-48.
- [Lóp+13] Victoria López et al. « An insight into classification with imbalanced data : Empirical results and current trends on using data intrinsic characteristics ». In : *Information sciences* 250 (2013), p. 113-141.
- [Low99] David G Lowe. « Object recognition from local scale-invariant features ». In : *Proceedings of the seventh IEEE international conference on computer vision*. T. 2. Ieee. 1999, p. 1150-1157.
- [LW07] Dengsheng Lu et Qihao Weng. « A survey of image classification methods and techniques for improving classification performance ». In : *International journal of Remote sensing* 28.5 (2007), p. 823-870.
- [LZF15] Cheng Li, Chao Zhao et Peng-Fei Fan. « White-cheeked macaque (*Macaca leucogenys*) : A new macaque species from Medog, southeastern Tibet ». In : *American Journal of Primatology* 77.7 (2015), p. 753-766.
- [Mac+02] Darryl I MacKenzie et al. « Estimating site occupancy rates when detection probabilities are less than one ». In : *Ecology* 83.8 (2002), p. 2248-2255.
- [MHN+13] Andrew L Maas, Awini Y Hannun, Andrew Y Ng et al. « Rectifier nonlinearities improve neural network acoustic models ». In : *Proc. icml*. T. 30. 1. Citeseer. 2013, p. 3.
- [Mit80] Tom M Mitchell. *The need for biases in learning generalizations*. Department of Computer Science, Laboratory for Computer Science Research . . ., 1980.
- [MP69] Marvin Minsky et Seymour Papert. « An introduction to computational geometry ». In : *Cambridge tiass., HIT* 479 (1969), p. 480.
- [MRT18] Mehryar Mohri, Afshin Rostamizadeh et Ameet Talwalkar. *Foundations of machine learning*. MIT press, 2018.
- [Nem+09] Arkadi Nemirovski et al. « Robust stochastic approximation approach to stochastic programming ». In : *SIAM Journal on optimization* 19.4 (2009), p. 1574-1609.
- [NFS18] Yoshihiro Nakashima, Keita Fukasawa et Hiromitsu Samejima. « Estimating animal density without individual recognition using information derivable exclusively from camera traps ». In : *Journal of Applied Ecology* 55.2 (2018), p. 735-744.
- [NH10] Vinod Nair et Geoffrey E Hinton. « Rectified linear units improve restricted boltzmann machines ». In : *Icmi*. 2010.
- [NHA20] Yoshihiro Nakashima, Shun Hongo et Etienne François Akomo-Okoue. « Landscape-scale estimation of forest ungulate density and biomass using camera traps : Applying the REST model ». In : *Biological Conservation* 241 (2020), p. 108381.

BIBLIOGRAPHIE

- [Nor+] Mohammed Sadegh Norouzzadeh et al. « Automatically identifying wild animals in camera trap images with deep learning ». In : *Proceedings of the National Academy of Sciences*. T. 115.
- [Nor+21] Mohammad Sadegh Norouzzadeh et al. « A deep active learning system for species identification and counting in camera trap images ». In : *Methods in ecology and evolution* 12.1 (2021), p. 150-161.
- [Nov63] Albert B Novikoff. *On convergence proofs for perceptrons*. Rapp. tech. STANFORD RESEARCH INST MENLO PARK CA, 1963.
- [NY83] Arkadij Semenovič Nemirovskij et David Borisovich Yudin. « Problem complexity and method efficiency in optimization ». In : (1983).
- [Nyi+11] Aisha Nyiramana et al. « Evidence for seed dispersal by rodents in tropical montane forest in Africa ». In : *Biotropica* 43.6 (2011), p. 654-657.
- [Ola96] Mikel Olazaran. « A sociological study of the official history of the perceptrons controversy ». In : *Social Studies of Science* 26.3 (1996), p. 611-659.
- [PTC02] Olivier SG Pauwels, A Kamdem Toham et Chuchéep Chimsunchart. « Recherches sur l'herpétofaune des Monts de Cristal, Gabon ». In : *Bulletin de l'Institut Royal des Sciences naturelles de Belgique, Biologie* 72 (2002), p. 59-66.
- [RC08] J Marcus Rowcliffe et Chris Carbone. « Surveys using camera traps : are we looking to a brighter future ? » In : *Animal conservation* 11.3 (2008), p. 185-186.
- [Ree+13] Jason Reed et al. « An exploratory factor analysis of motivations for participating in Zooniverse, a collection of virtual citizen science projects ». In : *2013 46th Hawaii International Conference on System Sciences*. IEEE. 2013, p. 610-619.
- [Ren+15] Shaoqing Ren et al. « Faster r-cnn : Towards real-time object detection with region proposal networks ». In : *Advances in neural information processing systems* 28 (2015).
- [RHW86] David E Rumelhart, Geoffrey E Hinton et Ronald J Williams. « Learning representations by back-propagating errors ». In : *nature* 323.6088 (1986), p. 533-536.
- [Ros58] Frank Rosenblatt. « The perceptron : a probabilistic model for information storage and organization in the brain. » In : *Psychological review* 65.6 (1958), p. 386.
- [Ros61] Frank Rosenblatt. *Principles of neurodynamics. perceptrons and the theory of brain mechanisms*. Rapp. tech. Cornell Aeronautical Lab Inc Buffalo NY, 1961.
- [Rov+08] F Rovero et al. « A new species of giant sengi or elephant-shrew (genus Rhynchocyon) highlights the exceptional biodiversity of the Udzungwa Mountains of Tanzania ». In : *Journal of Zoology* 274.2 (2008), p. 126-133.

BIBLIOGRAPHIE

- [Rov+13] Francesco Rovero et al. « " Which camera trap type and how many do I need ?" A review of camera features and study designs for a range of wildlife research applications. » In : *Hystrix* (2013).
- [Row+08] J Marcus Rowcliffe et al. « Estimating animal density using camera traps without the need for individual recognition ». In : *Journal of Applied Ecology* (2008), p. 1228-1236.
- [RSG16] Marco Tulio Ribeiro, Sameer Singh et Carlos Guestrin. « " Why should i trust you?" Explaining the predictions of any classifier ». In : *Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining*. 2016, p. 1135-1144.
- [RYH22] Daniel A Roberts, Sho Yaida et Boris Hanin. *The Principles of Deep Learning Theory : An Effective Theory Approach to Understanding Neural Networks*. Cambridge University Press, 2022.
- [SB14] Shai Shalev-Shwartz et Shai Ben-David. *Understanding machine learning : From theory to algorithms*. Cambridge university press, 2014.
- [SBB14] Pierre-Luc St-Charles, Guillaume-Alexandre Bilodeau et Robert Bergevin. « SuBSENSE : A universal change detection method with local adaptive sensitivity ». In : *IEEE Transactions on Image Processing* 24.1 (2014), p. 359-373.
- [Sch+19] Stephanie G Schuttler et al. « Citizen science in schools : students collect valuable mammal data for science, conservation, and community engagement ». In : *Bioscience* 69.1 (2019), p. 69-79.
- [Sha+09] Shai Shalev-Shwartz et al. « Stochastic Convex Optimization. » In : *COLT*. T. 2. 4. 2009, p. 5.
- [Sha18] Janelle Shane. « Do neural nets dream of electric sheep ». In : *AI Wierdness* (2018).
- [Shi00] Hidetoshi Shimodaira. « Improving predictive inference under covariate shift by weighting the log-likelihood function ». In : *Journal of statistical planning and inference* 90.2 (2000), p. 227-244.
- [Sil+04] Scott C Silver et al. « The use of camera traps for estimating jaguar *Panthera onca* abundance and density using capture/recapture analysis ». In : *Oryx* 38.2 (2004), p. 148-154.
- [Sil09] Jonathan Silvertown. « A new dawn for citizen science ». In : *Trends in ecology & evolution* 24.9 (2009), p. 467-471.
- [SKP11] Don E Swann, Kae Kawanishi et Jonathan Palmer. « Evaluating types and features of camera traps in ecological studies : a guide for researchers ». In : *Camera traps in animal ecology*. Springer, 2011, p. 27-43.
- [Sri+14] Nitish Srivastava et al. « Dropout : a simple way to prevent neural networks from overfitting ». In : *The journal of machine learning research* 15.1 (2014), p. 1929-1958.

BIBLIOGRAPHIE

- [Swa+04] Don E Swann et al. « Infrared-triggered cameras for detecting wildlife : an evaluation and review ». In : *Wildlife Society Bulletin* 32.2 (2004), p. 357-365.
- [Swa+15] Alexandra Swanson et al. « Snapshot Serengeti, high-frequency annotated camera trap images of 40 mammalian species in an African savanna ». In : *Scientific data* 2.1 (2015), p. 1-14.
- [Swi+14] Kristijn RR Swinnen et al. « A novel method to reduce time investment when processing videos from camera trap studies ». In : *PLoS one* 9.6 (2014), e98881.
- [SWI04] Terry Sunderland, Gretchen Walters et Y. Issembe. *ETUDE PRELIMINAIRE DE LA VEGETATION DU PARC NATIONAL DE MBE, MONTS DE CRISTAL, GABON*. Jan. 2004.
- [SZ14] Karen Simonyan et Andrew Zisserman. « Very deep convolutional networks for large-scale image recognition ». In : *arXiv preprint arXiv:1409.1556* (2014).
- [TH12] Tijmen Tieleman et Geoffrey Hinton. « Rmsprop : Divide the gradient by a running average of its recent magnitude. coursera : Neural networks for machine learning ». In : *COURSERA Neural Networks Mach. Learn* (2012).
- [Tob+15] Mathias W Tobler et al. « Spatiotemporal hierarchical modelling of species richness and occupancy using camera trap data ». In : *Journal of Applied Ecology* 52.2 (2015), p. 413-421.
- [Val84] Leslie G Valiant. « A theory of the learnable ». In : *Communications of the ACM* 27.11 (1984), p. 1134-1142.
- [Vap99] Vladimir Vapnik. *The nature of statistical learning theory*. Springer science & business media, 1999.
- [VC15] Vladimir N Vapnik et A Ya Chervonenkis. « On the uniform convergence of relative frequencies of events to their probabilities ». In : *Measures of complexity*. Springer, 2015, p. 11-30.
- [VC74] Vladimir Vapnik et Alexey Chervonenkis. *Theory of pattern recognition*. 1974.
- [Wel+16] Dustin J Welbourne et al. « How do passive infrared triggered camera traps operate and why does it matter ? Breaking down common misconceptions ». In : *Remote Sensing in Ecology and Conservation* 2.2 (2016), p. 77-83.
- [Wil+17] Ashia C Wilson et al. « The marginal value of adaptive gradient methods in machine learning ». In : *Advances in neural information processing systems* 30 (2017).
- [Wil90] Chris Wilks. *La conservation des écosystèmes forestiers du Gabon*. T. 14. IUCN, 1990.
- [Yos+14] Jason Yosinski et al. « How transferable are features in deep neural networks ? ». In : *Advances in neural information processing systems* 27 (2014).
- [ZS18] Zhilu Zhang et Mert Sabuncu. « Generalized cross entropy loss for training deep neural networks with noisy labels ». In : *Advances in neural information processing systems* 31 (2018).

Université de Montpellier
163 rue Auguste Broussonnet
34 090 Montpellier - FRANCE
Tel : 04 .67.41.74.00
www.umontpellier.fr

