

SYS843-01 RÉSEAUX DE NEURONES ET SYSTÈMES FLOUS (H2019)

DÉPARTEMENT DU GÉNIE DE LA PRODUCTION AUTOMATISÉE

SYNTHÈSE DE LITTÉRATURE : RECONNAISSANCE D'ESPÈCES ANIMALES

Soumis par

Hayat ANKOUR

Département du Génie de la Production Automatisée

École de Technologies Supérieures

Montréal, QC

Soumis à

Ismael BEN AYED

École de Technologies Supérieures

Montréal, QC



Le génie pour l'industrie

Table des matières

1 Mise en situation	3
1.1 Domaine d'application	3
1.2 Problématique	3
1.3 Objectif du projet	4
1.4 Méthodologie adoptée	4
1.5 Structure du document	4
2 Synthèse des techniques	5
2.1 Types de caractéristiques utilisées	5
2.2 Extraction des régions d'intérêts	6
2.3 Revue globale des approches utilisées en littérature	8
2.3.1 État de l'art de la classification d'image	8
2.3.2 Description détaillée du CNN	9
3 Analyse critique	12
3.1 Description des 2 articles	12
3.2 Comparaison qualitative des approches et étude analytique des performances	13
4 Conclusions	16

1 Mise en situation

1.1 Domaine d'application

Le but du projet est la reconnaissance d'espèce d'animaux dans une image, puis dans une vidéo si le temps le permet. Il existe plusieurs domaines d'application pour ce type d'algorithme et les informations nécessaires à la reconnaissance des espèces d'animaux comme la taille, la couleur de la fourrure ou encore la position de l'animal dans l'image dépendent du domaine d'application. Par exemple, en forêt, où le nombre de parasite autour de l'animal comme les arbres, buisson et autre est très important, on prendra des caractéristiques bien différentes que sur la banquise, ou l'on ne voit que l'animal sur un fond presque blanc.

L'importance de ce genre d'algorithme dépend également grandement des personnes qui l'utilisent. Il peut être utilisé à des fins de loisirs comme, par exemple, compter le nombre d'espèces croisées lors d'un safari, ou pour des applications telles que la surveillance du nombre d'individus d'une espèce en voie d'extinction. Les applications sont donc diverses et variées.



FIGURE 1: Photo de différentes espèces d'animaux dans des milieux environnementaux variés [1]

1.2 Problématique

L'identification d'espèce est un défi technologique complexe. De nos jours, ce sont des experts qui identifient et recensent les espèces. Cependant, cela prend beaucoup de temps, et les experts ne peuvent pas effectuer leur travail à grande échelle. De plus, c'est un travail visuel et manuel fastidieux et fatigant.

Une des solutions est d'assister l'expert par l'utilisation d'algorithmes. Donnée une image, comment identifier l'espèce des animaux présents sur l'image?

Une rapide recherche dans la littérature nous permet d'estimer que la méthode la plus utilisée est le réseau neuronal convolutif (Convolutional Neural Network, CNN). Cependant, celui-ci ne permet de faire que de la classification. Un CNN permet d'attribuer une classe à une image. Il faut donc faire un pré-traitement sur les images et obtenir les régions d'intérêts à envoyer au CNN, qui pourra classifier les différentes régions d'intérêt. Dans la littérature, il semblerait qu'une technique largement utilisée soit la recherche sélective, ou l'utilisation d'un autre réseau de neurones qui prédit la position et la taille d'une région d'intérêt.

Mon projet se découpe donc en deux problématiques :

1. Comment déterminer efficacement les régions d'intérêt des différents animaux dans une image?
2. Comment correctement classifier ces images?

1.3 Objectif du projet

Pour être efficace, il faut entraîner l'algorithme avec un grand nombre d'images. Cela lui permet de reconnaître un maximum d'espèces animales. Pour maximiser l'efficacité de l'algorithme, avoir des images de plusieurs profils différents est nécessaire.

Après l'entraînement de l'algorithme, il faut lui faire un test de validation qui permettra de légitimer les résultats obtenus et, dans le cas contraire, l'entraîner de nouveau afin d'augmenter la précision de classification de l'algorithme.

Il existe de nombreuses références sur la reconnaissance d'espèces animales à l'aide de la méthode CNN comme :

- <https://arxiv.org/abs/1603.06169> [2]
- <https://ieeexplore.ieee.org/abstract/document/7025172/> [3]

1.4 Méthodologie adoptée

Cette synthèse sur l'état de l'art est essentielle pour le bon déroulement du projet. En effet, pour mieux comprendre le fonctionnement du CNN et mieux me renseigner sur les autres méthodes existantes, il faut lire ce qu'il existe déjà sur les méthodes d'extraction de régions d'intérêt et sur la reconnaissance d'espèces animales.

Je vais ensuite me concentrer sur l'implémentation de l'algorithme à l'aide de Matlab et de Python. OpenCV pourra également être utilisé pour traiter les images. Les supports du cours SYS843 me seront aussi d'une grande aide.

Les databases que l'on peut rencontrer sont de deux types : elles sont soit créées par les personnes réalisant des articles et des projets sur la reconnaissance d'espèces animales comme [3], soit déjà existante comme la database suivante : <https://datadryad.org/resource/doi:10.5061/dryad.5pt92>. Je vais donc me focaliser sur une database existante.

1.5 Structure du document

Comme dit précédemment, faire un état de l'art est crucial avant le commencement d'un projet. Ce document recensera donc les techniques les plus utilisées d'une part pour l'extraction de la région d'intérêt comportant les animaux et leurs caractéristiques et les algorithmes de classification d'image les plus courants pour la reconnaissance d'espèces animales. Pour finir, il comportera un comparatif analytique et qualitatif des performances entre les méthodes couramment utilisées.

2 Synthèse des techniques

2.1 Types de caractéristiques utilisées

Il existe plusieurs techniques afin de récupérer les caractéristiques sur une image. Cette partie comportera seulement l'explication des Histograms of Oriented Gradients et des Bag-of-Visual-Words, deux parmi les plus utilisées.

- **Histograms of Oriented Gradients (HOG) [4]** : Le but du HOG est d'évaluer les histogrammes locaux normalisés à l'aide du gradient orienté d'une image dans une grille dense. En effet, l'apparence et la forme d'objet sur une image sont souvent caractérisées par la distribution locale des gradients d'intensités ou de la direction des contours de l'objet. Le principe HOG fonctionne de la façon suivante : l'image est divisée en plusieurs petites cellules, pour chaque cellule, on accumule l'histogramme local de la direction du gradient ou de l'orientation des contours sur les pixels de l'image. La combinaison de tous les histogrammes de l'image forme l'HOG de l'image. Pour de meilleurs résultats, on peut améliorer le contraste de l'image et le normaliser. En effet, on peut accumuler les histogrammes sur de plus grosses cellules et utiliser le résultat obtenu pour normaliser toutes les cellules initiales de l'image.



FIGURE 2: HOG appliqué à une image de chat [5]

- **Bag-of-Visual-Words [6]** : cette méthode est inspirée du bag-of-words, une technique permettant de catégoriser du texte. De base, cette technique utilise l'histogramme du nombre d'occurrence de chaque mot. Le texte est donc représenté par des collections de mots non ordonnées, où la grammaire et l'ordre des mots ne sont pas importants. Le Bag-of-Visual-Words fonctionne donc sur le même principe. L'image comporte donc des points locaux d'intérêts ou des points clés qui sont définis sur des petites régions de l'image et qui comportent beaucoup d'information locale sur l'image. En général, ces points clés sont situés sur des contours ou des coins dans l'image, par exemple autour des personnes ou des animaux. Les points clés permettent donc de décrire des patrons ou modèles locaux sur une image. Le nombre de points clés sur une image varie d'un détecteur à l'autre. Il n'est jamais fixe sur une image. Cela peut donc créer des soucis pour certains classificateurs supervisés qui ont besoin d'un vecteur de caractéristique à dimension précise.

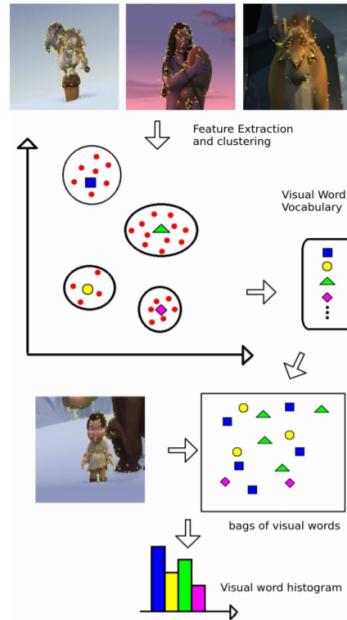


FIGURE 3: Bag-of-Visual-Words appliqu      des images [7]

2.2 Extraction des r  gions d'int  r  ts

Il existe plusieurs types d'extraction de r  gions d'int  r  t. Dans ce document, nous allons essentiellement voir deux m  thodes qui sont les plus utilis  es dans l'  tat de l'art, la recherche s  lective et l'utilisation des edge boxes.

- **Recherche s  lective :** La recherche s  lective [8] est une alternative de la recherche exhaustive. En effet, la recherche exhaustive est un principe qui n'est en g  n  ral pas utilis      cause de sa consommation en puissance de calcul. C'est un algorithme qui utilise des fen  tres glissantes qui parcourt toute l'image au fur et    mesure en commen  tant par une fen  tre d'une taille donn  e. Les fen  tres glissantes commencent    parcourir l'image petit    petit. En g  n  ral, ce sont les caract  ristiques HOG qui sont le plus utilis  es avec cette technique. Elle est souvent utilis  e en pr  -s  lection d'un classificateur en cascade. Cet algorithme est cependant coûteux en temps et infaisable. La recherche s  lective s'effectue quant    elle en plusieurs   tapes [9] : segmentation de l'image en utilisant un algorithme de type graph de Felzenszwalb et Huttenlocher; ensuite, regrouper les r  gions de m  mes intensit  , couleurs, forme, taille, ou encore texture afin d'obtenir moins de r  gions; ajout des bounding boxes par r  gion; regroupement des bounding boxes selon leur similarit  ; recommencer    partir de l'  tape d'ajout des bounding boxes. L'algorithme de recherche s  lective est donc beaucoup moins gourmand en termes de temps de calcul.



FIGURE 4: Résultat obtenu par recherche sélective [8]

- **Edge box :** Comme on peut le voir dans l'article [10], le but de la technique par "edge boxes" permet de générer des bounding boxes en utilisant les contours présents dans une image. En effet, le but de cette technique est de réduire au maximum la taille de l'image en entrée, pour avoir moins de position à analyser dans l'image. L'algorithme fonctionne de la manière suivante : au lieu de chercher un objet à chaque emplacement et échelle d'image, on va directement appliquer l'edge boxing pour récupérer les bounding boxes des objets présents sur l'image qui ont besoin d'être analysée.

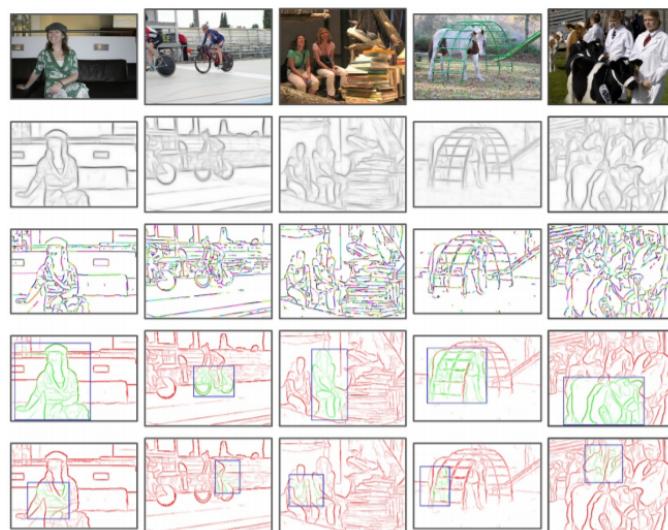


FIGURE 5: Résultat obtenu par edge boxing [10]

Comme on peut le voir sur la figure précédente, on récupère les contours présents dans l'image, on les regroupe et on crée la bounding box. La quatrième colonne montre un edge boxing qui serait bon tandis que la cinquième montre ce qui serait faux. Les pixels verts sont prédits pour faire partie de l'objet, les rouges sont prédits à ne pas faire partie de l'objet.

2.3 Revue globale des approches utilisées en littérature

Il existe plusieurs types de classification d'image. Les plus couramment utilisées sont l'algorithme YOLO, You Only Look Once [11], la plus récente version de R-CNN, Region-Convolutional Neural Network, le Faster R-CNN ainsi que CNN, Convolutional Neural Network. Je présenterai d'abord brièvement les algorithmes YOLO et R-CNN d'une part et j'expliquerai de manière détaillée l'algorithme CNN d'autre part, car c'est l'algorithme que je souhaiterais utiliser comme base dans mon projet pour l'instant.

2.3.1 État de l'art de la classification d'image

- **L'algorithme YOLO[11]** : Comme son nom l'indique, le principe de cette technique est de ne regarder l'image qu'une seule fois. En effet, l'algorithme divise en K^2 cellules l'image en entrée. Il comporte une seule couche de CNN (que j'explique dans la deuxième partie de sous partie du 2.2.3). Le but de l'algorithme est de déterminer les bounding boxes des objets présents sur une image en même temps que leur classe avec le CNN. Le tout n'étant effectué qu'une fois.

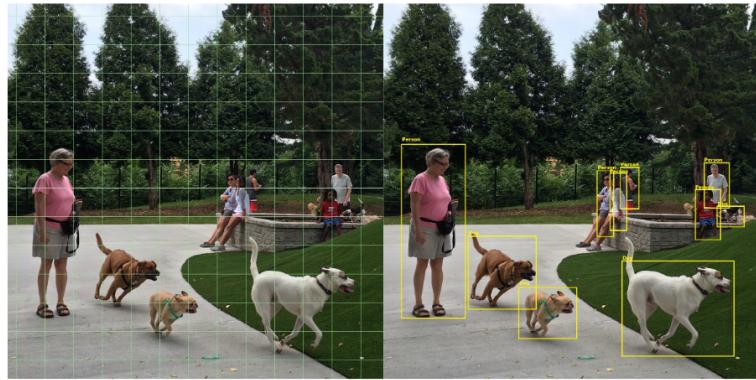


FIGURE 6: Division de l'image par l'algorithme YOLO et sa sortie

C'est un des algorithmes les plus rapides pour la reconnaissance d'objet dans une image. En effet, la plus récente version de cet algorithme peut aller jusqu'à 150 images par secondes. Avoir directement l'image en entrée crée des informations contextuelles qui facilitent la reconnaissance de la classe d'un objet présent sur l'image. Cela permet aussi d'éviter d'avoir l'arrière-plan qui est perçu comme un objet. De plus, quand le réseau est entraîné, il apprend plus facilement à généraliser des objets. Cela rend la détection de nouveau domaine d'objet plus facile. Mais le principal souci de cet algorithme est sa précision, surtout pour les petits objets. En effet, la bounding box qui comprend de petits objets peut être perçue comme une erreur par l'algorithme.

- **L'algorithme R-CNN [12]** : Il est en fait constitué de plusieurs étapes qu'on appelle des modules. Tout d'abord, le premier module sert à générer des catégories de régions indépendantes à l'aide de par exemple la recherche sélective. Ceux seront donc dans ces régions où la probabilité pour qu'il y ait un objet est la plus grande. Ensuite, un

CNN composé de plusieurs couches est appliqué afin d'extraire un vecteur de caractéristiques présent dans chaque région. Enfin, on applique un Support Vector Machine (SVM) afin de récupérer la classe des objets présents sur l'image. Un SVM est un classificateur linéaire qui permet de séparer de manières optimales des classes à l'aide d'un plan hyperbolique. Il contient de plus une marge d'erreur afin de maximiser la séparation entre les classes.

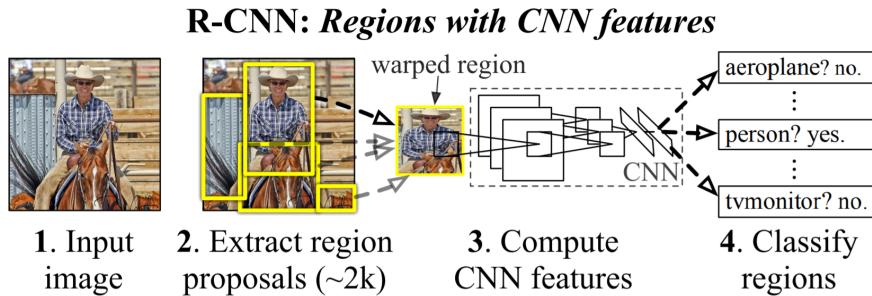


FIGURE 7: Étape successive du R-CNN [12]

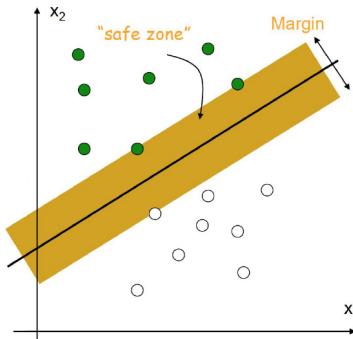


FIGURE 8: Visualisation d'un SVM (Cours de SYS828)

2.3.2 Description détaillée du CNN

La technique de classification la plus utilisée pour la reconnaissance d'espèces animales ou la détection d'objet dans l'état de l'art est le réseau neuronal convolutif (Convolutional Neural Network, CNN). Comme son nom l'indique, c'est un réseau neuronal qui s'appuie fortement sur le fonctionnement du cerveau.

En effet, dans le cerveau humain, les neurones sont regroupés sous forme de clusters. C'est grâce à ces clusters que l'on peut reconnaître par exemple un visage. Chaque cluster comporte un nombre de caractéristiques données.

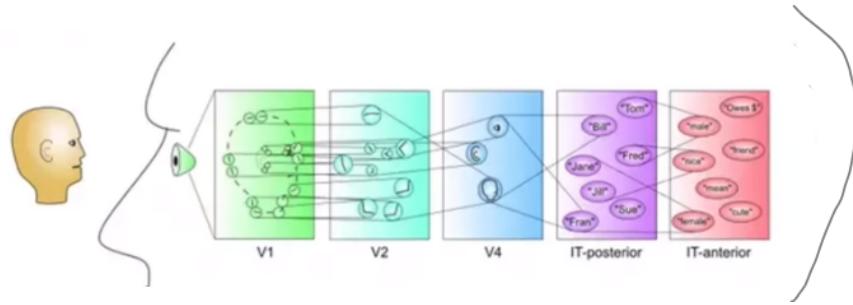


FIGURE 9: Constitution des clusters neuronaux chez l'humain [13]

Comme on peut le voir dans le schéma ci-dessus, les premiers clusters comportent des caractéristiques très basiques, et plus on avance vers les derniers clusters, plus les caractéristiques deviennent sophistiquées.

Le réseau neuronal convolutif fonctionne donc de la même manière.

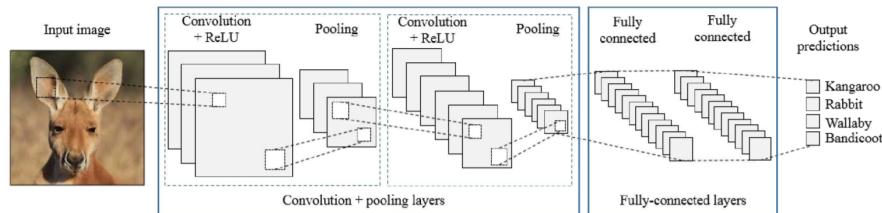


FIGURE 10: Illustration d'une architecture typique d'un réseau neuronal convolutif [14]

En entrée, le CNN prend une partie de l'image. On réduit donc l'entrée du réseau à une seule région donnée de l'image. Le réseau comporte des couches convolutionnelles qui correspondent aux clusters. Chaque couche convolutionnelle comporte un nombre donné de caractéristiques. C'est ce qu'on appelle des filtres. Ils peuvent détecter des contours, des cercles ou autres formes simples ou complexes. Plus une couche convolutionnelle comporte des filtres, plus elle est capable de reconnaître des caractéristiques particulières. La sortie des couches convolutionnelles est un produit de convolution entre les données de l'image et les filtres.

Les couches convolutionnelles sont appelées des "*hidden layers*". À l'issue de la convolution, les résultats sont transmis à une fonction d'activation, la plus couramment utilisée est la ReLU, Rectifier Linear Units, mais il en existe d'autre comme Tanh, la tangente hyperbolique.

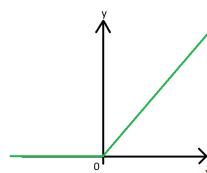


FIGURE 11: Fonction d'activation ReLU

ReLU est une fonction d'activation assez simple, avant 0, elle est égale à 0, ensuite $y = x$. Cette fonction nous permet de récupérer seulement les données qui nous intéressent, afin

de les transmettre à la couche suivante. Le principe est donc une alternance entre des convolutions et des fonctions d'activation afin d'avoir le meilleur résultat possible. Après quelques alternances, on réduit la dimension de l'image avec un Pool, une couche de pooling.

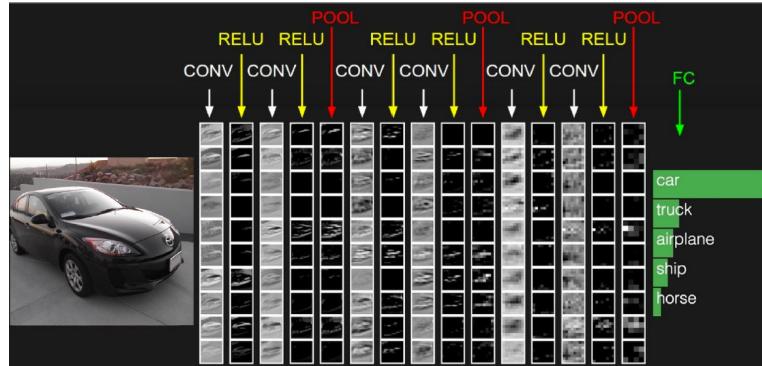


FIGURE 12: Résultat des couches convolutionnelles [15]

3 Analyse critique

Le but final de mon projet est la reconnaissance des espèces animales à l'aide de l'algorithme CNN. Je vais donc reprendre deux articles qui ont utilisé l'algorithme CNN pour la reconnaissance d'espèces animales, dont un des deux articles qui fait la comparaison entre une architecture de CNN (AlexNET) et Fast R-CNN (une alternative à R-CNN qui n'utilise pas de SVM) et l'autre qui compare des CNN avec un nombre de couches de convolutions différents.

3.1 Description des 2 articles

Le premier article, **Animal Recognition and Identification with Deep Convolutional Neural Networks for Automated Wildlife Monitoring** [16], par Nguyen et al. en 2017, compare 3 architectures de réseaux de neurones convolutionnelles, AlexNET (avec seulement 5 couches convolutionnelles), VGG-16 (16 couches convolutionnelles) et ResNET-50 (50 couches convolutionnelles). Ils ont utilisé le dataset Wildlife Spotter qui comporte 107,022 images labélisées, dont 72,498 images animales de 18 espèces différentes. Ils ont fait 3 tests différents : le premier permettant de détecter la présence d'un animal, le second pour identifier les 3 espèces prédominantes dans la base de données, et le troisième pour identifier les 6 espèces prédominantes dans la base de données. Ils ont donc divisé leur méthode en deux parties : identification de la présence d'un animal et ensuite reconnaissance de celui-ci.

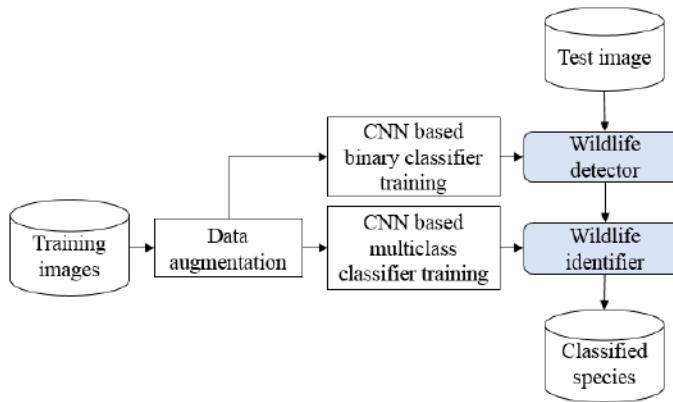


FIGURE 13: Architecture proposée par [16]

Le second article, **Fast animal detection in UAV images using convolutional neural networks** [17], par Kellenberger et al., utilise une instance pré-entraînée d'AlexNET au quels ils ont ajouté des nouvelles couches convolutionnelles. Leur méthode est divisée en 2 parties : détection locale de l'animal et ensuite taille de l'animal qui s'effectue en parallèle. Les images sont découpées en plusieurs cellules et chaque cellule reçoit un score de confiance sur la présence d'un animal et sur une estimation de la longueur et largeur de sa bounding box la plus probable. Ils ont comparé leur modèle à un Fast R-CNN comme mentionné précédemment, en utilisant le même réseau pré-entraîné AlexNET avec comme proposition d'objet les résultats d'une recherche sélective.

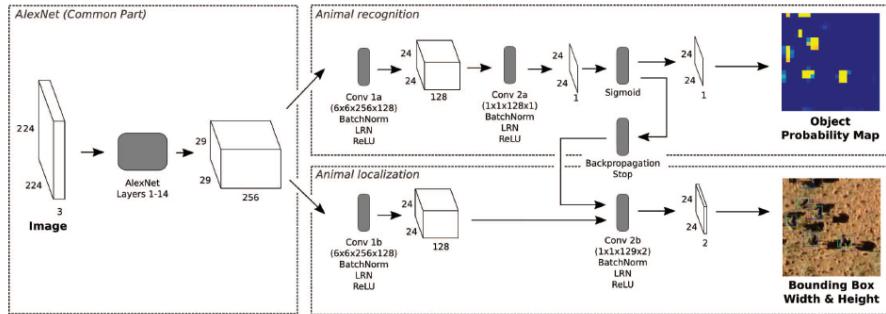


FIGURE 14: Architecture proposée par [17]

3.2 Comparaison qualitative des approches et étude analytique des performances

Le CNN peut être utilisé de plusieurs manières différentes : tout seul comme les réseaux AlexNET ou encore GoogLeNET.

Model	Trainable layers	Main specifications
AlexNet	8	5 convolutional layers and 3 fully-connected layers. [20]
VGG-16	16	13 convolutional layers with 3x3 filters, and 3 fully-connected layers. [18]
GoogLeNet	22	Developed an <i>Inception Module</i> that dramatically reduces the number of parameters while achieving high accuracy. Average pooling is used at top of CNN instead of fully-connected layers. [19]
ResNet-50	50	A deep residual learning framework, skip connections and batch normalization. Much deeper than VGG-16 (50 compared to 16) but having lower complexity and higher performance. [21]

FIGURE 15: Architectures CNN les plus communes [16]

Le point fort du CNN est sa robustesse. En effet, même en cas de données déséquilibrées, l'algorithme permet d'avoir des résultats fiables et précis. Cela permet notamment d'avoir une souplesse quant aux nouveaux datasets que l'on pourrait donner au CNN. La qualité principale que l'on ressort de cette souplesse est donc l'adaptation du CNN aux nouvelles bases de données. Cette fonction est aussi retrouvée par exemple dans le Fast R-CNN. En effet, comme une de ces étapes consiste en un CNN, il acquiert donc cette même qualité. La différence réside essentiellement dans la précision des deux algorithmes et dans leur rapidité d'exécution.

Voici ce que les deux articles ont obtenu comme résultats :

- Les résultats de la comparaison de plusieurs réseaux de convolutions avec un nombre de couches de convolutions différentes sont les suivants :

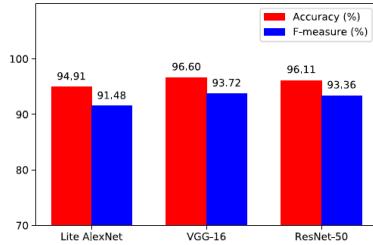


Figure 5: Animal vs. Non-animal image detection accuracy on Wildlife Spotter dataset of South-central Victoria, Australia. The data are imbalanced; the training set contains 55,000 animal images and 25,000 non-animal images, the validation set contains 18,500 animal images and 8,500 non-animal images. F-measure was used, in addition to accuracy, to evaluate the system's robustness.

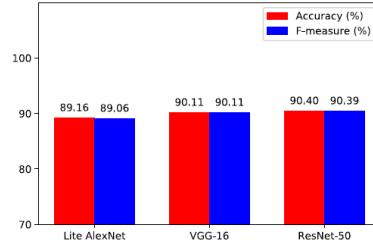


Figure 6: Animal identification accuracy on Wildlife Spotter dataset of the three most common species (*bird*, *rat*, and *bandicoot*). The training set is imbalanced as listed in Table II, 80% images of each class are used for training, 20% for validation.

FIGURE 16: Résultats obtenus par [16]

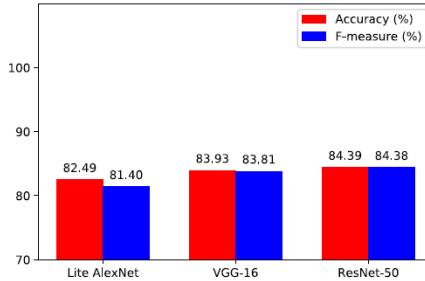


Figure 7: Animal identification accuracy on Wildlife Spotter dataset of the six most common species. The dataset is imbalanced as listed in Table II. From each class 80% images are used for training and the 20% for validation.

FIGURE 17: Résultats obtenus par [16]

Comme on peut le voir, à chaque test il compare les résultats entre les 3 architectures et elles-mêmes avec une f-measure (un score qui représente la moyenne harmonique entre la précision de l'algorithme et le recall, qui est le nombre de vrai positive divisée par le nombre d'échantillons, ce score permettant d'évaluer la robustesse d'un algorithme). Les résultats qu'ils ont obtenu sont excellents. En effet, ils ont obtenus un taux de précision allant jusqu'à 96.6% pour le VGG-16 pour la première étape. Les résultats des 3 architectures sont semblables, mais avec VGG-16 qui les surpasse pour cette étape. Pour la seconde étape, l'identification des 3 espèces prédominantes, il y a une légère détérioration de la précision. Ceci est dû par le grand nombre de classes alors que l'apprentissage est composé d'un plus petit nombre d'échantillons, cela devient donc plus difficile au modèle de s'adapter au dataset. Pour cette étape, c'est ResNET-50 qui a eu la meilleure précision. Pour la dernière étape, l'identification des 6 espèces

dominantes, c'est encore ResNET-50 qui montre les meilleurs résultats malgré une nouvelle chute de la précision. Ils ont aussi comparé VGG-16 et ResNET-50 pré-entraîné par ImageNET (fine-tuning) aux résultats précédents. On remarque que la vitesse d'exécution de l'algorithme est divisée par 61 (4000 secondes sans pré-entraînement préalable contre 65 secondes avec le fine-tuning). La conclusion que l'on peut tirer de cet article est donc : plus il y a de couches de convolution dans un CNN, meilleures sera la performance et la précision des résultats dans les problèmes de reconnaissances complexes. Il faut cependant faire attention à l'utilisation finale que l'on va en faire, pour ne pas complexifier notre problème initial.

- La comparaison de CNN avec Fast R-CNN a mené aux résultats suivant : Comme on

	Fast R-CNN (baseline)	Proposed Model
Ground Truth Objects	509	509
True Positives	429	379
False Positives	843	254
False Negatives	80	130
Precision (UA)	0.34	0.60
Recall (PA)	0.84	0.74
F1 Score	0.48	0.66
Avg. Speed [Hz]	2.96	73.62

FIGURE 18: Résultats obtenus par [17]

peut le voir, et d'après l'article, la précision est grandement augmentée. En effet, la méthode suggérée par l'article est nettement plus précise que le Fast R-CNN (0.34 vs 0.6). Cela est expliqué par le grand nombre de faux positives éliminé par leur méthode (254 vs. 843). Il y a aussi des difficultés pour le Fast R-CNN à détecter des petits animaux. En effet, le modèle proposé par l'article permet de prédire plus précisément le nombre de petite bounding boxes par animal et donc de jouer une fois de plus sur la précision des résultats obtenus. Comme dit précédemment, on remarque une nette supériorité de vitesse d'exécution : 2.96 images par seconde pour le Fast R-CNN contre 72.65 pour le modèle proposé. Le modèle peut donc être considéré comme tournant en temps réel et peut donc grandement réduire les latences lors de la surveillance des animaux sauvages.

4 Conclusions

Le but de mon projet est la reconnaissance d'espèce animale, je vais donc utiliser des databases comportant des images d'espèces animal avec plusieurs types d'espèces comme la Wildlife Spotter dataset en éliminant au préalable les images où il n'y a pas d'animaux. Pour la suite du projet, je vais plus me pencher sur la combinaison d'un CNN avec une méthode telle que YOLO pour avoir de meilleure performance. En effet, à l'issue de l'analyse critique des différentes approches, on s'aperçoit de l'importance de la combinaison de l'extraction des régions d'intérêts avec le CNN. Je pense de plus faire une comparaison de cette méthode avec une autre méthode comme R-CNN ou encore SSD, Single Shot Detectors, une autre méthode que je n'ai pas cité précédemment, mais qui est très utilisée pour la détection d'objet et sur laquelle je suis aussi en train de me documenter.

Les hypothèses que je pourrai faire pour la partie expérimentale sont de deux types :

- Concernant la vitesse d'exécution des deux méthodes, d'après l'état de l'art, c'est l'algorithme YOLO qui sera le plus rapide;
- Concernant la performance et la précision des deux méthodes, et toujours d'après l'état de l'art, c'est SSD ou R-CNN qui surpassera YOLO.

Table des figures

1	Photo de différentes espèces d'animaux dans des milieux environnementaux variés [1]	3
2	HOG appliqué à une image de chat [5]	5
3	Bag-of-Visual-Words appliqué à des images [7]	6
4	Résultat obtenu par recherche sélective [8]	7
5	Résultat obtenu par edge boxing [10]	7
6	Division de l'image par l'algorithme YOLO et sa sortie	8
7	Étape successive du R-CNN [12]	9
8	Visualisation d'un SVM (Cours de SYS828)	9
9	Constitution des clusters neuronaux chez l'humain [13]	10
10	Illustration d'une architecture typique d'un réseau neuronal convolutif [14]	10
11	Fonction d'activation ReLU	10
12	Résultat des couches convolutionnelles [15]	11
13	Architecture proposée par [16]	12
14	Architecture proposée par [17]	13
15	Architectures CNN les plus communes [16]	13
16	Résultats obtenus par [16]	14
17	Résultats obtenus par [16]	14
18	Résultats obtenus par [17]	15

Références

- [1] Ours polaire sur une banquise. <https://www.flickr.com/photos/marthaenpiet/2890352130/>. Accessed : 01-30-2019.
- [2] Alexander Gomez, Augusto Salazar, and Francisco Vargas. Towards automatic wild animal monitoring : Identification of animal species in camera-trap images using very deep convolutional neural networks. *arXiv preprint arXiv:1603.06169*, 2016.
- [3] Guobin Chen, Tony X Han, Zhihai He, Roland Kays, and Tavis Forrester. Deep convolutional neural network based species recognition for wild animal monitoring. In *Image Processing (ICIP), 2014 IEEE International Conference on*, pages 858–862. IEEE, 2014.
- [4] Navneet Dalal and Bill Triggs. Histograms of oriented gradients for human detection. In *international Conference on computer vision & Pattern Recognition (CVPR'05)*, volume 1, pages 886–893. IEEE Computer Society, 2005.
- [5] Weiwei Zhang, Jian Sun, and Xiaou Tang. From tiger to panda : animal head detection. *IEEE Transactions on Image Processing*, 20(6) :1696–1708, 2011.
- [6] Sheng Xu, Tao Fang, Deren Li, and Shiwei Wang. Object classification of aerial images with bag-of-visual words. *IEEE Geoscience and Remote Sensing Letters*, 7(2) :366–370, 2010.
- [7] Bruno Lopes and Rudinei Goularte. Multimodal late fusion bag of features applied to scene detection. pages 15–22, 11 2013.
- [8] Jasper RR Uijlings, Koen EA Van De Sande, Theo Gevers, and Arnold WM Smeulders. Selective search for object recognition. *International journal of computer vision*, 104(2) :154–171, 2013.
- [9] Selective search for object detection (c++ / python). <https://www.learnopencv.com/selective-search-for-object-detection-cpp-python/>. Accessed : 03-06-2019.
- [10] C Lawrence Zitnick and Piotr Dollár. Edge boxes : Locating object proposals from edges. In *European conference on computer vision*, pages 391–405. Springer, 2014.
- [11] Joseph Redmon, Santosh Divvala, Ross Girshick, and Ali Farhadi. You only look once : Unified, real-time object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 779–788, 2016.
- [12] Ross Girshick, Jeff Donahue, Trevor Darrell, and Jitendra Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 580–587, 2014.
- [13] Neuroscience : How do synapses form? <https://www.quora.com/Neuroscience-How-do-synapses-form>. Accessed : 03-06-2019.

- [14] Hung Nguyen, Sarah J Maclagan, Tu Dinh Nguyen, Thin Nguyen, Paul Flemons, Kylie Andrews, Euan G Ritchie, and Dinh Phung. Animal recognition and identification with deep convolutional neural networks for automated wildlife monitoring. In *2017 IEEE International Conference on Data Science and Advanced Analytics (DSAA)*, pages 40–49. IEEE, 2017.
- [15] Fei-Fei Li, Justin Johnson, and Serena Yeung. Stanford cs231n : Convolutional neural networks for visual recognition. *Assignment3* <http://cs231n.github.io/assignments2017/assignment3/> http://cs231n.stanford.edu/assignments/2017/spring1617_assignment3_v3.zip http://cs231n.stanford.edu/squeezezenet_tf.zip.
- [16] Hung Nguyen, Sarah J Maclagan, Tu Dinh Nguyen, Thin Nguyen, Paul Flemons, Kylie Andrews, Euan G Ritchie, and Dinh Phung. Animal recognition and identification with deep convolutional neural networks for automated wildlife monitoring. In *2017 IEEE International Conference on Data Science and Advanced Analytics (DSAA)*, pages 40–49. IEEE, 2017.
- [17] Benjamin Kellenberger, Michele Volpi, and Devis Tuia. Fast animal detection in uav images using convolutional neural networks. In *2017 IEEE International Geoscience and Remote Sensing Symposium (IGARSS)*, pages 866–869. IEEE, 2017.
- [18] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In F. Pereira, C. J. C. Burges, L. Bottou, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems 25*, pages 1097–1105. Curran Associates, Inc., 2012.
- [19] Benjamin Kellenberger, Michele Volpi, and Devis Tuia. Fast animal detection in uav images using convolutional neural networks. In *2017 IEEE International Geoscience and Remote Sensing Symposium (IGARSS)*, pages 866–869. IEEE, 2017.