



Rapport de stage

Mention : E3A

Parcours : M2 Réalité Virtuelle et Systèmes Intelligents

Analyse de la saillance et application à la détection et la classification d'images



EL MHAMDI EL ALAOUI Abdellah
Stage effectué du 01/04/2023 au 29/09/2023

Tuteur entreprise : M. Patrick Horain

Tuteur pédagogique : M. Hicham Hadj Abdelkader

Remerciements

Je tiens à exprimer ma gratitude envers toutes les personnes qui ont contribué, de près ou de loin, au succès de mon stage et qui m'ont apporté leur soutien dans la rédaction de ce rapport.

En tout premier lieu, je souhaite adresser mes remerciements à mon tuteur de stage, Monsieur Patrick Horain de l'école Télécom SudParis, pour sa confiance, sa convivialité et les connaissances précieuses qu'il a partagées avec moi.

Je tiens également à adresser mes sincères remerciements aux professeurs de l'Université d'Evry Val d'Essonne pour avoir mis à ma disposition les outils nécessaires à la réussite de mon stage.

Un grand merci s'adresse également à toute l'équipe du département Artemis ainsi qu'au personnel de l'école TSP pour leur accueil chaleureux au sein de l'entreprise. Leur amabilité et le temps qu'ils ont généreusement consacré à répondre à mes questions ont grandement contribué à mon expérience.

Enfin, je souhaite exprimer ma profonde gratitude envers mes parents, mon frère, ma sœur et mes amis. Leur soutien moral et financier constant, ainsi que leurs conseils avisés, ont été des éléments essentiels tout au long de mon parcours scolaire.

Table des matières

1	Introduction	7
2	Présentation de l'établissement d'accueil	8
2.1	Secteur d'activité	8
2.2	Importance	9
2.3	Schéma d'organisation succinct	9
2.4	Organisation du service	10
2.4.1	SAMOVAR	10
2.4.2	ARTEMIS	11
2.5	Rôle du tuteur entreprise	11
3	Contexte de la mission	12
3.1	Présentation du contexte de la mission	12
3.2	Description de la mission	13
3.3	Problématique	13
4	État de l'art	16
4.1	Datasets	16
4.1.1	Imagenet	16
4.2	Modèles	17
4.3	Carte d'activation de classe (CAM)	17
4.3.1	Historique des CAM	17
4.3.2	Fonctionnement des CAM	18
4.3.3	Grad-Cam et Guided Grad-Cam	18
4.3.4	ScoreCam et Faster ScoreCam	19
4.4	Approche Antérieure	20
5	Approche et travail réalisé	22
5.1	Mise en oeuvre	22
5.2	Essais et Résultats	24
5.2.1	Essai 1 : Produit	24
5.2.2	Essai 2 : Division	26
5.2.3	Essai 3 : Somme pondérée	26
6	Difficultés Rencontrées	33
6.1	Code existant peu lisible et non optimisé	33
6.2	Bugs et limitations de mémoire	33
6.3	Migration vers GOOGLE COLAB et limitations associées	33
6.4	Transition vers VS CODE avec des scripts PYTHON	34
6.5	Migration vers la plateforme de calcul départementale Vulkan	34
7	Conclusion	36
8	Perspectives	37

9 Annexes	39
9.1 Travaux Additionnels Post-Soutenance	39
9.1.1 Intégration du MANGTR	39

Table des figures

2.1	Campus de Télécom SudParis	8
2.2	organigramme de Télécom SudParis	10
3.1	Exemple d'image de cellule infectée	12
3.2	Activations par rapport à la classe de vérité terrain (Papier toilette) : non pertinente (en haut) et pertinente (en bas), avec la boîte de vérité de terrain mise en évidence en vert.	14
4.1	Exemples d'image provenants d'IMAGENET avec leurs boîtes de vérité terrain	16
4.2	Image d'IMAGENET avec sa boîte de vérité terrain (en vert) et sa CAM par rapport à la classe de vérité terrain (Tanche)	17
4.3	Aperçu de Grad-CAM[12]	19
4.4	Aperçu de Score-Cam[15]	20
4.5	Fonction de perte composé de CE et HAGTR	21
5.1	Fonction de perte composé de CE multipliée par MAGTR	24
5.2	Courbes d'apprentissage avec la fonction de perte "produit" 5.1	25
5.3	Fonction de perte composé de CE divisée par MAGTR	26
5.4	Courbes d'apprentissage avec la fonction de perte "division" 5.3	26
5.5	Fonction de perte de la somme pondérée de CE et MAGTR	27
5.6	Courbes d'apprentissage avec la fonction de perte "somme-pondérée" 5.5	28
5.7	Cartes d'activation par rapport à la classe de vérité terrain (1ère image : Noeud papillon, 2ème image : Noeud papillon, 3ème image : blouse de laboratoire, 4ème image : Mamba vert, 5ème image : Grand requin blanc, 6ème image : Rainette) pour les images correctement classées par les modèles après les époques 0 et 18, tout en présentant un MAGTR faible. À gauche, les images originales ; au milieu, les CAMs de l'époque 0 ; à droite, les CAMs de l'époque 18.	29
5.8	Matrice de confusion du modèle Resnet50 [3] entraîné sur l'entropie croisée pour 14 classes d'ImageNet (époque 0 de l'essai : somme pondérée 5.6).	30
5.9	Matrice de confusion du modèle Resnet50 [3] entraîné sur la somme pondérée de l'entropie croisée et le MAGTR pour 14 classes d'ImageNet (époque 14 de l'essai : somme pondérée 5.6).	31
9.1	Fonction de perte de la somme pondérée de CE, MAGTR et MANGTR	39
9.2	Courbes d'apprentissage avec la fonction de perte "somme-pondérée (33,33,34)" 9.1	40
9.3	Courbes d'apprentissage avec la fonction de perte "somme-pondérée (5,5,90)" 9.1	40
9.4	Courbes d'apprentissage avec la fonction de perte "somme-pondérée (5,0,95)" 9.1	41

9.5	Cartes d'activation par rapport à la classe de vérité terrain pour des images qui ont été mal classées et sont devenues correctement classées. À gauche, les images originales ; au milieu, les CAMs de l'époque 0 ; à droite, les CAMs de l'époque 15 de l'expérience 9.4.	42
9.6	Cartes d'activation par rapport à la classe de vérité terrain pour des images qui ont été correctement classées à l'époque 0 et sont devenues mal classées. À gauche, les images originales ; au milieu, les CAMs de l'époque 0 ; à droite, les CAMs de l'époque 15 de l'expérience 9.4. . . .	43

Chapitre 1

Introduction

Les avancées en matière d'intelligence artificielle (IA) ont permis des progrès significatifs dans le domaine de la vision par ordinateur, notamment dans des tâches telles que la classification d'images. Cependant, les résultats obtenus restent souvent difficilement explicables. En particulier, l'activation des réseaux convolutifs est parfois très éloignée de la localisation véritable des objets détectés dans les images. Cette limitation est particulièrement préoccupante dans des domaines sensibles tels que la médecine, où il est essentiel de pouvoir avoir confiance en les décisions prises par les applications d'IA.

Dans le cadre de ce stage, l'objectif est d'explorer des méthodes visant à renforcer la fiabilité des décisions de classification en prenant en compte l'activation des réseaux neuronaux pendant l'apprentissage. L'idée sous-jacente est d'utiliser les informations fournies par ces réseaux pour améliorer l'explicabilité et ainsi la confiance accordée aux résultats obtenus.

Le travail de recherche comprendra une comparaison entre les approches traditionnelles basées sur les cartes de saillance, telles que GradCam et ScoreCam. L'évaluation de ces méthodes sera réalisée sur deux ensembles de données distincts. Tout d'abord, une évaluation sera effectuée sur la base de données ImageNet, qui est un référentiel couramment utilisé pour évaluer les performances des modèles de vision par ordinateur. Ensuite, une évaluation spécifique sera réalisée sur des images de frottis sanguins pour la détection des parasites responsables de la paludisme, une maladie grave qui affecte de nombreuses personnes à travers le monde.

Les résultats attendus de ce stage sont une meilleure compréhension de l'activation des réseaux neuronaux pendant l'apprentissage et le développement de méthodes plus robustes pour la prise de décisions en utilisant ces informations. L'accent sera mis sur l'utilisation de cartes de saillance pour identifier les régions d'intérêt dans les images, afin d'accroître la confiance dans les résultats de classification obtenus.

Chapitre 2

Présentation de l'établissement d'accueil

Mon stage se déroulait à l'école Télécom SudParis. Fondée en 1979, TSP est une grande école publique d'ingénieurs reconnue au meilleur niveau des sciences et technologies du numérique. La qualité de ses formations est basée sur l'excellence scientifique de son corps professoral et une pédagogie mettant l'accent sur les projets d'équipes, l'innovation de rupture et l'entrepreneuriat. Dans ce chapitre, je vais vous présenter l'entreprise et son service d'accueil.



FIGURE 2.1 – Campus de Télécom SudParis

2.1 Secteur d'activité

Télécom SudParis est une école d'ingénieurs en télécommunications et technologies de l'information située à Évry, en France. Elle est classée dans le secteur de l'enseignement supérieur et est identifiée par le code APE 8542Z. Ce code APE désigne les établissements d'enseignement supérieur qui proposent des formations dans divers domaines tels que l'informatique, l'électronique, la télécommunication, etc.

En tant qu'école d'ingénieurs spécialisée dans les technologies de l'information et de la communication, Télécom SudParis offre une formation de haut niveau à ses étudiants, qui peuvent se spécialiser dans des domaines tels que la sécurité des réseaux, la communication sans fil, l'intelligence artificielle, etc.

2.2 Importance

Télécom SudParis est une école d'ingénieurs à but non lucratif, le chiffre d'affaires n'est pas un indicateur pertinent de son importance. Cependant, l'école emploie environ 200 personnes, dont 103 enseignants-chercheurs permanents. Les autres membres du personnel travaillent dans les services administratifs, les laboratoires de recherche, les centres de formation continue, etc. Télécom SudParis accueille également plus de 2 000 étudiants chaque année, ce qui en fait une institution importante dans le domaine de l'enseignement supérieur en France.

En outre, l'école TSP fait partie de l'iMT. TSP est l'une des 5 écoles membres de l'Institut Polytechnique de Paris, un groupe d'écoles d'ingénieurs spécialisées dans les technologies de l'information et de la communication, ce qui renforce encore son importance dans le secteur.

2.3 Schéma d'organisation succinct

Télécom SudParis est structuré en une Direction, 6 directions métiers, 6 départements d'enseignement et de recherche, 4 services communs avec IMT-BS. Le schéma ci-dessous présente de manière synthétique la structure organisationnelle de Télécom SudParis :

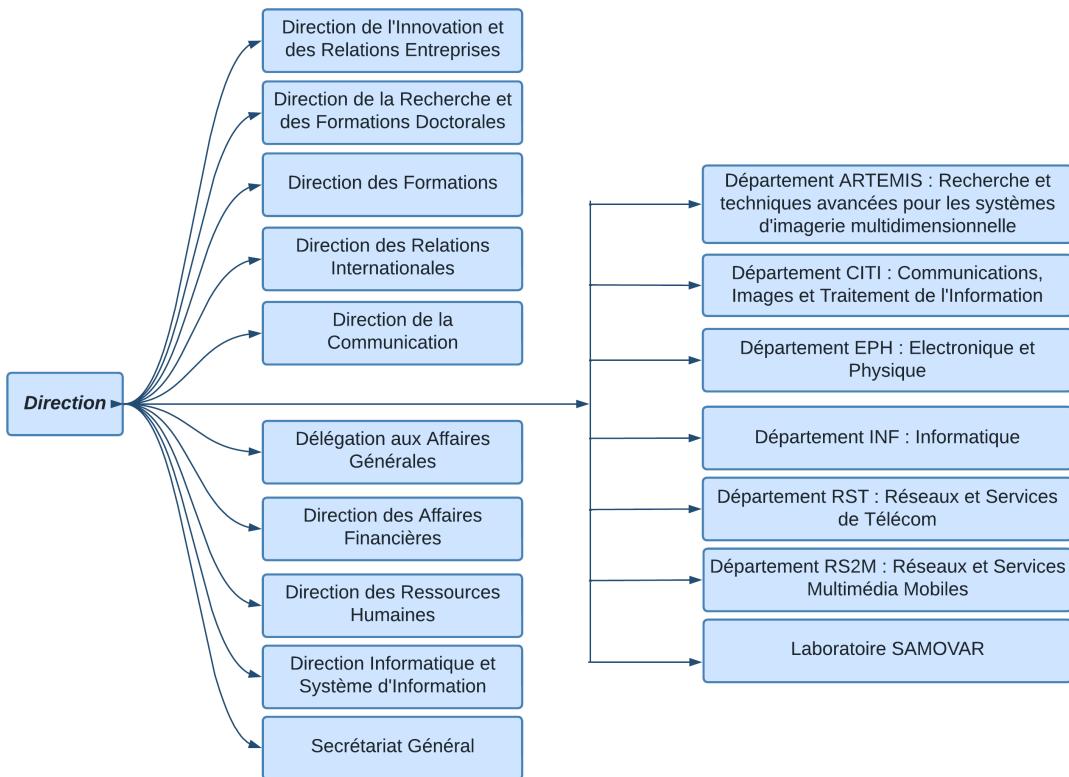


FIGURE 2.2 – organigramme de Télécom SudParis

2.4 Organisation du service

Dans Télécom SudParis, il existe un laboratoire appelé SAMOVAR, qui regroupe plusieurs équipes de chercheurs travaillant dans le domaine des services, des réseaux et des télécommunications. En plus de cela, il y a également différents départements de recherche qui rassemblent des enseignants-chercheurs issus du laboratoire SAMOVAR. J'ai eu l'opportunité d'effectuer mon stage dans le département ARTEMIS.

2.4.1 SAMOVAR

Le laboratoire SAMOVAR (Services répartis, Architectures, Modélisation, Validation, Administration des Réseaux) regroupe plusieurs équipes de chercheurs qui travaillent dans le domaine des services, des réseaux et des télécommunications.

SAMOVAR est un des rares laboratoires à couvrir l'ensemble des thématiques des systèmes de communications. Menées en relation avec des industriels, ses activités allient qualité scientifique et originalité.

2.4.2 ARTEMIS

Le département ARTEMIS a été créé en 1999. Il regroupe aujourd’hui une trentaine de personnes. Le cœur des recherches d’ARTEMIS relève des sciences et technologies de l’image numérique. Celles-ci lancent en effet de nombreux défis aux acteurs économiques et académiques afin de concevoir méthodes innovantes et nouveaux services pour la société de l’information. L’image numérique d’aujourd’hui couvre le large éventail des données visuelles : multimédia (ex : photos, télévision), biologiques (scanner, microscopie...) ou 3D (réalité virtuelle et augmentée...) en environnement fixe ou mobile.

ARTEMIS traite de la chaîne de l’image depuis la création des contenus numériques jusqu’à leur diffusion. L’enjeu est de créer, modéliser, analyser, indexer, animer, sécuriser, manipuler, enrichir, coder, distribuer et visualiser des contenus hétérogènes et complexes pour des services d’intermédiation économiquement réalistes.

2.5 Rôle du tuteur entreprise

Mon tuteur entreprise, Monsieur Patrick Horain est membre du département ARTEMIS à Telecom SudParis. Ses activités d’enseignement consistent à créer des unités de cours et des conférences pour les étudiants en ingénierie et en master. Il coordonne le domaine académique de l’image, du multimédia et des applications à Telecom SudParis.

Ses intérêts de recherche portent sur le traitement d’images et la vision par ordinateur en général. Il possède une longue expérience dans les interfaces perceptives pour l’interaction homme-machine. Au sein du réseau de recherche du CNRS sur l’Information, le Signal, les Images et la Vision (ISIS), il a coanimé une action spéciale sur le visage, le geste, l’action et le comportement (2009-2016). Il a coorganisé et présidé les conférences internationales IHCI 2014 et IHCI 2017 sur l’interaction homme-machine intelligente. Il a également présidé le comité d’organisation de la conférence Web3D 2022.

Chapitre 3

Contexte de la mission

3.1 Présentation du contexte de la mission

Le paludisme, également connu sous le nom de malaria, est une maladie infectieuse grave transmise par les moustiques et causée par les parasites du genre *Plasmodium* qui attaquent les globules rouges du sang. Ses symptômes courants tels que la fièvre, les frissons, les maux de tête et les douleurs musculaires nécessitent une détection et un traitement rapides pour éviter des complications graves.



FIGURE 3.1 – Exemple d'image de cellule infectée

Traditionnellement, le diagnostic du paludisme se fait par la méthode du frottis sanguin au microscope, mais la détection précise des parasites est un défi en raison de leur variabilité et de leur faible concentration dans le sang. C'est là que les méthodes automatiques, comme l'utilisation d'algorithmes de détection tels que les réseaux de neurones, peuvent apporter une aide précieuse. Ces approches automatisées permettent de traiter de vastes quantités de données et de rechercher des caractéristiques spécifiques des parasites, offrant ainsi une alternative prometteuse pour améliorer la détection précoce et le traitement rapide de cette maladie potentiellement mortelle.

Cependant, malgré les défis liés à l'interprétation des résultats des algorithmes de détection (notamment les réseaux de neurones), leur utilisation peut apporter des informations précieuses pour orienter les professionnels de la santé dans l'analyse d'échantillons sanguins et dans la prise de décisions cliniques en vue d'un diagnostic plus précis et d'un traitement rapide du paludisme. Il convient de souligner qu'une condition pour établir la santé d'un patient est d'examiner au moins 200 000 globules rouges et de ne pas y détecter de parasites.

3.2 Description de la mission

Ce rapport de stage porte sur une mission passionnante réalisée au sein du département ARTEMIS de l'école Telecom SudParis. Notre objectif principal était d'étudier les cartes de saillance dans le domaine de la vision par ordinateur et de trouver des moyens de renforcer la fiabilité des décisions de classification obtenues par les réseaux convolutifs.

Pour remédier à ce problème, notre objectif principal dans ce stage était d'explorer des approches qui prennent en compte l'activation des cartes de saillance lors de l'apprentissage. Pour évaluer ces approches, nous avons utilisé deux ensembles de données distincts. Tout d'abord, nous avons utilisé la base d'images ImageNet, qui est largement utilisée dans la recherche en vision par ordinateur. Ensuite, nous nous sommes concentrés sur des images de frottis sanguins pour la détection des parasites Plasmodium, responsables du paludisme.

Une chose à noter est que ce stage fait suite aux travaux réalisés par deux anciens stagiaires. Le premier stagiaire, Adam Allah, a réalisé un état de l'art sur les modèles de classification et les approches de génération des cartes d'activation[1]. Le deuxième stagiaire, Wyctor Fogos da Rochas, a tenté d'entraîner un modèle (VGG16[13]) en utilisant un entraînement personnalisé pour améliorer les cartes d'activation[10](nous reviendrons sur ces travaux dans les sections suivantes).

Dans les chapitres qui suivent, nous expliquons les travaux réalisés par Adam et Wyctor. Nous détaillons également les différentes étapes que nous avons suivies tout au long du stage, notamment la collecte des données, la mise en œuvre des modèles de réseaux convolutifs, la génération des cartes de saillance, les expérimentations réalisées et l'évaluation des performances de nos approches. Nous présenterons en détail notre approche, les résultats obtenus et les analyses effectuées.

3.3 Problématique

Dans les réseaux de neurones convolutifs (CNN), les activations font référence aux régions spécifiques d'une image qui sont fortement responsables de la décision prise par le modèle. Ces régions sont détectées par le CNN en réponse à des caractéristiques visuelles significatives liées à la classe de l'objet. Cependant, certaines techniques l'explication de la catégorie peuvent parfois donner des résultats incohérents, où les zones activées ne correspondent pas toujours à la véritable région de l'objet de la catégorie prédite.

Voici deux exemples 3.2 d'images correctement classées, accompagnés de leurs activations par rapport à la classe de vérité de terrain, illustrant à la fois une activation pertinente et une activation moins précise :



FIGURE 3.2 – Activations par rapport à la classe de vérité terrain (Papier toilette) : non pertinente (en haut) et pertinente (en bas), avec la boîte de vérité de terrain mise en évidence en vert.

Cette discordance entre la classification et les activations engendre un manque de confiance dans les modèles, car leurs résultats ne sont pas toujours explicables de manière cohérente. En conséquence, il devient crucial d'améliorer l'explicabilité de ces modèles afin que les activations détectées soient davantage pertinentes par rapport à la classification réalisée.

Le principal objectif de ce stage, comme mentionné dans l'introduction, est donc de proposer des améliorations pour rendre les activations des modèles plus pertinentes et concordantes avec la classification effectuée. Cela permettra de mieux comprendre le processus de décision du modèle et de gagner en confiance quant à l'exactitude de ses prédictions et leur explicabilité.

Pour atteindre cet objectif, diverses approches peuvent être envisagées. Il est possible d'explorer de nouvelles méthodes d'attribution d'importance aux caractéristiques apprises par le modèle, ou encore d'introduire des mécanismes d'apprentissage supplémentaires visant à renforcer la corrélation entre les activations et la classification. De plus, il peut être intéressant d'investiguer les raisons sous-jacentes des incohérences observées et de proposer des stratégies pour les atténuer.

Le résultat attendu de ce travail serait donc une meilleure compréhension de la façon dont les modèles de classification prennent leurs décisions, rendant ainsi leurs activations plus fiables et explicables. Ces améliorations pourraient contribuer à accroître l'adoption et l'utilisation de ces modèles dans des domaines critiques tels que la santé, la sécurité ou encore l'automobile autonome, où une interprétabilité fiable est primordiale.

Chapitre 4

État de l'art

4.1 Datasets

4.1.1 Imagenet

ImageNet[11] est une base de données d'images organisée selon la hiérarchie WordNet (actuellement uniquement pour les noms), dans laquelle chaque nœud de la hiérarchie est représenté par des centaines et des milliers d'images. Ce projet a joué un rôle essentiel dans l'avancement de la recherche en vision par ordinateur et en apprentissage profond. Les données sont disponibles gratuitement pour les chercheurs à des fins non commerciales.

On a utilisé cette base de données pour 3 raisons : premièrement, parce qu'elle est très grande (plus de 14 millions d'images). Deuxièmement, nous disposons des boîtes de vérité terrain pour des centaines d'images par classe. Et troisièmement, les modèles de classification avec lesquels nous travaillons ont déjà été entraînés sur ImageNet.

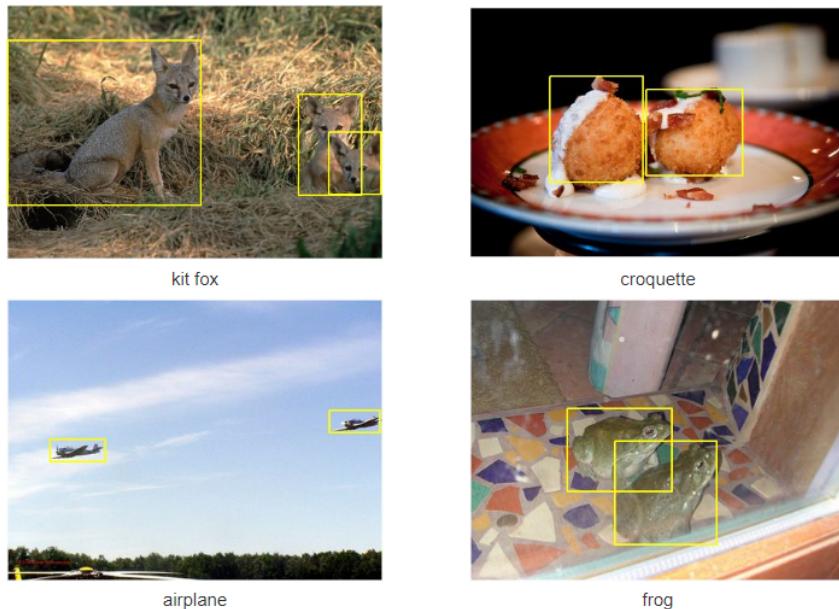


FIGURE 4.1 – Exemples d'image provenants d'IMAGENET avec leurs boîtes de vérité terrain

4.2 Modèles

Adam et Wyctor[1, 10] ont travaillé avec le VGG16[13]. J'ai choisi le ResNet50V2[3] car il est plus performant en termes de précision de classification que le VGG16[13], tout en occupant moins d'espace mémoire avec moins de paramètres. De plus, le ResNet50V2 nécessite moins de temps par étape d'inférence[5].

4.3 Carte d'activation de classe (CAM)

Les CAM (pour Class Activation Maps) sont une technique en vision par ordinateur pour visualiser et comprendre les régions d'intérêt des images utilisées par les réseaux de neurones convolutifs lors de la prise de décision. Cette approche permet de localiser visuellement les zones d'une image qui ont le plus contribué à la classification d'une classe spécifique par le modèle.

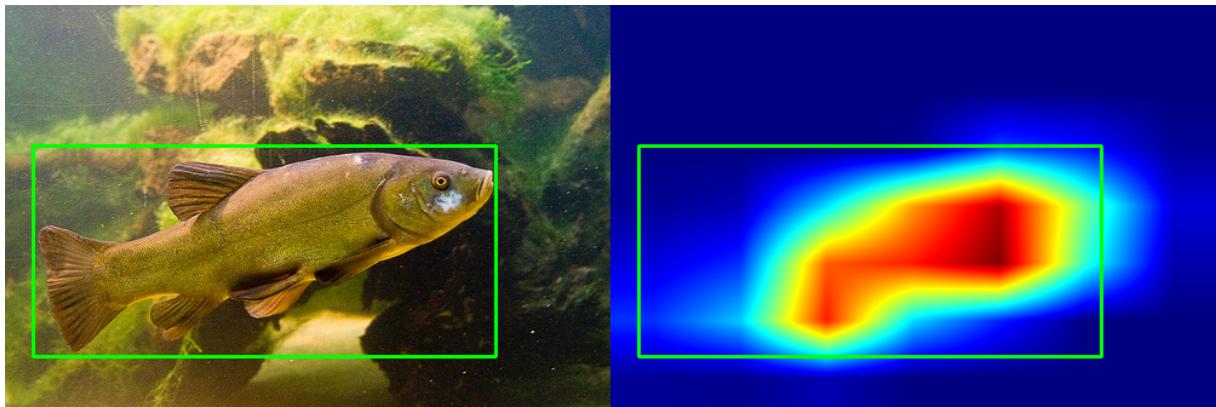


FIGURE 4.2 – Image d'IMAGENET avec sa boîte de vérité terrain (en vert) et sa CAM par rapport à la classe de vérité terrain (Tanche)

Je vais présenter l'évolution des Cartes d'Activation Maximale (CAM) et leur fonctionnement, en mettant en évidence les avancées majeures de la recherche dans ce domaine. Je discuterai des avantages, des limitations et de leur utilisation concrète dans la classification d'images. En outre, je vais explorer la pertinence des CAM dans notre étude et comment elles peuvent améliorer la compréhension et l'interprétation de notre modèle de classification.

4.3.1 Historique des CAM

Les CAM ont été introduites en 2015 par B. Zhou et al. dans leur article "Learning Deep Features for Discriminative Localization"[16]. Depuis lors, les CAM ont suscité un intérêt considérable dans la communauté de la recherche en vision par ordinateur.

4.3.2 Fonctionnement des CAM

Les CAM permettent de mettre en évidence les régions d'une image qui contribuent le plus à la classification d'une classe spécifique par le modèle de réseau de neurones convolutifs. Pour cela, la dernière couche convulsive est modifiée de manière à ce que ses activations soient directement liées aux classes d'intérêt. Cela permet de créer une carte de chaleur qui indique les régions d'activation importantes pour chaque classe.

Pour obtenir les CAM, le réseau est entraîné sur un ensemble de données étiqueté. Une fois l'apprentissage terminé, les activations de la dernière couche convulsive sont pondérées par les poids des neurones dans la couche entièrement connectée correspondant à la classe d'intérêt. Ces activations pondérées sont ensuite agrégées pour créer la carte d'activation pour cette classe spécifique.

4.3.3 Grad-Cam et Guided Grad-Cam

Selvaraju et al. [12] ont introduit la méthode "Grad-CAM" (Gradient-weighted Class Activation Mapping) dans leur article "Grad-CAM : Visual Explanations from Deep Networks via Gradient-based Localization" en 2017. Cette approche a permis d'obtenir des cartes d'activation plus interprétables en utilisant les gradients de la sortie de classe plutôt que les poids des couches.

À partir d'une image et d'une classe d'intérêt en entrée, ils propagent l'image à travers la partie CNN du modèle, puis effectuent les calculs spécifiques à la tâche pour obtenir un score brut pour la catégorie. Les gradients sont fixés à zéro pour toutes les classes, sauf la classe désirée (tiger cat), qui est fixée à 1. Ce signal est ensuite rétropropagé aux cartes de caractéristiques convolutionnelles rectifiées d'intérêt, qu'ils combinent pour calculer la localisation grossière Grad-CAM (carte de chaleur bleue), qui représente où le modèle doit regarder pour prendre la décision particulière. Enfin, ils multiplient point par point la carte de chaleur avec la propagation guidée pour obtenir des visualisations Guided Grad-CAM, qui sont à la fois haute résolution et spécifiques au concept.

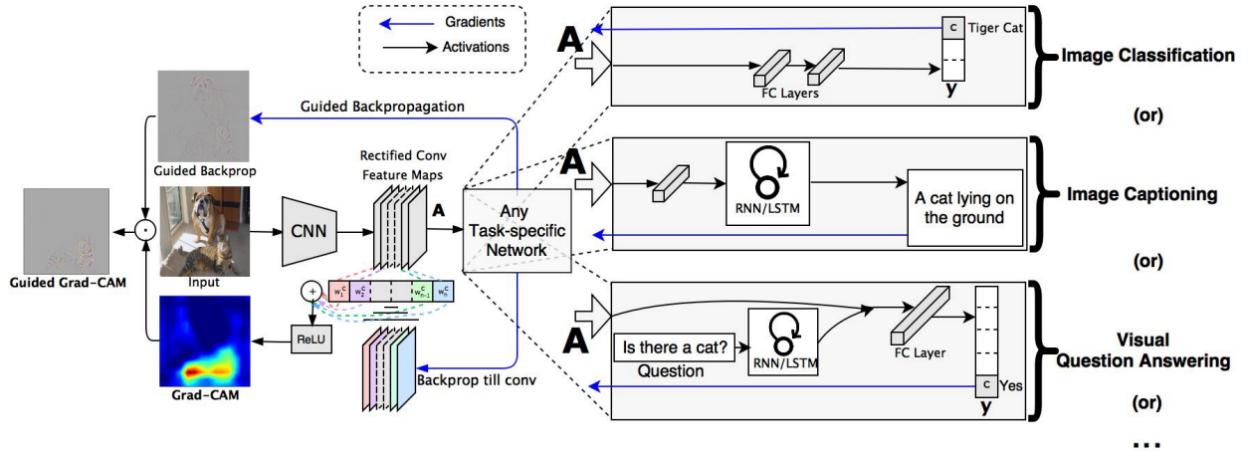


FIGURE 4.3 – Aperçu de Grad-CAM[12]

4.3.4 ScoreCam et Faster ScoreCam

Comme le Grad-CAM, le Score-CAM permet de visualiser les régions importantes des images pour la classification effectuée par le modèle. Cette approche a été introduite par Wang et autres dans leur papier intitulé "Score-Weighted Visual Explanations for Convolutional Neural Networks"[15].

Le pipeline proposé du Score-CAM se déroule en deux phases. Dans la première phase, les cartes d'activation sont extraites à partir des couches convolutives du modèle CNN. Chaque carte d'activation agit ensuite comme un masque sur l'image d'origine, et le modèle calcule son score de passage avant (forward-passing score) pour la classe cible. La deuxième phase se répète N fois, où N est le nombre de cartes d'activation. À chaque itération, une nouvelle carte d'activation est utilisée pour mettre en évidence les régions importantes de l'image pour la classe cible. Ainsi, plusieurs cartes d'activation sont obtenues, chacune montrant différentes régions significatives. Enfin, le résultat final est généré en réalisant une combinaison linéaire des poids basés sur les scores obtenus dans la première phase, ainsi que des cartes d'activation elles-mêmes. Cette combinaison permet de mettre en évidence les zones les plus importantes de l'image qui ont contribué à la prédiction du modèle pour la classe d'intérêt.

Les phases 1 et 2 partagent le même module CNN en tant qu'extracteur de caractéristiques, ce qui rend le processus plus efficace et évite la duplication des calculs. Ainsi, le Score-CAM fournit des explications visuelles pertinentes et compréhensibles pour les décisions prises par le modèle CNN, améliorant ainsi la transparence et la confiance dans ses prédictions.

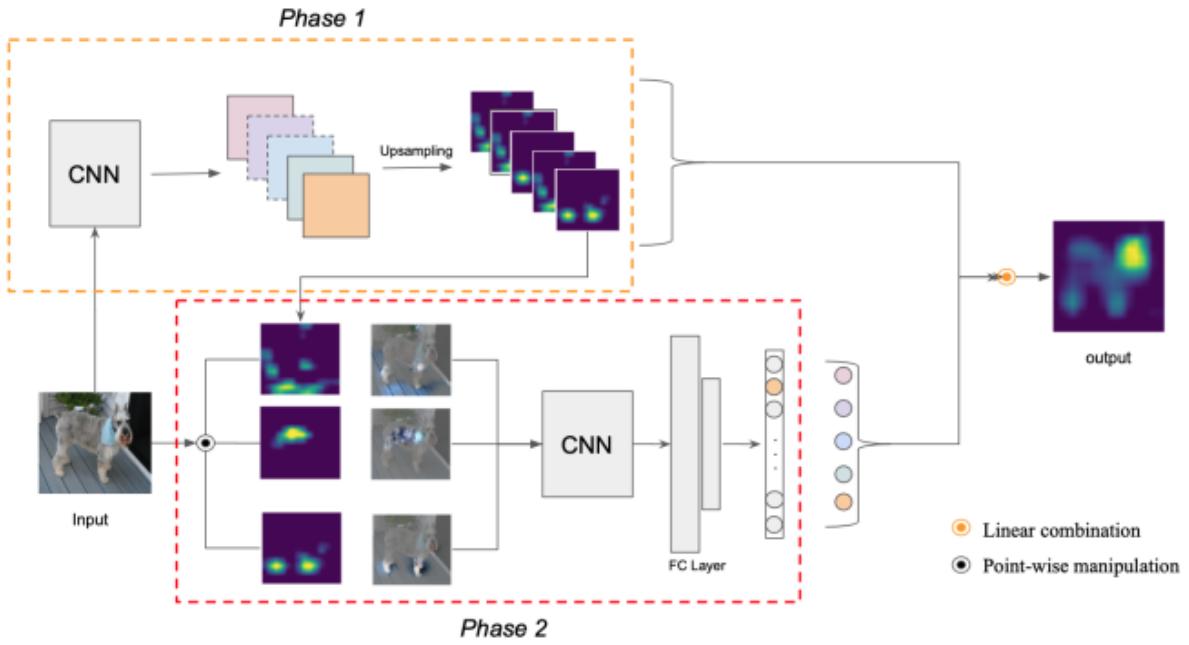


FIGURE 4.4 – Aperçu de Score-Cam[15]

Le Faster SCORECAM[15] est une amélioration du modèle Score-CAM que nous avons précédemment présenté. Il apporte une optimisation au niveau de la rapidité de calcul par rapport à SCORECAM. En d'autres termes, il est plus efficace en termes de temps d'exécution tout en fournissant des résultats d'interprétation de qualité similaire.

Cette amélioration est essentielle car elle permet une analyse plus rapide des modèles de vision complexes, ce qui est particulièrement utile lorsque nous traitons de grandes quantités de données ou lorsque nous avons besoin d'obtenir des résultats d'interprétation en temps réel.

4.4 Approche Antérieure

Les approches discutées ici ont pour objectif d'améliorer la manière dont un modèle identifie et localise les zones importantes dans une image, plutôt que de se limiter à la simple classification. Cette modification de l'objectif d'optimisation découle de la nécessité de mieux comprendre les décisions du modèle, en particulier dans des domaines où la localisation précise des objets ou des caractéristiques est cruciale, comme la vision par ordinateur.

Lorsqu'un modèle est optimisé uniquement pour la classification à l'aide d'une seule fonction de perte (généralement l'entropie croisée), il peut privilégier les caractéristiques globales de l'image au détriment de la localisation précise des objets d'intérêt. Cela peut conduire à des situations où le modèle réussit la classification globale, mais échoue à identifier correctement les zones importantes dans l'image.

Au lieu de simplement optimiser la classification, on l'optimise conjointement avec un autre critère, ce qui a conduit à l'introduction du terme HAGTR (ou plus tard le MAGTR) dans la fonction de perte. Cela vise à guider le modèle vers une focalisation accrue sur les zones de l'image correspondant aux emplacements réels des objets d'intérêt, tout en sachant que cela peut potentiellement entraîner une légère diminution de la précision. Cependant, l'objectif est que cette dégradation de la précision soit minime.

Le HAGTR, qui correspond à l'activation la plus élevée dans la région de vérité terrain[10]. La fonction de perte est alors égale au produit entre l'entropie croisée et $(1 - \text{HAGTR})$. Ils ont utilisé $(1 - \text{HAGTR})$ parce qu'ils minimisent la fonction de perte, donc s'ils veulent maximiser le HAGTR, il faut minimiser $-\text{HAGTR}$. De plus, ils ont ajouté un 1 car le HAGTR est normalisé entre 0 et 1.

```
loss = Cross_Entrop * (1-HAGTR)
```

FIGURE 4.5 – Fonction de perte composé de CE et HAGTR

Le HAGTR a été calculé à partir des CAM (Class Activation Maps) calculées par la méthode GradCam. Cela consiste à prendre la CAM, à l'échantillonner à la taille de l'image d'origine, puis à calculer l'activation la plus élevée dans la région de vérité terrain. Cette approche a été expérimentée sur 2 classes d'IMAGENET (Sac de boxe et épis de maïs), puis sur 14 classes d'IMAGENET. Il est généralement préférable de commencer par une petite base de données pour obtenir rapidement des résultats, car les entraînements prennent beaucoup de temps, et c'est plus facile à gérer et à analyser les résultats.

Cependant, les résultats de ces expériences ont abouti à des légères dégradations au niveau de la précision de classification avec une petite amélioration de l'activation dans la région de vérité terrain. Pour cette raison, ils ont essayé une autre approche, en remplaçant le HAGTR par le MAGTR, qui correspond à l'activation moyenne dans la région de vérité terrain. Cette solution n'était pas complète, et c'est à partir de ce point que j'ai continué ces travaux.

Chapitre 5

Approche et travail réalisé

On a étudié différentes façons d'améliorer à la fois la classification et la localisation de l'activation, dont l'ajout du terme MAGTR à la fonction de perte en combinaison avec l'entropie croisée, on minimise l'entropie croisée tout en maximisant le MAGTR.

5.1 Mise en oeuvre

La mise en œuvre de ces approches nécessite plusieurs étapes, que je vais détailler ci-dessous :

- La préparation de données : Il convient de noter que les travaux antérieurs sur ce projet portaient sur 2 classes d'ImageNet[11], Cependant, ma tâche consistait à travailler sur 14 classes d'ImageNet, puis à passer à une échelle plus grande (plus de classes). Pour optimiser les calculs, il était nécessaire de préparer les données avant de démarrer l'entraînement. Cette préparation incluait le chargement des images à partir de notre base de données, leur conversion en tableaux compatibles avec les tenseurs, leur mise à l'échelle et leur normalisation afin d'obtenir une moyenne proche de zéro et une échelle relativement petite. Ces étapes peuvent contribuer à améliorer la stabilité et les performances de l'apprentissage. Ensuite, les étiquettes ont été préparées en les encodant pour qu'elles correspondent à l'entrée du modèle. De plus, un fichier CSV a été préparé pour contenir les coordonnées des boîtes de vérité terrain des images. Enfin, la base de données a été divisée en ensembles d'entraînement, de validation et de test, avec des proportions de 70%, 15% et 15 % respectivement.
- Une fois la préparation des données terminée, la mise en place du modèle à entraîner est intervenue (ResNet50 [3]). À ce stade, nous avons mis en œuvre la méthode du transfert d'apprentissage, laquelle consiste à utiliser un modèle préalablement entraîné pour identifier des caractéristiques générales de bas niveau présentes dans les données, et ensuite l'adapter pour résoudre une tâche spécifique en ajoutant de nouvelles couches conçues spécifiquement pour cette tâche. Cette approche permet de gagner du temps et d'obtenir de bons résultats, même avec

peu de données pour effectuer un entraînement pour une nouvelle tâche. Dans notre cas, nous avons utilisé les couches d'extraction de caractéristiques de ResNet50, préalablement entraîné sur les 1000 classes d'ImageNet [11], auxquelles nous avons ajouté des couches de classification adaptées au nombre de classes (14). Cette étape visait à établir un point de référence que nous désignerons comme "époque 0", afin de pouvoir comparer les résultats obtenus avec nos approches ultérieures. Ci-dessous 5.1, nous présentons les performances du modèle après cette phase d'entraînement (époque 0), que nous utiliserons comme base de référence pour nos essais ultérieurs.

Données	Précision (%)	Entropie-croisée	MAGTR
Validation	95.84	0.1605	0.3648

TABLE 5.1 – Tableau de référence (époque 0)

- Entraînement multitâches : Suite à la préparation des données et à la mise en place du modèle, une boucle d'apprentissage personnalisée a été élaborée en utilisant TensorFlow[7] et Keras[5]. Cette approche sur mesure implique l'utilisation de l'objet "GradientTape" pour enregistrer les opérations effectuées pendant la propagation avant (forward pass) du modèle. À l'intérieur de cette boucle, les prédictions sont calculées, la perte est évaluée en utilisant une fonction de perte définie, et les gradients sont calculés par rapport aux paramètres du modèle. Ces gradients sont ensuite utilisés pour mettre à jour les poids du modèle via un optimiseur sélectionné. Cette boucle est répétée pour chaque lot d'entraînement, permettant ainsi un ajustement progressif des paramètres du modèle et l'amélioration de ses performances sur la tâche donnée.
- Ajustement des hyperparamètres : L'ajustement des hyperparamètres consiste à régler des valeurs clés qui ne sont pas directement apprises par le modèle, telles que les taux d'apprentissage et la taille des lots d'entraînement, entre autres. L'objectif est d'optimiser les performances du modèle sur les données d'entraînement et de validation en expérimentant différentes combinaisons de valeurs. Cette étape nécessite généralement une approche itérative, où les performances du modèle sont évaluées sur un ensemble de validation. L'ajustement des hyperparamètres joue un rôle crucial dans l'obtention de modèles d'apprentissage automatique performants et généralisables, adaptés aux spécificités de chaque tâche
- L'évaluation : L'évaluation du modèle consiste à mesurer ses performances et sa capacité à effectuer des prédictions précises sur de nouvelles données. Cela implique de tester le modèle sur un ensemble de données distinct (ensemble de test ou de validation) qui n'a pas été utilisé pendant l'entraînement. Les métriques d'évaluation, telles que la précision, l'exactitude, le rappel, etc., permettent de quantifier la qualité des prédictions du modèle. L'évaluation est essentielle pour

déterminer si le modèle généralise bien au-delà des données d'entraînement et pour identifier d'éventuelles lacunes ou erreurs à corriger. Nous avons utilisé la précision et le terme MAGTR pour évaluer la qualité de la précision et de l'activation, respectivement.

Nous présenterons les résultats des approches que nous avons testées. Chacune de ces expériences a contribué à approfondir notre compréhension du problème et à orienter nos efforts vers des stratégies plus prometteuses, comme le démontrera la discussion à venir.

5.2 Essais et Résultats

5.2.1 Essai 1 : Produit

La fonction de perte est le produit de l'entropie croisée par (1-MAGTR).

```
loss_value = Cross_Entrop*(1-MAGTR)
```

FIGURE 5.1 – Fonction de perte composé de CE multiplié par MAGTR

Il est important de noter que l'ajustement des hyperparamètres lors de l'entraînement prend beaucoup de temps, pouvant s'étendre sur plusieurs jours voire des semaines. Dans le cadre de notre expérience, un autre hyperparamètre crucial est le nombre de canaux N_max (ou max_N), qui est un paramètre de Faster-Scorecam, comme expliqué dans le chapitre État de l'art [sous-section 4.3.4](#). Ce paramètre correspond au nombre maximal de canaux à prendre en compte pour calculer la carte d'activation. Son utilisation accélère significativement l'entraînement et évite les contraintes liées à la mémoire. Par conséquent, je ne présenterai ici que les expériences ayant donné les meilleurs résultats en termes d'hyperparamètres.

En ce qui concerne le produit, nous avons initialement essayé d'ajuster la taille des lots d'entraînement. Cependant, le code n'était pas compatible avec différentes tailles de lots, car il ne traitait qu'une seule image par lot. Pour résoudre ce problème, nous avons entrepris des modifications substantielles dans le code. Nous avons revu et adapté toutes les fonctions utilisées dans la boucle d'entraînement afin qu'elles puissent traiter efficacement les images par lot de tailles variables. Ces ajustements nous ont permis d'assurer la compatibilité du code avec différentes tailles de lots, améliorant ainsi la flexibilité et les performances globales du système. Ensuite, nous avons cherché à déterminer la meilleure valeur pour N_max. Malheureusement, en raison des contraintes de mémoire, nous n'avons pas pu explorer des valeurs étendues de N_max (au-delà de 50 canaux) ni utiliser des tailles de lots importantes (au-delà de 6 images) dans le cadre de cette approche. Cependant, j'ai réussi à accroître la

taille du lot en utilisant une technique d'accumulation de gradient, ce qui a permis de contourner ces limitations. Cette méthode fonctionne en divisant un lot en sous-lots plus petits, calculant les gradients pour chaque sous-lot, puis en accumulant ces gradients avant de mettre à jour les poids du modèle. Cela permet d'exploiter des lots plus grands sans surcharger la mémoire, améliorant ainsi les performances globales du processus d'apprentissage.

Cette expérience a permis de corriger de nombreuses erreurs présentes dans le programme, que ce soit lors de la préparation des données, de la boucle d'entraînement ou de l'évaluation.

Voici le meilleur résultat que nous avons obtenu pour le produit :

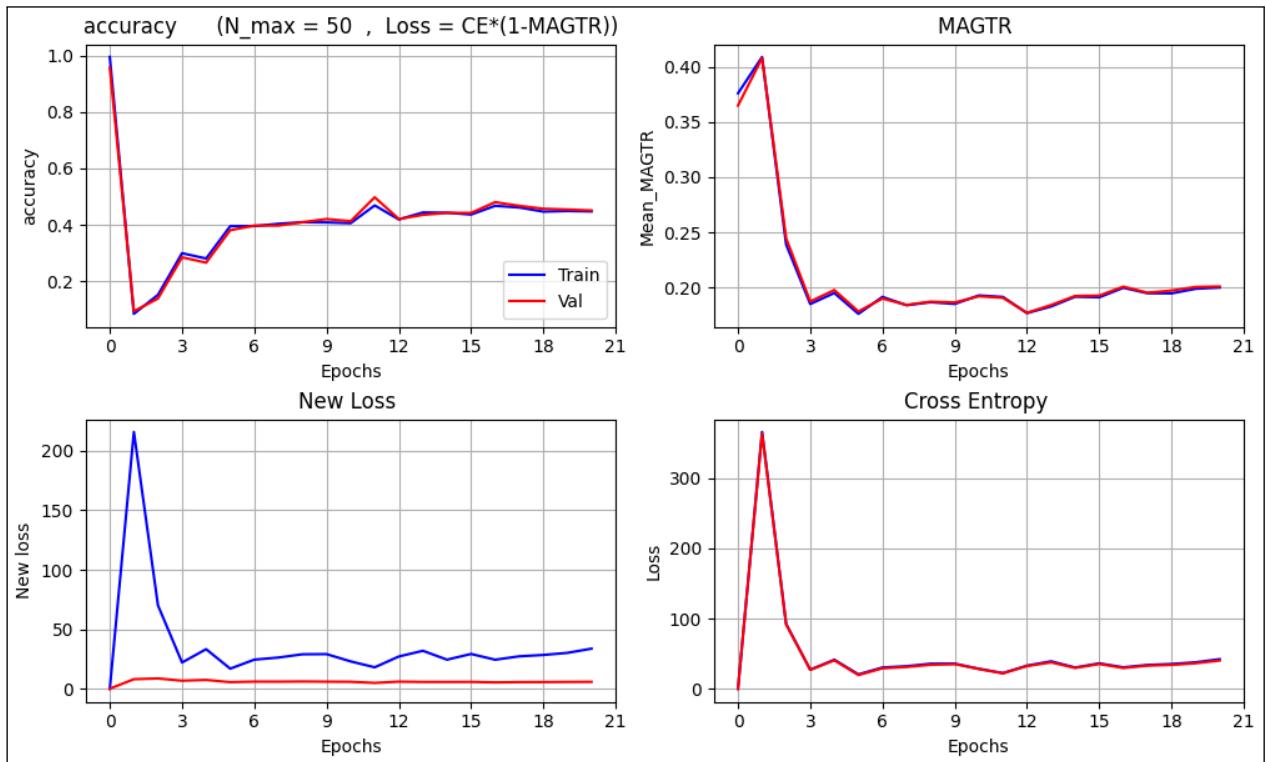


FIGURE 5.2 – Courbes d'apprentissage avec la fonction de perte "produit" 5.1

Pour toutes les expériences on analyse les 4 courbes ci-dessus, la précision, le MAGTR qu'on maximise, l'entropie croisée et la fonction perte, pour savoir la qualité de la précision, la qualité de l'activation et comment le model apprend au cours des époques.

On observe sur la courbe du MAGTR une diminution, tout comme une dégradation significative de la précision. L'analyse des courbes issues de cette expérience révèle une particularité intrigante : les valeurs de perte constatées sur les données de validation se révèlent en effet inférieures à celles enregistrées lors de l'entraînement, comme présenté dans la figure 5.2. Ces constatations indiquent la nécessité d'envisager un changement d'approche.

5.2.2 Essai 2 : Division

Après l'essai 1 nous avons changé le produit par la division. Par conséquent, la différence réside dans la formule de la fonction de perte, où nous avons exploré la diviser l'entropie croisée par le MAGTR. On a suivi les mêmes étapes de l'expérience précédente, pour ajuster les hyperparamètres.

$$\text{loss} = \text{Cross_Entrop} / \text{MAGTR}$$

FIGURE 5.3 – Fonction de perte composé de CE divisée par MAGTR

Voici le résultat le plus optimal que nous avons pu obtenir pour la division :

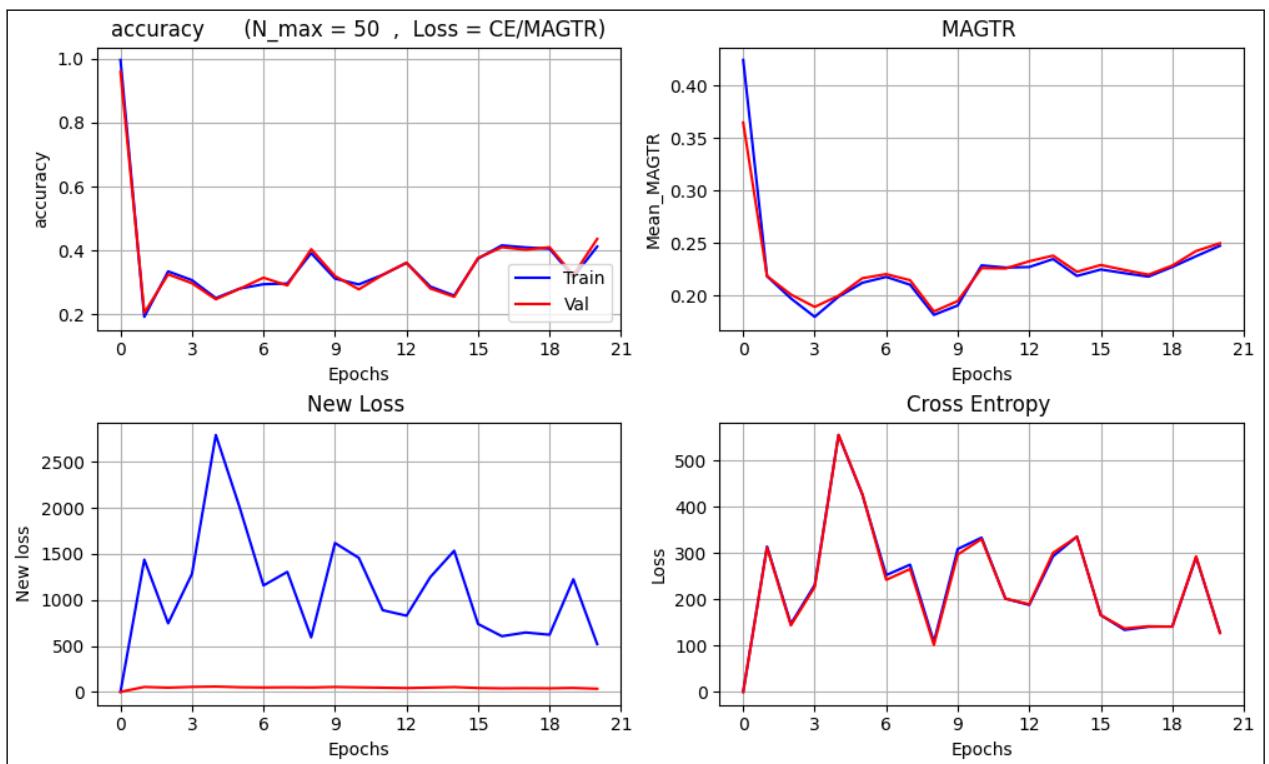


FIGURE 5.4 – Courbes d'apprentissage avec la fonction de perte "division" 5.3

Les courbes d'apprentissage de cette expérience mettent en évidence une mauvaise précision et un MAGTR faible. Les résultats demeurent globalement insatisfaisants. Cette convergence souligne la persistance des problèmes identifiés précédemment, nécessitant une réévaluation attentive de nos approches et paramètres expérimentaux.

5.2.3 Essai 3 : Somme pondérée

Après tous ces tests, nous avons découvert plusieurs travaux sur internet qui utilisaient la somme pondérée pour un entraînement multitâche, lorsqu'un modèle doit apprendre deux tâches ou plus simultanément, dont un article [4]. Nous avons suivi

cette approche en effectuant la somme pondérée des deux termes que nous souhaitons optimiser, à savoir l'entropie croisée et le MAGTR.

$$\text{loss_value} = k1 * \text{cross_Entrop} + k2 * (1 - \text{MAGTR})$$

FIGURE 5.5 – Fonction de perte de la somme pondérée de CE et MAGTR

Lors de la conduite de cette expérience, nous avons suivi les mêmes étapes méthodologiques que celles précédemment entreprises pour ajuster les hyperparamètres du modèle. Cependant, pour surmonter des instabilités et converger vers un optimum, il a fallu accorder une attention particulière à deux hyperparamètres essentiels : la taille des lots (batch size) et le taux d'apprentissage (learning rate).

Étant limités par la capacité de mémoire RAM, nous avons initialement restreint le nombre d'images par lot, ce qui a eu un impact négatif sur la convergence du modèle. Pour résoudre ce problème, nous avons adopté une stratégie de gradient accumulatif, accumulant les gradients sur plusieurs lots avant la mise à jour des poids du modèle. De plus, nous avons réajusté le taux d'apprentissage pour compenser les changements introduits par la modification de la taille des lots. Ces ajustements conjoints ont permis de rétablir la convergence de l'apprentissage et d'améliorer les performances du modèle,

Deux nouveaux hyperparamètres ont été pris en considération exclusivement pour cette expérimentation, à savoir les coefficients ($k1$ et $k2$) de la somme pondérée^{5.5}. Les résultats les plus optimaux que nous avons obtenus pour ces expériences sont les suivants :

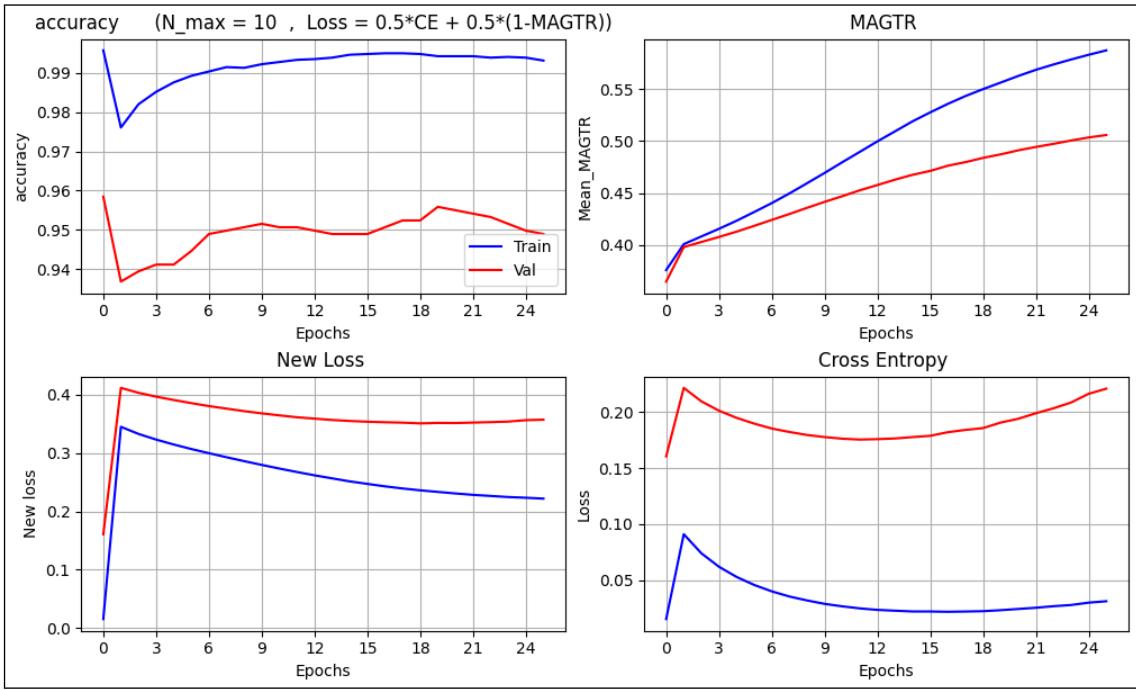


FIGURE 5.6 – Courbes d'apprentissage avec la fonction de perte "somme-pondérée"[5.5](#)

Les courbes de la figure 5.6 illustrent une légère dégradation de la précision, ainsi qu'une nette amélioration du MAGTR. Toutefois, afin de déterminer si cette augmentation du MAGTR après l'entraînement correspond à une activation significative ou non, il est nécessaire de générer les cartes d'activation pour les images correctement classées avant l'entraînement (époque 0) et après l'entraînement (en sélectionnant la meilleure époque, soit l'époque 18 selon les courbes d'apprentissage [5.6](#)).

Ainsi, nous recherchons ces images dans des fichiers CSV générés après l'entraînement, contenant les valeurs du MAGTR. Nous ciblons spécifiquement les images qui ont des valeurs faibles du MAGTR, et ayant été correctement classées par les modèles après les deux époques.

Ci-dessous, des exemples de ces images accompagnées de leurs activations correspondantes :

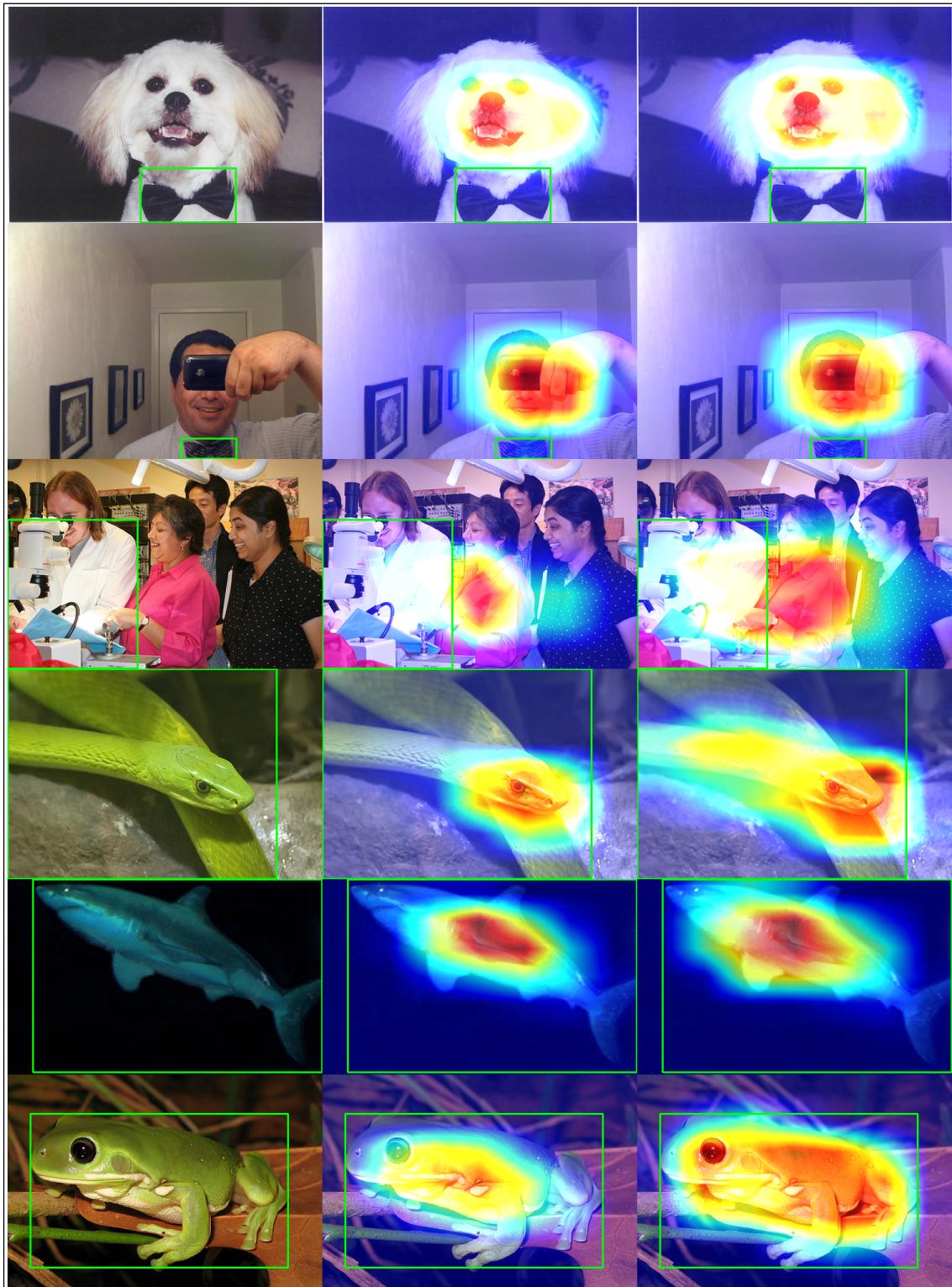


FIGURE 5.7 – Cartes d'activation par rapport à la classe de vérité terrain (1ère image : Noeud papillon, 2ème image : Noeud papillon, 3ème image : blouse de laboratoire, 4ème image : MAmba vert, 5ème image : Grand requin blanc, 6ème image : Rainette) pour les images correctement classées par les modèles après les époques 0 et 18, tout en présentant un MAGTR faible. À gauche, les images originales; au milieu, les CAMs de l'époque 0; à droite, les CAMs de l'époque 18.

Les observations que nous pouvons tirer de ces CAMs indiquent que le MAGTR modifie peu la localisation des activations, en particulier les mauvaises localisations, mais il étend leur étendue. Cependant, la précision n'a pas subi une dégradation remarquable, on peut mieux voir cela sur les matrices de confusion de l'époque 0 et 18 des données de test.

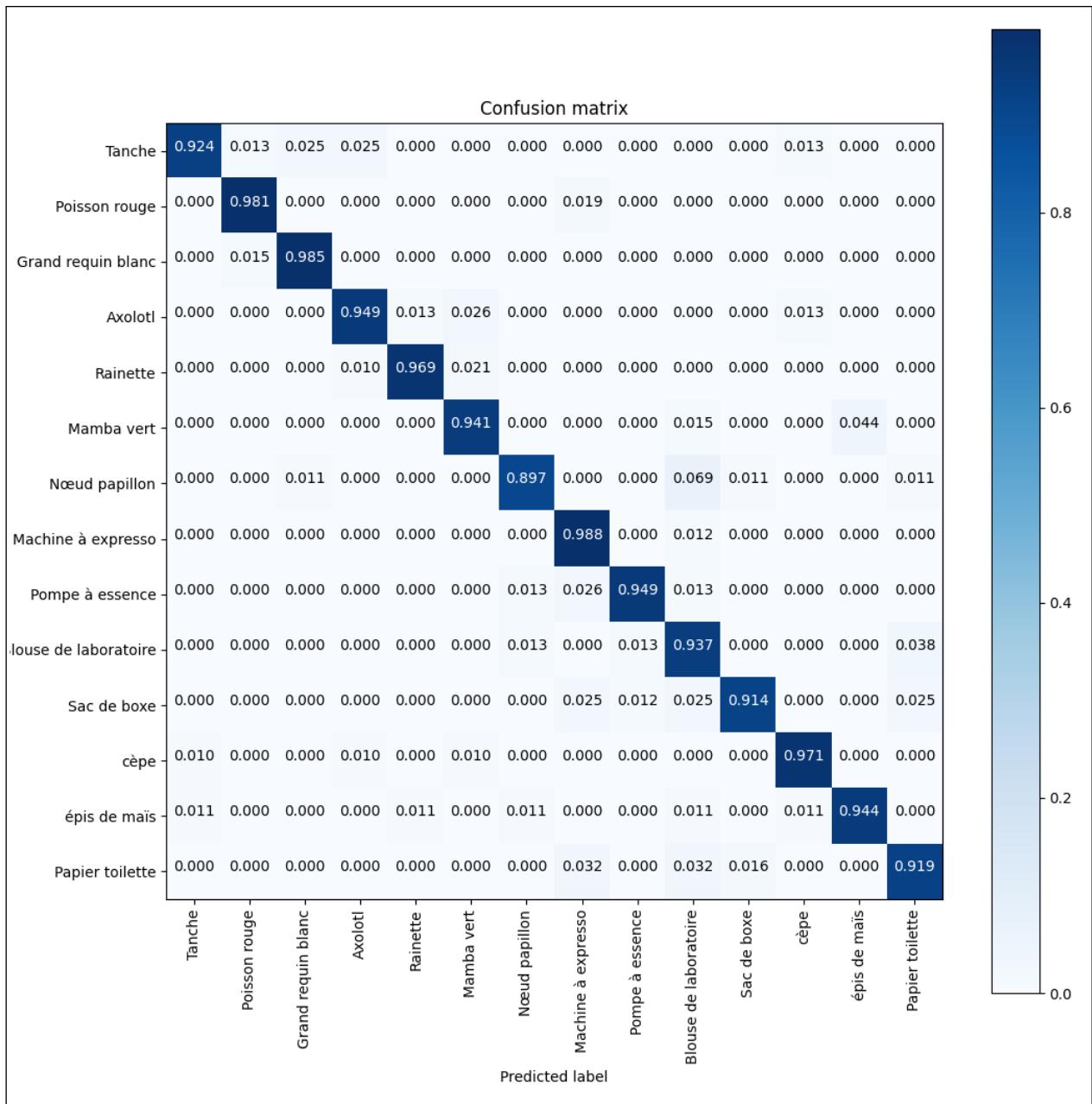


FIGURE 5.8 – Matrice de confusion du modèle Resnet50 [3] entraîné sur l'entropie croisée pour 14 classes d'ImageNet (époque 0 de l'essai : somme pondérée 5.6).

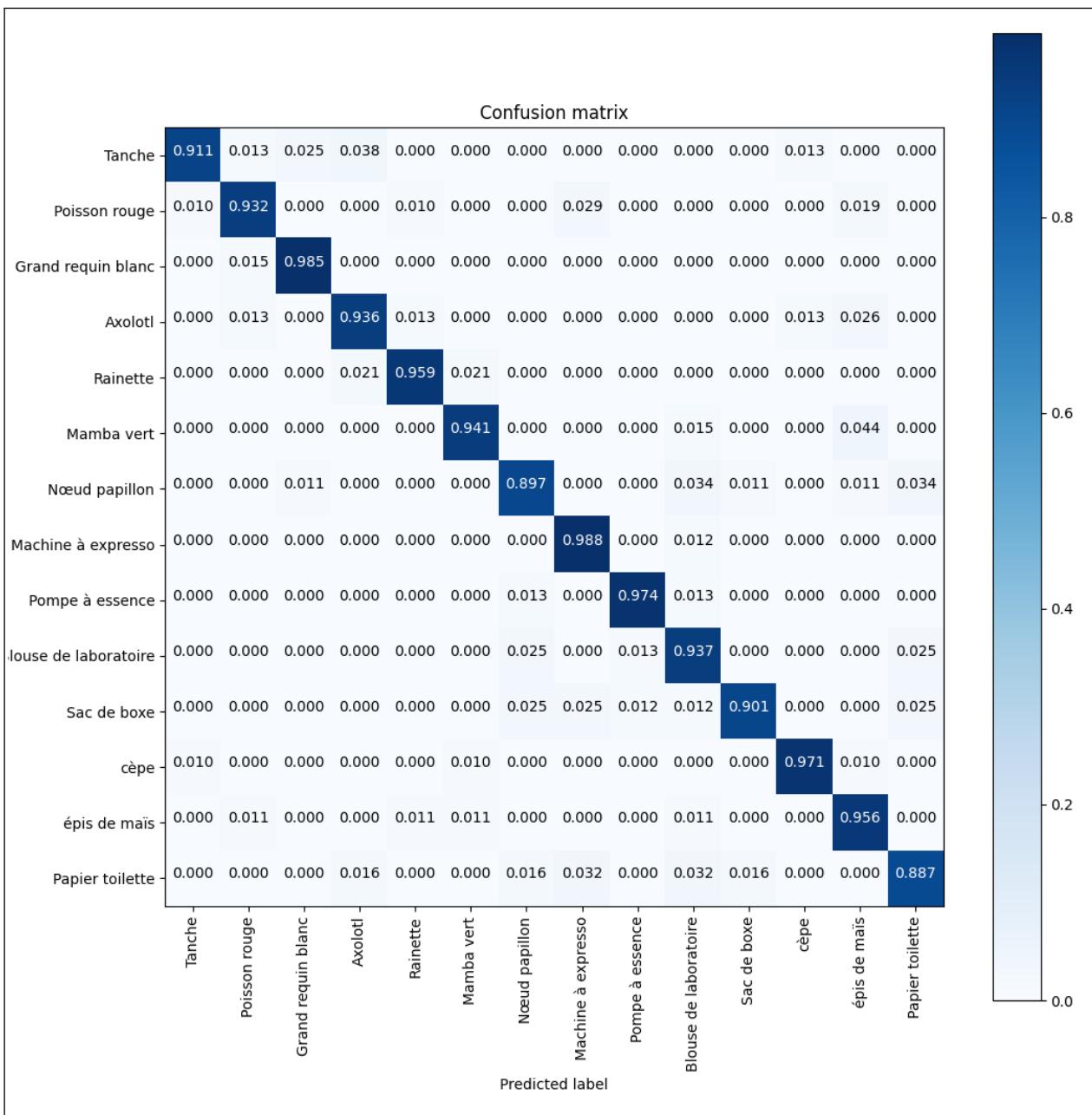


FIGURE 5.9 – Matrice de confusion du modèle Resnet50 [3] entraîné sur la somme pondérée de l'entropie croisée et le MAGTR pour 14 classes d'ImageNet (époque 14 de l'essai : somme pondérée 5.6).

Les conclusions que nous pouvons tirer de cet essai (somme pondérée) est la précision est peu affectée par le compromis (optimisation multi-tâches : la CE et le MAGTR). Toutefois, cela ne répond toujours pas à notre objectif d'obtenir des localisations précises des activations dans le cas de prédictions correctes.

Chapitre 6

Difficultés Rencontrées

6.1 Code existant peu lisible et non optimisé

Au début de mon stage, j'ai été confronté à un code implémenté par de précédents stagiaires Adam Aalah[1] et Wyctor Fogos da Rocha [10] qui présentait des problèmes de lisibilité et de structuration. Cela rendait ma compréhension du code difficile et m'a causé des difficultés pour effectuer des modifications et des améliorations. En particulier, j'ai simplifié le code de W. Fogos da Rochas qui effectuait un calcul explicite des gradients en utilisant partout des tenseurs gérés par le mécanisme de différentiation automatique de Tensorflow. De plus, j'ai dû faire face à des incompatibilités entre les versions des bibliothèques et frameworks utilisés, comme TENSORFLOW[7] et CUDA[8], ce qui compliquait la tâche de faire fonctionner le code correctement.

6.2 Bugs et limitations de mémoire

Après avoir acquis une meilleure compréhension de la problématique et des approches à suivre, j'ai dû résoudre des bugs présents dans le code existant. Ces erreurs ont nécessité des efforts supplémentaires pour les identifier et les corriger. Par ailleurs, en raison de limitations de mémoire, j'ai dû optimiser le code pour résoudre les problèmes de limitation et garantir le bon fonctionnement des calculs.

6.3 Migration vers GOOGLE COLAB et limitations associées

Ensuite on a essayé de changer la plateforme en utilisant GOOGLE COLAB[2] au lieu de Jupyter[9] dans la machine locale. L'exécution était plus rapide par contre il y avait d'autres limitations l'espace mémoire de stockage et la session d'exécution s'arrête après 12 heures, de plus les ressources ne sont pas toujours disponibles surtout le GPU (qui est nécessaire pour nos calculs).

6.4 Transition vers VS CODE avec des scripts PYTHON

Après avoir surmonté ces limitations j'ai continué le travail sur Jupyter pour quelque semaines, avant de passer à VS CODE[14] en utilisant des scripts PYTHON plutôt qu'un NOTEBOOK Jupyter pour plusieurs avantages :

- Les scripts Python peuvent être intégrés dans des applications ou des projets plus vastes, ce qui permet une utilisation plus souple et une meilleure maintenance du code.
- Les scripts Python peuvent être facilement automatisés pour s'exécuter à intervalles réguliers ou en réponse à des événements spécifiques. De plus, en organisant le code en fonctions et modules, vous pouvez le réutiliser plus facilement dans d'autres projets.
- Dans certaines situations, exécuter un script Python peut être plus rapide qu'un notebook Jupyter, car les cellules du notebook ont une surcharge supplémentaire en raison de leur nature interactive.
- Lorsque vous exécutez un script Python, les variables sont encapsulées dans l'espace de noms du script. Cela réduit le risque de pollution des variables globales, ce qui peut être plus problématique dans un notebook Jupyter où les cellules partagent un espace de noms global.

6.5 Migration vers la plateforme de calcul départementale Vulkan

J'ai rencontré des difficultés lors de mes tentatives pour exécuter les scripts sur ma machine locale. Les problèmes de compatibilité entre TENSORFLOW et CUDA sous Windows se sont révélés complexes à résoudre. Cependant, une solution alternative s'est présentée sous la forme de la plateforme de calcul départementale, Vulkan. Avec l'aide de collègues expérimentés dans la gestion de cette plateforme, nous avons mis en place un environnement virtuel minutieusement configuré. Cet environnement incluait toutes les bibliothèques nécessaires pour exécuter sans encombre mes scripts.

La réussite de cette solution a été déterminante. Les scripts ont pu être exécutés de manière fluide et performante au sein de cet environnement Vulkan. Devant cette efficacité, j'ai poursuivi mon travail exclusivement sur la plateforme de calcul de l'école, sans résoudre les problèmes de ma machine locale jusqu'à ce que j'avance davantage dans mes travaux. Cette décision m'a permis de continuer à progresser efficacement dans mes projets sans être entravé par les difficultés de compatibilité que j'avais rencontrées initialement sur ma machine locale.

Le nœud de calcul attribué pour mes travaux était spécifiquement configuré comme suit :

- Mémoire RAM : 26 Go
- Système d'Exploitation : Linux

- Carte Graphique : NVIDIA GeForce RTX 1080Ti (Capacité de GPU : 11 Go)

Cette configuration a été essentielle pour maximiser les performances de calcul et de traitement graphique. En conclusion, ces difficultés rencontrées lors de mon stage ont été essentielles pour mon apprentissage, m'a aidant à mieux comprendre les enjeux du projet et à développer mes compétences professionnelles tout au long du processus d'adaptation.

Chapitre 7

Conclusion

En conclusion, cette mission au sein du département ARTEMIS de l'école Telecom SudParis a été une opportunité enrichissante d'explorer le domaine de la vision par ordinateur et d'étudier les cartes de saillance pour améliorer la fiabilité des décisions de classification obtenues par les réseaux convolutifs. Malgré des obstacles tels que le code peu lisible, les limitations de mémoire et les ajustements aux environnements GOOGLE COLAB [2], VS CODE [14] et la plateforme de calcul départementale Vulkan, nos efforts ont abouti à l'élaboration d'une approche prometteuse basée sur l'intégration des cartes de saillance dans le processus d'apprentissage des réseaux convolutifs.

Les résultats des essais menés ont été significatifs, en particulier dans l'expérimentation relative à la somme pondérée, qui a renforcé notre confiance en l'efficacité de notre méthode pour améliorer les performances de classification. Cependant, l'essai de la somme pondérée demeure encore insuffisant, soulignant la nécessité de rechercher des améliorations, comme suggéré dans les perspectives. À cet égard, notre travail futur consistera à explorer davantage cette voie dans le cadre de cette mission.

En somme, cette mission a été une expérience marquante qui témoigne de notre engagement envers le domaine de la vision par ordinateur, mettant en avant notre désir de contribuer à l'amélioration constante des méthodes et solutions.

Chapitre 8

Perspectives

Les travaux à effectuer pour le reste du stage, sur une période de six semaines, comprennent :

- Intégration du terme MANGTR : Une avenue prometteuse serait l'intégration du terme MANGTR (Mean Activation Not Ground Truth Region) à la fonction de perte de l'essai Somme pondérée. Cette démarche viserait à pénaliser les activations en dehors de la région de vérité terrain, potentiellement améliorant ainsi la performance globale de la méthode en réduisant les erreurs de classification.
- Tests avec Fast ScoreCam [6] et GradCAM[12] : Il serait opportun de répéter les tests en utilisant les méthodes alternatives Fast ScoreCam et GradCAM. Cela permettrait une comparaison approfondie des performances des différentes approches de génération de cartes de saillance, en prenant en compte le coût computationnel élevé associé à ScoreCAM [15].
- Évaluation étendue : Étendre l'évaluation de la méthode choisie serait une étape cruciale. Actuellement, basée sur des essais portant sur 14 classes, une évaluation à plus grande échelle avec environ 200 classes de la base de données ImageNet fournirait des conclusions plus solides et une meilleure compréhension des performances.

Ces perspectives rapprochent notre travail des objectifs visés et reflètent notre dévouement à l'amélioration continue des méthodes de vision par ordinateur. En mettant l'accent sur l'optimisation du code, la gestion des contraintes matérielles et l'exploration de nouvelles avenues, nous visons à consolider notre contribution à ce domaine en constante évolution.

Bibliographie

- [1] Adam AALAH. « Saliency analysis and visual explanations for convolutional neural networks in object detection ». In : (2021). UVSQ, Master TRIED.
- [2] *Google Colaboratory*. Google. 2023. URL : <https://colab.research.google.com>.
- [3] Kaiming HE et al. « Identity Mappings in Deep Residual Networks ». In : *Computer Vision – ECCV 2016*. Sous la dir. de Bastian LEIBE et al. Cham : Springer International Publishing, 2016, p. 630-645. ISBN : 978-3-319-46493-0.
- [4] Alex KENDALL, Yarin GAL et Roberto CIPOLLA. « Multi-Task Learning Using Uncertainty to Weigh Losses for Scene Geometry and Semantics ». In : *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. Juin 2018.
- [5] *Keras Documentation*. 2023. URL : <https://keras.io/api/applications/#usage-examples-for-image-classification-models>.
- [6] Jing LI et al. « FIMF score-CAM : Fast score-CAM based on local multi-feature integration for visual interpretation of CNNS ». In : *IET Image Processing* 17.3 (2023), p. 761-772.
- [7] MARTÍN ABADI et al. *TensorFlow : Large-Scale Machine Learning on Heterogeneous Systems*. Software available from tensorflow.org. 2015. URL : <https://www.tensorflow.org/>.
- [8] *NVIDIA CUDA Programming Guide*. URL : <https://developer.nvidia.com/cuda-zone>. NVIDIA Corporation. 2023.
- [9] *Project Jupyter*. Jupyter Development Team. 2023. URL : <https://jupyter.org>.
- [10] Wyctor FOGOS DA ROCHA. « Analysis of activation and detection enhancing ». In : (2022). UVSQ, Master TRIED.
- [11] Olga RUSSAKOVSKY et al. « ImageNet Large Scale Visual Recognition Challenge ». In : *International Journal of Computer Vision (IJCV)* 115.3 (2015), p. 211-252. DOI : [10.1007/s11263-015-0816-y](https://doi.org/10.1007/s11263-015-0816-y).
- [12] Ramprasaath R. SELVARAJU et al. « Grad-CAM : Visual Explanations From Deep Networks via Gradient-Based Localization ». In : *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*. Oct. 2017.
- [13] Karen SIMONYAN et Andrew ZISSERMAN. *Very Deep Convolutional Networks for Large-Scale Image Recognition*. 2015. arXiv : [1409.1556 \[cs.CV\]](https://arxiv.org/abs/1409.1556).
- [14] *Visual Studio Code*. Microsoft. 2023. URL : <https://code.visualstudio.com>.
- [15] Haofan WANG et al. « Score-CAM : Score-Weighted Visual Explanations for Convolutional Neural Networks ». In : *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*. Juin 2020.
- [16] Bolei ZHOU et al. « Learning Deep Features for Discriminative Localization ». In : *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. Juin 2016.

Chapitre 9

Annexes

9.1 Travaux Additionnels Post-Soutenance

9.1.1 Intégration du MANGTR

Comme mentionné dans les perspectives, nous avons essayé d'intégrer un nouveau terme dans la fonction de perte, le MANGTR (activation moyenne hors de la région de vérité de terrain). Pour ce faire, nous avons suivi les mêmes étapes que lors des expériences précédentes pour ajuster les hyperparamètres et les poids de chacun des termes (CE, MAGTR et MANGTR) de la fonction de perte.

$$\text{loss} = k1 * \text{CE} + k2(1 - \text{MAGTR}) + k3 * \text{MANGTR}$$

FIGURE 9.1 – Fonction de perte de la somme pondérée de CE, MAGTR et MANGTR

Lors de ces expériences, nous avons remarqué que lorsque nous attribuons un poids élevé au terme MAGTR, le MANGTR ne s'améliore pas suffisamment, comme illustré dans l'exemple ci-dessous où nous avons attribué un poids élevé. Cependant, lorsque nous attribuons un faible poids au MAGTR, le MAGTR montre une dégradation, tandis que le MANGTR n'enregistre pas d'amélioration significative, comme le démontre l'exemple ci-dessous où nous avons attribué un poids faible.

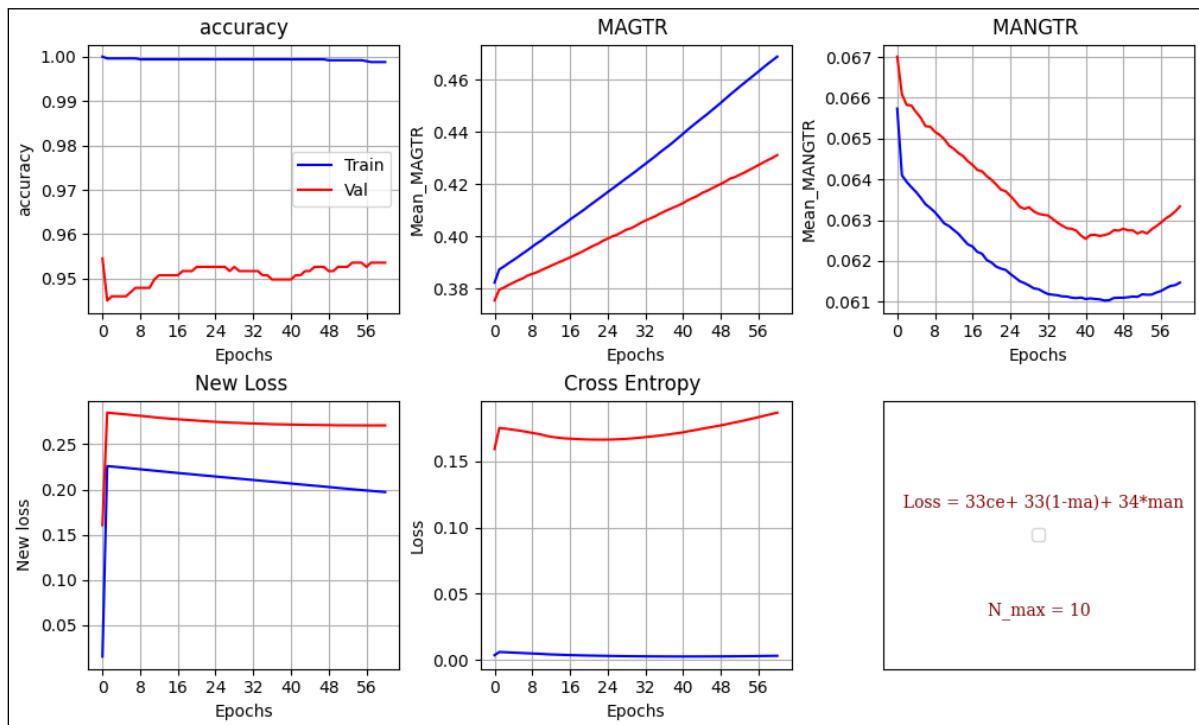


FIGURE 9.2 – Courbes d'apprentissage avec la fonction de perte "somme-pondérée (33,33,34)" 9.1

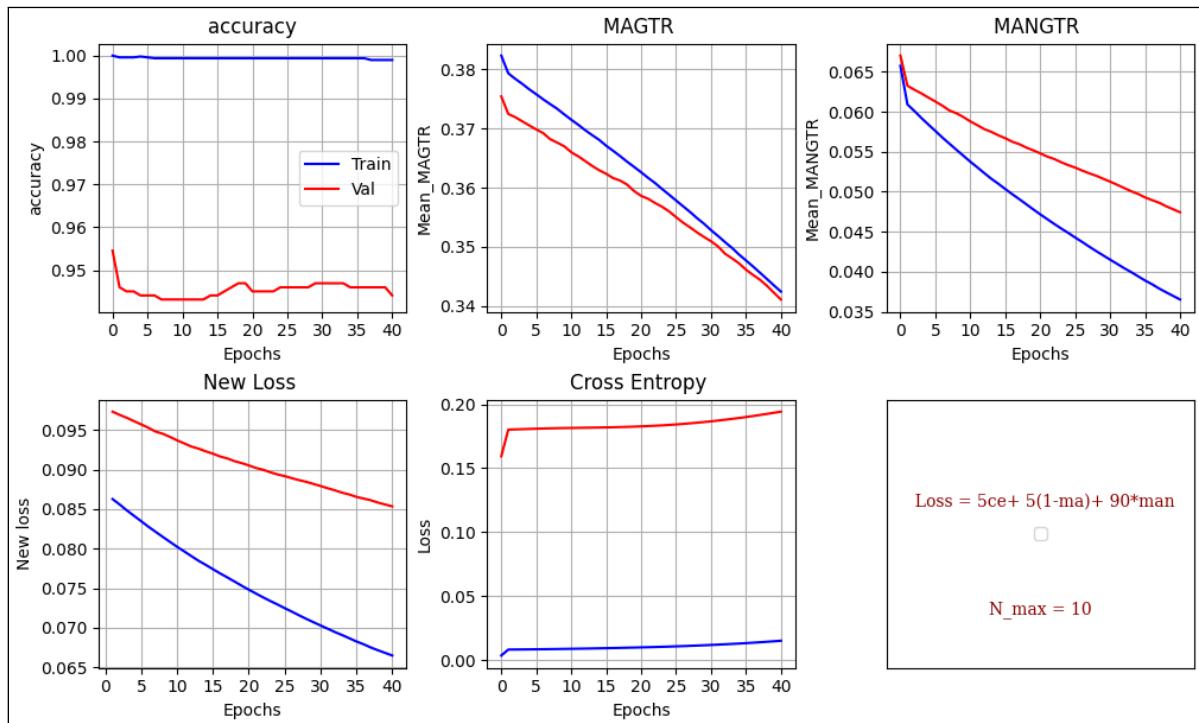


FIGURE 9.3 – Courbes d'apprentissage avec la fonction de perte "somme-pondérée (5,5,90)" 9.1

Donc, un poids faible sur le MAGTR avec un poids fort sur le MANGTR n'empêche pas la dégradation du MAGTR, en revanche, il dégrade bien le MANGTR. C'est pourquoi nous avons essayé d'attribuer un poids nul au MAGTR et un poids fort sur le MANGTR, comme le démontrent les courbes d'apprentissage ci-dessous :

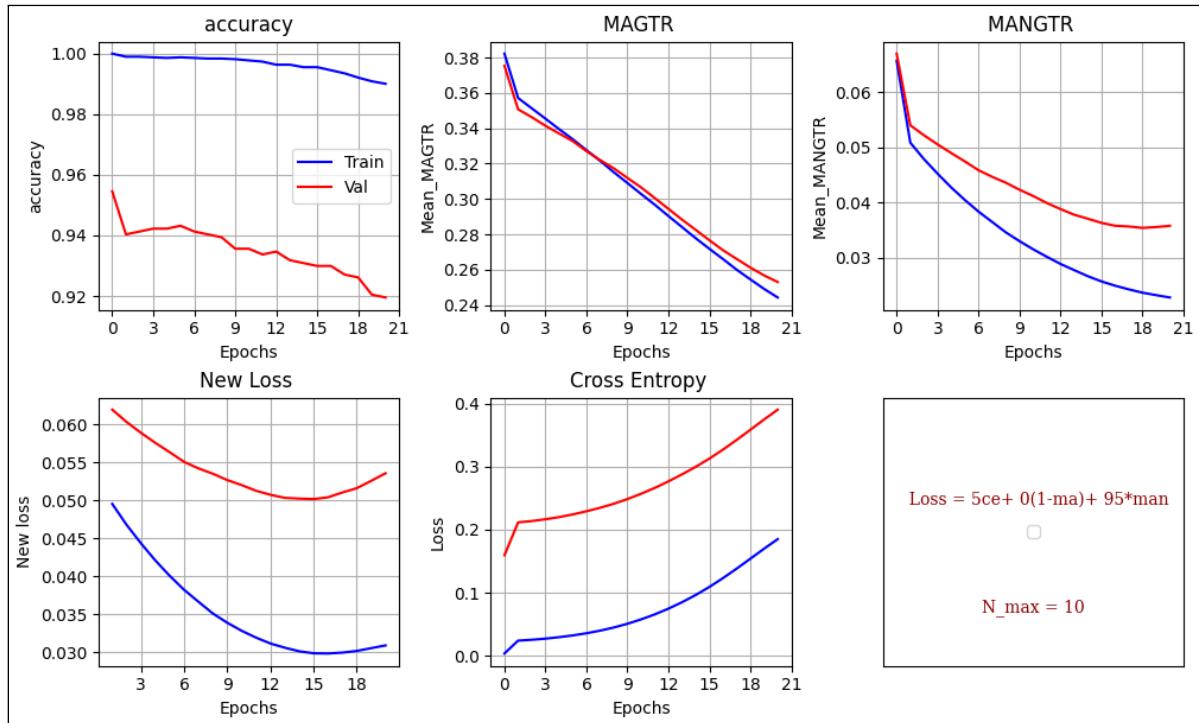


FIGURE 9.4 – Courbes d'apprentissage avec la fonction de perte "somme-pondérée (5,0,95)" 9.1

Comme on a fait avec les expériences précédentes, après l'analyse des courbes d'apprentissage on passe au analyse des cartes d'activation, sauf ici on a changé la méthode du choix des images. On a pris les images qui ont été correctement classées à l'époque 0 et sont devenues mal classées, ainsi que les images qui ont été mal classées et sont devenues correctement classées.

Ci-dessous, des exemples (de l'expérience de la figure 9.4) de ces images accompagnées de leurs activations correspondantes :

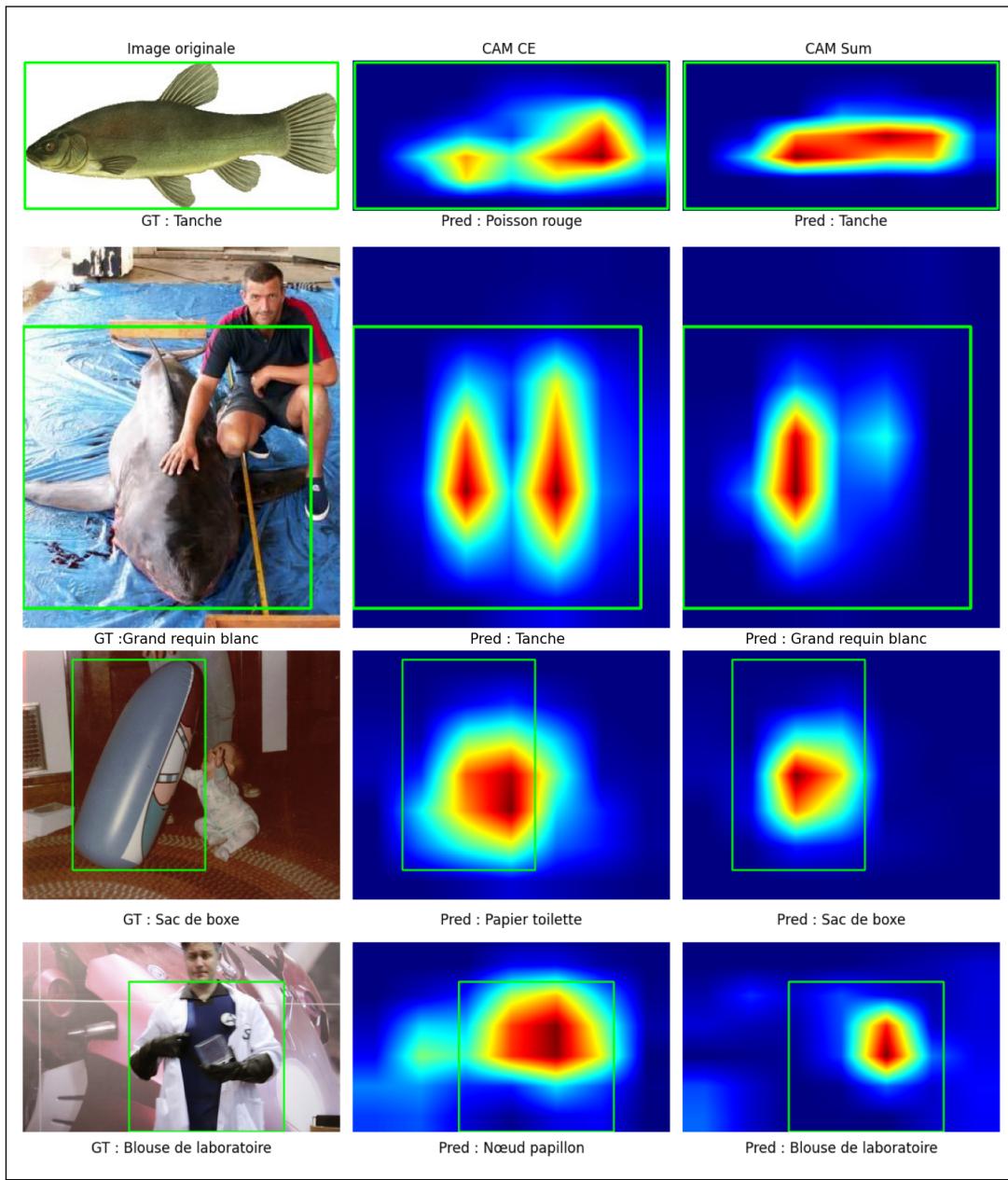


FIGURE 9.5 – Cartes d’activation par rapport à la classe de vérité terrain pour des images qui ont été mal classées et sont devenues correctement classées. À gauche, les images originales; au milieu, les CAMs de l’époque 0; à droite, les CAMs de l’époque 15 de l’expérience 9.4.

On observe des améliorations dans les cartes d’activations de ces images 9.5, qui avaient été initialement mal classées mais sont maintenant correctement classées.

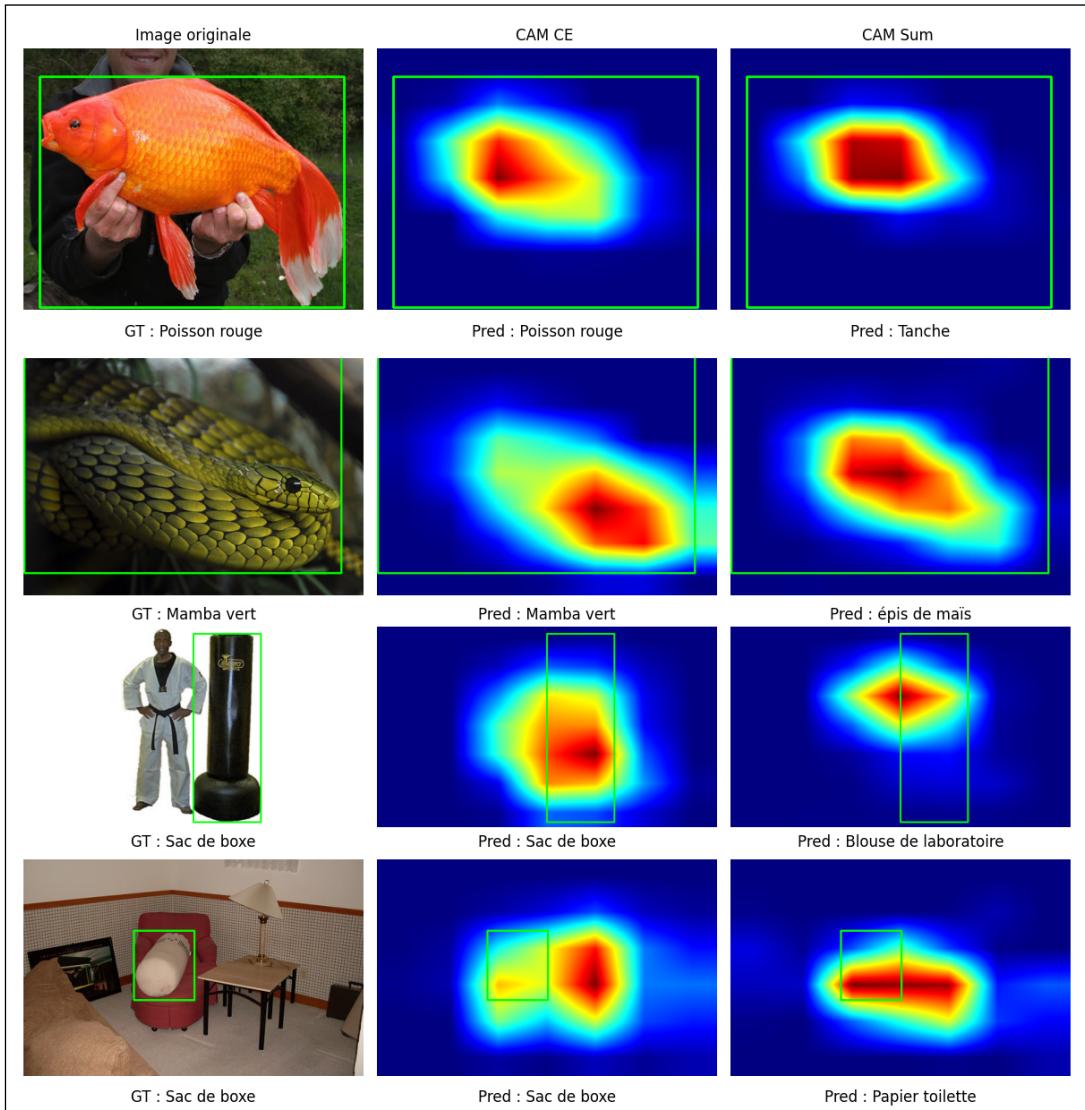


FIGURE 9.6 – Cartes d’activation par rapport à la classe de vérité terrain pour des images qui ont été correctement classées à l’époque 0 et sont devenues mal classées. À gauche, les images originales; au milieu, les CAMs de l’époque 0; à droite, les CAMs de l’époque 15 de l’expérience 9.4.

De même, nous constatons des améliorations dans les cartes d’activations des images 9.6, qui étaient à l’origine correctement classées mais sont désormais mal classées. Les prédictions pour ces images ne sont pas arbitraires, mais présentent des similitudes avec les images de la classe prédictive, notamment en ce qui concerne les textures, les couleurs et les formes des objets dans l’image.

Prenons par exemple la première image, qui représente un poisson rouge incorrectement prédict comme une tanche. Cette confusion s'explique par le fait que la plupart des images de la classe "tanche" montrent des personnes tenant une tanche, tout comme cette image montre une personne tenant un poisson rouge dans ses deux mains. De plus, la forme de ce poisson rouge (un peu plus grand que les autres poissons rouges dans les autres images de la classe poisson rouge) est similaire à celle d'une tanche.

Dans le cas de la deuxième image, un mamba vert a été incorrectement prédict comme un épi de maïs. La ressemblance réside dans la texture de la peau du mamba vert, qui rappelle celle des épis de maïs, ainsi que dans leur couleur. Des similitudes de ce type se retrouvent également dans les deux dernières images de la figure 9.6.

En conclusion, il est possible de constater que les erreurs de classification ne sont pas aléatoires, mais plutôt explicables, étant donné les similitudes observées. Cependant, il reste à prouver ces observations en utilisant une base de données propre et rigoureuse, telle que la base de données sur le paludisme, afin de valider nos conclusions.

Abstract

Résumé

Ce rapport de stage témoigne de notre engagement au sein du département ARTEMIS de l'école Telecom SudParis, où nous avons entrepris une mission passionnante visant à améliorer la fiabilité des classifications obtenues par les réseaux convolutifs grâce à l'exploration des cartes de saillance dans le domaine de la vision par ordinateur. Notre démarche a été axée sur l'intégration de l'activation des cartes de saillance dans le processus d'apprentissage, en utilisant des ensembles de données tels qu'ImageNet. Ce travail s'inscrit dans la continuité des recherches entreprises par les précédents stagiaires, Adam Allah et Wyctor Fogos da Rochas. Ce rapport détaille nos efforts, des étapes initiales de collecte des données à l'évaluation approfondie des performances de nos approches.

Mots-clés

Vision par ordinateur, Réseaux de neurones convolutifs, Classification d'images, Carte d'activation de classe, Explicabilité du modèle, ScoreCam.

Summary

This internship report reflects our commitment within the ARTEMIS department at Telecom SudParis, where we embarked on an exciting mission to enhance the reliability of classifications obtained through convolutional networks by exploring saliency maps in the field of computer vision. Our approach focused on integrating saliency map activations into the learning process, utilizing datasets such as ImageNet. This work builds upon the research pursued by previous interns, Adam Allah and Wyctor Fogos da Rochas. This report elaborates on our endeavors, from the initial data collection steps to the comprehensive evaluation of our approaches' performances.

Keywords

Computer Vision, Convolutional Neural Networks, Image Classification, Class Activation Mapping, Model Explainability, ScoreCam