Gradients Intégrés Renforcés

Caroline Mazini Rodrigues[*,**]
Nicolas Boutry[*], Laurent Najman[**]

[*]Laboratoire de Recherche d'EPITA (LRE)
cmr@lrde.epita.fr,
nicolas.boutry@lrde.epita.fr
[**]Laboratoire d'Informatique Gaspard-Monge (LIGM)
laurent.najman@esiee.fr

Résumé. Les visualisations fournies par les techniques d'Intelligence Artificielle Explicable (xAI) pour expliquer les réseaux de neurones convolutionnels (CNN's) sont parfois difficiles à interpréter. La richesse des motifs d'une image qui sont fournis en entrées (les pixels d'une image) entraîne des corrélations complexes entre les classes. Les techniques basées sur les gradients, telles que les gradients intégrés, mettent en évidence l'importance de ces caractéristiques. Cependant, lorsqu'on les visualise sous forme d'images, on peut se retrouver avec un bruit excessif et donc une difficulté à interpréter les explications fournies. Nous proposons la méthode intitulée Gradients Intégrés Renforcés (*RIG*), une variation des gradients intégrés, qui vise à mettre en évidence les régions influentes des images dans la décision des réseaux. Cette méthode vise à réduire la surface des zones à analyser lors de la visualisation des résultats, générant ainsi moins de bruit apparent. Des expériences à base d'occlusions démontrent que les régions choisies par notre méthode jouent effectivement un rôle important en terme de classification.

1 Introduction

Les méthodes d'xAI dites *post-hoc* et *modèle-agnostiques* peuvent être utiles pour expliquer un nombre considérable de modèles d'apprentissage automatique. Cette catégorie de méthodes comprend des subdivisions telles que les explications basées sur des exemples (Erhan et al. (2009), Bien et Tibshirani (2012), Kim et al. (2016)), les modèles de substitution (Ribeiro et al. (2016), Thiagarajan et al. (2016)), et les méthodes d'influence (Simonyan et al. (2014), Merrick (2019), Zeiler et Fergus (2014), Bach et al. (2015), Lundberg et Lee (2017)). Le plus intéressant dans cette dernière, c'est qu'elle entend, en quelque sorte, fournir des explications en utilisant le support de structures internes du modèle à expliquer. Si nous y réfléchissons, ces explications fournies seront, à différents niveaux, fidèles à la façon dont les modèles raisonnent en interne.

Le problème est que, même si nous fournissons des explications fiables à l'aide de ces méthodes, il est parfois difficile d'en interpréter le sens. Les explications visuelles reposant sur des cartes d'attribution ou de saillance en sont un exemple. Elles peuvent donner lieu à des

cartes bruitées, comme le montre la figure 1. Ces visualisations ont été obtenues en utilisant la technique des gradients intégrés (Sundararajan et al. (2017)) appliquée à un modèle entraîné (ResNet-18) afin d'expliquer localement un échantillon d'ensemble de données.

Dans ce travail, nous prévoyons de présenter des explications visuelles simplifiées se concentrant sur un seul concept : *la distanciation des classes*. L'idée est de mettre en évidence les caractéristiques qui sont importantes lorsque l'on veut que les prédictions d'appartenance d'une image à *sa* classe soit la plus différente possible de la probabilité d'appartenance aux *autres* classes (on augmente l'aspect discriminant du réseau). Ainsi, nous tentons de fournir les explications visuelles les plus explicites possibles (dans le sens dénuées de bruit) en proposant la méthode Gradients Intégrés Renforcés (*RIG*). Celle-ci utilise une méthode basée sur le gradient d'influence, les bien connus gradients intégrés, comme base de notre approche itérative. Cette approche s'appuie sur des régressions du support pour corroborer la décision de savoir quelles caractéristiques sont importantes : chaque modèle de support est un réseau neuronal différent formé pour fournir plus de distance entre les classes en se basant, non plus sur une tâche de classification (avec des valeurs discrètes de classe comme sorties), mais sur une tâche de régression (avec des valeurs continues comme sorties). Nous considérons comme importantes les caractéristiques qui contribuent à tous les modèles, originaux et de support, au cours du processus.

Les expériences sont présentées en deux parties : visualisations de l'attribution des cartes et analyse de la sortie du réseau pour les images excluant les régions d'intérêt. Avec cette méthode, nous avons l'intention de contribuer à une première analyse de la dépendance des classes à des concepts spécifiques et des relations interclasses. De plus, nous espérons être en mesure d'améliorer les visualisations de la relation entre les classes, en fournissant des cartes d'attribution moins bruitées. Dans la Section 3, nous présentons la méthode de base des gradients intégrés ; la Section 4 contient les trois étapes de la méthode *RIG* ; nous décrivons les résultats obtenus dans la Section 5 et nous concluons ce document en présentant les travaux futurs dans la Section 6.

2 Modèles d'influence

Diverses méthodes ont été proposées afin de fournir des explications. Elles peuvent être classées en méthodes *intrinsèques* et *spécifiques au modèle*, ou en méthodes *post-hoc* et (généralement) *agnostiques au modèle*. Des exemples de méthodes *intrinsèques* sont les arbres de décision Quinlan (1986), les réseaux d'attention Gu et al. (2021) et l'entraînement conjoint avec explication de texte Park et al. (2018). Pour ces méthodes, nous disposons d'explications directement issues du modèle d'apprentissage analysé. Certains exemples de méthodes *post-hoc* et *agnostic de modèle* sont l'analyse d'influence, l'analyse de sensibilité, les méthodes basées sur des exemples et les modèles de substitution Adadi et Berrada (2018). Pour ces méthodes, nous appliquons les techniques xAI à une méthode d'apprentissage déjà entraînée. Nous nous concentrerons sur les méthodes post-hoc et les méthodes agnostiques de modèle, car nous prévoyons d'analyser des architectures CNN plus générales.

À l'intérieur de cette classe de méthodes, nous pouvons trouver quelques subdivisions telles que les modèles d'influence, les modèles basés sur des exemples et les modèles de substitution. Une méthode d'influence tente d'expliquer le modèle appris en présentant les influences des entrées ou des composants internes sur la sortie (Adadi et Berrada (2018)). Parmi les mé-

thodes d'influence, citons l'analyse de sensibilité avec des cartes de saillance Simonyan et al. (2014) ou des techniques d'occlusion Merrick (2019), Zeiler et Fergus (2014) ; *Layer-wise Relevance Propagation* (LRP) Bach et al. (2015), et les méthodes d'importance des caractéristiques, sur lesquelles nous avons décidé de nous concentrer dans ce papier. Parmi les méthodes d'importance des caractéristiques les plus utilisées, on peut citer : Déconvolution Zeiler et Fergus (2014), Guided-Backpropagation Springenberg et al. (2015), DeepLIFT Shrikumar et al. (2017), Integrated Gradients (IG) Sundararajan et al. (2017), CAM Zhou et al. (2016), Grad-CAM, Guided-GradCAM Selvaraju et al. (2017), GradCAM++ Chattopadhyay et al. (2018), les valeurs de Shapley Lundberg et Lee (2017).

3 Gradients Intégrés pour l'analyse de la séparation des classes

Dans cet article, nous nous sommes concentrés sur un type spécifique de méthode d'explication basée sur l'influence *post-hoc*, celle basée sur le gradient. La méthode basée sur les gradients utilisée est celle des gradients intégrés (IG) proposée par Sundararajan *et al.* Sundararajan et al. (2017) qui présente l'équation 1. L'idée est d'intégrer les gradients obtenus par un chemin de variation (selon β) depuis un échantillon de base $\mathcal{X}'$ jusqu'à l'échantillon cible $\mathcal{X}$ par rapport à la fonction F apprise par le réseau. Dans ce contexte, nous considérons une image comme un échantillon ($\mathcal{X}$). Les gradients sont obtenus pour chaque caractéristique x_i de l'échantillon $\mathcal{X}$. Le résultat de l'intégration est multiplié par la différence entre la valeur de la caractéristique x_i et la caractéristique correspondante x'_i de l'échantillon de base.

$$IG_i(\mathcal{X}) ::= (x_i - {x'}_i) \times \int_{\beta=0}^{1} \frac{\delta F(\mathcal{X}' + \beta(\mathcal{X} - \mathcal{X}'))}{\delta x_i} d\beta \tag{1}$$

Généralement, lorsqu'on travaille avec des images, la ligne de base est une image noire ou le fond des images. Le choix de cette méthode comme notre représentant basé sur le gradient était dû à la réduction du bruit dans les cartes d'attribution finales (l'importance des pixels finaux) en raison de l'ajout d'un chemin d'intégration dans le processus.

Les résultats des gradients intégrés dépendent de la classe analysée, ils sont obtenus par pixel et ils sont appelés *attributions*. Nous pouvons visualiser l'ensemble des attributions sous forme d'image, en mettant en évidence l'importance des pixels pour la décision de classe, comme présenté dans la Figure 1 en exprimant visuellement les pixels les plus importants pour la classe *chat* et *chien*.

Dans ce travail, au lieu d'utiliser les attributions originales des gradients intégrés pour un modèle spécifique entraîné, nous proposerons une adaptation appelée *Gradients Intégrés Renforcés* (*RIG*). Avec cette méthode, nous avons l'intention de mettre en évidence les régions de l'image qui sont plus influentes pour séparer deux classes.

4 Mise en évidence des gradients dépendant de la classe

Pour trouver les régions les plus dépendantes de la classe, nous avons travaillé sur les probabilités de classe à partir des échantillons d'entraînement. Considérons un problème de clas-

FIG. 1 – *Cartes d'attribution obtenues par la méthode des gradients intégrés. L'importance des pixels est décrite du blanc au noir (moins important à plus important) selon une classe choisie : chat ou chien. Les personnes ont été anonymisées.*

sification avec m classes $\mathcal{C} = \{c_d\}_{d=1}^{m}$, un jeu de données $\mathcal{D} = \bigcup_{d=1}^{m} \mathcal{D}_{c_d}$, avec n_d échantillons $\mathcal{D}_{c_d} = \{\mathbf{I}_{c_d,i}\}_{i=1}^{n_d}$ de la classe c_d. Pour mettre en évidence les différences entre les classes, nous voulons les éloigner *artificiellement* sans perdre la relation trouvée par le réseau entre les échantillons à la fin de l'entrainement. En éloignant les échantillons de classe et en gardant la même structure apprise, nous tentons de faire ressortir les caractéristiques les plus importantes pour la classification. Le processus de distanciation artificielle des classes est composé de trois étapes.

4.1 Choisir deux classes d'intérêt et augmenter les distances

L'idée ici est de choisir deux classes d'intérêt parmi les $\mathcal{C}$ classes entraînées, afin d'analyser les caractéristiques responsables de leur distance. Prenons l'exemple des classes *chat* et *chien*. Nous voulons augmenter les distances en réduisant la probabilité qu'un chien soit un chat et vice-versa, sans changer la structure apprise ou la relation de ces classes avec les autres classes.

Par conséquent, nous avons directement modifié les probabilités non normalisées, d'être un chien (avant l'activation du softmax), sur les échantillons $\mathcal{D}_{cat}$ (images de chats) en soustrayant une valeur α_{cat}. De plus, nous soustrayons une valeur α_{dog} des probabilités d'être un chat des échantillons $\mathcal{D}_{dog}$ (images de chiens), comme l'illustre la figure 2. De cette manière, on obtient des chats moins considérés comme des chiens, et vice-versa.

4.2 Entraînement des régressions

Cette distanciation, comme nous l'avons mentionné, est artificielle. Pour amener les réseaux à présenter ces nouvelles probabilités chat/chien, nous devons entraîner de nouveaux modèles. Cette fois, il s'agit d'un problème de régression. Nos entrées x restent les images de $\mathcal{D}$, nous utilisons les poids du réseau de classification analysé (pour l'initialisation), cependant, les sorties attendues y sont maintenant les probabilités modifiées. L'idée ici n'est pas d'obtenir les modèles les plus généralisés. Nous voulons obtenir les meilleures approximations de ces nouvelles probabilités, afin de comprendre quelles caractéristiques restent importantes. Nous avons remplacé l'activation softmax (utilisée dans la dernière couche pour une classification) par une activation linéaire (utilisée pour une régression) et nous avons utilisé l'erreur

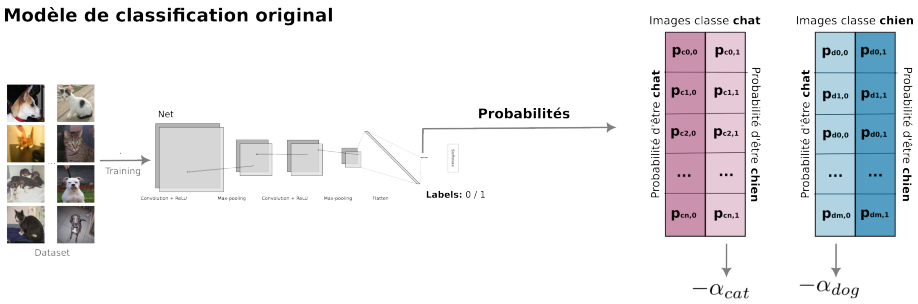


FIG. 2 – *En se basant sur les probabilités non normalisées (c'est-à-dire celles avant le softmax) pour les deux classes, nous éloignons les échantillons des différentes classes. Nous définissons les valeurs α_{cat} et α_{dog} qui seront soustraites des probabilités. La valeur α_{cat} réduira la probabilité que les images de chats soient des chiens (colonne rose clair) et α_{dog} réduira la probabilité que les images de chiens soient des chats (colonne bleu clair). Ces nouvelles probabilités sont artificielles, mais elles préservent les relations entre les échantillons d'une même classe.*

quadratique moyenne (EQM) pour entraîner la régression. La figure 3 illustre ce processus de modification du processus d'apprentissage du modèle avec les étiquettes converties en un vecteur de probabilités.

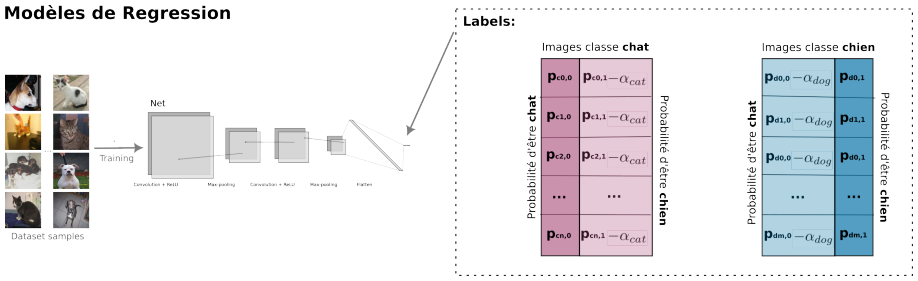


FIG. 3 – *Au lieu d'analyser un problème de classification, nous avons utilisé les probabilités modifiées pour entraîner un problème de régression. Le nouveau modèle est entraîné à prédire les probabilités précédemment obtenues (de la classification) en soustrayant les valeurs α de la probabilité de la classe opposée pour chaque échantillon.*

4.3 Choisir les caractéristiques importantes

Pour être plus précis dans notre choix de caractéristiques importantes, nous proposons l'exécution des étapes 1 et 2 (Sections 4.1 et 4.2) plusieurs fois, en augmentant la valeur de α_{cat} et α_{dog}. Il en résultera de multiples réseaux de régression. Après les avoir entraînés, nous appliquons les gradients intégrés à chacun d'eux lors de l'analyse des images. Les caractéristiques importantes doivent représenter le modèle original (celui qui doit être expliqué). Cependant, nous nous attendons à ce que les caractéristiques qui sont importantes pour tous

les réseaux, soient les plus cohérentes à considérer lors de la distanciation des deux classes. La figure 4 présente un exemple de l'utilisation de quatre réseaux de régression de support pour choisir les régions importantes. L'image dans laquelle les gradients intégrés ont été appliqués a été incorrectement classée comme chien par le réseau analysé.

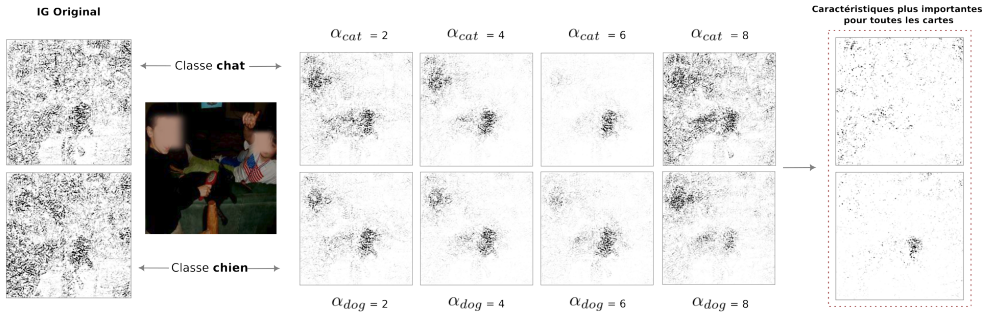


FIG. 4 – *Exemple de la façon dont nous proposons de choisir les caractéristiques importantes. Après avoir formé des réseaux de régression, nous appliquons la méthode des gradients intégrés pour chaque modèle résultant et pour les deux classes. La carte d'attribution finale comprend uniquement les caractéristiques présentes dans les cinq cartes d'attribution, la carte initiale (modèle de classification analysé) et les quatre modèles de régression de support. Nous obtenons des cartes d'attribution filtrées, incluant les régions de l'image qui séparent le plus les deux classes.*

Comme présenté dans cette section, la méthodologie est axée sur les réseaux neuronaux convolutifs. Cependant, l'idée de la distance entre classes à l'aide de réseaux de régression de support est également applicable à d'autres types de réseaux et de données pour un problème de classification. Il suffit de disposer d'une méthode basée sur l'influence adaptée au modèle analysé et aux données correspondantes. Par exemple, les réseaux de neurones entièrement connectés pourraient également être analysés par RIG avec les importances attribuées aux vecteurs de caractéristiques.

5 Résultats

Nous présenterons quelques résultats de la visualisation de deux réseaux formés dans le même ensemble de données. Ces réseaux sont : ResNet-18 proposé par He et al. (2016) et VGG16 proposé par Simonyan et Zisserman (2015). Le jeu de données est une classification chat/chien [1]. Nous avons utilisé les valeurs $\alpha = \{0, 2, 4, 6, 8\}$ pour les deux classes. Les réseaux de régression de support ont initié le processus d'entraînement avec les poids du modèle de classification analysé. Nous avons entraîné chaque modèle de support pendant 10 époques avec l'optimiseur Adam et un taux d'apprentissage égal à 4×10^{-5}.

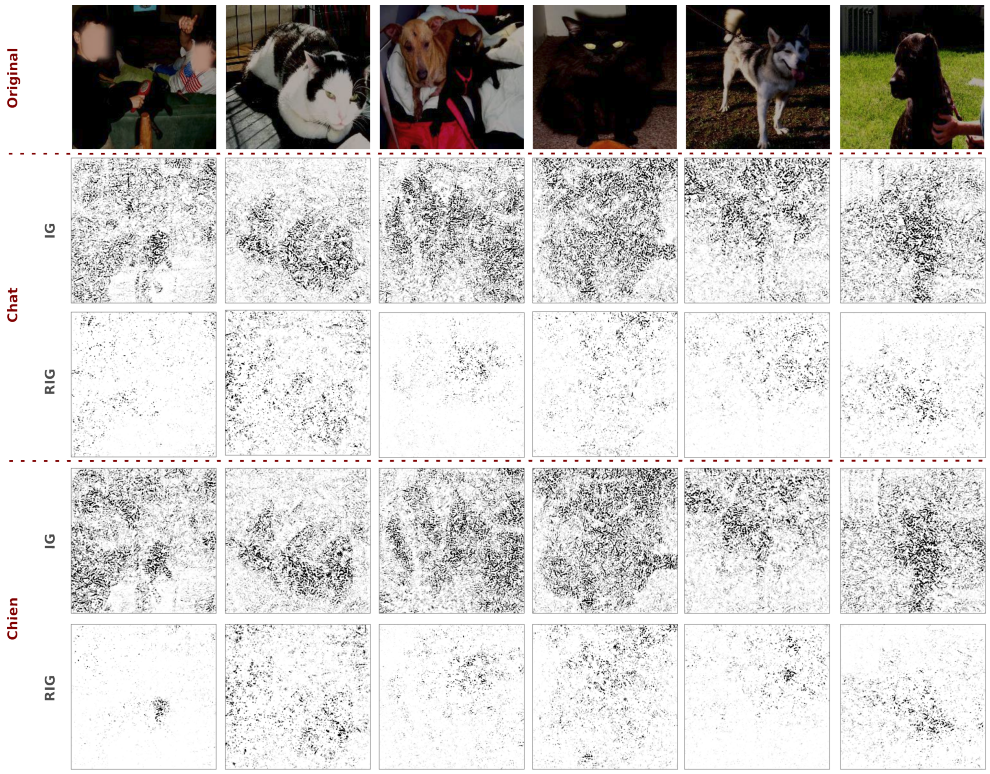


FIG. 5 – *Cartes d'attribution obtenues à partir de ResNet-18 pour six images du jeu de données. Nous présentons dans la première ligne les images originales, suivies, dans les deuxième et troisième lignes, de la carte d'attribution des gradients intégrés et de RIG respectivement, pour la classe chat. De plus, dans les troisième et quatrième rangées, la carte d'attribution des gradients intégrés et de RIG respectivement, pour la classe chien. RIG réduit les régions d'intérêt spécialement dans les images comme celles de la première et de la cinquième colonne.*

5.1 Cartes d'attribution

Après avoir formé les quatre réseaux de régression de support, nous présentons dans cette section les résultats des cartes d'attribution avec les caractéristiques choisies. Nous les comparons aux cartes d'attribution (via les gradients intégrés) obtenues à partir du réseau de classification. Nous présentons dans les Figures 5 et 6 cette comparaison pour les deux classes en utilisant six exemples.

Au cours de l'analyse des figures 5 et 6, nous avons observé une différence entre les deux modèles entraînés. Les caractéristiques importantes du VGG16 présentent une meilleure délimitation des animaux. La précision totale du VGG16 est plus élevée dans les ensembles d'entraînement et de validation par rapport à ResNet-18. De plus, dans l'ensemble de validation, la classe *chat* a atteint la meilleure précision que la classe *chien* pour ResNet-18 et, la

1. https ://www.kaggle.com/competitions/dogs-vs-cats-redux-kernels-edition/data

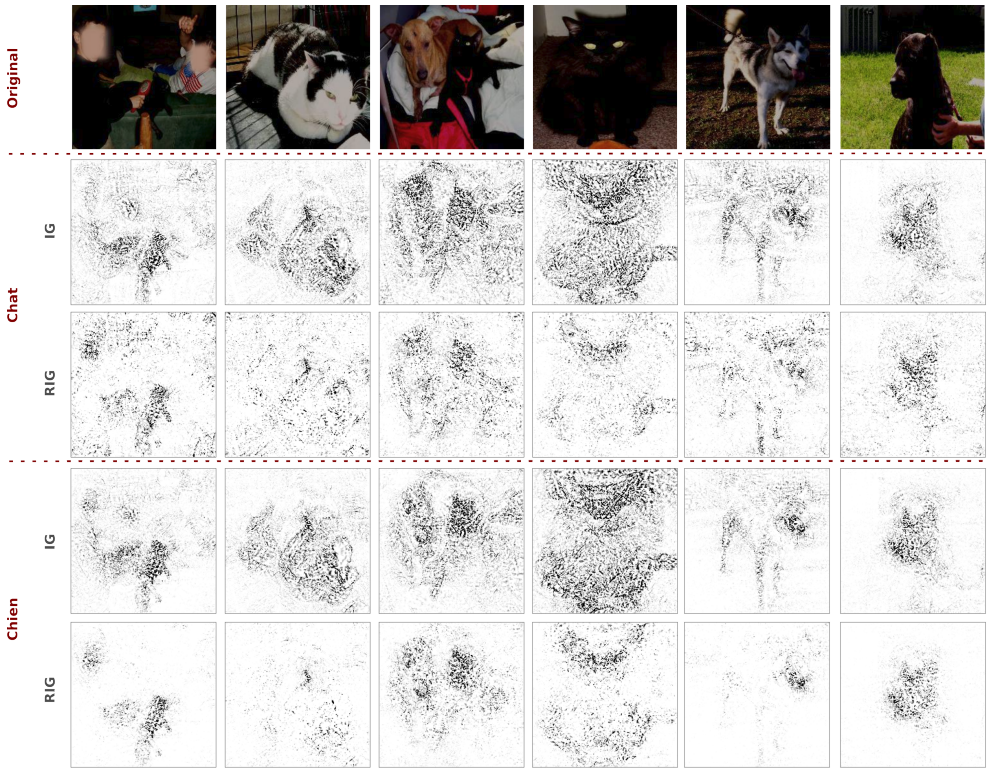


FIG. 6 – *Cartes d'attribution obtenues à partir du VGG16 pour six images de l'ensemble de données. Nous présentons dans la première ligne les images originales, suivies, dans les deuxième et troisième lignes, de la carte d'attribution des gradients intégrés et du RIG respectivement, pour la classe chat. Enfin, aux troisième et quatrième lignes, la carte d'attribution des gradients intégrés et du RIG respectivement, pour la classe chien. RIG réduit l'intérêt pour toutes les images, en particulier pour la classe chien.*

TAB. 1 – *Précision pour la prédiction des ensembles d'entraînement et de validation en utilisant ResNet-18 et VGG16. Les résultats sont présentés pour chaque classe séparément et ensemble (*accuracy *(précision) totale). VGG16 a présenté la meilleure* accuracy.

	Train			**Val**		
	Chat	**Chien**	**Total**	**Chat**	**Chien**	**Total**
ResNet	98.6010	97.8201	98.2102	**97.9329**	97.7996	97.8665
VGG	99.0942	99.0055	99.0455	98.4789	**98.7426**	**98.6102**

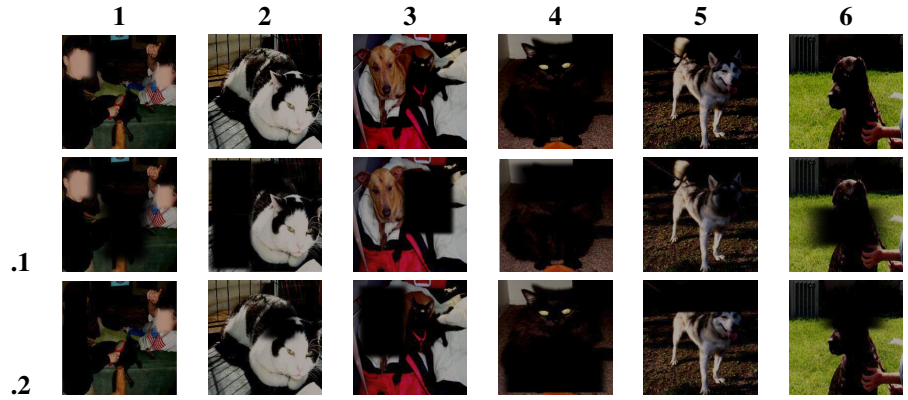
classe *chien* a atteint la meilleure précision pour VGG16. En fait, la classe *chien* était la plus précise (compte tenu de la petite quantité de caractéristiques importantes concentrées) pour le VGG16.

En outre, les cartes finales obtenues sont moins bruitées que les cartes de gradients intégrés

originales, en particulier celle de la classe *chien* (VGG16). Ce résultat pourrait être utile pour mettre en évidence exclusivement les régions de séparation de classes les plus importantes et, par conséquent, pour améliorer la lisibilité. Néanmoins, nous devons vérifier si ces régions sont, dans les faits, importantes pour la classe correspondante.

5.2 Probabilités après occlusion

Les résultats que nous avons présentés dans la section 5.1 indiquent que nous sommes capables d'obtenir des régions d'intérêt plus spécifiques, car nous avons obtenu des cartes d'attribution moins bruitées. Cependant, nous devons vérifier si les régions indiquées sont importantes pour les classes déterminées. De plus, nous devons tester si, sans ces régions, la distance entre les deux classes est perturbée.



TAB. 2 – *Images originales utilisées pour obtenir les cartes d'attribution dans la première rangée. Nous montrons dans les deuxième et troisième rangées les deux occlusions différentes appliquées à l'image originale ci-dessus (première rangée) afin de valider les régions désignées comme importantes.*

Pour tester ces régions d'intérêt, en nous inspirant de l'analyse de sensibilité à l'occlusion proposée par Zeiler et Fergus (2014), nous avons généré deux images occluses pour chaque échantillon. Le critère choisi pour l'occlusion était de cacher les régions avec des pixels plus concentrés pour les classes *chat* et *chien* (une image chacune). Nous avons choisi d'utiliser la même couleur pour occlure les régions que celle de l'image de base de l'IG : le noir. Nous avons également considéré les similarités entre les deux réseaux pour générer l'occlusion. Nous présentons dans la Figure 2 les images originales (première ligne) numérotées de 1 à 6 et les images dérivées correspondantes, occlusion .1 dans la deuxième ligne et occlusion .2 dans la troisième ligne.

Suite à ces occlusions, nous avons obtenu les sorties du réseau de classification analysé qui sont présentées dans le Tableau 3. Les colonnes **Chat (cach.)** et **Chien (cach.)** présentent les sorties pour chacune de ces deux classes (avant softmax) pour les images occluses (.1 et .2). Les colonnes **Chat (orig.)** et **Chien (orig.)** sont les résultats pour les images originales (sans aucune occlusion).

TAB. 3 – *Nous présentons les sorties de classification originales (avant softmax) pour les images échantillons (Im) de la Figure 2, originales (orig.) et occluses (cach.). Les résultats sont représentatifs des classes chat et chien dans les deux réseaux : ResNet-18 et VGG16.*

	ResNet				VGG				
Im	**Chat (cach.)**	**Chien (cach.)**	**Chat (orig.)**	**Chien (orig.)**	**Chat (cach.)**	**Chien (cach.)**	**Chat (orig.)**	**Chien (orig.)**	**Im**
1.1	-0.238	0.613	-0.789	0.906	-1.130	-0.932	**-1.322**	**-0.501**	**1**
1.2	-0.723	0.782			**0.257**	**-0.905**			
2.1	**1.786**	**-1.122**	**1.309**	**-0.638**	**3.649**	**-4.049**	**3.493**	**-4.551**	**2**
2.2	1.543	-0.568			**2.640**	**-3.585**			
3.1	-1.445	2.398	-1.803	2.546	-4.254	4.046	-5.444	4.835	**3**
3.2	-0.011	0.358			0.578	-1.112			
4.1	**0.266**	**0.279**	**1.821**	**-0.641**	0.217	-0.776	1.634	-2.458	**4**
4.2	1.706	-0.649			1.601	-1.660			
5.1	-0.602	0.656	-1.126	0.901	**0.546**	**-0.436**	**-2.233**	**1.629**	**5**
5.2	-1.740	1.501			**-3.467**	**2.793**			
6.1	-0.041	-0.173	-0.620	0.608	-1.538	1.047	-2.132	1.327	**6**
6.2	-0.334	0.067			-1.438	0.782			

Cette expérience d'occlusion nous a permis de constater que les régions choisies étaient en fait importantes. Voici deux exemples intéressants : les occlusions dans l'image **1** pour VGG16 ont confirmé les cartes *RIG*, l'image était mal classée et en occultant le visage d'une des personnes dans l'image (mis en évidence dans les deux cartes de classes), nous avons obtenu la classe correcte (**1. 2** en rouge) ; et dans l'image **5**, en occultant le museau du chien, nous avons changé de classe, passant du chien au chat (**5.1** en rouge), en occultant le haut de l'image (près des oreilles) nous avons renforcé la classe chien (**5.2** en bleu). D'autres exemples, comme l'image **2**, l'occlusion des notes (**2.1**) renforce la classe chat (chiffres en gras) et, l'occlusion d'une oreille de chat pour VGG réduit la distance de classe chat/chien (chiffres en bleu). Pour ResNet-18, l'image **4** occluse dans la région des oreilles subit un léger changement de classe (chiffres en gras). Cela nous donne des indications que les oreilles proéminentes sont très considérées pour augmenter la classe *chat* et que le museau augmente la classe *chien*.

6 Conclusion

Ces premières expériences ont montré que *RIG* est capable de capturer les principales régions responsables de la relation de classe. Sur la base des expériences d'occlusion, nous avons observé que certaines images, telles que l'image du chien blanc, lorsqu'elles sont occultées dans la région indiquée par *RIG*, changent même de classe. Cela pourrait indiquer que les réseaux se concentrent sur des concepts très spécifiques lors de la mise en relation des classes. Notre principale hypothèse est la suivante : les deux réseaux utilisent principalement les régions des oreilles et du nez pour distinguer les chats des chiens. Cette hypothèse devrait être vérifiée a posteriori par de futures expérimentations. Dans le cadre d'un travail futur, nous avons l'intention d'étendre les expériences à d'autres architectures et ensembles de données avec des classes multiples. De plus, nous espérons vérifier différentes valeurs pour α et com-

ment choisir automatiquement la meilleure plage de valeurs de α pour optimiser l'impact des modèles de régression de support.

Références

Adadi, A. et M. Berrada (2018). Peeking inside the black-box : A survey on explainable artificial intelligence (XAI). *IEEE Access 6*, 1–23.

Bach, S., A. Binder, G. Montavon, F. Klauschen, K.-R. Muller, et W. Samek (2015). On pixel-wise explanations for non-linear classifier decisions by layer-wise relevance propagation. *PLoS One 10*(7), 1–46.

Bien, J. et R. Tibshirani (2012). Prototype selection for interpretable classification. *Annals of Applied Statistics 5*(4), 1–23.

Chattopadhyay, A., A. Sarkar, P. Howlader, et V. Balasubramanian (2018). Grad-cam++ : Generalized gradient-based visual explanations for deep convolutional networks. In *4th IEEE Winter Conference on Applications of Computer Vision (WACV)*, pp. 839–847.

Erhan, D., Y. Bengio, A. Courville, et P. Vincent (2009). Visualizing higher-layer features of a deep network. *Technical Report, Univeristé de Montréal*, 1–13.

Gu, R., G. Wang, T. Song, R. Huang, M. Aertsen, J. Deprest, S. Ourselin, T. Vercauteren, et S. Zhang (2021). Ca-net : Comprehensive attention convolutional neural networks for explainable medical image segmentation. *IEEE Transactions on Medical Imaging 40*, 699–711.

He, K., X. Zhang, S. Ren, et J. Sun (2016). Deep residual learning for image recognition. In *29th IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 770–778.

Kim, B., R. Khanna, et O. Koyejo (2016). Examples are not enough, learn to criticize ! Criticism for interpretability. In *30th International Conference on Neural Information Processing Systems (NeurIPS)*, pp. 2288–2296.

Lundberg, S. M. et S.-I. Lee (2017). A unified approach to interpreting model predictions. In *31st International Conference on Neural Information Processing Systems (NeurIPS)*, pp. 4768–4777.

Merrick, L. (2019). Randomized ablation feature importance.

Park, D. H., L. A. Hendricks, Z. Akata, A. Rohrbach, B. Schiele, T. Darrell, et M. Rohrbach (2018). Multimodal explanations : Justifying decisions and pointing to the evidence. In *31st International Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 8779–8788.

Quinlan, J. R. (1986). Induction of decision trees. *Machine Learning 1*, 81–106.

Ribeiro, M. T., S. Singh, et C. Guestrin (2016). "Why Should I Trust You ?" : Explaining the predictions of any classifier. In *22nd International Conference on Knowledge Discovery and Data Mining (KDD)*, pp. 1135–1144.

Selvaraju, R. R., M. Cogswell, A. Das, R. Vedantam, D. Parikh, et D. Batra (2017). Grad-cam : Visual explanations from deep networks via gradient-based localization. In *16th International Conference on Computer Vision (ICCV)*, pp. 618–626.

Shrikumar, A., P. Greenside, et A. Kundaje (2017). Learning important features through propagating activation differences. In *34th International Conference on Machine Learning (ICML)*, pp. 3145–3153.

Simonyan, K., A. Vedaldi, et A. Zisserman (2014). Deep inside convolutional networks : Visualising image classification models and saliency maps. In *Workshop at 2nd International Conference on Learning Representations (ICLR)*, pp. 1–8.

Simonyan, K. et A. Zisserman (2015). Very deep convolutional networks for large-scale image recognition. In *3rd International Conference on Learning Representations (ICLR)*.

Springenberg, J. T., A. Dosovitskiy, T. Brox, et M. Riedmiller (2015). Striving for simplicity : The all convolutional net. In *3rd International Conference on Learning Representations (ICLR)*, pp. 1–14.

Sundararajan, M., A. Taly, et Q. Yan (2017). Axiomatic attribution for deep networks. In *34th International Conference on Machine Learning (ICML)*, pp. 3319–3328.

Thiagarajan, J. J., B. Kailkhura, P. Sattigeri, et K. N. Ramamurthy (2016). Treeview : Peeking into deep neural networks via feature-space partitioning.

Zeiler, M. D. et R. Fergus (2014). Visualizing and understanding convolutional networks. In *13th European Conference on Computer Vision (ECCV)*, pp. 818–833.

Zhou, B., A. Khosla, A. Lapedriza, A. Oliva, et A. Torralba (2016). Learning deep features for discriminative localization. In *29th IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 2921–2929.

Summary

Visualizations provided by xAI techniques to explain convolutional neural networks could be difficult to interpret. The richness of the patterns in an image that are provided as inputs (the pixels of an image) leads to complex correlations between the classes. Gradient-based techniques, such as Integrated Gradients (IG), perform a good work by evincing the importance of these features. However, when visualizing them as images, we can end up with excessive noise, thus some issues to interpret the provided image. In this work we propose Gradients Intégrés Renforcés (*RIG*), a variation of IG, which aims at highlighting influential regions took into account by the network. This method intends to reduce the areas to be analyzed when visualizing the results, consequently, generating less apparent noise. Experiments with occluded images demonstrate that the regions chosen by our method indeed played an important role in the classification.