Le réseau U-Net exploite-t-il des relations directionnelles entre objets pour les segmenter et les reconnaître ?

Mateus RIVA[1], Pietro GORI[1], Florian YGER[2], Isabelle BLOCH[3,1]

[1]LTCI, Télécom Paris, Institut Polytechnique de Paris, France

[2]LAMSADE, Université Paris-Dauphine, PSL Research University, France

[3]Sorbonne Université, CNRS, LIP6, Paris, France

{mateus.riva, pietro.gori}@telecom-paris.fr
florian.yger@dauphine.fr, isabelle.bloch@sorbonne-universite.fr

Résumé – Nous proposons dans cet article une méthode expérimentale permettant d'identifier dans quelle mesure un réseau de neurones (ici le U-Net) utilise des relations spatiales directionnelles entre des objets d'une scène pour les segmenter et les reconnaître. Ce travail s'inscrit dans la ligne de la recherche d'explications aux prédictions d'un réseau de neurones.

Abstract – We propose an experimental method to identify whether neural networks (here the classical U-Net) makes use of directional spatial relations between objects in a scene in order to segment and recognize them. This work contributes to the field of explainable AI, where explanations to the predictions of a network are searched.

1 Introduction

Avec le développement des réseaux de neurones convolutifs (CNN) et le développement de l'intelligence artificielle explicable (XAI), plusieurs approches ont été proposées pour expliquer les prédictions des CNN, en recherchant principalement les informations locales (régions ou caractéristiques) impliquées dans une décision [11]. Cependant, les informations structurelles telles que les relations spatiales se sont avérées utiles pour analyser et interpréter une scène et pour reconnaître les objets qu'elle contient (voir par exemple [1] et les références qui y sont citées, ou encore [7, 8]). On suppose souvent que les CNN ont la capacité intrinsèque d'apprendre des relations pertinentes tant qu'elles s'inscrivent dans le champ réceptif [4, 5, 8, 10]. Cependant, à notre connaissance, cet encodage implicite des relations spatiales n'a pas été étudié en détail. D'autres travaux supposent que cette capacité n'est pas toujours garantie, et forcent les relations en utilisant des techniques externes au CNN [2, 3, 7, 12], en les modélisant de manière explicite, ou en combinant informations textuelles et visuelles. Ces approches sortent du cadre de cette étude. De plus, l'utilisation de certaines mesures de performance ne met pas en évidence le processus de raisonnement conduisant à une décision. Pour toutes ces raisons, il devient difficile de dire si, quand ou comment un CNN donné apprend une relation d'objet particulière.

C'est à ces questions que nous cherchons à répondre ici. Dans ce travail, nous nous concentrons sur *les relations directionnelles*, où les objets d'une scène sont distribués dans des directions spécifiques les uns par rapport aux autres (par exemple, « le cercle est à 20 pixels à gauche du carré, à la même hauteur »). Contrairement aux techniques mentionnées ci-dessus, notre travail vise à explorer de manière expérimentale l'hypothèse implicite qu'un CNN peut raisonner entre les objets de son champ réceptif, d'une manière contrôlée. L'objectif de cet article est de déterminer si un réseau U-Net de base, entraîné pour une tâche de segmentation multi-objets avec des fonctions de coût communes, est capable d'apprendre et d'utiliser les relations directionnelles entre des objets distincts pour aider à leur segmentation et leur reconnaissance. Nous entraînons le réseau populaire U-Net [6], en utilisant des hyperparamètres couramment utilisés (voir section 2.2), dans un contexte où les informations sur les relations directionnelles sont essentielles pour une bonne segmentation des objets d'intérêt. Nous contribuons ainsi au domaine de l'explicabilité des réseaux de neurones en présentant les performances de ce réseau dans un tel contexte. Notre code est disponible publiquement sur `https://github.com/maree0/satann_synth`.

2 Méthodes

Dans cette section, nous présentons des méthodes expérimentales pour évaluer les capacités de raisonnement spatial directionnel du réseau U-Net, en l'entraînant sur une tâche de segmentation qui nécessite d'utiliser les relations directionnelles entre les objets pour obtenir une réponse correcte. À cette fin, nous présentons le jeu de données synthétiques Nuage d'Objets Structurés (NOS).

2.1 Le jeu de données Nuage d'Objets Structurés

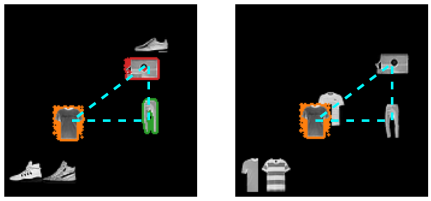
Le jeu de données Nuage d'Objets Structurés (NOS) proposé utilise des images simples (issues de Fashion-MNIST [9]) pour générer une scène structurée. Une donnée NOS est une image avec des objets d'intérêt (OI) appartenant à des classes spécifiques et distribués de manière structurée, ainsi que plusieurs instances d'un ensemble donné d'objets distribués de manière aléatoire et appelés bruit. Les OI (et seulement les OI) sont les cibles de la segmentation, et sont toujours au premier plan (c'est-à-dire qu'ils ne sont jamais occultés par les objets de bruit). Les OI ont une boîte englobante de taille de 28×28 pixels. Nous utilisons une configuration composée d'images 2D de taille 160×160 contenant trois objets d'intérêt, chacun appartenant à une classe différente (spécifiquement, les « chemises », « pantalons », et « sacs » de Fashion-MNIST). Ces objets forment les sommets d'un triangle rectangle de côtés $48 \times 64 \times 80$, dont la cathète majeure est horizontale (voir figure 1), ce qui détermine les relations directionnelles entre les objets. L'ensemble de la structure d'OI est déplacé par une translation d'un nombre aléatoire de pixels, tirés indépendamment selon une distribution uniforme pour chaque axe dans un intervalle spécifique. Nous utilisons les configurations de distribution de bruit suivantes.

Facile : trois éléments de bruit sont ajoutés à l'image, appartenant à une classe différente de celles des objets d'intérêt (« chaussures » dans nos expériences). La translation appliquée au triangle prend ses valeurs dans l'intervalle $[-32, 32]$ pixels. La translation de chaque OI est tirée indépendamment selon une loi uniforme dans l'intervalle de $[-16, 16]$ pixels, rendant ainsi le triangle légèrement imparfait et ajoutant du bruit aux relations directionnelles.

Stricte : trois éléments de bruit sont ajoutés à l'image, appartenant à la classe « chemise ». Le triangle des OI est toujours parfait. La translation appliquée au triangle prend ses valeurs dans l'intervalle $[-40, 40]$ pixels. De plus, les éléments de bruit sont distribués uniquement dans la région inférieure gauche de la figure (à l'intérieur d'un carré de taille 80×80), de sorte que les informations de position absolue sont inutiles pour segmenter l'OI. Une segmentation correcte n'est possible que si les relations directionnelles entre les objets sont apprises. Enfin, seul l'objet de la classe dont certaines instances sont du bruit (« chemises ») est considéré comme une cible pour la tâche de segmentation.

Des exemples d'images NOS avec des objets de Fashion-MNIST sont illustrés dans la figure 1 pour deux configurations.

La configuration « Stricte » présente un problème conjoint de segmentation et de détection. Les réseaux doivent apprendre à détecter et à segmenter correctement les objets (une tâche simple), mais ils doivent également apprendre à raisonner pour déterminer quel est le bon objet. Une bonne segmentation de l'objet correct implique un taux élevé de vrais positifs (TP) par rapport aux faux négatifs (FN). Cependant, les résultats de la segmentation qui pointent vers des objets incorrects entraîneront un faible taux de vrais positifs (TP) par rapport aux faux positifs (FP).



Facile Stricte

FIGURE 1 – Exemples d'images du Nuage d'Objets Structurés, pour deux configurations. Les cibles de la segmentation sont mises en évidence (contours en couleurs). Les relations directionnelles sont représentées par le triangle en pointillés.

2.2 Apprentissage

L'apprentissage du modèle commence par le choix d'une configuration NOS et la définition de la taille de l'ensemble de données D, dont 70% sont utilisés pour l'apprentissage, et les 30% restant pour la validation. Nous utilisons un réseau U-Net standard [6] à quatre niveaux. Le champ réceptif est de 61×61 pixels au niveau du goulet (couche la plus basse) et 101×101 pixels au niveau de la sortie [1], et peut donc s'adapter à tous les OI. Nous initialisons aléatoirement les poids du réseau avec 5 graines distinctes. La division de l'ensemble de données en ensemble d'apprentissage et ensemble de validation est répétée aléatoirement cinq fois. Pour chaque configuration NOS, nous entraînons un réseau pour 100 passages sur l'ensemble des exemples de la base d'apprentissage en utilisant un optimiseur ADAM et l'entropie croisée comme fonction de coût. Nous choisissons le modèle avec le plus faible coût de validation.

Pour évaluer les modèles, nous générons un ensemble de test contenant 100 nouvelles images de la même configuration NOS que le modèle, et utilisons deux mesures : la précision, définie comme la valeur prédictive positive par pixel $\frac{TP}{TP+FP}$, et le rappel, défini comme le taux de vrais positifs par pixel $\frac{TP}{TP+FN}$. Nous calculons la précision et le rappel moyens sur l'ensemble de test pour la classe « chemise », sur toutes les initialisations où le modèle a convergé (défini comme les cas où la précision et le rappel sont supérieurs à $0,5$). Nous indiquons également combien de modèles entraînés ont convergé. Les résultats sont résumés dans la table 1. Des exemples de résultats sont présentés dans la figure 2.

Nous pouvons voir que segmenter et reconnaître correctement les objets dans la configuration « Stricte » est difficile lorsque le jeu de données contient peu d'exemples (petite valeur de D). Les scores de précision plus faibles dans ces cas indiquent que le réseau est incapable d'éviter complètement les éléments de bruit. Cependant, avec suffisamment de données, le modèle apprend à reconnaître les OI. Le nombre de modèles

1. Calculé en utilisant la bibliothèque *receptivefield*, disponible sur `https://github.com/shelfwise/receptivefield`

TABLE 1 – Précision et rappel moyens ($\pm$ un écart-type) pour la classe « chemise », et écart-type, pour différentes tailles et configurations d'ensembles de données, lorsque les modèles convergent.

Config.	D	Classe « chemise » Précision	Classe « chemise » Rappel	Convergences
Facile	100	$0,97 \pm 0,09$	$0,95 \pm 0,11$	25/25
	1000	$1,00 \pm 0,00$	$1,00 \pm 0,00$	25/25
	10000	$0,99 \pm 0,03$	$0,99 \pm 0,04$	25/25
Stricte	1000	$0,65 \pm 0,32$	$0,71 \pm 0,33$	6/25
	5000	$0,79 \pm 0,29$	$0,79 \pm 0,30$	14/25
	10000	$0,87 \pm 0,19$	$0,86 \pm 0,21$	21/25
	50000	$0,91 \pm 0,14$	$0,90 \pm 0,15$	22/25

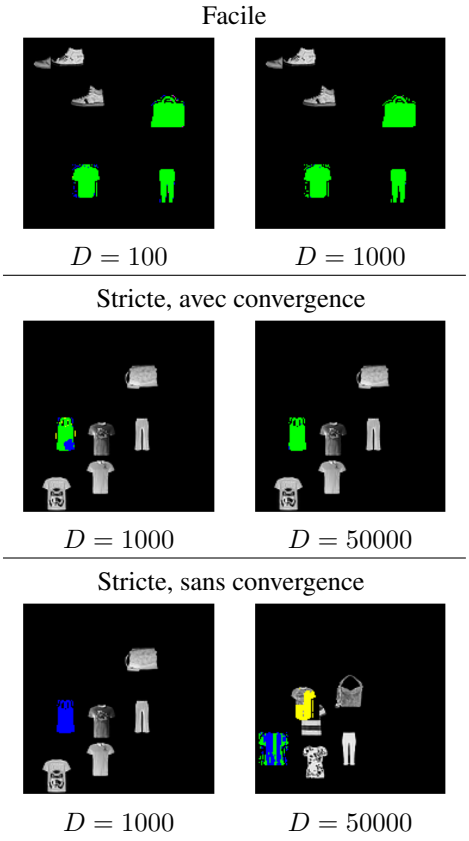


FIGURE 2 – Exemples de résultats de certains des modèles entraînés. Les régions vertes indiquent les vrais positifs ; les régions bleues indiquent les faux négatifs ; les régions jaunes indiquent les faux positifs de la classe « chemise ». Le bruit autour des OI est hérité de la base de données Fashion-MNIST.

ayant convergé montre qu'il y a peu de garantie de réussir à résoudre la tâche « Stricte » sans beaucoup plus de données que pour la tâche « Facile ».

L'analyse des exemples de sortie des réseaux, dans la figure 2, permet de mieux comprendre les mesures de la table 1. Pour la configuration « Facile » (première ligne), le réseau fonctionne parfaitement, ce qui indique que, sans bruit déroutant, il s'agit d'une tâche simple. Dans la configuration « Stricte », les modèles qui n'ont pas convergé (dernière ligne) produisent encore quelques prédictions, car le réseau n'avait qu'une seule cible de segmentation. Cependant, ils ne parviennent pas à détecter correctement et à segmenter complètement la bonne chemise. Dans les cas de convergence (ligne centrale), nous pouvons observer la tendance attendue vers de meilleures segmentations lorsque le nombre de données augmente ; il est clair qu'avec suffisamment de données, le réseau peut résoudre cette tâche. Il doit donc être capable de raisonner sur les relations spatiales directionnelles.

3 Mesure de l'exploitation des relations directionnelles

Si le modèle apprend à segmenter un OI en utilisant un autre objet comme référence, nous pouvons nous attendre à ce que le déplacement de la référence affecte la segmentation. Pour le démontrer, nous générons des images de test où chacun des OI, un par un, est déplacé sur l'image avec un pas de 20 pixels, tandis que les autres OI restent fixes. L'OI qui est déplacé est appelé la référence. La référence est toujours au premier plan de l'image. Nous calculons le rappel et la précision de la segmentation de l'OI « chemise » (même lorsqu'il est utilisé comme référence). Pour toutes les positions de la référence, 20 images sont générées avec le triangle parfaitement centré et le bruit distribué selon la configuration considérée.

Nous construisons ensuite une carte d'évaluation dans laquelle la valeur en chaque point (x, y) est la mesure d'évaluation moyenne (soit la précision, soit le rappel) de la classe « chemise » lorsque l'objet de référence est à la position (x, y). Dans la configuration « Stricte », si le réseau a appris à utiliser d'autres classes pour la segmentation de l'OI, nous nous attendons à voir de mauvaises performances lorsque les références ne sont pas positionnées aux endroits attendus.

La figure 3 montre ces cartes sur le plus grand ensemble de données ($D = 50000$) pour la configuration « Stricte ». Pour faciliter l'interprétation, les cartes d'évaluation sont superposées sur une image fictive montrant les OI centrés, et la référence n'est pas affichée.

Dans la première ligne, où la « chemise » elle-même est déplacée sur l'image, nous pouvons voir que sa segmentation ne peut se faire que dans une région spécifique de l'image. Cela peut être dû au fait que le réseau a besoin des autres OI pour segmenter la « chemise », qu'il apprend les positions absolues où la « chemise » peut être trouvée, ou une combinaison des deux. Dans la deuxième ligne, nous pouvons voir que le rappel est remarquablement plus faible lorsque le sac n'est pas parfaitement placé ; et aussi que la précision de la chemise bénéficie du positionnement correct du « sac », ce qui implique qu'il joue un certain rôle en permettant au réseau d'éviter de segmenter les mauvaises « chemises ». Tout cela est une preuve supplémentaire que le réseau U-Net a appris à utiliser d'autres objets lors du raisonnement sur la segmentation de l'OI « chemise ».

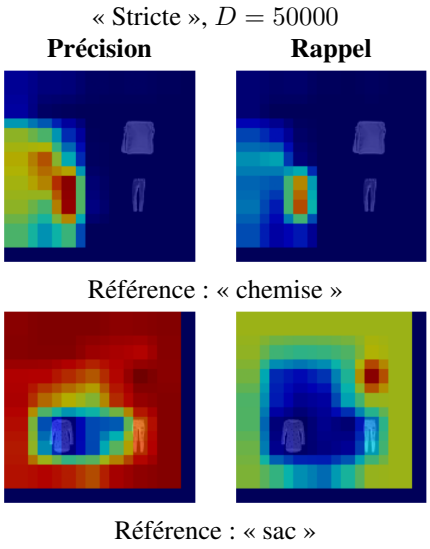


FIGURE 3 – Cartes de précision et de rappel de la classe « chemise » lorsque les objets de référence sont déplacés sur l'image, pour la configuration « Stricte ». Les effets de bord sont dus aux limites de la fenêtre de déplacement.

4 Conclusion

D'après les expériences présentées, on peut raisonnablement conclure que le réseau U-Net est effectivement capable d'exploiter différents objets dans son champ réceptif, et d'utiliser les relations spatiales directionnelles pour assurer une segmentation correcte. Lorsqu'il est entraîné à une tâche nécessitant un raisonnement relationnel directionnel, un simple réseau U-Net entraîné pour optimiser l'entropie croisée est capable d'obtenir des résultats satisfaisants, lorsque suffisamment de données sont fournies. Les tests montrent que la perturbation des relations directionnelles dans les données de test entraîne directement une moindre performance, ce qui permet d'expliquer la nature des relations apprises par le réseau.

Ce travail n'est qu'un premier pas vers l'amélioration de l'explicabilité des CNN en comprenant mieux comment les CNN de base peuvent raisonner sur les relations spatiales entre les objets contenus dans leurs champs réceptifs. Nous avons montré expérimentalement qu'un CNN peut apprendre à prendre en compte le contexte spatial des objets - plus précisément, il peut apprendre les relations spatiales directionnelles - dans son champ réceptif, tout en mettant en évidence la nécessité de grandes quantités de données, inhérente aux tâches de raisonnement complexes. D'autres travaux viseront à explorer cette question dans différentes directions : (i) quel est plus précisément le processus d'apprentissage des relations ? (ii) l'apprentissage relationnel peut-il être accéléré ? (iii) l'accélération de l'apprentissage des relations donnera-t-elle lieu à des réseaux plus performants ou réduira-t-elle le nombre de données nécessaires pour l'apprentissage ? (iv) quelles sont les limites du raisonnement relationnel (comme le comportement face à des champs réceptifs trop étroits ou épars) ?

Références

[1] I. Bloch. Fuzzy sets for image processing and understanding. *Fuzzy Sets and Systems*, 281 :280–291, 2015.

[2] J. Chopin, J.-B. Fasquel, H. Mouchère, R. Dahyot, and I. Bloch. Semantic image segmentation based on spatial relationships and inexact graph matching. In *2020 Tenth International Conference on Image Processing Theory, Tools and Applications (IPTA)*, pages 1–6, 2020.

[3] N. Krishnaswamy, S. Friedman, and J. Pustejovsky. Combining deep learning and qualitative spatial reasoning to learn complex structures from sparse examples with noise. In *AAAI Conference on Artificial Intelligence*, volume 33, pages 2911–2918, Jul. 2019.

[4] S. S. Mohseni Salehi, D. Erdogmus, and A. Gholipour. Auto-context convolutional neural network (Auto-Net) for brain extraction in magnetic resonance imaging. *IEEE Transactions on Medical Imaging*, 36(11) :2319–2330, 2017.

[5] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi. You only look once : Unified, real-time object detection. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 779–788, 2016.

[6] O. Ronneberger, P. Fischer, and T. Brox. U-Net : Convolutional Networks for Biomedical Image Segmentation. In N. Navab, J. Hornegger, W. M. Wells, and A. F. Frangi, editors, *Medical Image Computing and Computer-Assisted Intervention (MICCAI)*, pages 234–241, 2015.

[7] A. Santoro, D. Raposo, D. G. Barrett, M. Malinowski, R. Pascanu, P. Battaglia, and T. Lillicrap. A simple neural network module for relational reasoning. *Advances in Neural Information Processing Systems*, 30, 2017.

[8] M. Shaban, R. Awan, M. M. Fraz, A. Azam, Y.-W. Tsang, D. Snead, and N. M. Rajpoot. Context-aware convolutional neural network for grading of colorectal cancer histology images. *IEEE Transactions on Medical Imaging*, 39(7) :2395–2405, 2020.

[9] H. Xiao, K. Rasul, and R. Vollgraf. Fashion-MNIST : a novel image dataset for benchmarking machine learning algorithms. https ://github.com/zalandoresearch/fashion-mnist and arXiv :1708.07747, 2017. [Online ; accessed 24-February-2022].

[10] X. Yang, X. Li, Y. Ye, R. Y. K. Lau, X. Zhang, and X. Huang. Road detection and centerline extraction via deep recurrent convolutional neural network U-Net. *IEEE Transactions on Geoscience and Remote Sensing*, 57(9) :7209–7220, 2019.

[11] Q.-S. Zhang and S.-C. Zhu. Visual interpretability for deep learning : a survey. *Frontiers of Information Technology & Electronic Engineering*, 19(1) :27–39, 2018.

[12] B. Zhou, A. Andonian, A. Oliva, and A. Torralba. Temporal relational reasoning in videos. In *European Conference on Computer Vision (ECCV)*, 2018.