ICHEC - ALUMNI
BRUSSELS MANAGEMENT SCHOOL

ARDANS
Knowledge Consulting & Software Solutions

SRBS
KBVS Royal Statistical Society of Belgium

delaware

Atelier Apprentissage Profond : Théorie et Applications (APTA)

Atelier organisé dans le cadre de la conférence EGC 2020

28 janvier 2020 Bruxelles (Belgique)

Organisateurs

- Jonathan Weber, MCF, Université de Haute-Alsace, IRIMAS
- Camille Kurtz, MCF, Université Paris Descartes, LIPADE
- Cédric Wemmert, PU, Université de Strasbourg, ICUBE
- Germain Forestier, PU, Université de Haute-Alsace, IRIMAS

ICUBE

IRIMAS UNIVERSITÉ HAUTE-ALSACE
Institut de Recherche en Informatique, Mathématiques, Automatique et Signal

LIPADE
Laboratoire d'Informatique PAris DEscartes

L'ATELIER APTA

L'apprentissage profond révolutionne depuis quelques années l'apprentissage machine. Si les premiers résultats marquants ont été obtenus principalement en analyse d'images, les travaux actuels en apprentissage profond (deep learning) s'intéressent à présent à tous les types de données et presque tous les types de traitement (classification de séries temporelles, augmentations de données, analyse de texte, etc.). Son impact dans le domaine de la science des données et l'extraction de connaissances est considérable

Nous avons souhaité proposer dans le cadre de la conférence EGC 2020 un espace d'échanges autour de ce domaine, permettant d'aborder les défis théoriques et les possibilités applicatives offertes à notre discipline de l'extraction et de la gestion des connaissances. Dans le cadre de cet atelier, nous nous focalisons sur les applications de l'apprentissage profond dans différents domaines (analyse ou génération d'images, classification de données temporelles, extraction d'informations à partir de données hétérogènes, etc.) mais également permettre la présentation de travaux plus théoriques (nouvelles architectures, nouvelles fonctions de coût, interprétabilité des modèles, etc).

L'objectif de cet atelier est d'offrir un espace d'échange entre d'une part, des experts de l'apprentissage profond, développant de nouveaux modèles et éventuellement à la recherche de domaines d'application pour les valider, et d'autre part, des utilisateurs avec moins d'expérience et souhaitant appliquer les méthodes d'apprentissage profond à leurs données. Cet espace peut permettre également aux potentiels nouveaux utilisateurs, curieux de ces approches d'en comprendre les avantages et les limites.

CONTENU SCIENTIFIQUE DE L'ATELIER

Sur 7 propositions originales, 6 ont été retenues par le comité de programme de l'atelier. Nous avons été en mesure de proposer pour l'ensemble des propositions deux à trois rapports d'experts afin d'offrir un processus scientifique constructif aux auteurs.

Les articles qui vous sont proposés cette année dans les actes qui suivent explorent une grande variété de thématiques relatives à l'apprentissage profond, aussi bien dans les données que dans les processus méthodologiques proposées.

REMERCIEMENTS

Les responsables de l'atelier souhaitent remercier vivement toutes les personnes ayant contribué à la tenue de cet atelier. En particulier :

- les auteurs pour la qualité de leurs contributions constituant la base essentielle de discussions fructueuses ;
- les membres du comité de programme et plus généralement tous les relecteurs de cet atelier dont le travail d'évaluation était crucial pour assurer la qualité de l'atelier ;
- les organisateurs d'EGC 2020 ainsi que les responsables du comité de programme qui ont mis en place l'environnement et les moyens

Table des matières

GAN based data augmentation for histopathological image segmentation

Florian Allender*, Rémi Allègre*, Cédric Wemmert*, Jean-Michel Dischler*

* Laboratoire ICube
300 bd Sébastien Brant
CS 10413
F-67412 Illkirch Cedex

Abstract In the context of kidney transplant, histopathological images and deep learning are powerful tools to help keep track of diseases related with transplant rejection. However, the training of neural networks require huge amounts of data that are not always available due to the lack of annotation. The use of synthetic data to train algorithms has been proven effective even in medical imaging. We aim at providing a pipeline composed of Generative Adversarial Networks to produce high resolution glomeruli patches that can be used to train a segmentation network. As it is still a work in progress, we focus in this article on the second part of the pipeline : generating images from segmentation masks. We use image translation techniques and in particular the Pix2Pix network. We show that adding structure maps to the input and a regularizing loss helps mitigate the issue of mode collapse and produce good looking results.

1 Introduction

Every year, thousands of patients around the world undergo kidney transplant surgery, while other thousands die while on the waiting list. In this context, the study of kidney transplant rejection is crucial in order to save more lives. Interstitial Fibrosis and Tubular Atrophy (IFTA) and glomerulosclerosis are two pathologies associated with chronic kidney transplant rejection. The study of these pathologies could lead to a better understanding of the mechanisms behind transplant rejection, and thus help reduce the loss of transplanted organs. To do so, researchers and practitioners can rely on Whole Slide Imaging (WSI) and the development of digital histopathology. The objects of interest in these extremely high resolution images are glomeruli, clusters of blood vessels allowing blood filtration. To this end, we need to detect and segment glomeruli on patches extracted from the complete images.

Deep Learning has revolutionized the area of image processing, providing powerful tools to automatically accomplish various tasks such as image recognition (Krizhevsky et al., 2012), voice generation (van den Oord et al., 2016) or self-driving cars (Santana and Hotz, 2016), with great success, sometimes outperforming humans (He et al., 2015; Mnih et al., 2015). In the field of medical images, Deep Learning is closing the gap between clinicians and AI performances (Liu et al., 2019) allowing them to process those images faster, with more accuracy. However the application of Deep Learning on medical images comes with its own set

of limitations, one of the main being the lack of annotated data. This is in particular true for histopathological images and the challenge we are tackling. The segmentation of many WSI requires expert knowledge and is an extremely time consuming task, and as a result an expensive one. Moreover, privacy issues make it difficult for researchers to share their data, making it difficult to collect large databases on which we could test the algorithms of the community, providing a common reference to evaluate their performances.

To cope with the lake of data, new architectures and training procedures have been proposed. A standard procedure in Deep Learning is to artificially augment the size of databases is the use of random affine transformations (rotations, translations, scaling) on the training set (Krizhevsky et al., 2012). In (Ronneberger et al., 2015), the authors propose a deep neural network, U-Net, to segment medical images, as well as elastic transformations to augment their database. U-Net can be train with less data while outperforming previous architectures on the ISBI challenge. We will detail the architecture in section 3. In (Lampert et al., 2019), the authors propose a procedure to learn the segmentation of glomeruli on images with different staining. To go further, a new approach has aroused in recent years : training networks with synthetic data (Nikolenko, 2019).

This approach has been made far more efficient with the introduction of Generative Adversarial Networks (GANs) in 2014 (Goodfellow et al., 2014). Their goal is to learn an implicit representation of a dataset distribution that can be sampled to produce new data. GANs are composed of a pair of Neural Networks : a generator and a discriminator. The generator takes random noise as input and output a data (an image in our case). The discriminator takes a data (real or fake) as input, and outputs a digit indicating if the data is real or fake. Both networks are trained together in a competitive manner, until G learns the data distribution and D is not able to differentiate between fake and real data anymore. GANs have made quick progress and took many forms, from Convolutional GANs (Radford et al., 2015) to Conditional GANs (Odena et al., 2016). They are now able to synthesize high resolution images with astonishing realism (Karas et al., 2018; Karras et al., 2018; Park et al., 2019). GANs still suffer from a few drawbacks : they are notoriously difficult to train, require careful hyper-parameter tuning and have difficulties to produce diverse results (Goodfellow, 2017; Lucic et al., 2018; Salimans et al., 2017; Arjovsky and Bottou, 2017). Indeed, they have a tendency to focus on only a few mode of the data distribution. This issue is known as mode collapse. Many tricks have been proposed in the literature to solve it (Metz et al., 2016; Mao et al., 2019; Che et al., 2016; Arjovsky et al., 2017; Srivastava et al., 2017), but they seem to work only with specific architectures or datasets.

GANs have been used in medical images to help enhance the results of Deep Learning methods on various tasks (reconstruction, registration, segmentation, detection...), as highlighted by the recent review (Yi et al., 2018). Conditional GANs and in particular the ones performing image domain translation such as Pix2Pix (Isola et al., 2016) and CycleGAN (Zhu et al., 2017) are popular in the field. The use of synthetic data to train classification or segmentation algorithms in the medical field has been proven effective (Mahmood et al., 2018; Xiao et al., 2019; Frid-Adar et al., 2018; Hou et al., 2017; Senaras et al., 2018). Our aim in this paper is to propose a new pipeline able to generate high resolution synthetic glomeruli patches. We validate our results by two different means : a perception study with experts, and the training

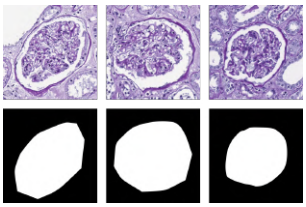


FIG. 1 – *Example of glomeruli patches with associated segmentation masks.*

of a U-Net with our synthetic data. [1]

2 Materials

We have at our disposal 10 annotated WSI from different patients and with different staining, distributed between training, testing and validating set. We extracted patches centered around glomeruli from those images with the corresponding annotation, which is a segmentation mask. See figure 1 for examples. These patches are usually used to train a U-Net for segmentation purposes (Lampert et al., 2019). In this article, we will use the training set to train GANs and then use the produced synthetic data to train the network from (Lampert et al., 2019). To simplify the problem and ease the conduct of our experimentations, we focused on only one staining. In the end, we have 660 pairs of glomeruli and mask with a 256*256 size.

Glomeruli patches are difficult to synthesize because the size of the object of interest is large with respect to the size of the image. we have to reproduce the global structure and local patterns. Thus we can not process it as a stochastic texture only. To do so, GANs are promising tools that may help us captures semantic information at different scale levels.

In order to use our synthetic data to train a U-Net, we need to generate a glomeruli patch and the corresponding mask at the same time. To do so, we first imagined to seperate the problem into two parts : generating a new mask, then generating a new glomerulus with respect to this mask. In this article, we focus on the second step. To generate a glomerulus that respects the constraints imposed by a mask, we use Pix2Pix (Isola et al., 2016), as it takes a semantic label map as input and outputs the corresponding image. However Pix2Pix severely suffers from mode collapse, has already mentioned in the original paper. As a result, the generator always outputs the same image, even when feed with different inputs, as highlighted by figure 2.

In order to tackle the issue of mode collapse, we propose to add information to the input of our generator, in the form of structure maps. Those structure maps are binary images obtained by thresholding the glomeruli images, as show in figure 3. Our input is now the concatenation of a mask and a structure map. By this mean we give more constraints to the generator and completely avoid mode collapse during the image translation phase, as will be shown in the Results section. The issue is not completely solved as we merely moved the burden of generating diversity to a structure and mask generation phase.

1. The first part of the pipeline and the complete analysis of the results are still a work in progress and will not be shown in this version of the paper. The conclusion will be modified accordingly.

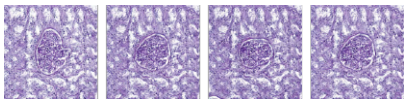


FIG. 2 – *Illustration of mode collapse. The generator always outputs the same texture content, no matter the input.*

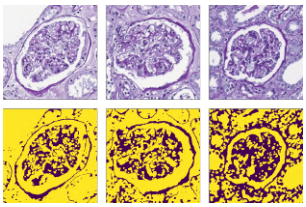


FIG. 3 – *Example of glomeruli patches with associated structure map.*

3 Methods

3.1 U-Net description

U-Net (Ronneberger et al., 2015) is a network with two branches : one that performs data compression with downsampling layers and the other that upscale the data back to its original resolution, just like auto-encoders (Rumelhart et al., 1986). The difference between auto-encoders and U-Net is the presence of skip connections that connect the two branches, allowing information to flow from one side to the other. See figure 4 for details on the original architecture.

3.2 GAN description

A GAN is composed of two networks : a generator (G) and a discriminator (D). We denote p_{data} the distribution of the data and p_g the distribution learned by G. We note p_z the distribution from which we sample a random vector z, used as input for G. G and D play a zero sum game, described by the following equation (Goodfellow et al., 2014) :

$$\min_G \max_D L(D,G) = \mathbb{E}_{\mathbf{x}\sim p_{data}(\mathbf{x})}[logD(\mathbf{x})] + \mathbb{E}_{\mathbf{z}\sim p_z(\mathbf{z})}[log(1-D(G(\mathbf{z})))] \quad (1)$$

Theoretically there exists a optimum for this game when $p_g = p_{data}$, but it is notoriously difficult to find, especially with the stochastic gradient descent techniques used to train neural networks.

3.3 Generating images using a U-Net : mode seeking Pix2pix

Pix2pix is an image translation model, a type of Conditional GAN that takes a label map as input and produce an image as output. It uses a U-Net as generator and can be used to generate images from sketches. The GAN equation is then modified. G now learns a mapping from an image x to an image y. In our case, x is the concatenation of a mask and a structure map.

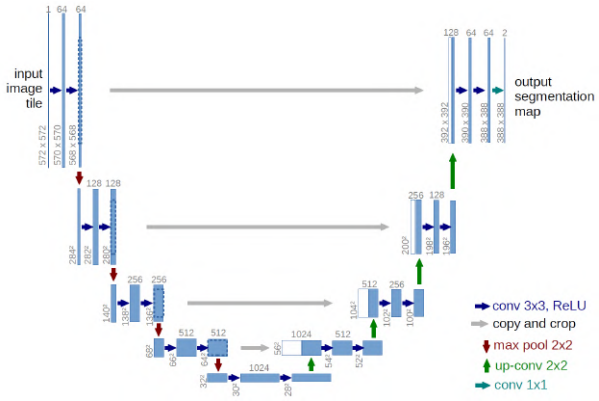


FIG. 4 – *Original U-Net architecture. (Ronneberger et al., 2015)*

In the Conditional GAN framework, D usually receives as input the concatenation of x and y, in order to give poor rating to an image that does not correspond with the condition, not matter how realistic it may be. However not feeding x to D helps reduce mode collapse, so we only feed y as for classic GANs, at the price of a slight loss in visual quality. It gives us the following modified equation :

$$\min_G \max_D L_{cGAN}(D,G) = \mathbb{E}_{\mathbf{x},\mathbf{y}\sim p_{data}(\mathbf{x},\mathbf{y})}[logD(\mathbf{y})] + \mathbb{E}_{\mathbf{x}\sim p_{data}(\mathbf{x})}[log(1-D(G(\mathbf{x})))] \quad (2)$$

The authors also found that adding a L1 distance between the output of G and the real image from the dataset helps to reduce blurring :

$$L_{L1}(G) = \mathbb{E}_{\mathbf{x},\mathbf{y}\sim p_{data}(\mathbf{x},\mathbf{y})}[||\mathbf{y} - G(\mathbf{x})||_1] \quad (3)$$

To that we add a regularizing loss inspired by Mode Seeking GANs (Mao et al., 2019). It aims at reducing mode collapse in the case of conditional GANs by penalizing the generator if it produces similar images for two different condition masks x_1 and x_2 :

$$L_{ms}(G) = \mathbb{E}_{\mathbf{x}_1\sim p_{data}(\mathbf{x}_1),\mathbf{x}_2\sim p_{data}(\mathbf{x}_2)} \frac{||\mathbf{x}_1 - \mathbf{x}_2||_1}{||G(\mathbf{x}_1) - G(\mathbf{x}_2)||_1} \quad (4)$$

Which gives us the expression of our optimal generator :

$$G* = \underset{G}{argmin}\, \underset{D}{max} L_{cGAN}(D,G) + \lambda L_{L1}(G) + \mu L_{ms}(G) \quad (5)$$

We set $\lambda = 10$ and $\mu = 10$. We train both our networks with the Adam optimizer, with the following parameters : learning rate $= 0.0002$, $\beta_1 = 0.5$ and $\beta_2 = 0.999$, batch size $= 1$. The complete model of G and D is given in the appendix.

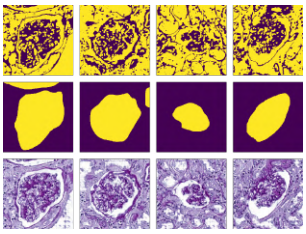


FIG. 5 – *Results from our modified Pix2Pix tested on real structure maps and masks. First row : input structure maps, second row : input masks, third row : results.*

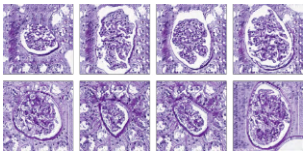


FIG. 6 – *First row : results when training with no regularizing loss. Second row : results when training with no structure maps. In both cases, mode collapse still occurs and the visual quality of the generated samples is not convincing enough.*

4 Results

4.1 Images generated by our mode seeking Pix2Pix

We visually selected twelve epochs with appealing results during training and tested the generator on unseen data. Those data are real masks and structure maps coming from the test set. We show in figure 5 four randomly selected samples from the results. As can be seen here and in the appendix, the images are of very high visual quality and mode collapse is non-existent. The texture created by the generator perfectly matches the structure given as input.

4.2 Ablation study

The use of structure maps and the L_{ms} term are both mandatory to obtain good results. We trained two different models, one using only masks and L_{ms} and the other using masks and structure maps but not L_{ms}. As we show in figure 6, both models collapse and output poor looking results.

We add results obtained when feeding the discriminator with the condition as mention in section 3. The mode collapse effect is not as prominent as in 6, but still visible, especially near the center of the image.

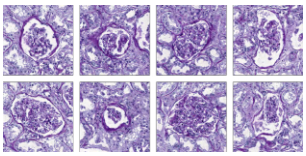


FIG. 7 – *Results when feeding the condition to the discriminator. Mode collapse is still an issue in this case, even if not highly visible.*

5 Conclusion

In this article we exposed the first part of a work in progress. We showed how we can modify an existing image translation model (Pix2Pix) to produce high quality images and alleviate the mode collapse issue. This improvement is made possible by two key ingredients : the use of structure maps to give more constraints to the input of our generator and a regularizing loss term to stabilize the model convergence. This constitutes the second part of our planned pipeline.

The first part of the pipeline is still worked on and will be crucial to our application. We have now to synthesize masks and structure maps of enough quality to feed our modified Pix2Pix, but also with enough diversity so that we can use the images produced by the complete pipeline as a training set.

When the pipeline is complete, we will be able to validate our method by two different means. First a perception study with histopatholgy experts to see if our images can fool the human eye, then a quantitative study to check if a segmentation network trained by our synthetic images can generalize well on real images.

References

Arjovsky, M. and L. Bottou (2017). Towards principled methods for training generative adversarial networks. *ArXiv abs/1701.04862*.

Arjovsky, M., S. Chintala, and L. Bottou (2017). Wasserstein gan. cite arxiv:1701.07875.

Che, T., Y. Li, A. P. Jacob, Y. Bengio, and W. Li (2016). Mode regularized generative adversarial networks. *CoRR abs/1612.02136*.

Frid-Adar, M., I. Diamant, E. Klang, M. Amitai, J. Goldberger, and H. Greenspan (2018). Gan-based synthetic medical image augmentation for increased cnn performance in liver lesion classification. *CoRR abs/1803.01229*.

Goodfellow, I. J. (2017). NIPS 2016 tutorial: Generative adversarial networks. *CoRR abs/1701.00160*.

Goodfellow, I. J., J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio (2014). Generative Adversarial Networks. *ArXiv e-prints*.

He, K., X. Zhang, S. Ren, and J. Sun (2015). Deep residual learning for image recognition. *CoRR abs/1512.03385*.

Hou, L., A. Agarwal, D. Samaras, T. M. Kurç, R. R. Gupta, and J. H. Saltz (2017). Unsupervised histopathology image synthesis. *ArXiv abs/1712.05021*.

Isola, P., J.-Y. Zhu, T. Zhou, and A. A. Efros (2016). Image-to-image translation with conditional adversarial networks. *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 5967–5976.

Karas, T., T. Aila, S. Laine, and J. Lehtinen (2018). Progressive growing of gans for improved quality, stability, and variation. *International Conference on Learning Representations*.

Karras, T., S. Laine, and T. Aila (2018). A style-based generator architecture for generative adversarial networks. *CoRR abs/1812.04948*.

Krizhevsky, A., I. Sutskever, and G. E. Hinton (2012). Imagenet classification with deep convolutional neural networks. In F. Pereira, C. J. C. Burges, L. Bottou, and K. Q. Weinberger (Eds.), *Advances in Neural Information Processing Systems 25*, pp. 1097–1105. Curran Associates, Inc.

Lampert, T., O. Merveille, J. Schmitz, G. Forestier, F. Feuerhake, and C. Wemmert (2019). Strategies for training stain invariant cnns. In *2019 IEEE 16th International Symposium on Biomedical Imaging (ISBI 2019)*, pp. 905–909.

Liu, X., L. Faes, A. U. Kale, S. K. Wagner, D. J. Fu, A. Bruynseels, T. Mahendiran, G. Moraes, M. Shamdas, C. Kern, J. R. Ledsam, M. K. Schmid, K. Balaskas, E. J. Topol, L. M. Bachmann, P. A. Keane, and A. K. Denniston (2019). A comparison of deep learning performance against health-care professionals in detecting diseases from medical imaging: a systematic review and meta-analysis. *The Lancet Digital Health 1*(6), e271 – e297.

Lucic, M., K. Kurach, M. Michalski, O. Bousquet, and S. Gelly (2018). Are gans created equal? a large-scale study. In *Proceedings of the 32Nd International Conference on Neural Information Processing Systems*, NIPS'18, USA, pp. 698–707. Curran Associates Inc.

Mahmood, F., R. Chen, and N. J. Durr (2018). Unsupervised reverse domain adaptation for synthetic medical images via adversarial training. *IEEE Transactions on Medical Imaging 37*(12).

Mao, Q., H. Lee, H. Tseng, S. Ma, and M. Yang (2019). Mode seeking generative adversarial networks for diverse image synthesis. *CoRR abs/1903.05628*.

Metz, L., B. Poole, D. Pfau, and J. Sohl-Dickstein (2016). Unrolled generative adversarial networks. *ArXiv abs/1611.02163*.

Mnih, V., K. Kavukcuoglu, D. Silver, A. A. Rusu, J. Veness, M. G. Bellemare, A. Graves, M. Riedmiller, A. K. Fidjeland, G. Ostrovski, S. Petersen, C. Beattie, A. Sadik, I. Antonoglou, H. King, D. Kumaran, D. Wierstra, S. Legg, and D. Hassabis (2015). Human-level control through deep reinforcement learning. *Nature 518*(7540), 529–533.

Nikolenko, S. I. (2019). Synthetic data for deep learning. *ArXiv abs/1909.11512*.

Odena, A., C. Olah, and J. Shlens (2016). Conditional image synthesis with auxiliary classifier gans. In *ICML*.

Park, T., M.-Y. Liu, T.-C. Wang, and J.-Y. Zhu (2019). Semantic image synthesis with spatially-adaptive normalization. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*.

Radford, A., L. Metz, and S. Chintala (2015). Unsupervised representation learning with deep convolutional generative adversarial networks. *CoRR abs/1511.06434*.

Ronneberger, O., P. Fischer, and T. Brox (2015). U-net: Convolutional networks for biomedical image segmentation. *CoRR abs/1505.04597*.

Rumelhart, D. E., G. E. Hinton, and R. J. Williams (1986). Learning internal representations by error propagation. In D. E. Rumelhart, J. L. McClelland, and C. PDP Research Group (Eds.), *Parallel Distributed Processing: Explorations in the Microstructure of Cognition, Vol. 1*, pp. 318–362. Cambridge, MA, USA: MIT Press.

Salimans, T., I. J. Goodfellow, W. Zaremba, V. Cheung, A. Radford, and X. Chen (2017). Improved techniques for training gans. *CoRR abs/1606.03498*.

Santana, E. and G. Hotz (2016). Learning a driving simulator. *CoRR abs/1608.01230*.

Senaras, C., M. K. K. Niazi, B. Sahiner, M. P. Pennell, G. Tozbikian, G. Lozanski, and M. N. Gurcan (2018). Optimized generation of high-resolution phantom images using cgan: Application to quantification of ki67 breast cancer images. *PLOS ONE 13*(5), 1–12.

Srivastava, A., L. Valkov, C. Russell, M. U. Gutmann, and C. Sutton (2017). Veegan: Reducing mode collapse in gans using implicit variational learning. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett (Eds.), *Advances in Neural Information Processing Systems 30*, pp. 3308–3318. Curran Associates, Inc.

van den Oord, A., S. Dieleman, H. Zen, K. Simonyan, O. Vinyals, A. Graves, N. Kalchbrenner, A. W. Senior, and K. Kavukcuoglu (2016). Wavenet: A generative model for raw audio. *CoRR abs/1609.03499*.

Xiao, Y., E. Decencière, S. Velasco-Forero, H. Burdin, T. Bornschlögl, F. Bernerd, E. Warrick, and T. Baldeweck (2019). A NEW COLOR AUGMENTATION METHOD FOR DEEP LEARNING SEGMENTATION OF HISTOLOGICAL IMAGES. In *2019 IEEE 16th International Symposium on Biomedical Imaging (ISBI)*, Venise, France.

Yi, X., E. Walia, and P. Babyn (2018). Generative adversarial network in medical imaging: A review. *ArXiv abs/1809.07294*.

Zhu, J., T. Park, P. Isola, and A. A. Efros (2017). Unpaired image-to-image translation using cycle-consistent adversarial networks. In *2017 IEEE International Conference on Computer Vision (ICCV)*, pp. 2242–2251.

Appendix 1 : Model description

Layer	Output shape	Number of parameters
input	(1, 256, 256, 2)	0
conv2d	(1, 128, 128, 64)	1664
leaky ReLU	(1, 128, 128, 64)	0
conv2d	(1, 64, 64, 128)	205184
leaky ReLU	(1, 64, 64, 128)	0
conv2d	(1, 32, 32, 256)	819968
leaky ReLU	(1, 32, 32, 256)	0
conv2d	(1, 16, 16, 512)	3278336
leaky ReLU	(1, 16, 16, 512)	0
conv2d	(1, 8, 8, 512)	6555136
leaky ReLU	(1, 8, 8, 512)	0
conv2d	(1, 4, 4, 512)	6555136
leaky ReLU	(1, 4, 4, 512)	0
conv2d	(1, 2, 2, 512)	6555136
leaky ReLU	(1, 2, 2, 512)	0
conv2d	(1, 1, 1, 512)	655513
ReLU	(1, 1, 1, 512)	0
deconv2d	(1, 2, 2, 512)	6555136
dropout	(1, 2, 2, 512)	0
concatenation	(1, 2, 2, 1024)	0
ReLU	(1, 2, 2, 1024)	0
deconv2d	(1, 4, 4, 512)	13108736
dropout	(1, 4, 4, 512)	0
concatenation	(1, 4, 4, 1024)	0
ReLU	(1, 4, 4, 1024)	0
deconv2d	(1, 8, 8, 512)	13108736
dropout	(1, 8, 8, 512)	0
concatenation	(1, 8, 8, 1024)	0
ReLU	(1, 8, 8, 1024)	0
deconv2d	(1, 16, 16, 512)	13108736
concatenation	(1, 2, 2, 1024)	0
ReLU	(1, 16, 16, 1024)	0
deconv2d	(1, 32, 32, 256)	6554368
concatenation	(1, 2, 2, 512)	0
ReLU	(1, 32, 32, 512)	0
deconv2d	(1, 64, 64, 128)	1638784
concatenation	(1, 2, 2, 256)	0
ReLU	(1, 64, 64, 256)	0
deconv2d	(1, 128, 128, 64)	409792
concatenation	(1, 2, 2, 128)	0
ReLU	(1, 128, 128, 128)	0
deconv2d	(1, 256, 256, 3)	9603

TAB. 1 – *Pix2Pix generator model*

Layer	Output shape	Number of parameters
input	(1, 256, 256, 3)	0
conv2d	(1, 128, 128, 64)	4864
leaky ReLU	(1, 128, 128, 64)	0
conv2d	(1, 64, 64, 128)	205184
leaky ReLU	(1, 64, 64, 128)	0
conv2d	(1, 32, 32, 256)	819968
leaky ReLU	(1, 32, 32, 256)	0
conv2d	(1, 32, 32, 512)	3278336
leaky ReLU	(1, 32, 32, 512)	0
reshape	(1, 524288)	0
linear	(1, 1)	0

TAB. 2 – *Pix2Pix discriminator model*

Appendix 2 : Additional results

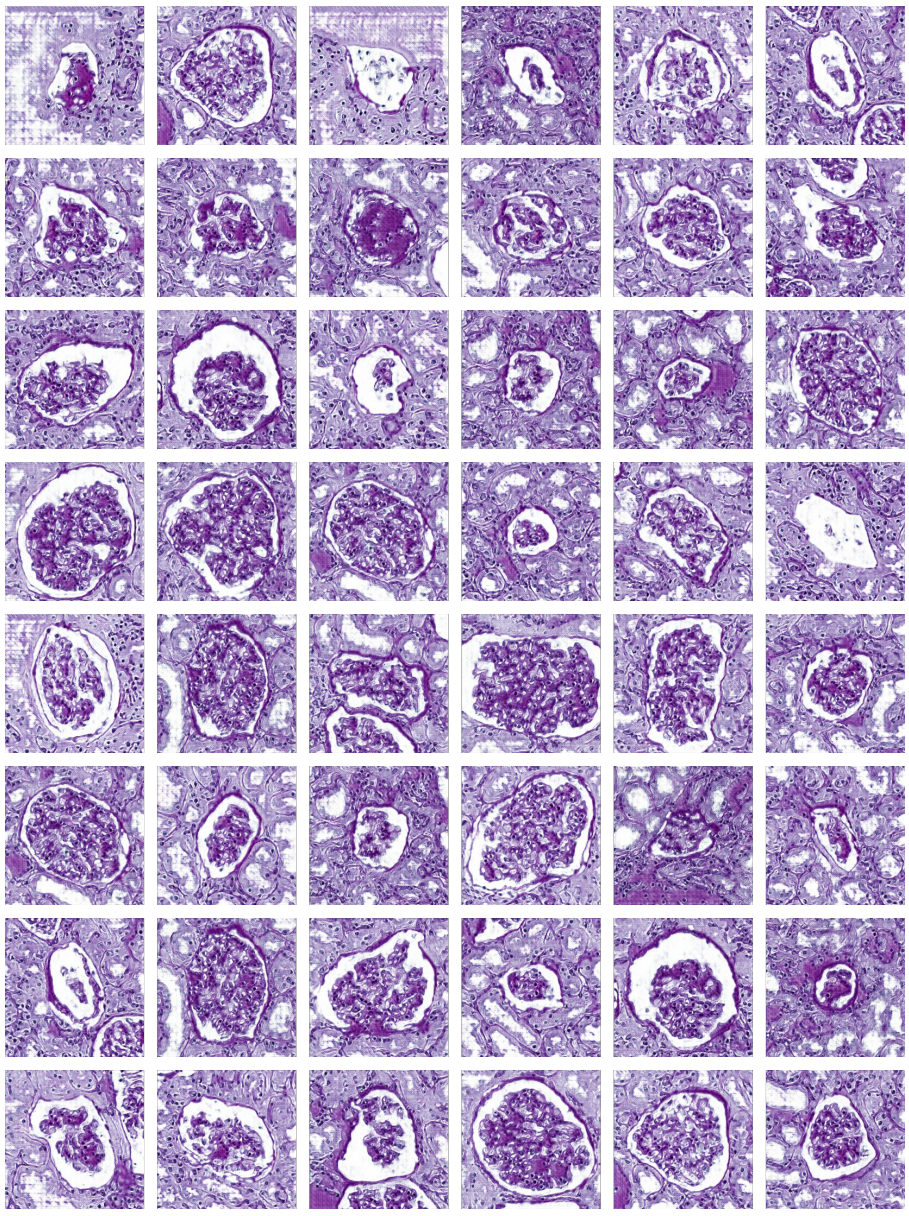


FIG. 8 – *Batch of 48 randomly selected results.*

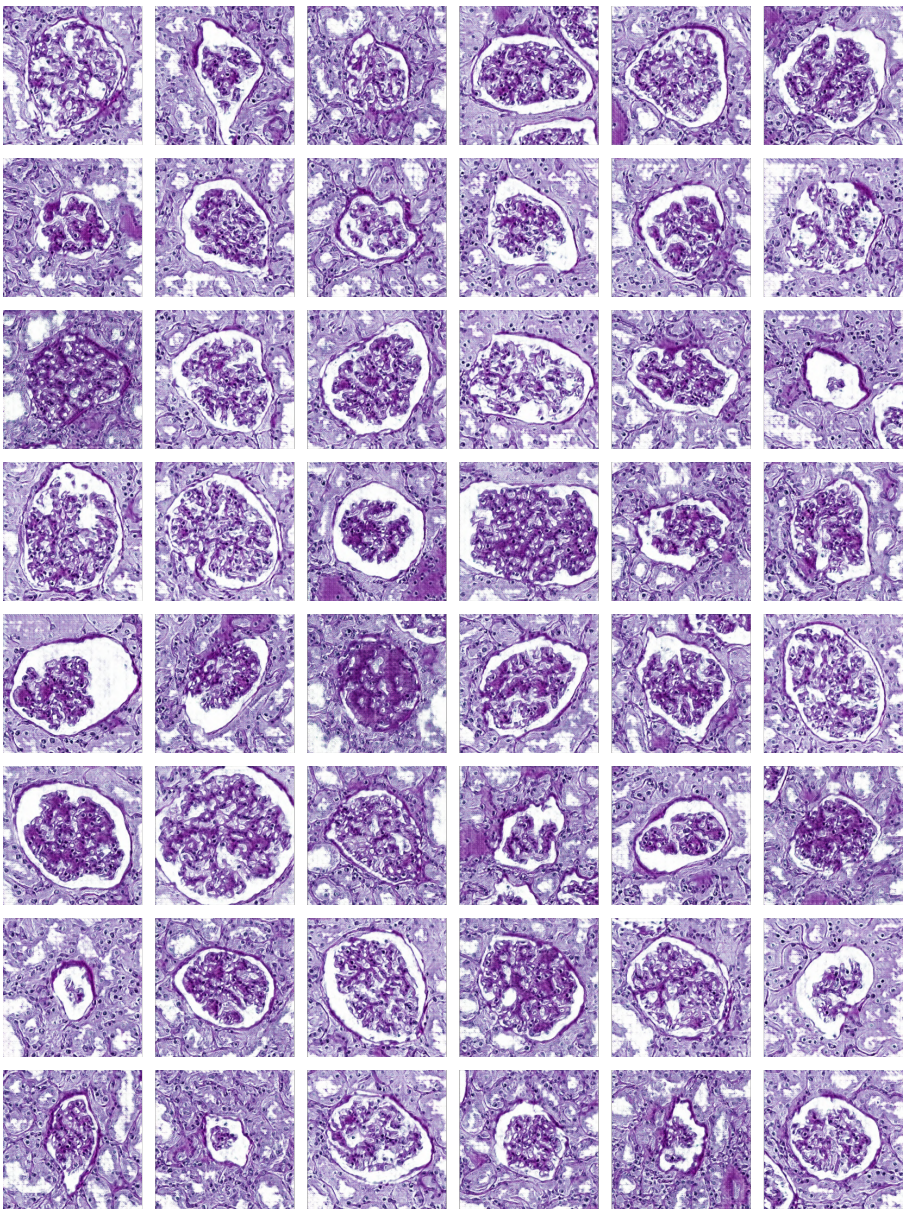


FIG. 9 – *Second batch of 48 randomly selected results.*

Apprentissage par transfert pour la classification de séquences vidéo de mouvements de foule

Mounir Bendali-Braham, Jonathan Weber
Germain Forestier, Lhassane Idoumghar, Pierre-Alain Muller

IRIMAS, Université de Haute-Alsace, 68100 Mulhouse, France
prénom.nom@uha.fr

Résumé. La reconnaissance automatique d'un mouvement de foule, capturé par une caméra de vidéo-protection, peut être d'une aide considérable pour les forces de l'ordre qui ont pour mission d'assurer la sécurité des personnes sur la voie publique. Dans ce cadre, nous proposons d'entraîner un modèle issu de l'architecture *TwoStream Inflated 3D*, pré-entrainé sur les jeux de données sources ImageNet et Kinetics, à classer des séquences vidéo de mouvements de foule issues du jeu de données cible Crowd-11. En comparant nos résultats à l'état-de-l'art, notre modèle supplante son score de précision.

1 Introduction

Afin de gérer un mouvement de foule en amont, les forces de l'ordre peuvent recourir à l'usage des caméras de vidéo-protection (Drews et al., 2010; Sultani et al., 2018; Porikli et al., 2013). L'installation de ces caméras couvre une grande partie de l'espace public (Kritter et al., 2019). Bien que l'un de leurs usages le plus commun est la récupération d'images attestant d'une activité criminelle et leur emploi par la suite à des fins judiciaires, l'usage qui commence à en être fait est l'analyse des comportements de foule afin de prédire des situations anormales (Sultani et al., 2018). Toutefois, malgré l'abondance d'images brutes, provenant des caméras de vidéo-protection, il n'existe pas à ce jour de modèle issu de l'apprentissage profond qui serve dans tous les scénarios possibles de scènes de foule. Ceci est dû à la rareté de données annotées disponibles publiquement (Carreira et Zisserman, 2017).

Récemment, une équipe du CEA (Commissariat à l'énergie atomique et aux énergies alternatives) a créé un jeu de données appelé Crowd-11 (Dupont et al., 2017). Ce jeu de données, de plus de 6000 séquences vidéo, constitue une contribution majeure pour l'analyse du comportement de foule car il décrit une dizaine de comportements observables dans la voie publique.

Dans ce travail, nous appliquons l'apprentissage par transfert pour classer les séquences vidéo de mouvements de foule. Dans ce cadre, notre tâche consiste à étiqueter une vidéo. Pour ce faire, nous affinons un modèle issu de l'architecture *TwoStream Inflated 3D ConvNet* (*TwoStream-I3D*) (Carreira et Zisserman, 2017) pré-entrainé sur les jeux de données ImageNet (Deng et al., 2009) et Kinetics (Kay et al., 2017), sur ce qui a pu être récupéré du jeu de données Crowd-11. Le modèle *TwoStream-I3D* affiné est comparé à un modèle issu de l'architecture *3D Convolutional Networks* (*C3D*) (Tran et al., 2015), pré-entrainé sur le jeu de

données Sports-1m avant d'être affiné sur Crowd-11. La suite du papier est organisée comme suit : dans la Sous-section 2.1, nous parlons brièvement de ce qui se fait dans le domaine de l'analyse des foules. Dans la Sous-section 2.2, nous présentons le jeu de données Crowd-11. Dans la Section 3, nous introduisons l'apprentissage par transfert dans le cadre de la classification de vidéos, et nous présentons les architectures pour lesquelles nous l'avons appliqué. Dans la Section 4, nous présentons les différents modèles que nous avons entrainés, et puis évalués, sur le jeu de données Crowd-11.

2 Contexte

2.1 État de l'art

Depuis plus de deux décennies, l'analyse des foules fait partie de la recherche en vision par ordinateur. Les travaux réalisés dans ce domaine se subdivisent en deux grandes catégories : le calcul des statistiques de foule, et l'analyse des comportements de foule (Zhan et al., 2008; Lamba et Nain, 2017; Grant et Flynn, 2017).

Calcul des statistiques de foule :

— **Comptage du nombre de personnes d'une foule :** comme son nom l'indique, cette branche de l'estimation des statistiques de foule consiste à compter le nombre de personnes constituant la ou les foules d'individus capturées dans une scène (Ranjan et al., 2018).

— **Estimation de la densité des foules :** les travaux qui estiment la densité d'une scène de foule, tel que Xu et al. (2017), peuvent être d'une aide considérable pour les forces de l'ordre ou les organisations de gestion des mouvements de foule.

Analyse des comportements de foule :

— **Analyse des trajectoires :** l'analyse des trajectoires fait partie de ce qui se fait le plus en analyse des comportements de foule (Lu et al., 2017). L'analyse des trajectoires peut aider à détecter des groupes d'individus (Solera et al., 2015), détecter des trajectoires anormales (Coşar et al., 2016), ou prédire l'évolution des trajectoires (Alahi et al., 2016).

— **Reconnaissance des actions de groupes :** suite à une détection des groupes à partir des formations qu'ils observent, certains travaux se penchent sur la reconnaissance des actions menées en groupe (Ibrahim et al., 2016). La détection et l'étude des comportements de groupes font partie des approches mésoscopiques en analyse des foules, car un groupe est à mi-chemin entre l'individu et la foule (Shao et al., 2018).

— **Détection d'anomalies :** souvent considérée comme un sujet à part entière ou couplée avec un autre sous-sujet de l'analyse des foules, l'on peut faire de la détection d'anomalies pour n'importe quelle tâche de l'analyse des foules (Zhou et al., 2018).

L'analyse des foules peut recourir à l'extraction manuelle d'un certain nombre d'indices visuels (Zhan et al., 2008; Lamba et Nain, 2017; Grant et Flynn, 2017; Li et al., 2015). Cette tâche difficile, et sujette à un certain nombre d'omissions, peut être déléguée aux réseaux de neurones profonds qui sont souvent capables de mieux repérer les indices visuels significatifs (Tripathi et al., 2018).

2.2 Le jeu de données Crowd-11

Créé par une équipe du CEA-LIST (Dupont et al., 2017), ce jeu de données récent et totalement annoté, contient plus de 6000 séquences vidéo. Les séquences vidéo disposent de résolutions variables allant de 220×400 à 700×1250, et proviennent à la base d'une multitude de sources pré-existantes. Les vidéos sont classées en 11 catégories.

Dans ce qui suit, nous décrivons les comportements correspondants aux 11 classes contenues dans le jeu de données Crowd-11 :

0. **Gas Free :** Individus marchant dans toutes les directions sans rencontrer d'obstacles.
1. **Gas Jammed :** Foule congestionnée.
2. **Laminar Flow :** Individus marchant dans une seule direction.
3. **Turbulent Flow :** Foule marchant dans une seule direction perturbée par un individu marchant à contresens.
4. **Crossing Flows :** Deux foules qui se croisent.
5. **Merging Flows :** Deux foules qui convergent.
6. **Diverging Flow :** Une foule qui se subdivisent en deux foules.
7. **Static Calm :** Une foule d'individus statiques et calmes.
8. **Static Agitated :** Une foule d'individus statiques et agités.
9. **Interacting Crowd :** Deux foules d'individus qui s'opposent. Cette classe contient des scènes de conflits.
10. **No Crowd :** Aucune présence humaine dans la scène.

Les vidéos proviennent principalement de trois sites d'hébergement de vidéos et qui sont Youtube [1], Pond5 [2], et GettyImages [3].

Le reste provient des jeux de données suivants : UMN SocialForce, AgoraSet, PETS-2009, Violent-Flows, Hockey Fights and Movies, WWW Crowd, CUHK Crowd, et Shanghai WorldExpo'10 Crowd.

La plupart de ces jeux de données sont publiquement disponibles et facilement accessibles. Toutefois, certains ne le sont plus tels que WWW Crowd, CUHK Crowd, et Shanghai WorldExpo'10 Crowd. À cause de cela, nous n'avons pas pu récupérer le jeu de données Crowd-11 dans sa totalité. Ce qui a pu être récupéré représente approximativement 90% du jeu de données initial. Une estimation de la répartition des séquences récupérées par classe permet de constater qu'il n'y a pas eu une perte majeure par rapport au jeu de données initial, comme nous pouvons l'observer dans le tableau 1.

3 Apprentissage par transfert

Le but de l'apprentissage par transfert est de transmettre les connaissances apprises par un modèle à partir d'un jeu de données source vers un jeu de données cible (Pan et Yang, 2010). Dans des travaux récents, l'apprentissage par transfert pour la classification des clips vidéo a été appliqué pour la reconnaissance d'actions dans des scènes individuelles (Carreira et Zisserman, 2017; Tran et al., 2015). Dans cette situation, l'objectif est de transférer les connaissances

1. Youtube : https://www.youtube.com/
2. Pond5 : https://www.pond5.com/
3. GettyImages : https://www.gettyimages.fr/

Étiquette	Nom de la classe	#vidéos (qté originale)	#vidéos récupérées
0	Gas Free	529	477
1	Gas Jammed	520	508
2	Laminar Flow	1304	1189
3	Turbulent Flow	892	862
4	Crossing Flows	763	717
5	Merging Flow	295	267
6	Diverging Flow	184	189
7	Static Calm	737	686
8	Static Agitated	410	351
9	Interacting Crowd	248	153
10	No Crowd	390	370

TAB. 1 – *Tableau comparatif entre le nombre de vidéos récupérées et le nombre de vidéos original par classe pour le jeu de données Crowd-11.*

acquises d'un jeu de données source vers un jeu de données cible appartenant au même domaine. Dupont et al. (2017) a appliqué cette opération en transférant les connaissances qu'un modèle a apprises d'un jeu de données source de reconnaissance d'actions à un jeu de données cible illustrant des mouvements de foule. Afin de surpasser les problèmes liés à l'apprentissage par transfert, en passant d'un domaine à un autre, nous appliquons l'apprentissage par transfert en lançant la procédure d'ajustement sur un nombre important d'époques (entre 30 et 40).

3.1 Architectures implémentées

Nous avons sélectionné trois modèles à affiner de deux architectures : *C3D* et *TwoStream-I3D*. Le choix de l'architecture *TwoStream-I3D* est principalement motivé par les bons résultats obtenus par ses modèles par rapport aux modèles *C3D* lorsqu'ils effectuent la reconnaissance d'actions dans des scènes individuelles à partir des jeux de données UCF-101 et HMDB-51 (Carreira et Zisserman, 2017). L'équipe du CEA ayant obtenu les meilleurs résultats avec l'architecture *C3D*, son choix dans nos expériences est naturel car nous n'avons pas été en mesure de récupérer le jeu de données Crowd-11 dans son intégralité. Un modèle *C3D* pré-entrainé sur Sports-1m a obtenu ses meilleurs résultats en classant les vidéos de Crowd-11 (Dupont et al., 2017). Ce modèle représente donc pour nous le résultat de base à améliorer au cours de nos expériences. Plus de détails sur les architectures implémentées peuvent être trouvés dans ce papier Bendali-Braham et al. (2019).

3.1.1 Réseaux de neurones *3D Convolutional Neural Network*

Nous avons décidé de ré-implémenter une version des réseaux de neurones convolutifs 3D correspondant à l'architecture décrite dans Tran et al. (2015).

Comme nous l'avons déjà mentionné, l'équipe du CEA obtient sa meilleure performance avec *C3D* après avoir pré-entrainé le modèle sur le jeu de données Sports-1m (Karpathy et al., 2014).

3.1.2 Réseaux de neurones *Two-Stream Inflated 3D*

Carreira et Zisserman proposent l'architecture *Two-Stream Inflated 3D Neural Network* (Carreira et Zisserman, 2017). Cette architecture a été utilisée pour apprendre la reconnaissance d'actions dans des scènes individuelles, où elle a obtenu de très bons résultats par rapport à *C3D*. Nous l'utilisons pour apprendre à reconnaître les mouvements de foule.

Carreira et Zisserman ont pré-entrainé un modèle *TwoStream-I3D* sur ImageNet (Deng et al., 2009) et Kinetics (Kay et al., 2017). En testant ce modèle sur les jeux de données UCF-101 et HMDB-51, ils ont considérablement dépassé les performances des modèles *C3D* qui ont été pré-entraînés sur Sports-1m (Carreira et Zisserman, 2017). Dans notre cas, nous avons décidé de transférer les connaissances acquises d'une branche RVB de l'architecture *I3D* sur les jeux de données sources ImageNet et Kinetics vers le jeu de données cible Crowd-11. Nous avons fait la même chose pour le modèle *TwoStream-I3D* en transférant les connaissances apprises de la branche RVB et de la branche flux optique de l'architecture au jeu de données cible. Nous avons extrait le flux optique de chaque clip vidéo en utilisant l'algorithme TV-L1 (Zach et al., 2007).

4 Expérimentations sur Crowd-11

Dans les expériences que nous avons réalisées, nous avons décidé pour chaque architecture d'affiner un modèle pré-entrainé et d'entraîner un modèle à partir de zéro sur Crowd-11. Dans le cas du modèle pré-entraîné *C3D*, le pré-entraînement a été réalisé sur le jeu de données Sports-1m. Dans le cas des modèles *I3D*/*TwoStream-I3D*, le pré-entraînement a été effectué sur ImageNet, puis sur la version RVB de Kinetics pour la branche RVB, et la version flux optique de Kinetics pour la branche du flux optique.

En prenant en compte les paramètres d'apprentissages trouvés sur Tran et al. (2015) et Carreira et Zisserman (2017) respectivement pour les modèles *C3D* et *TwoStream-I3D*, nous avons choisi d'appliquer la descente du gradient stochastique (SGD) comme fonction d'optimisation, et avons fixé le taux d'apprentissage initial à $0,003$. La fonction de perte choisie pour ces expériences est l'entropie croisée catégorielle. Afin d'être très proche des hyper-paramètres utilisés pour *C3D* par Dupont et al. (2017), nous avons divisé le taux d'apprentissage par 10 toutes les 4 époques. Cependant, nous n'avons pas reproduit cette opération lors de l'entraînement des modèles *I3D* et *TwoStream-I3D*. Pour ces derniers, nous avons choisi de diviser le taux d'apprentissage par 10 uniquement si la valeur de l'erreur augmente sur l'ensemble de validation. Pendant la phase d'entraînement, le nombre d'époques a été fixé à 40 pour les modèles *C3D* et à 30 pour les autres, afin de maximiser les chances des modèles *C3D* d'obtenir de meilleurs scores. Un modèle est enregistré à la fin de chaque époque. À la fin de la phase d'apprentissage, nous avons choisi de sauvegarder le modèle minimisant la fonction de perte lors de la phase de validation. Lors de l'affinement des modèles, nous avons décidé de ne geler aucune couche des réseaux, car les jeux de données sources sur lesquels nos modèles ont été pré-entraînés diffèrent beaucoup de ceux que nous voulons apprendre. Par conséquent, nous avons décidé de rétropropager la mise à jour des poids des réseaux sur l'ensemble des architectures des réseaux lors des phases d'apprentissage. Contrairement à Dupont et al. nous n'avons pas appliqué de méthodes d'augmentation des données pour entraîner nos modèles. Sachant que l'augmentation des données est une méthode de régularisation, nous voulons voir si nos mo-

Modèle	Condition d'entraînement	Précision
Notre *C3D*	Sans pré-entraînement	31.88%
C3D Dupont et al.	Sans pré-entraînement	46.9%
Notre *C3D*	Pré-entrainé	58.29%
C3D Dupont et al.	Pré-entrainé	61.6%

TAB. 2 – *Comparaison entre notre version de C3D et celle de Dupont et al. (2017)*

Architecture	Condition d'entraînement	Moyenne	Min	Max
I3D	Sans pré-entraînement	47.01%	40%	53.36%
C3D	Sans pré-entraînement	31.88%	28.82%	36.43%
TwoStream-I3D	Sans pré-entraînement	47.85%	43.91%	52.42%
I3D	Pré-entrainé	58.97%	56.33%	60.17%
C3D	Pré-entrainé	58.29%	57.19%	60%
TwoStream-I3D	Pré-entrainé	**68.2%**	**66.01%**	**70.34%**

TAB. 3 – *Précision obtenue à la suite de la validation croisée avec K=5.*

dèles ne souffrent pas d'un sur-apprentissage sur la version basique du jeu de données (Dvornik et al., 2018). Par ailleurs, nous voulons déterminer quelles classes nuisent à l'apprentissage de nos modèles, sans parer à ce problème en utilisant l'augmentation des données. Comme nous comptons tester plusieurs méthodes d'augmentation des données vidéo, nous préférons nous consacrer à ce problème ultérieurement.

4.1 Validation croisée à 5 échantillons

Notre version de Crowd-11 est composée de 1641 scènes. Ces scènes ont été divisées en 5769 clips vidéo. Pour éviter que des échantillons se chevauchent, nous avons décidé de conserver tous les clips d'une même scène dans un même échantillon. Lorsque nous sélectionnons une scène à ajouter à un échantillon, notre sélection fait en sorte de maintenir une similarité approximative des échantillons en termes de nombre de clips par classe. Pour entraîner ou ajuster nos modèles, nous avons divisé le jeu de données en 5 échantillons, et avons décidé d'appliquer la validation croisée 5 fois. Pour chaque itération de la validation croisée, nous avons choisi 3 échantillons pour constituer l'ensemble d'apprentissage, un pour constituer l'ensemble de validation et un dernier pour l'ensemble de test. À chaque itération de la validation croisée, l'ensemble de test change. L'ensemble de validation est choisi de manière aléatoire parmi les 4 échantillons restants.

Comme nous avons appliqué une validation croisée 5 fois pour chacun de nos trois modèles en prenant en compte les deux conditions d'entraînement : l'entraînement à partir de zéro, et l'ajustement d'un modèle pré-entraîné ; nous avons lancé 30 procédures d'entraînement [4].

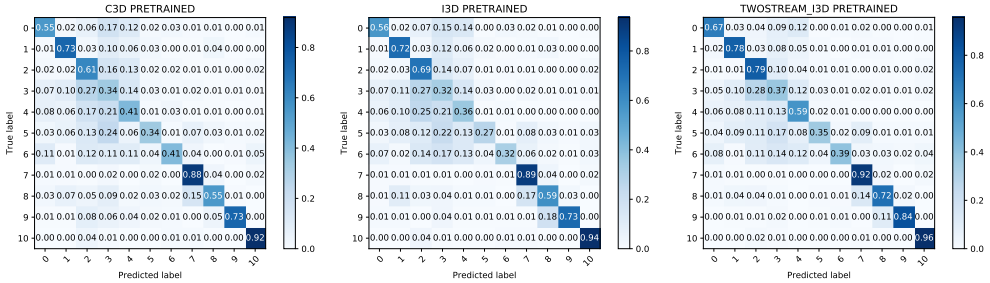


FIG. 1 – *Matrices de confusion globales, des modèles pré-entrainés, calculées à la suite de la validation croisée à 5 échantillons*

4.2 Discussion des résultats obtenus

Selon les résultats affichés sur le tableau 2, nous observons que le modèle *C3D* entrainé à partir de zéro n'est pas aussi performant que le modèle entraîné par Dupont et al. (2017). Cela peut avoir plusieurs raisons : une possible différence entre les hyperparamètres que nous utilisons et ceux qui sont utilisés par les auteurs de Crowd-11 pour l'entraînement de leur modèle, la différence entre nos deux jeux de données, et le fait que nous n'ayons pas recours à l'augmentation des données vidéo. Selon les résultats affichés dans le tableau 3, nous constatons que les modèles *C3D* et *I3D* obtiennent des résultats presque identiques lors de la classification des clips vidéo lors de phase de test. *C3D* n'est dépassé que d'environs $0,6\%$ de précision par le modèle *I3D*. Cette légère différence de performances peut s'expliquer par le fait que l'architecture *C3D* doit entraîner 78 millions de paramètres, tandis que l'architecture *I3D* compte 12 millions de paramètres ainsi qu'une structure profonde. De plus, nous observons que le modèle *TwoStream-I3D* arrive bien à tirer profit du flux optique lors de l'affinement. Cela n'est pas le cas lorsqu'il est entraîné à partir de zéro. Globalement, les modèles *TwoStream-I3D* obtiennent les meilleurs scores.

À partir des matrices de confusion affichées sur la figure 1, nous observons que chaque modèle éprouve des difficultés face aux mêmes classes indicées de 3 à 6, qui sont respectivement : *Turbulent Flow*, *Crossing Flows*, *Converging Flow* et *Diverging Flow*. Nous constatons, également, que les clips appartenant à ces classes, y compris la classe *Laminar Flow*, sont fréquemment confondus. Alors que la classe *Laminar Flow* n'est pas une grande source de confusion, car la foule y suit une direction unique, les multiples transitions clés observables dans les quatre autres classes peuvent perturber la décision du classifieur. Par exemple, nous observons que la classe *Merging Flow* n'est pas confondue avec la classe *Diverging Flow*, ce qui montre que le classifieur apprend bien à différencier entre ces deux comportements. Cependant, ces deux classes sont fréquemment confondues avec la classe *Crossing Flows*. Lorsqu'une foule se croise avec une autre, des comportements de convergence et de divergence sont observés. De plus, alors que *Crossing Flows* est composée par ≈ 850 clips, les classes *Merging Flow* et *Diverging Flow* sont composées par ≈ 200 clips chacune (comme indiqué dans le tableau 1). Cette situation peut amener deux classes à être englouties par une classe plus globale, telle que la classe *Crossing Flows*.

4. Le code source de ce travail est disponible ici :
https://github.com/MounirB/Crowd-movements-classification

5 Conclusion et perspectives

Dans ce travail, nous avons étudié la capacité du réseau *TwoStream Inflated 3D* à tirer profit de son pré-entraînement sur les jeux de données ImageNet et Kinetics pour la classification des comportements de foule sur le jeu de données Crowd-11. Après avoir transféré les connaissances apprises des jeux de données sources vers le jeu de données cible, le modèle produit surpasse l'état-de-l'art, sur Crowd-11, avec une marge conséquente de $\approx 10\%$ de précision. Cependant, du fait du score qu'il a obtenu, le classifieur ne peut pas être, pour l'instant, considéré comme un outil de classification précis pour la gestion des mouvements de foule. Sur la base des résultats obtenus, nous avons l'intention de voir dans quelle mesure nous pouvons les améliorer en testant les méthodes suivantes :

- Appliquer l'augmentation des données vidéo ;
- Remédier aux classes défectueuses du jeu de données Crowd-11 en leur ajoutant des clips vidéo ;
- Tester des modèles issus des architectures *Temporal 3D ConvNets (T3D)* (Diba et al., 2017) et *ActionVLAD* (Girdhar et al., 2017), car les modèles de ces architectures obtiennent des scores supérieurs à 90% de précision sur les jeux de données UCF-101 et HMDB-51 ;
- Modifier l'architecture *Inflated 3D* via :
  - L'ajout de nouveaux modules Inception ;
  - L'hybridation de l'architecture *I3D* avec l'une des deux architectures *T3D* ou *ActionVLAD*.
- Prendre en compte des entrées d'une étape de prétraitement, comme l'extraction des trajectoires denses **(iDT)** (Wang et Schmid, 2013), avant de procéder à l'entraînement des modèles.

Remerciements

Les auteurs tiennent à remercier NVIDIA Corporation pour nous avoir fourni des GPUs et le Mésocentre de Strasbourg pour leur avoir permis de mener des calculs sur le cluster de GPUs. Ce travail a été soutenu par le projet ANR OPMoPS (subvention ANR-16-SEBM-0004) financé par l'Agence nationale de la recherche.

Références

Alahi, A., K. Goel, V. Ramanathan, A. Robicquet, L. Fei-Fei, et S. Savarese (2016). Social lstm : Human trajectory prediction in crowded spaces. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 961–971.

Bendali-Braham, M., J. Weber, G. Forestier, L. Idoumghar, et P.-A. Muller (2019). Transfer learning for the classification of video-recorded crowd movements. In *IEEE International Symposium on Image and Signal Processing and Analysis*, pp. 271–276.

Carreira, J. et A. Zisserman (2017). Quo vadis, action recognition ? a new model and the kinetics dataset. In *proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 6299–6308.

Coşar, S., G. Donatiello, V. Bogorny, C. Garate, L. O. Alvares, et F. Brémond (2016). Toward abnormal trajectory and event detection in video surveillance. *IEEE Transactions on Circuits and Systems for Video Technology 27*(3), 683–695.

Deng, J., W. Dong, R. Socher, L.-J. Li, K. Li, et L. Fei-Fei (2009). Imagenet : A large-scale hierarchical image database. In *IEEE conference on computer vision and pattern recognition*, pp. 248–255.

Diba, A., M. Fayyaz, V. Sharma, A. H. Karami, M. M. Arzani, R. Yousefzadeh, et L. Van Gool (2017). Temporal 3d convnets : New architecture and transfer learning for video classification. *ArXiv*.

Drews, P., J. Quintas, J. Dias, M. Andersson, J. Nygårds, et J. Rydell (2010). Crowd behavior analysis under cameras network fusion using probabilistic methods. In *International Conference on Information Fusion*, pp. 1–8.

Dupont, C., L. Tobias, et B. Luvison (2017). Crowd-11 : A dataset for fine grained crowd behaviour analysis. In *IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, Volume 2017-July, Honolulu, United States, pp. 2184–2191.

Dvornik, N., J. Mairal, et C. Schmid (2018). On the importance of visual context for data augmentation in scene understanding. *ArXiv*.

Girdhar, R., D. Ramanan, A. Gupta, J. Sivic, et B. Russell (2017). Actionvlad : Learning spatio-temporal aggregation for action classification. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 971–980.

Grant, J. M. et P. J. Flynn (2017). Crowd scene understanding from video : a survey. *ACM Transactions on Multimedia Computing, Communications, and Applications (TOMM) 13*(2), 19.

Ibrahim, M. S., S. Muralidharan, Z. Deng, A. Vahdat, et G. Mori (2016). A hierarchical deep temporal model for group activity recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1971–1980.

Karpathy, A., G. Toderici, S. Shetty, T. Leung, R. Sukthankar, et L. Fei-Fei (2014). Large-scale video classification with convolutional neural networks. In *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1725–1732.

Kay, W., J. Carreira, K. Simonyan, B. Zhang, C. Hillier, S. Vijayanarasimhan, F. Viola, T. Green, T. Back, P. Natsev, et al. (2017). The kinetics human action video dataset. *ArXiv*.

Kritter, J., M. Brévilliers, J. Lepagnot, et L. Idoumghar (2019). On the optimal placement of cameras for surveillance and the underlying set cover problem. *Applied Soft Computing 74*, 133 – 153.

Lamba, S. et N. Nain (2017). Crowd monitoring and classification : a survey. In *Advances in Computer and Computational Sciences*, pp. 21–31. Springer.

Li, T., H. Chang, M. Wang, B. Ni, R. Hong, et S. Yan (2015). Crowded scene analysis : A survey. *IEEE transactions on circuits and systems for video technology 25*(3), 367–386.

Lu, W., X. Wei, W. Xing, et W. Liu (2017). Trajectory-based motion pattern analysis of crowds. *Neurocomputing 247*, 213–223.

Pan, S. J. et Q. Yang (2010). A survey on transfer learning. *IEEE Transactions on knowledge and data engineering 22*(10), 1345–1359.

Porikli, F., F. Bremond, S. L. Dockstader, J. Ferryman, A. Hoogs, B. C. Lovell, S. Pankanti, B. Rinner, P. Tu, et P. L. Venetianer (2013). Video surveillance : past, present, and now the future [dsp forum]. *IEEE Signal Processing Magazine 30*(3), 190–198.

Ranjan, V., H. Le, et M. Hoai (2018). Iterative crowd counting. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pp. 270–285.

Shao, J., N. Dong, et Q. Zhao (2018). A real-time algorithm for small group detection in medium density crowds. *Pattern Recognition and Image Analysis 28*(2), 282–287.

Solera, F., S. Calderara, et R. Cucchiara (2015). Socially constrained structural learning for groups detection in crowd. *IEEE transactions on pattern analysis and machine intelligence 38*(5), 995–1008.

Sultani, W., C. Chen, et M. Shah (2018). Real-world anomaly detection in surveillance videos. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 6479–6488.

Tran, D., L. Bourdev, R. Fergus, L. Torresani, et M. Paluri (2015). Learning spatiotemporal features with 3d convolutional networks. In *Proceedings of the IEEE international conference on computer vision*, pp. 4489–4497.

Tripathi, G., K. Singh, et D. K. Vishwakarma (2018). Convolutional neural networks for crowd behaviour analysis : a survey. *The Visual Computer*, 1–24.

Wang, H. et C. Schmid (2013). Action recognition with improved trajectories. In *Proceedings of the IEEE international conference on computer vision*, pp. 3551–3558.

Xu, X., D. Zhang, et H. Zheng (2017). Crowd density estimation of scenic spots based on multifeature ensemble learning. *Journal of Electrical and Computer Engineering 2017*.

Zach, C., T. Pock, et H. Bischof (2007). A duality based approach for realtime tv-l1 optical flow. In *Pattern Recognition*, pp. 214–223.

Zhan, B., D. N. Monekosso, P. Remagnino, S. A. Velastin, et L. Q. Xu (2008). Crowd analysis : A survey. *Machine Vision and Applications 19*(5-6), 345–357.

Zhou, P., Q. Ding, H. Luo, et X. Hou (2018). Violence detection in surveillance video using low-level features. *PLoS one 13*(10), e0203668.

Summary

The automatic recognition of a crowd movement captured by a CCTV camera can be of considerable help to security forces whose mission is to ensure the safety of people on the public area. In this context, we propose to fine-tune a model from the TwoStream Inflated 3D architecture, pre-trained on the ImageNet and the Kinetics source datasets, to classify video sequences of crowd movements from the Crowd-11 target dataset. The evaluation of our model demonstrates its superiority over the state-of-the-art in terms of classification accuracy.

Classification de séries temporelles d'images basée sur une représentation planaire spatio-temporelle

Mohamed Chelali*, Camille Kurtz*, Anne Puissant**, Nicole Vincent*

*LIPADE, Université de Paris, Paris, France
firstname.lastname@u-paris.fr
**LIVE, Université de Strasbourg, Strasbourg, France
firstname.lastname@unistra.fr

Résumé. Les séries temporelles d'images, telles que les séquences fonctionnelles IRM ou les séries temporelles d'images satellites (STIS), fournissent des informations précieuses pour l'analyse automatique de motifs complexes dans le temps. Un problème majeur lors de l'analyse de telles données est de considérer à la fois leurs dimensions temporelle et spatiale. Dans cet article, nous présentons une nouvelle représentation des données qui rend l'étude des séries temporelles d'images compatible avec un modèle d'apprentissage profond classique, tel que les réseaux de neurones à convolution 2D (CNN). L'approche proposée est basée sur une nouvelle représentation plane de la série temporelle d'images qui convertit les données $2D+t$ en images $2D$ sans perdre trop d'informations spatiales ou temporelles. Ce faisant, CNN peut apprendre en même temps les paramètres des filtres $2D$ impliquant des connaissances temporelles et spatiales. Les résultats préliminaires dans le domaine de la télédétection soulignent la capacité de notre approche à discriminer des classes complexes de couverture du sol (en agriculture) à partir d'une STIS.

1 Introduction

Les séries temporelles d'images sont quotidiennement produites par divers capteurs tels que l'IRM (imagerie fonctionnelle), les satellites, les drones ou les caméras classiques observant des classes particulières d'occupation du sol conduisant à une grande quantité d'images ($2D+t$). Dans le contexte de l'observation de la Terre, de nouvelles constellations de satellites acquièrent des images de haute résolution spatiale, spectrale et temporelle dans le monde entier. Par exemple, la constellation Sentinel-2 produit des séries temporelles d'images satellitaires (STIS) avec une durée de re-visite de 5 jours et une résolution spatiale de 10 à 20 mètres.

Parmi les applications potentielles des STIS, on peut citer la cartographie de la couverture terrestre (e.g. les zones agricoles, les zones urbaines) et l'identification de changements d'occupation des sols (e.g. l'urbanisation, la déforestation). La disponibilité croissante de ces données temporelles permet de produire et de mettre à jour des cartes précises de la couverture terrestre d'un territoire (Inglada et al., 2017). Afin de gérer efficacement l'énorme quantité

de données générée par ces nouveaux capteurs, des méthodes adaptées à l'analyse des STIS doivent être développées. Ces méthodes devraient permettre à l'utilisateur final d'obtenir des résultats satisfaisants avec un minimum de temps et d'efforts.

Un problème majeur lors de l'analyse des séries temporelles d'images est de prendre en compte simultanément les dimensions temporelle et spatiale du cube de données $2D+t$. La prise en compte simultanée de ces deux aspects peut, par exemple, faciliter la distinction entre différentes classes complexes de couverture agricole (par exemple, les vergers, les prairies) à partir des STIS. Cet article se concentre sur ce problème spécifique. Pour le traiter, nous définissons une nouvelle représentation spatio-temporelle de séries temporelles d'images qui permet de bénéficier du cadre classique de l'apprentissage profond (initialement proposé pour les images $2D$). Notre contribution principale est la proposition d'une stratégie pour représenter les données $2D+t$ sous forme d'images $2D$ sans perdre trop d'informations spatiales ou temporelles. Ce faisant, les réseaux de neurones convolutionnels (CNN) peuvent apprendre des filtres $2D$ impliquant à la fois des informations temporelles et spatiales. Ici, nous n'avons pas pour objectif de produire des cartes temporelles de la couverture terrestre ni d'étudier les changements d'occupation des sols, mais notre objectif est de cartographier des classes complexes de couverture terrestre sujettes aux confusions lorsqu'une seule image est employée.

Cet article est organisé comme suit. La section 2 rappelle certaines méthodes de l'état de l'art dédiées à l'analyse des STIS. La section 3 présente notre représentation spatio-temporelle pour l'analyse de STIS basée sur les CNN. La section 4 décrit les expériences liées à la classification de parcelles agricoles dans le domaine de la télédétection. La section 5 dresse un bilan et quelques perspectives de recherche.

2 Méthodes de l'état de l'art

Les STIS permettent l'observation et l'analyse de phénomènes terrestres avec une large gamme d'applications telles que l'étude de l'occupation du sol ou même la cartographie des dommages suite à une catastrophe. Ces changements peuvent être de différents types, origines et durées. Pour une étude détaillée, voir (Coppin et al., 2004).

Les méthodes pionnières d'analyse des STIS fonctionnaient sur des images simples ou des piles d'images. Sur chaque image, les différentes mesures par pixel étaient considérées comme des caractéristiques indépendantes et impliquées dans les procédures classiques basées sur l'apprentissage automatique. Dans de telles approches, la date des mesures était ignorée dans l'espace des caractéristiques. L'analyse bi-temporelle a ensuite permis de localiser et d'étudier les changements intervenant entre deux observations (Bruzzone et Prieto, 2000).

Une autre catégorie d'approches était directement conçue pour traiter les séries temporelles d'images. La plupart d'entre elles sont basées sur des approches de classification multi-dates comme l'analyse de trajectoires radiométriques (Verbesselt et al., 2010). Ces approches exploitent la notion selon laquelle la couverture du sol peut varier dans le temps (en raison des saisons, de l'évolution de la végétation (Senf et al., 2015)) et prennent en compte l'ordre des mesures à l'aide de méthodes d'analyse de séries temporelles (Bagnall et al., 2017). Chaque pixel est considéré comme une série de mesures ordonnées dans le temps (et alignées), et les modifications des mesures dans le temps sont analysées pour rechercher des motifs (temporels).

En ce qui concerne le type de caractéristiques, les approches dans le domaine fréquentiel incluent l'analyse spectrale, l'analyse d'ondelettes (Andres et al., 1994), tandis que les approches dans le domaine temporel impliquent des analyses de corrélation. En ce qui concerne la méthode de classification, la méthode classique consiste à mesurer la similarité entre un échantillon entrant et l'ensemble d'apprentissage, puis attribuer l'étiquette de la classe la plus similaire. Pour ce faire on peut utiliser, par exemple, la distance euclidienne basée sur un algorithme de plus plus proche voisin ou une méthode de distance élastique comme DTW (Petitjean et al., 2012a). Certaines méthodes proposent d'abord une projection de la STIS dans un nouvel espace, plus riche, afin d'en extraire des caractéristiques discriminantes (Petitjean et al., 2012b; Chelali et al., 2019) et la classification est réalisée dans ce nouvel espace.

Plus récemment, des approches d'apprentissage profond ont également été envisagées pour classifier les images de télédétection et générer des cartes d'occupation du sol. Dans de nombreux travaux, les réseaux de neurones convolutionnels (CNN) sont pris en compte, traitant généralement le domaine spatial des données en appliquant des convolutions $2D$ (Huang et al., 2018). Lorsqu'il s'agit de séries d'images temporelles, les convolutions sont souvent appliquées dans le domaine temporel (Pelletier et al., 2019). D'autres types d'architectures conçues pour les données temporelles sont les réseaux de neurones récurrents (RNN), comme les LSTM, utilisés avec succès dans (Ienco et al., 2017). Dans ce contexte, les approches d'apprentissage profond surpassent les algorithmes de classification traditionnels tels que les Random Forest (Ismail Fawaz et al., 2019), mais elles ne tiennent pas directement compte de la dimension spatiale des données car elles considèrent les pixels de manière indépendante. Quelques tentatives ont été réalisées pour considérer à la fois les dimensions temporelle et spatiale du cube $2D+t$ (Di Mauro et al., 2017). Une stratégie commune consiste à créer deux modèles (un pour la dimension spatiale et un pour la dimension temporelle), puis de fusionner leurs résultats au niveau de la décision. Dans le domaine de l'analyse vidéo, les caractéristiques spatio-temporelles sont apprises à l'aide de convolutions $3D$ (Tran et al., 2015), mais une telle stratégie nécessite l'apprentissage d'un nombre important de paramètres.

Dans cet article, notre stratégie consiste à classer une STIS en utilisant un modèle classique de CNN $2D$, mais nous proposons une nouvelle représentation des séries temporelles d'images intégrant simultanément les dimensions temporelle et spatiale des données. Nous proposons plusieurs représentations basées sur des stratégies variées pour prendre en compte des variations locales de pixels de manières différentes. Le CNN apprend simultanément avec des convolutions $2D$ des informations temporelles et spatiales.

3 Approche proposée

Cette section présente notre méthode dédiée à la classification des séries temporelles d'images basée sur une représentation planaire spatio-temporelle. Ce travail a fait l'objet d'une publication en conférence internationale (Chelali et al., 2020). Après avoir fourni une vue d'ensemble de la chaîne de traitement du processus global, nous détaillerons les différentes étapes de la méthode.

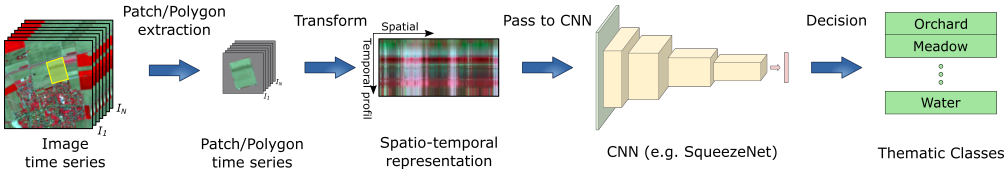


FIG. 1: Organigramme de notre méthode de classification de STIS basée sur une représentation planaire spatio-temporelle.

3.1 Chaîne de traitement

La méthode proposée repose sur l'utilisation d'une architecture classique de réseau de neurones profond. Mais l'entrée n'est pas une structure $3D$ comme dans (Tran et al., 2015) ni une structure $1D$ comme dans (Pelletier et al., 2019), approches fréquentes pour les méthodes de l'état de l'art qui étudient les séries temporelles associées à chaque pixel. Dans notre cas, nous proposons de considérer les pixels d'une région d'intérêt (par exemple, un patch d'image ou un polygone) dans son ensemble et d'appliquer d'abord une transformation de ces données $2D+t$ fournissant une image $2D$ (structure planaire) contenant toutes les données spatio-temporelles. Cela correspond à la partie gauche de l'organigramme présenté dans la Figure 1. Une telle structure est ensuite transférée en tant qu'entrée d'un réseau de neurones classique pour permettre la classification. Le réseau peut être conçu pour apprendre les étiquettes à partir des informations spatiales et temporelles contenues dans les données. La partie droite de l'organigramme illustré en Figure 1 illustre ce processus.

3.2 Représentation planaires des données : du $2D+t$ au $2D$

Afin de réduire la complexité de la structure de données, nous proposons de transformer la représentation spatiale des pixels en une structure $1D$. Initialement, un pixel est défini par sa position (un couple d'entiers) dans une image de hauteur $\mathbb{H}$ et de largeur $\mathbb{W}$. Maintenant, il sera défini par un seul entier donné par un index spécifiant la position du pixel dans un chemin couvrant la région d'intérêt. La fonction $\mathfrak{R}$

$$\begin{aligned} \mathfrak{R} : [1,\mathbb{W}] \times [1,\mathbb{H}] &\rightarrow & [1,\mathbb{W}\times\mathbb{H}] \\ (x,y) &\mapsto & i = \mathfrak{R}(x,y) \end{aligned}$$

associe à un pixel de coordonnées (x,y) sa position i dans un espace mono-dimensionnel.

Ce qui est important dans le plan, c'est la notion de voisinage. Un pixel a généralement 8 ou 4 voisins selon la topologie considérée. Dans une chaîne $1D$, chaque élément n'a que 2 voisins les plus proches. Ensuite, bien sûr, en transformant un espace $2D$ en un espace $1D$, les informations spatiales seront réduites, mais l'objectif est de conserver les informations les plus représentatives pendant la transformation.

Lorsqu'une transformation particulière est choisie (quelques exemples seront proposés ci-après), elle sera appliquée de la même manière à toutes les N images (ou à une région d'intérêt particulière) de la série. Donc, nous obtenons N chaînes qui seront considérées comme les lignes d'une nouvelle image. La nouvelle hauteur de l'image est égale au nombre N des

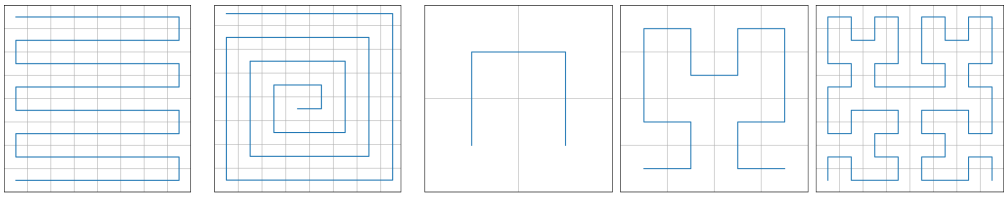


FIG. 2: Illustrations des différentes courbes (en bleu) couvrant un espace $2D$ (en noir, une grille de pixels) ; (a) Courbe serpent ; (b) Courbe spirale ; (c, d, e) Les trois premiers ordres de la courbe de Hilbert.

images de la STIS et sa largeur est égale au nombre de pixels de la région que nous voulons représenter. Cette nouvelle image constitue alors une représentation spatio-temporelle $2D$ d'une série temporelle d'images $2D+t$.

Afin de conserver certains voisins significatifs dans cette nouvelle représentation, le problème est alors de remplir un espace discret $2D$ avec une courbe discrète. En suivant les pixels le long de la courbe, tous les pixels de la région seront numérotés une seule fois et, par construction, deux pixels adjacents dans la courbe sont des pixels voisins dans le plan. Dans la littérature, de nombreuses méthodes ont été proposées pour réaliser une telle transformation, mais le but est de considérer des voisins statistiquement représentatifs sans aucun biais en raison du tracé choisi dans le plan.

Nous avons comparé expérimentalement plusieurs stratégies :

— la première représentation est la plus naïve, notée $\Re_{snake}$. L'espace est rempli par une simple courbe qui scanne l'image, ligne par ligne, en serpentant (Figure 2 (a)). Les lignes sont liées de manière intelligente, de sorte que les informations de voisinage spatial sont préservées : les extrémités des lignes impaires sont liées aux têtes des lignes paires, et vice versa. Les pixels sont alors numérotés en fonction de la courbe.

— la deuxième représentation est basée sur la spirale d'Archimède, notée $\Re_{spiral}$. La grille de pixels est associée à une courbe en spirale qui remplit un carré (Figure 2 (b)). La courbe commence à partir du point central $(0,0)$ d'un carré et de son voisin droit puis tourne autour. La construction de cette courbe se fait en fixant deux variables qui indiquent le prochain point de la courbe, $(x+dx, y+dy)$. dx, dy sont initialisés à 0 et 1 respectivement. Les points angulaires sont ceux qui vérifient $x = y$, $x = -y$ et $y > 0$, $x-1 = -y$ et $x > 0$. La courbe doit aller à droite, à gauche, en bas ou en haut selon les directions de (dx, dy). Les valeurs (dx, dy) sont successivement $(0,1), (-1,0), (0,-1)$ et enfin $(1,0)$.

— la troisième représentation est basée sur des courbes de remplissage de l'espace, notée $\Re_{Hilbert}$. Notre choix est la courbe de Hilbert, qui est une courbe fractale remplissant l'espace (Butz, 1971) et qui remplit un carré (une surface $2D$). Pour définir cette courbe, un processus récursif est appliqué à partir d'un domaine carré, le domaine étant divisé en quatre carrés égaux. Les quatre petits carrés sont liés de manière à ce que deux parties avec une arête commune aient deux index consécutifs. Cette règle est appliquée de manière récursive sur les carrés dont la largeur est une puissance de 2. L'ordre des pixels est finalement donné par la courbe de Hilbert. L'intérêt principal de ce type de

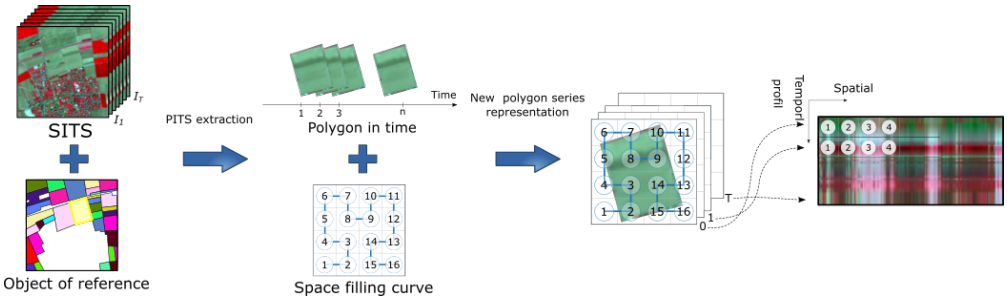


FIG. 3: Représentation d'une série temporelle de polygones basée sur la courbe de Hilbert.

courbe est la préservation de la relation de voisinage spatial de points successifs de la courbe. La Figure 2 (c–e) illustre les trois premiers ordres de courbes de Hilbert.

En appliquant le processus aux N images (ou à une région d'intérêt spécifique) de la STIS, nous obtenons N lignes de longueur égale au nombre N_r de pixels de la région. Ces lignes sont utilisées pour remplir une matrice et une nouvelle représentation de la STIS est obtenue sous forme d'une image avec N lignes et N_r colonnes. Maintenant, cette nouvelle image peut également être interprétée en termes de colonnes. Chaque colonne est associée à un pixel et à sa série temporelle dans la STIS, un pixel temporel $p = \{< p_t(x,y) > | t = 1 \ldots N\}$ est contenu dans une colonne de la nouvelle image. La Figure 3 illustre la construction de la nouvelle représentation.

3.3 Architecture profonde employée

Les réseaux de neurones convolutionnels sont utilisés dans la plupart des méthodes appartenant à la famille des algorithmes d'apprentissage profond. Les CNN sont composés, dans la partie gauche, de couches de neurones calculant les convolutions des sorties des couches précédentes. Les neurones de chaque couche sont activés par des fonctions non linéaires permettant l'extraction de caractéristiques d'ordre élevé de l'entrée. Il existe également des couches de regroupement maximal entre les couches de convolution afin de réduire progressivement le nombre d'entrées et le nombre de paramètres à calculer pour définir le réseau et pour contrôler également le sur-apprentissage. Dans la dernière partie droite du réseau, pour résoudre les problèmes de classification, nous trouvons généralement une couche entièrement connectée fournissant un vecteur de probabilité, couplée à une fonction softmax permettant de prédire une classe.

Dans notre approche, nous considérons le modèle SqueezeNet (Iandola et al., 2016). Ce modèle est un petit réseau composé de peu de paramètres à apprendre. Dans notre cas, il s'agit d'un modèle intéressant, car il s'adapte à notre contexte applicatif et à notre jeu de données (taille réduite des exemples d'apprentissage). Ce CNN conduit au même niveau de précision que le modèle AlexNet, lorsqu'il est évalué sur le jeu de données ImageNet.

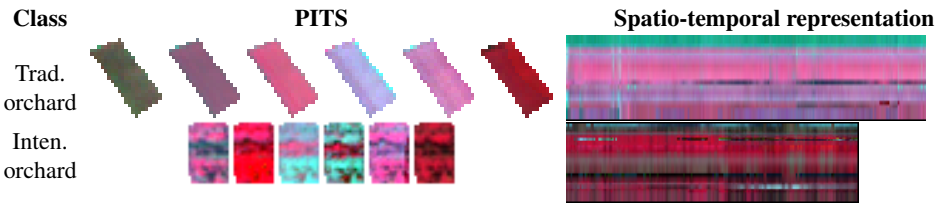


FIG. 4: Exemple de STP représentant des vergers ; (gauche) Évolution d'un verger traditionnel / intensif ; (droite) Représentations spatio-temporelles associées (ici avec la stratégie de Hilbert $\Re_{Hilbert}$).

4 Étude expérimentale

L'approche proposée a été évaluée dans le cadre d'une application de télédétection, à savoir la classification de parcelles agricoles à partir de STIS. Notre objectif est de différencier certaines classes thématiques agricoles (ici les vergers traditionnels par rapport aux vergers intensifs difficilement différentiables à l'échelle des images Sentinel-2). L'aspect visuel de ces parcelles agricoles est hétérogène car les vergers font l'objet de nombreuses pratiques agricoles, dépendant de la saison, et leur identification automatique reste une tâche complexe et importante pour différents besoins de gestion des territoires et de l'environnement. Afin de différencier ces deux classes, les caractéristiques spatio-temporelles peuvent contenir des informations utiles pour mieux discriminer les pratiques agricoles.

4.1 Données

Les données utilisées dans cette étude expérimentale sont des STIS optiques, captées par le satellite Sentinel-2 (Est de la France). Les données acquises ont été corrigées et orthorectifiées par le programme français Theia afin de pouvoir être comparables radiométriquement. Les images sont distribuées avec leurs masques de nuages associés. Un prétraitement a été appliqué aux images avec une interpolation linéaire sur les pixels masqués pour garantir la chérence de tous les pixels.

Nous disposons d'un STIS de $N = 50$ images capturées en 2017 sur la même zone géographique. Pour chaque image, seules trois bandes sont conservées : proche infrarouge (Nir), rouge (R) et vert (G). Toutes ces bandes ont une résolution spatiale de 10 mètres.

En plus des images, nous disposons de données de référence composées des délinéations de parcelles agricoles de référence (dans notre contexte les vergers) représentées sous forme de polygones vectoriels. Ces polygones sont extraits du RPG de l'IGN. Dans notre cas, les polygones ont été rasterisés en fonction de la résolution spatiale de chaque image, ce qui a conduit à une nouvelle série temporelle de polygones, notée STP.

Les données de référence utilisées dans notre expérience sont les étiquettes sémantiques de ces polygones (vergers traditionnels ou intensifs). La Figure 4 présente un exemple de l'évolution temporelle de deux vergers à travers la STIS. Enfin, nous disposons de 100 polygones par classe. Afin d'obtenir plus de données annotées, nous avons employé une technique d'aug-

mentation de données (AD) en appliquant des rotations sur les images avec les angles : 45°, 90°, 135° and 180°.

4.2 Protocole expérimental

Nous avons appliqué la méthode proposée pour classifier les deux classes de vergers (traditionnel et intensif). D'un point de vue intuitif, les vergers intensifs devraient avoir une texture plus homogène dans le domaine spatial puisque les arbres fruitiers sont généralement alignés, ce qui n'est pas toujours le cas dans les vergers traditionnels.

4.2.1 Préparation des données

Premièrement, les données d'entrée sont préparées grâce aux représentations spatio-temporelles d'images proposées. Ceci est opéré au niveau polygone. Chaque STP est traité de 3 manières différentes selon les fonctions $\Re_{snake,spiral,Hilbert}$ présentées précédemment. Pour souligner l'intérêt de considérer la relation spatiale entre les pixels, nous avons ajouté (comme base naïve de référence) une stratégie aléatoire pour former la représentation spatio-temporelle du STP, notée $\Re_{random}$.

En fonction de la taille d'entrée du CNN, qui est de 224×224, nous adaptons les images générées à cette taille. Pour la dimension temporelle (axe Y), nous proposons deux stratégies. La première consiste à centrer verticalement les informations d'origine des images d'entrée N ($N = 50$). Les lignes supérieures et inférieures restantes sont fixées à la valeur zéro. Pour la seconde, nous avons choisi de traiter une série temporelle de longueur 224, c'est-à-dire de remplir tout l'espace vertical restant. Pour ce faire, nous avons appliqué une interpolation linéaire sur les informations temporelles. Nous supposons que l'information temporelle entre deux dates consécutives est monotone et linéaire. L'interpolation est ensuite effectuée en considérant que nous n'avons que 224 jours dans l'année, de sorte qu'un jour a une durée d'environ 39 heures. Pour les dates initiales, nous affectons les informations temporelles de la première date dans la STIS. Pour les dernières dates, nous affectons les dernières informations temporelles dans la STIS. Pour les autres valeurs de date inconnues, nous les calculons en appliquant une fonction linéaire qui prend en compte deux dates disponibles consécutives (prises à partir du jeu de $N = 50$ images de la STIS). Enfin, nous avons 224 dates qui complètent la hauteur de l'image. Ces deux stratégies (avec dates originales ou avec interpolation temporelle) seront évaluées séparément.

Pour la dimension spatiale (axe X), la taille des polygones étant rarement égale à 224, nous avons adopté la stratégie suivante. Pour les polygones dont le nombre de pixels est inférieur à 224, nous répétons la séquence. Pour ceux composés de plus de 224 pixels, nous découpons la nouvelle représentation en différentes images avec 224 colonnes, ce qui conduit potentiellement à un nombre de données à classer supérieur au nombre de polygones.

Les données images ont été normalisées en fonction des valeurs maximales et minimales du jeu de données. Dans notre cas, nous avons limité les valeurs à 2% (ou 98%), comme proposé dans (Pelletier et al., 2019).

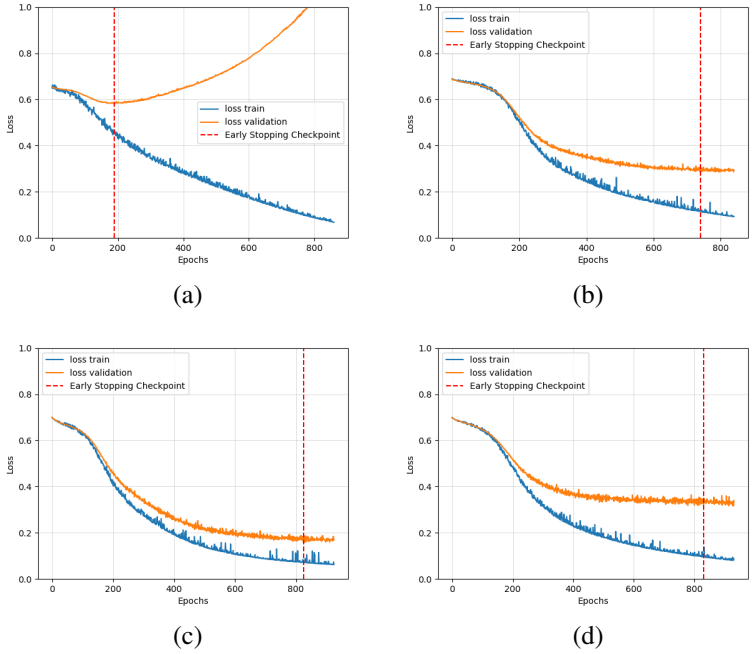


FIG. 5: Courbes de perte liées à l'entraînement de notre modèle avec les différentes représentations spatio-temporelles ; (a) stratégie aléatoire $\Re_{random}$; (b) stratégie serpent $\Re_{snake}$; (c) stratégie spirale $\Re_{spiral}$; et (d) stratégie Hilbert $\Re_{Hilbert}$.

4.2.2 Protocole d'apprentissage et de validation

Pour valider ces expériences, une stratégie de validation croisée (5 fold) est utilisée. Dans chaque cas, le jeu de données est divisé de manière aléatoire en 3 jeux, au niveau polygone, et nous répétons 5 fois le processus. La taille de ces ensembles est de 60%, 20% et 20% de toutes les données disponibles représentant respectivement les ensembles d'apprentissage, de validation et de test. Dans chaque expérience, les mêmes découpages sont pris en compte afin de rendre les résultats plus comparables. Le modèle est formé et évalué 5 fois en fonction de chaque division. Pour une division, nous considérons le système qui donne le meilleur résultat sur l'ensemble de validation. Notez que la décision de la sortie du classificateur est prise au niveau du polygone. Nous avons déjà expliqué que pour les grands polygones (qui ont plus de 224 pixels), nous construisons plusieurs images différentes dans notre processus (voir la Section 4.2.1). Ensuite, plusieurs images sont associées à un seul polygone. Pour prendre une décision dans ce cas, le modèle renvoie les probabilités de classes pour chaque image associée au polygone. Ensuite, nous faisons la moyenne de ces probabilités pour chaque classe et nous affectons au polygone l'étiquette de la classe avec la probabilité la plus élevée. Nous rapportons la précision globale qui correspond à la valeur moyenne des résultats sur les ensembles de test en fonction des 5 divisions et de l'écart type.

Nous entraînons le modèle en utilisant *Adam* comme optimiseur avec un taux d'apprentis-

sage de 10^{-6} et les valeurs par défaut des autres paramètres ($\beta_1 = 0.9$, $\beta_2 = 0.999$ et $\varepsilon = 10^{-8}$) avec une taille de batch de 8. Nous limitons le nombre d'époques à 2 000, à la suite d'une technique d'arrêt précoce avec un nombre de patience de 100.

La taille de l'ensemble de données disponible étant limitée, nous entraînons le réseau selon deux stratégies : (1) *from scratch* et (2) avec un *fine-tuning* (le modèle a été pré-entraîné sur ImageNet dans le cadre d'un problème de classification). Nous avons également procédé à une augmentation des données (AD).

4.3 Résultats et discussions

Les représentations spatio-temporelles proposées des STP ont été utilisées pour nourrir le CNN. Nous avons également utilisé l'ordre $\Re_{random}$ de pixels afin d'évaluer l'importance des informations spatiales. Deux pixels successifs dans la représentation $1D$ sont des voisins dans l'espace $2D$. Ceci est une propriété des différentes courbes de remplissage d'espace que nous avons considérées. À des fins de visualisation, la Figure 4 illustre deux STP avec leurs représentations spatio-temporelles résultantes, basées ici sur la stratégie $\Re_{Hilbert}$.

Le modèle CNN a été entraîné conformément au protocole d'apprentissage, avec et sans fine-tuning. Nous avons également évalué l'impact de la prise en compte des dates originales ou de l'application d'une interpolation temporelle pour correspondre à la taille d'entrée d'image (224×224) requise par SqueezeNet.

La Figure 5 illustre les courbes de perte résultantes lorsque SqueezeNet est entraîné (suivant une technique d'arrêt précoce) avec des images associées respectivement aux stratégies $\Re_{random}$, $\Re_{snake}$, $\Re_{spiral}$ et $\Re_{Hilbert}$, ici avec les dates originales. À partir de ces courbes, nous remarquons que les valeurs de perte les plus élevées sont obtenues avec la stratégie $\Re_{random}$ comme prévu. De plus, la courbe de perte de la stratégie $\Re_{random}$ commence à se stabiliser dès 200 époques, par rapport aux autres stratégies qui commencent à se stabiliser à partir d'environ 600 époques. Intuitivement, cela signifie que la stratégie $\Re_{random}$ ne fournit pas une bonne représentation des STP avec une bonne capacité à généraliser lors de l'entraînement. Les autres représentations permettent de faire un meilleur entraînement. Nous pouvons également voir que les meilleures courbes d'apprentissage sont obtenues en (c), en utilisant $\Re_{spiral}$, avec les meilleurs résultats sur l'ensemble de validation. Ce classement n'est pas conservé au niveau de l'ensemble de test global.

Le Tableau 1 présente les résultats de la classification (précision globale) obtenus avec nos représentations spatio-temporelles (avec les dates originales). Nous remarquons que $\Re_{random}$ fournit toujours les scores les plus bas comparés aux autres représentations, avec et sans AD, avec et sans fine-tuning. Ceci est attendu car la discrimination entre vergers traditionnels et intensifs repose sur des informations spatiales et ces informations sont partiellement préservées avec les courbes de remplissage d'espace fournissant des informations spatiales en plus des informations temporelles. Dans le Tableau 1, nous remarquons également que, avec l'AD, tous les scores sont légèrement augmentés et que les meilleurs ont été obtenus en combinant AD et le fine-tuning. Enfin, les meilleures représentations oscillent entre $\Re_{snake}$, $\Re_{spiral}$ et $\Re_{Hilbert}$.

A titre comparatif, nous avons comparé nos résultats à ceux obtenus avec la méthode TempCNN dédiée à la classification des séries temporelles, proposée dans (Pelletier et al., 2019). Cette approche repose sur l'utilisation d'un CNN, dans lequel les convolutions sont appliquées dans le domaine temporel (convolutions $1D$). Les tailles de filtre sont fixées en fonction du critère indiqué dans (Pelletier et al., 2019) : avec une taille de noyau de 5 lors de

TAB. 1: Résultats de classification (précision globale – OA et écart type – STD) obtenus avec nos représentations spatio-temporelles (avec dates originales); (première / deuxième ligne) Sans / avec augmentation des données.

		From scratch		Fine tuning	
	Rep.	**OA**	**STD**	**OA**	**STD**
sans AD	$\mathfrak{R}_{random}$	71.50	7.17	81.00	8.15
	$\mathfrak{R}_{snake}$	78.00	4.30	90.50	7.96
	$\mathfrak{R}_{spiral}$	76.00	8.74	**92.00**	**3.31**
	$\mathfrak{R}_{Hilbert}$	**79.00**	**5.61**	91.00	2.00
avec AD	$\mathfrak{R}_{random}$	80.50	3.67	87.00	4.58
	$\mathfrak{R}_{snake}$	83.50	7.00	**93.50**	**2.54**
	$\mathfrak{R}_{spiral}$	**84.50**	**5.33**	93.00	1.87
	$\mathfrak{R}_{Hilbert}$	81.50	6.44	91.00	2.54

TAB. 2: Résultats de la classification (précision globale – OA et écart type – STD) avec les architectures TempCNN (avec les dates d'origine et un noyau de taille 5).

nb filt.	16	32	64	128	256	512	1024
OA	78.81	77.38	81.66	78.45	**85.37**	81.73	84.80
STD	6.08	6.51	4.59	4.79	**3.44**	5.75	6.48

la prise en compte des dates d'origine et de 11 lors de la prise en compte des dates interpolées. À des fins de comparaison, nous avons entraîné et validé le modèle TempCNN en utilisant le même protocole de validation. Dans le code proposé par les auteurs du modèle, la décision au niveau polygone est obtenue via un vote majoritaire pondéré par les probabilités issues du réseau. Notez que le modèle TempCNN est proposé avec différentes architectures (profondeurs), conduisant à un nombre différent de filtres.

Le Tableau 2 contient les résultats de TempCNN. Les meilleurs scores ont été obtenus avec 256 filtres. Les scores obtenus suggèrent que les résultats obtenus avec TempCNN surpassent ceux obtenus avec notre méthode lorsque nous entraînons le réseau à partir d'une initialisation aléatoire. Cependant, avec une initialisation du réseau avec les poids obtenus lors d'un pré-entraînant sur ImageNet, nous obtenons de meilleurs scores. Cela met en évidence, pour notre contexte applicatif, l'avantage de considérer un modèle classique de CNN $2D$ pour classifier les images $2D+t$ combinées à nos représentations spatio-temporelles.

Le Tableau 3 présente les résultats de la classification avec la stratégie d'interpolation temporelle. Nous remarquons qu'avec plus d'informations temporelles, les scores globaux sont augmentés par rapport au cas où moins d'informations temporelles (images avec dates originales) sont disponibles (Tableau 1). Cela s'explique par la distribution non régulière des dates d'origine. Avec l'interpolation, nous obtenons une information temporelle avec une régularité égale pour obtenir 224 dates. Ceci provient également du comportement monotone réel entre les dates consécutives utilisées pour l'interpolation. Nous observons à nouveau que la stratégie

TAB. 3: Résultats de classification (précision globale – OA et écart type – STD) obtenus avec nos représentations spatio-temporelles (avec interpolation temporelle) ; (première / deuxième ligne) Sans / avec augmentation des données.

		From scratch		Fine tuning	
	Rep.	**OA**	**STD**	**OA**	**STD**
sans AD	$\mathfrak{R}_{random}$	84.00	9.02	87.00	4.30
	$\mathfrak{R}_{snake}$	85.00	4.18	**92.50**	**3.16**
	$\mathfrak{R}_{spiral}$	85.00	3.53	91.00	2.54
	$\mathfrak{R}_{Hilbert}$	**89.00**	**3.39**	91.00	2.54
avec AD	$\mathfrak{R}_{random}$	82.00	8.71	83.50	4.35
	$\mathfrak{R}_{snake}$	86.50	5.38	90.50	1.87
	$\mathfrak{R}_{spiral}$	86.50	3.00	**91.50**	**3.74**
	$\mathfrak{R}_{Hilbert}$	**92.50**	**1.58**	89.00	3.39

TAB. 4: Résultats obtenus (précision globale – OA et écart type – STD) avec les architectures TempCNN (avec interpolation temporelle et un noyau de taille 11).

nb filt.	16	32	64	128	256	512	1024
OA	78.96	81.40	83.96	81.86	85.93	84.23	**87.21**
STD	7.34	6.32	7.14	5.18	8.03	6.23	**8.28**

$\mathfrak{R}_{random}$ conduit aux pires scores. Cela confirme que l'information spatiale est importante et pas seulement temporelle. Nous voyons aussi que l'AD augmente légèrement les scores en cas d'apprentissage from scratch mais n'est pas en mesure d'améliorer les résultats en cas de fine-tuning. Dans cette expérience, la stratégie $\mathfrak{R}_{Hilbert}$ conduit à la représentation qui fournit les meilleurs scores lorsque nous entraînons le réseau from scratch (avec ou sans AD). Mais lorsque nous employons le fine-tuning, les meilleures représentations oscillent entre $\mathfrak{R}_{snake}$ et $\mathfrak{R}_{spiral}$.

Les résultats obtenus avec TempCNN et la stratégie d'interpolation temporelle sont répertoriés dans le Tableau 4. Les scores initiaux sont dans le même intervalle que notre méthode lorsque nous entraînons from scratch. Mais avec AD et / ou fine-tuning, nos scores sont plus élevés.

5 Conclusion

Dans cet article, nous présentons une nouvelle stratégie pour transformer une série temporelle d'images en une représentation planaire spatio-temporelle. Ceci permet de réduire la complexité de la structure de la série temporelle d'images (de $2D+t$ à $2D$) tout en conservant (partiellement) les relations spatiales et temporelles des pixels. Ces représentations sont utilisées pour alimenter un CNN classique afin d'effectuer une classification. Les convolutions $2D$

peuvent alors conduire à une extraction de caractéristiques spatio-temporelles. En comparaison aux approches 1*D* dédiées aux séries temporelles, nous avons un nombre moins élevé de données annotées, mais ceci est compensé par une stratégie d'augmentation des données. En considérant des convolutions 2*D*, nous pouvons également bénéficier d'un modèle pré-entraîné sur ImageNet. Une telle initialisation des poids du CNN est moins facile à réaliser pour les approches 1*D*, car aucun jeu de données semblable à ImageNet n'est disponible.

L'approche proposée a été évaluée en télédétection pour la classification de parcelles agricoles à partir de STIS. Dans notre expérimentation, nous étudions l'impact de la transformation spatio-temporelle en utilisant différentes courbes de remplissage de l'espace. Les résultats obtenus reflètent l'utilité et l'impact de la prise en compte des informations spatiales et temporelles. Dans notre étude thématique, nous observons que les scores de classification sont plus élevés lorsque l'on considère les représentations spatio-temporelles avec plus d'informations temporelles (en utilisant l'interpolation temporelle) que celles qui en ont moins, même si elles sont construites à partir des mêmes données initiales. Il est donc plus important d'avoir beaucoup de données dans le domaine temporel que d'optimiser la façon dont le plan 2*D* est rempli par les courbes de remplissage de l'espace.

Dans notre étude comparative, nous remarquons que la méthode TempCNN (Pelletier et al., 2019) s'applique au niveau des pixels alors que notre approche s'applique au niveau des polygones. Cela signifie que pour TempCNN le nombre d'échantillons d'entraînement est supérieur à celui de notre méthode où les pixels d'un polygone sont tous résumés dans une seule image spatio-temporelle. Malgré le faible nombre de données disponibles, l'accroissement de la précision par notre processus est de l'ordre de 8% basé sur les données d'origine et de l'ordre de 5% sur les données interpolées.

Dans nos futurs travaux, l'ordre des pixels sera étudié plus précisément. Nous avions jusqu'à présent analysé les pixels d'une surface carrée dans laquelle le polygone était inclus, mais nous devons définir un ordre adapté à la géométrie du polygone lui-même. Nous allons également augmenter le nombre d'instances de vergers et appliquer la même approche à des problèmes impliquant un plus grand nombre de classes afin de générer des cartes d'occupation du sol.

Remerciements

Ces travaux ont été financés par l'ANR sous le numéro de projet ANR-17-CE23-0015.

Références

Andres, L., W. Salas, et D. Skole (1994). Fourier analysis of multi-temporal AVHRR data applied to a land cover classification. *Int. J. Remote Sens. 15*(5), 1115–1121.

Bagnall, A., J. Lines, A. Bostrom, J. Large, et E. Keogh (2017). The great time series classification bake off : a review and experimental evaluation of recent algorithmic advances. *DMKD 31*(3), 606–660.

Bruzzone, L. et D. Prieto (2000). Automatic analysis of the difference image for unsupervised change detection. *IEEE Transactions on Geoscience and Remote Sens. 38*(3), 1171–1182.

Butz, A. (1971). Alternative algorithm for Hilbert's space-filling curve. *IEEE Trans. on Computers 20*(4), 424–426.

Chelali, M., C. Kurtz, A. Puissant, et N. Vincent (2019). Urban land cover analysis from satellite image time series based on temporal stability. In *JURSE, Procs.*, pp. 1–4.

Chelali, M., C. Kurtz, A. Puissant, et N. Vincent (2020). Image time series classification based on a planar spatio-temporal data representation. In *VISAPP, Procs.*, pp. XX–XX.

Coppin, P., I. Jonckheere, K. Nackaerts, B. Muys, et E. Lambin (2004). Digital change detection methods in ecosystem monitoring : A review. *Int. J. Remote Sens.*, 1565–1596.

Di Mauro, N., A. Vergari, T. M. A. Basile, F. G. Ventola, et F. Esposito (2017). End-to-end learning of deep spatio-temporal representations for satellite image time series classification. In *DC@PKDD/ECML, Procs.*, pp. 1–8.

Huang, B., K. Lu, N. Audebert, A. Khalel, Y. Tarabalka, J. Malof, et A. Boulch (2018). Large-scale semantic classification : Outcome of the first year of inria aerial image labeling benchmark. In *IGARSS, Procs.*, pp. 6947–6950.

Iandola, F., M. Moskewicz, K. Ashraf, S. Han, W. Dally, et K. Keutzer (2016). Squeezenet : AlexNet-level accuracy with 50x fewer parameters and <1MB model size. *CoRR abs/1602.07360*.

Ienco, D., R. Gaetano, C. Dupaquier, et P. Maurel (2017). Land cover classification via multitemporal spatial data by deep recurrent neural networks. *IEEE Geosci. Remote Sens. Lett. 14*(10), 1685–1689.

Inglada, J., A. Vincent, M. Arias, B. Tardy, D. Morin, et I. Rodes (2017). Operational high resolution land cover map production at the country scale using satellite image time series. *Remote Sens. 9*(1), 95–108.

Ismail Fawaz, H., G. Forestier, J. Weber, L. Idoumghar, et P. Muller (2019). Deep learning for time series classification : A review. *DMKD 33*(4), 917–963.

Pelletier, C., G. Webb, et F. Petitjean (2019). Temporal convolutional neural network for the classification of satellite image time series. *Remote Sens. 11*(5), 523–534.

Petitjean, F., J. Inglada, et P. Gançarski (2012a). Satellite image time series analysis under time warping. *IEEE Transactions on Geoscience and Remote Sens. 50*(8), 3081–3095.

Petitjean, F., C. Kurtz, N. Passat, et P. Gançarski (2012b). Spatio-temporal reasoning for the classification of satellite image time series. *PRL 33*(13), 1805–1815.

Senf, C., P. Leitao, D. Pflugmacher, S. Van der Linden, et P. Hostert (2015). Mapping land cover in complex mediterranean landscapes using landsat : Improved classification accuracies from integrating multi-seasonal and synthetic imagery. *Remote Sens. Environ. 156*, 527–536.

Tran, D., L. Bourdev, R. Fergus, L. Torresani, et M. Paluri (2015). Learning spatiotemporal features with 3D convolutional networks. In *ICCV, Procs.*, pp. 4489–4497.

Verbesselt, J., R. Hyndman, G. Newnham, et D. Culvenor (2010). Detecting trend and seasonal changes in satellite image time series. *Remote Sens. Environ. 114*(1), 106–115.

Summary

Image time series such as MRI functional sequences or Satellite Image Time Series (STIS) provide valuable information for the automatic analysis of complex patterns through time. A major issue when analyzing such data is to consider at the same time their temporal and spatial dimensions. In this article we present a novel data representation that makes image times series compatible with classical deep learning model, such as Convolutional Neural Networks (CNN). The proposed approach is based on a novel planar representation of image time series that converts $2D+t$ data as $2D$ images without loosing too much spatial or temporal information. Doing so, CNN can learn at the same time the parameters of $2D$ filters involving temporal and spatial knowledge. Preliminary results in the remote sensing domain highlight the ability of our approach to discriminate complex agricultural land-cover classes from a STIS.

Réseaux antagonistes génératifs pour la reconstruction super-résolution et la segmentation en IRM

Quentin Delannoy[*], Chi-Hieu Pham[**], Clément Cazorla[*], Carlos Tor-Díez[**], Guillaume Dollé[***], Hélène Meunier[****], Nathalie Bednarek[*,****], Ronan Fablet[♯], Nicolas Passat[*], François Rousseau[**]

[*] Université de Reims Champagne Ardenne, CReSTIC, EA 3804, 51097 Reims, France
[**] IMT Atlantique, LaTIM U1101 INSERM, UBL, Brest, France
[***] Université de Reims Champagne Ardenne, CNRS, LMR UMR 9008, 51097 Reims, France
[****] Service de médecine néonatale et réanimation pédiatrique, CHU de Reims, France
[♯] IMT Atlantique, Lab-STICC UMR CNRS 6285, Brest, France

Résumé. La faible résolution et l'anisotropie des données induisent des difficultés pour l'analyse cérébrale néonatale en IRM (imagerie par résonance magnétique). Dans la plupart des chaînes de traitement d'analyse IRM, les données sont d'abord rééchantillonnées, puis segmentées par des approches (semi-)automatiques. En d'autres termes, la reconstruction et la segmentation de l'image sont effectuées séparément. Nous proposons une méthodologie permettant d'effectuer simultanément la reconstruction haute résolution et la segmentation des données IRM du cerveau néonatal. Notre stratégie s'appuie principalement sur des réseaux antagonistes génératifs (GAN). Nous décrivons et discutons cette architecture et les résultats qu'elle permet d'obtenir.

1 Introduction

Des études à long terme sur les conséquences de la prématurité ont démontré que la majorité des nouveau-nés prématurés peuvent présenter des déficits moteurs, cognitifs et comportementaux importants (Hack et Fanaroff, 2000; Marlow et al., 2005). Au demeurant, notre compréhension de la nature des anomalies cérébrales sous-jacentes à ces séquelles neurologiques reste limitée. Dans ce contexte, l'imagerie par résonance magnétique (IRM) offre des possibilités uniques d'investigation in vivo du cerveau humain. En raison de leur faible résolution et de leur anisotropie, l'analyse de ces images cérébrales néonatales reste difficile. Ainsi, l'amélioration de la résolution des images et la segmentation du cerveau à partir de celles-ci est une condition indispensable pour permettre des analyses morphométriques robustes.

Lorsqu'il s'agit d'images anisotropes à faible résolution, l'une des premières composantes clés d'une chaîne de traitement de données IRM cliniques est l'estimation d'images isotropes, par suréchantillonnage. La super-résolution (SR) (Greenspan, 2008) est une technique qui vise à améliorer la résolution d'une image après son acquisition. Cependant, la SR constitue un problème inverse difficile à résoudre ; en particulier, l'estimation de la texture et des détails est délicate. Dans ce contexte, les techniques d'apprentissage profond supervisé ont permis de

réelles améliorations par rapport aux approches basées sur des modèles. Ainsi, l'application de réseaux neuronaux convolutionnels 3D (CNNs) aboutit à des résultats prometteurs pour les données IRM (Pham et al., 2017; Chen et al., 2018).

Cependant, l'utilisation d'une fonction de coût basée uniquement sur une comparaison point à point de l'image estimée peut conduire à un lissage excessif des images haute résolution obtenues (Johnson et al., 2016). Dans l'optique de rendre ces images plus « réalistes », une composante perceptuelle (Ledig et al., 2017) peut être ajoutée à la fonction de coût. Dans notre cas, cette dernière sera basée sur la fonction de coût d'un GAN (*generative adversarial network*). Il est à noter que l'usage de GAN a été récemment proposé pour la segmentation d'IRM cérébrales (Moeskops et al., 2017).

Parmi les structures cérébrales d'intérêt, les structures minces et en particulier la matière grise corticale, restent difficiles à analyser, principalement à cause de leur forte dégradation dans les images basse résolution. Néanmoins, le cortex est une région d'intérêt, comme le soulignent des travaux récents portant par exemple sur la gyrification (Dubois et al., 2008; Lefèvre et al., 2016; Orasanu et al., 2016), la connectivité corticale (Ball et al., 2013a) ou le développement cortical (Ball et al., 2013b; Yu et al., 2016).

Notre objectif est de proposer une méthodologie dédiée à l'analyse d'images IRM anisotropes basse résolution. En particulier, nous visons à traiter des structures anatomiques complexes, dont le cortex. Pour ce faire, nous proposons une approche basée sur les GAN, nommée SegSRGAN, qui génère à la fois une image super-résolue et une carte de segmentation corticale à partir d'une seule image basse résolution.

2 Super-resolution et segmentation d'image : formulation

2.1 Super-résolution

Le but d'une méthode de SR est d'estimer une image haute résolution (HR) $\mathbf{X} \in \mathbb{R}^m$ à partir d'une image basse résolution (LR) observée $\mathbf{Y} \in \mathbb{R}^n$, avec $m > n$. Un tel problème peut être formulé à l'aide du modèle d'observation linéaire :

$$\mathbf{Y} = H_{\downarrow} B\mathbf{X} + N = \Theta\mathbf{X} + N \tag{1}$$

où $N \in \mathbb{R}^n$ est un bruit additif, $B \in \mathbb{R}^{m\times m}$ est une matrice de dispersion (« flou »), $H_{\downarrow} \in \mathbb{R}^{n\times m}$ est une matrice de décimation (sous-échantillonnage) et $\Theta = H_{\downarrow}B \in \mathbb{R}^{n\times m}$.

Une façon usuelle de résoudre ce problème de SR consiste à définir la matrice Θ^{-1} comme la combinaison d'un opérateur de restauration $F \in \mathbb{R}^{m\times m}$ et d'un opérateur d'interpolation $S^{\uparrow} \in \mathbb{R}^{m\times n}$ qui calcule l'image LR interpolée $\mathbf{Z} \in \mathbb{R}^m$ associée à $\mathbf{Y}$ (i.e. $\mathbf{Z} = S^{\uparrow}\mathbf{Y}$). Dans le contexte d'un apprentissage supervisé, étant donné un ensemble d'images HR $\mathbf{X}_i$ et leurs images LR correspondantes $\mathbf{Y}_i$, l'opérateur de restauration F peut être estimé tel que :

$$\widehat{F} = \arg\min_{F} \sum_{i} d(\mathbf{X}_i - F(\mathbf{Z}_i)) \tag{2}$$

où d peut être, par exemple, une norme ℓ_2, une norme ℓ_1 ou une variante dérivable de la norme ℓ_1 telle que définie par Charbonnier et al. (1997). À l'instar de Lai et al. (2017), notre fonction de coût sera de type « Charbonnier ». Dans (Pham et al., 2017; Chen et al., 2018), il a été démontré que les CNNs 3D pouvaient être utilisés pour estimer avec précision la fonction de restauration $\widehat{F}$ pour les images IRM du cerveau.

2.2 Segmentation

Afin d'équilibrer les contributions de l'image SR et de la segmentation dans la fonction de coût, la segmentation de l'image est considérée comme un problème de régression supervisée :

$$\mathbf{S_X} = R(\mathbf{X}) \tag{3}$$

où R désigne une fonction non linéaire de l'image interpolée $\widehat{\mathbf{X}}$ vers la carte de segmentation $\mathbf{S_X}$. Comme pour le problème de SR, en supposant que nous avons un ensemble d'images interpolées $\widehat{\mathbf{X}}$ et leurs cartes de segmentation correspondantes $\mathbf{S_{X_i}}$, une approche générale pour résoudre un tel problème de segmentation consiste à trouver la correspondance R tel que :

$$\widehat{R} = \arg\min_R \sum_i d(\mathbf{X}_i - R(\mathbf{X}_i)) \tag{4}$$

3 Description de la méthode

Les approches basées sur les GAN consistent à former un premier réseau G, dit générateur, qui estime pour une image d'entrée interpolée donnée, l'image HR et la carte de segmentation correspondante. Un second réseau D, dit discriminateur, est conçu pour différencier les couples d'images HR et de segmentations réels des couples générés d'images SR et de segmentations.

3.1 Fonction de coût

Afin d'éviter de potentiels problèmes de saturation du gradient, que nous pouvons rencontrer avec la fonction de coût traditionnelle [1] « minimax » des GAN, la fonction de coût appelée WGAN-GP (Gulrajani et al., 2017) est utilisée. Ce type de GAN vise à minimiser la distance de Wasserstein entre les deux distributions $\mathbb{P}_r$ et $\mathbb{P}_g$ (resp. données réelles et générées) :

$$W(\mathbb{P}_r, \mathbb{P}_g) = \inf_{\gamma \in \Pi(\mathbb{P}_r, \mathbb{P}_g)} \mathbb{E}_{(x,y)\sim\gamma}\left[||x-y||\right] \tag{5}$$

$$= \sup_{||f||_L \leq 1} \mathbb{E}_{x\sim\mathbb{P}_r}[f(x)] - \mathbb{E}_{x\sim\mathbb{P}_g}[f(x)] \tag{6}$$

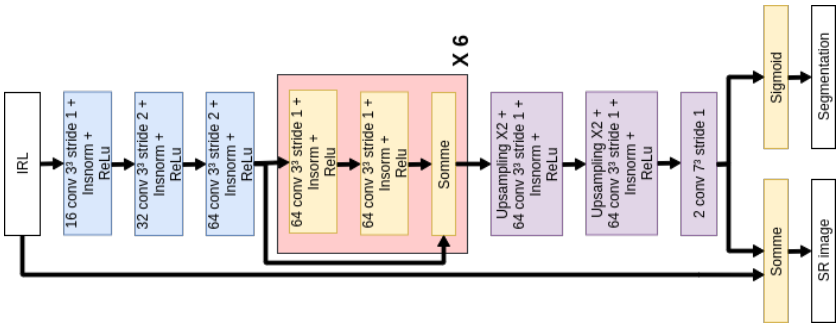
où, $\Pi(\mathbb{P}_r, \mathbb{P}_g)$ est l'ensemble de toutes les distributions dont les marginales sont respectivement $\mathbb{P}_r$ et $\mathbb{P}_g$, et le supremum est calculé sur toutes les fonctions 1-lipschitziennes f.

Dans ce GAN, le discriminateur apprend la fonction paramétrée f tandis que le générateur vise à minimiser cette distance. Ainsi, la partie antagoniste de la fonction de coût est :

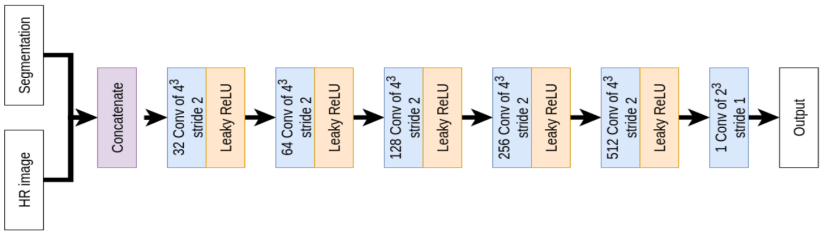
$$\mathcal{L}_{adv} = \mathbb{E}_{\mathbf{X}\sim\mathbb{P}_\mathbf{X}, \mathbf{S_X}\sim\mathbb{P}_{\mathbf{S_X}}}[D((\mathbf{X}, \mathbf{S_X}))] - \mathbb{E}_{\mathbf{Z}\sim\mathbb{P}_\mathbf{Z}}[D(G(\mathbf{Z}))] \tag{7}$$

où $\mathbf{X}$ et $\mathbf{S_X}$ sont les vraies image HR et carte de segmentation, respectivement, D est le discriminateur, G le générateur et $\mathbf{Z}$ l'image interpolée. C'est à travers ce terme que sera exprimé le jeu entre le générateur (minimiser $\mathcal{L}_{adv}$) et le discriminateur (maximiser $\mathcal{L}_{adv}$).

1. C'est-à-dire en résolvant $\min_G \max_D V(D,G) = \mathbb{E}_{x\in p_{data}(x)}[\log D(x)] + \mathbb{E}_{z\in p_z(z)}[\log(1-D(G(z)))]$.



(a) Architecture du générateur.



(b) Architecture du discriminateur.

FIG. 1 – *Architecture des réseaux de neurones. (a) Générateur. (b) Discriminateur.*

Rappelons que le discriminateur vise à estimer une fonction 1-lipschitzienne ; un terme de régularisation sur la valeur de son gradient est donc ajouté. Finalement, la fonction de coût (à minimiser) du discriminateur est :

$$\mathcal{L}_{dis} = \lambda_{gp}\mathbb{E}_{\widehat{\mathbf{XS}}}[(\|(\nabla_{\widehat{\mathbf{XS}}}D(\widehat{\mathbf{XS}})\|_2 - 1)^2] - \mathcal{L}_{adv} \tag{8}$$
$$\text{avec } \widehat{\mathbf{XS}} = (1-\varepsilon)(\mathbf{X}, \mathbf{S_X}) + \varepsilon G(\mathbf{Z}) \text{ et } \varepsilon \sim U[0,1]$$

où $\lambda_{gp} > 0$ et ∇ désignent respectivement le coefficient de pénalité du gradient et l'opérateur de gradient.

La fonction de coût du générateur est construite en ajoutant un terme de comparaison point par point ρ (Charbonnier et al., 1997) des images cibles en sortie et des images estimées :

$$\mathcal{L}_{gen} = \lambda_{adv}\mathcal{L}_{adv} + \mathbb{E}_{\mathbf{X}\sim\mathbb{P}_\mathbf{X}, \mathbf{S_X}\sim\mathbb{P}_{\mathbf{S_X}}}\left[\rho\left((\mathbf{X}, \mathbf{S_X}) - G(\mathbf{Z})\right)\right] \tag{9}$$
$$\text{avec } \rho(x_1, \ldots, x_n) = \frac{1}{n}\sum_{i=1}^{n}\sqrt{(x_i^2 + \nu^2)} \text{ et } \nu = 10^{-3}$$

3.2 Architecture du réseau

Architecture du générateur Le réseau du générateur est un réseau basé sur la convolution, avec des blocs résiduels. Il prend en entrée l'image LR interpolée. Il est composé de 18 couches de convolution : trois pour la partie encodage, deux fois six pour la partie résiduelle et trois

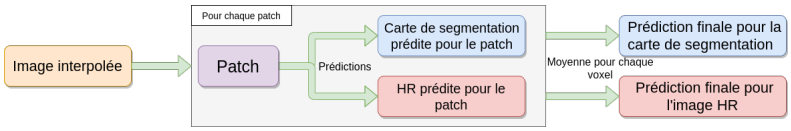


FIG. 2 – *Fabrication de la prédiction durant la phase de test.*

pour la partie décodage (voir Figure 1(a)). Afin d'améliorer les performances de la procédure d'entraînement, des couches de normalisation d'instance sont utilisées sur le résultat de chaque convolution, avant application de la fonction d'activation.

Lors de l'encodage, la taille des cartes de caractéristiques est divisée par 4 ; leur nombre passe successivement de 1 (entrée) à 64. La partie résiduelle est composée de six blocs. Chaque bloc transforme l'entrée grâce à deux couches de convolution puis somme le résultat avec l'entrée. Enfin, la partie décodeur double la taille de chaque carte de caractéristiques puis les transforme par le biais d'une couche de convolution. La dernière couche de convolution produit deux images 3D : la première sera transformée en carte de probabilité de classe (en utilisant une activation sigmoïde) ; la seconde sera additionnée avec l'image interpolée originale.

Architecture du discriminateur Le réseau du discriminateur est entièrement convolutif. Il prend en entrée une image HR et une carte de segmentation. Le discriminateur contient cinq couches de convolution avec un nombre croissant de noyaux, augmentant par un facteur 2 de 32 à 512 noyaux (voir Figure 1(b)). Chaque convolution (sauf la dernière) est effectuée avec un pas de deux, ce qui permet de réduire successivement la taille des cartes de caractéristiques par deux jusqu'à arriver à une représentation sous forme d'un scalaire (lorsque l'entrée est de taille 64^3).

3.3 Entraînement et évaluation

Avant chaque application du réseau, chaque image LR est normalisée (division par la plus grande intensité de l'image) et interpolée via des splines cubiques. Ces deux étapes sont effectuées lors de l'entraînement et de l'évaluation de la méthode. Avant d'appliquer le réseau, les images sont également divisées en patchs d'une taille donnée.

Entraînement Les données d'entraînement sont constituées de patchs d'images interpolées (entrée du réseau) et de patchs de cartes de segmentation et d'images HR. Le choix de traiter les images par patch permet de limiter la mémoire RAM nécessaire pour l'entraînement et ainsi permettre d'effectuer l'entraînement en GPU. À partir de ces données d'entraînement, le discriminateur et le générateur sont entraînés de la manière suivante. Pour chaque mise à jour des poids du générateur, les poids du discriminateur sont mis à jour cinq fois. La méthode d'optimisation choisie est Adam avec les mêmes paramètres que ceux de l'article Kingma et Ba (2014). La taille de batch est de 32 patchs de taille 64^3, créés avec un décalage de 20 entre chacun. Enfin λ_{gp} et λ_{adv} sont respectivement fixés à 100 et 0.001. Pour chaque entraînement effectué, le nombre maximal d'*epochs* est fixé à 200 et les poids finaux (parmi les poids de chaque *epoch*) sont ceux qui maximisent la performance sur un ensemble de données de test.

Evaluation La figure 2 illustre le processus d'évaluation de la méthode.

4 Resultats

4.1 Données

Nous travaillons sur la base de données IRM dHCP [2]. Cette base contient 40 IRM cérébrales de nouveau-nés imagés à terme. En plus de ces acquisitions, la base dHCP fournit, pour chaque sujet, des cartes de segmentation du cerveau. Nous nous intéressons ici à la carte de segmentation (binaire) du cortex cérébral.

4.2 Pré-traitement : génération d'images LR

Toutes les images de la base dHCP sont isotropes et de résolution suffisante ($0.5 \times 0.5 \times 0.5$ mm^3) pour être considérées de haute résolution. Par conséquent, afin de former un modèle de super-résolution, nous devons déterminer des images LR correspondantes pour ces couples images HR / cartes de segmentation. Les images LR sont générées en utilisant le modèle proposé par Greenspan (2008). En notant X l'image HR et X_{LR} l'image LR associée :

$$X_{LR} = H_{\downarrow} B X \tag{10}$$

où B est une matrice floue et $H_{\downarrow}$ est une décimation par sous-échantillonnage. En particulier, nous considérons un filtre gaussien B avec un écart-type :

$$\sigma = \frac{\text{res}}{2\sqrt{2 \log 2}} \tag{11}$$

où res est la résolution de l'image LR, ici fixé à $0.5 \times 0.5 \times 3$ mm^3 (avec une forte anisotropie destinée à prendre en compte les conditions réelles d'acquisition des données cliniques).

4.3 Métriques utilisées

Segmentation Afin d'évaluer la qualité des résultats de segmentation, nous considérons le score de Dice, qui est une mesure standard (comprise entre $[0, 1]$) pour évaluer une segmentation S par rapport à une vérité-terrain G :

$$\text{Dice}(S, G) = \frac{2|S \cap G|}{|S| + |G|} \tag{12}$$

En particulier, plus le Dice est proche de 1, plus S et G sont semblables.

Super-résolution Mesurer la performance des algorithmes SR est moins simple. En effet, pour mesurer la similitude d'aspect visuel entre deux images, une distance entre l'intensité des voxels SR et HR peut ne pas être suffisante. La performance de la reconstruction SR est alors mesurée par deux indices : le PSNR et le SSIM (Wang et al., 2004), définis comme :

$$\text{PSNR}(X, Y) = 10 \log_{10} \frac{\left(\max_i X_i\right)^2}{\frac{1}{|X|} \sum_i |X_i - Y_i|^2} \tag{13}$$

2. `http://www.developingconnectome.org`

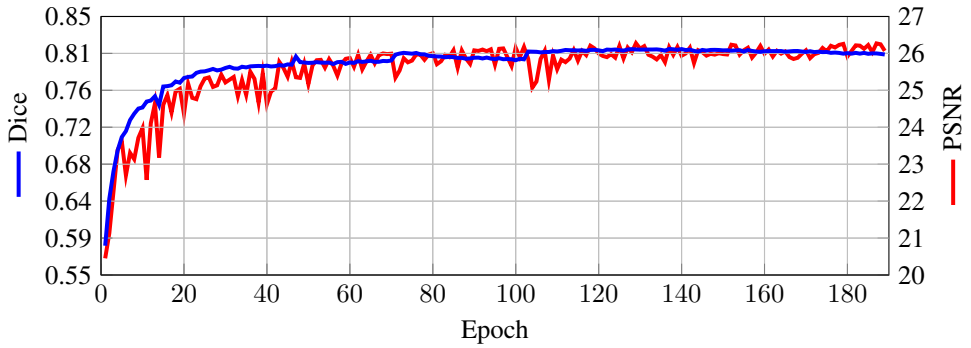


FIG. 3 – *Dice et PSNR moyens à chaque* epoch*, mesurés sur les huit images test de dHCP.*

où X_i et Y_i sont les valeurs de X et Y au point i, respectivement, et :

$$\text{SSIM}(X,Y) = \frac{(2\mu_X\mu_Y + c_1) + (2\sigma_{XY} + c_2)}{(\mu_X^2 + \mu_Y^2 + c_1)(\sigma_X^2 + \sigma_Y^2 + c_2)} \tag{14}$$

où μ_X (resp. μ_Y) est la moyenne des valeurs X (resp. Y), σ_X (resp. σ_Y) est l'écart-type des valeurs X (resp. Y), σ_{XY} est la covariance entre les valeurs X et Y, et c_1, c_2 sont des stabilisateurs numériques liés quadratiquement à la dynamique de l'image.

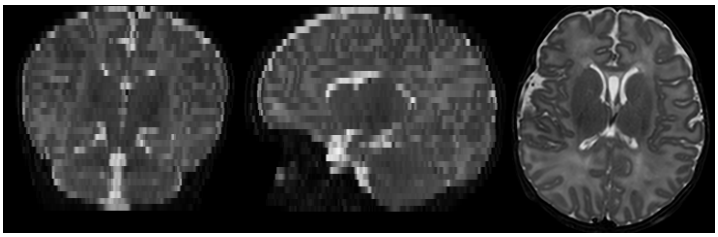
Plus la valeur de ces indices est élevée, meilleure est la similitude entre les deux images.

4.4 Convergence (entraînement)

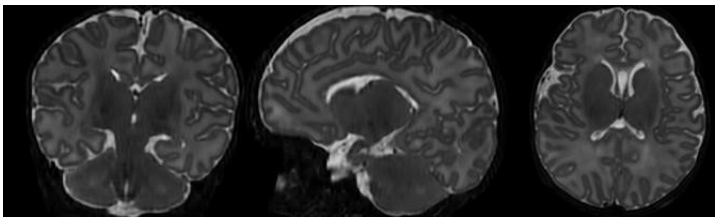
Tout d'abord, nous observons l'évolution des scores Dice et PSNR, tout au long de l'entraînement. Les patchs sur les image tests sont créés de la même façon que ceux de la base d'entraînement. Le PSNR et le Dice sont ensuite calculés pour chaque patch. Les PSNR et Dice finaux sont obtenus en établissant la moyenne des valeurs calculées dans chaque patch. Les résultats sont représentés dans la Figure 3, qui fournit l'évolution des scores de Dice et de PSNR à la fin de chaque *epoch*. Le Dice initial a une valeur très faible, proche de $0,5$, et augmente ensuite jusqu'à $0,8$. Il semble converger aux alentours de l'*epoch* 100. Le PSNR converge également, mais de façon plus bruitée. Cependant, la taille des pics diminue alors que le score tend à se stabiliser, suivant le même comportement que le Dice.

4.5 Résultats obtenus sur une base test

Les résultats présentés dans cette partie ont été obtenus avec des patchs de taille 128^3 voxels (pour des considérations de capacité lors de traitement sur GPU) et un décalage de 30 voxels entre patchs successifs. La figure 4 fournit un exemple de résultat obtenu sur une image de la base test de dHCP. Les résultats quantitatifs présentés ci-après ont été calculés à partir des 8 images de dHCP utilisées comme base de données test. Le tableau 1 résume les Dices de notre méthode (SegSRGAN), de la méthode IMAPA (Tor-Díez et al., 2018) et de DrawEM (Makropoulos et al., 2014). Comme dans un contexte clinique typique, ces méthodes



(a) Image LR (initiale)



(b) Image SR (reconstruite)

(c) Carte de segmentation du cortex

FIG. 4 – *Image LR de la base test de dHCP (a) et les résultat obtenus sur celle-ci (b) et (c)*

	SegSRGAN	IMAPA	DrawEM
Dice	$0.855(\pm 0.014)$	$0.786(\pm 0.023)$	$0.730(\pm 0.010)$

TAB. 1 – *Moyennes (et écarts-type) du Dice pour les cartes de segmentation estimées, calculées sur huit images test.*

ont été appliquées sur des images interpolées (en utilisant une spline cubique). On observe que, quantitativement, l'approche proposée conduit aux meilleurs résultats de segmentation corticale avec une amélioration significative par rapport aux deux autres méthodes. De plus, comme mentionné par Tor-Díez et al. (2018), l'utilisation de IMAPA appliquée aux images originales HR de dHCP conduit à un score moyen de 0,887 (écart-type de 0,011). Enfin, le résultat obtenu sur les images interpolées ne diminue que de 3% par rapport à IMAPA appliquée sur les images HR. Le tableau 2 résume le PSNR et le SSIM de notre méthode versus une interpolation par spline cubique. On peut observer que les deux scores de qualité pour la reconstruction de l'image SR traduisent de meilleurs résultats avec SegSRGAN qu'avec l'interpolation par spline cubique, qui constitue une base standard de comparaison et est l'entrée du réseau.

	SegSRGAN	Interpolation par spline cubique
PSNR	26.96	24.22
SSIM	0.73	0.63

TAB. 2 – *Valeurs moyennes du PSNR et du SSIM, calculées sur huit images test.*

5 Conclusion

Dans cet article, nous avons proposé une nouvelle solution méthodologique pour effectuer la reconstruction et la segmentation d'images IRM 3D complexes. Les résultats obtenus tendent à prouver la pertinence de notre approche, avec des résultats satisfaisants tant en termes de reconstruction SR que de segmentation. Cependant, il est important de garder en tête que la méthode a ici été entraînée et évaluée pour une reconstruction super-résolution d'une résolution axiale initiale fixée (3 mm) à une autre (0.5 mm). Cependant, prendre en compte une variabilité de la résolution initiale peut également être intéressant d'un point de vue clinique. Cet aspect, ainsi que la segmentation d'autres zones du cerveau, constituent les travaux à venir.

Remerciements Ces travaux ont été soutenus par l'Agence Nationale de la Recherche (contrat ANR-15-CE23-0009) ; l'INSERM et l'Institut Mines Télécom Atlantique (Chaire "Imagerie médicale en thérapie interventionnelle") ; la Fondation pour la Recherche Médicale (contrat DIC2016123636453) ; et l'*American Memorial Hospital Foundation*. Nous remercions NVIDIA pour la fourniture de la carte GPU Titan Xp utilisée dans le cadre de ces recherches.

Références

Ball, G., J. P. Boardman, P. Aljabar, A. Pandit, T. Arichi, N. Merchant, D. Rueckert, A. D. Edwards, et S. J. Counsell (2013a). The influence of preterm birth on the developing thalamocortical connectome. *Cortex 49*(6), 1711–1721.

Ball, G., L. Srinivasan, P. Aljabar, S. J. Counsell, G. Durighel, J. V. Hajnal, M. A. Rutherford, et A. D. Edwards (2013b). Development of cortical microstructure in the preterm human brain. *Proceedings of the National Academy of Sciences of the United States of America 110*(23), 9541–9546.

Charbonnier, P., L. Blanc-Féraud, G. Aubert, et M. Barlaud (1997). Deterministic edge-preserving regularization in computed imaging. *IEEE Transactions on Image Processing 6*(2), 298–311.

Chen, Y., Y. Xie, Z. Zhou, F. Shi, A. G. Christodoulou, et D. Li (2018). Brain MRI super resolution using 3D deep densely connected neural networks. In *ISBI*, pp. 739–742.

Dubois, J., M. Benders, A. Cachia, F. Lazeyras, R. Ha-Vinh Leuchter, S. V. Sizonenko, C. Borradori-Tolsa, J. F. Mangin, et P. S. Hüppi (2008). Mapping the early cortical folding process in the preterm newborn brain. *Cerebral Cortex 18*(6), 1444–1454.

Greenspan, H. (2008). Super-resolution in medical imaging. *The Computer Journal 52*(1), 43–63.

Gulrajani, I., F. Ahmed, M. Arjovsky, V. Dumoulin, et A. C. Courville (2017). Improved training of Wasserstein GANs. In *NIPS*, pp. 5769–5779.

Hack, M. et A. A. Fanaroff (2000). Outcomes of children of extremely low birthweight and gestational age in the 1990s. *Seminars in Neonatology 5*, 89–106.

Johnson, J., A. Alahi, et L. Fei-Fei (2016). Perceptual losses for real-time style transfer and super-resolution. In *ECCV*, pp. 694–711.

Kingma, D. P. et J. Ba (2014). Adam: A method for stochastic optimization. *CoRR abs/1412.6980*.

Lai, W.-S., J.-B. Huang, N. Ahuja, et M.-H. Yang (2017). Deep Laplacian pyramid networks for fast and accurate super-resolution. *CoRR abs/1704.03915*.

Ledig, C., L. Theis, F. Huszár, J. Caballero, A. Cunningham, A. Acosta, A. Aitken, A. Tejani, J. Totz, Z. Wang, et W. Shi (2017). Photo-realistic single image super-resolution using a generative adversarial network. In *CVPR*, pp. 105–114.

Lefèvre, J., D. Germanaud, J. Dubois, F. Rousseau, I. de Macedo Santos, H. Angleys, J.-F. Mangin, P. S. Hüppi, N. Girard, et F. De Guio (2016). Are developmental trajectories of cortical folding comparable between cross-sectional datasets of fetuses and preterm newborns? *Cerebral Cortex 26*(7), 3023–3035.

Makropoulos, A., I. S. Gousias, C. Ledig, P. Aljabar, A. Serag, J. V. Hajnal, A. D. Edwards, S. J. Counsell, et D. Rueckert (2014). Automatic whole brain MRI segmentation of the developing neonatal brain. *IEEE Transactions on Medical Imaging 33*(9), 1818–1831.

Marlow, N., D. Wolke, M. A. Bracewell, M. Samara, et E. S. Group (2005). Neurologic and developmental disability at six years of age after extremely preterm birth. *The New England Journal of Medicine 352*, 9–19.

Moeskops, P., M. Veta, M. W. Lafarge, K. A. J. Eppenhof, et J. P. W. Pluim (2017). Adversarial training and dilated convolutions for brain MRI segmentation. In *DLMIA and ML-CDS*, pp. 56–64.

Orasanu, E., A. Melbourne, M. J. Cardoso, H. Lomabert, G. S. Kendall, N. J. Robertson, N. Marlow, et S. Ourselin (2016). Cortical folding of the preterm brain: A longitudinal analysis of extremely preterm born neonates using spectral matching. *Brain and Behavior 6*(8), e00488.

Pham, C.-H., A. Ducournau, R. Fablet, et F. Rousseau (2017). Brain MRI super-resolution using deep 3D convolutional networks. In *ISBI*, pp. 197–200.

Tor-Díez, C., N. Passat, I. Bloch, S. Faisan, N. Bednarek, et F. Rousseau (2018). An iterative multi-atlas patch-based approach for cortex segmentation from neonatal MRI. *Computerized Medical Imaging and Graphics 70*, 73–82.

Wang, Z., A. C. Bovik, H. R. Sheikh, et E. P. Simoncelli (2004). Image quality assessment: From error visibility to structural similarity. *IEEE Transactions on Image Processing 13*(4), 600–612.

Yu, Q., A. Ouyang, L. Chalak, T. Jeon, J. Chia, V. Mishra, M. Sivarajan, G. Jackson, N. Rollins, S. Liu, et H. Huang (2016). Structural development of Human fetal and preterm brain cortical plate based on population-averaged templates. *Cerebral Cortex 26*(11), 4381–4391.

Segmentation of axillary lymph nodes in PET/CT scans: First experiments

Diana Lucia Farfan Cabrera[*,**], Nicolas Gogin[**], David Morland[*,***], Dimitri Papathanassiou[*,***], Nicolas Passat[*]

[*] Université de Reims Champagne Ardenne, CReSTIC, 51097 Reims, France
[**] General Electric Healthcare, Buc, France
[***] Département de Médecine Nucléaire, Institut Godinot, Reims, France

Abstract. The analysis of axillary lymph nodes is of crucial importance for the staging of breast cancer. As a consequence, an accurate segmentation of the nodes reached by cancer can constitute a precious help for computer-aided diagnosis. However, due to the size of axillary lymph nodes in PET/CT images, and to the low resolution of PET data where their abnomal metabolic hyperactivity may be observed, segmentation remains a challenging task. We investigate the relevance of considering axillary lymph nodes segmentation from PET/CT images, based on Convolutional Neural Networks (CNNs). To this end, our initial working hypotheses were twofold: first, taking advantage of both anatomical information from CT, for detecting the nodes, and from functional information from PET for detecting the inflammatory ones; second, considering region-based attributes extracted from component-tree analysis of PET images in order to enrich the information natively carried by PET, with features that can hardly be inferred by CNNs directly from the images. We describe our first results, and discuss about the validity of these working hypotheses.

1 Introduction

Breast cancer is one of the most common diseases in women and one of the principal causes of death in females. Approximately 1.38 million cases are detected worldwide per year and as a consequence causes 458,000 deaths. This type of cancer develops from breast tissue; lymph nodes near these regions are then among the first structures to be affected. This motivates the involvement of lymph nodes in the usual TNM protocol dedicated to the staging of breast, that relies on three criteria: size of tumor (T); number of lymph nodes reached by cancer (N); and metastasis state (M).

Positron Emission Tomography (PET), generally coupled with X-ray Computed Tomography (CT) is widely used for imaging purpose in cancer, and in particular in the case of breast cancer (Vercher-Conejero et al., 2015; Krammer et al., 2015; Kaseda et al., 2016; Piva et al., 2017). Whereas PET provides information on the high metabolism of cancerous cells, CT provides anatomical information on the structures of interest, with a high spatial resolution. However, it was observed by Groheux et al. (2016) that PET/CT data experiment two major

limitations. First, patients in early stages of cancer may have a very small quantity of cancerous cells. In this context, PET/CT may not easily allow to detect these few cells. Second, as inflammatory cells have a metabolism similar to cancerous cells, the putative presence of such cells may lead to false positives.

Despite these difficulties, PET/CT data constitute an important source of information that may be used for computer-aided diagnosis in the case of breast cancer. The challenging properties of these bimodal images (low resolution of PET, possible presence of false positives) also argue in favour of developing robust lymph node segmentation methods.

However, the literature specifically dedicated to lymph node segmentation is still rather limited. In this context, Deep Learning (DL) has recently emerged as a promising segmentation paradigm (Ehteshami Bejnordi et al., 2017), that seems to outperform other standard machine learning approaches for this specific task. In particular, it was shown by Wang et al. (2017) that Convolutional Neural Networks (CNNs) (Long et al., 2015), already used for PET/CT co-segmentation by Zhong et al. (2018), is a potentially relevant paradigm.

A second, recent approach dedicated to PET/CT analysis consists of considering hierarchical image models in order to emphasize the mixed spatial-spectral information carried by PET data. In this context, some morphological trees, and in particular the component-tree (Salembier et al., 1998), have been involved in segmentation methods, e.g. for interactive PET segmentation (Grossiord et al., 2015), lymphoma lesion segmentation from PET/CT (Grossiord et al., 2017), or coupled PET/enhanced CT co-segmentation (Alvarez Padilla et al., 2018). These methods rely on the hypothesis that hierarchical image models constitute an efficient data-structure for extracting high-level, region-based features that can be hardly computed by other strategies (Machairas et al., 2016; Conze et al., 2017).

Our initial purpose was to consider jointly both approaches, namely CNNs and hierarchical image models, in order to develop a lymph node automatic segmentation method that will take advantage not only of the complementary information carried by PET and CT, but also of high-level features that could not be implicitly discovered from the native images by deep-learning architectures. We present our methodology, and experimental results that shed light on the successes and failures of the proposed approach, and emphasize the next steps and challenges to be tackled in our work.

2 CNN architecture

Our proposed CNN is based on the U-Net architecture (Ronneberger et al., 2015); see Figure 1. The first 3 layers act as an encoder; they perform volume downsampling that correspond to the CNN feature extraction. The last 3 layers act, symetrically, as a decoder; they perform volume up-sampling. During this up-sampling, some skip connections issued from the input PET image provide information for the output volume reconstruction. The loss function is the Dice score whereas the used optimizer is Nadam (Dozat, 2016).

This architecture is designed for considering multiple inputs, by duplicating the encoder part for each of these inputs. Basically, it uses two inputs, namely the PET image and its associated CT image. Indeed, cancer lymph nodes are structures with a globally round shape, which are perceived in PET as high-intensity, compact structures. However, other tissues and organs with round shape can present a high metabolic activity in PET, thus leading to possible false positives segmentation. In order to discriminate both kinds of structures, nuclear medicine

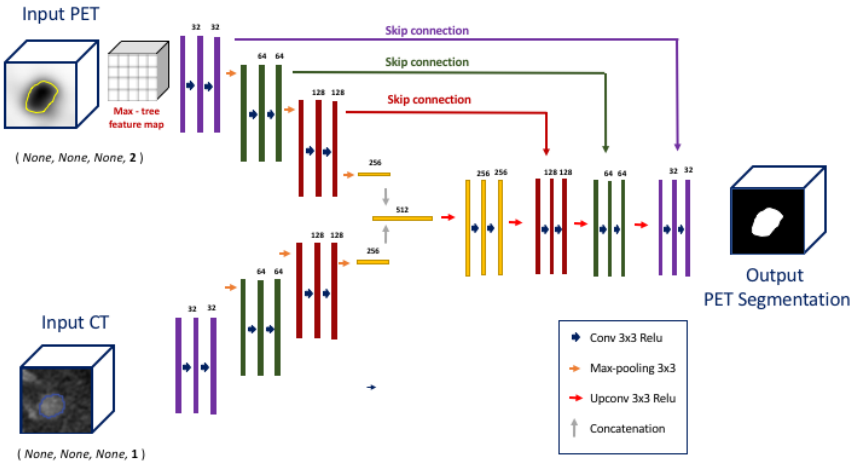


FIG. 1 – *The CNN architecture, illustrated here for two inputs, namely the PET and CT images.*

doctors generally identify cancer lymph nodes by observing both PET and CT data. This is a strategy we chose to reproduce in our CNN, in order to also take into account morphological information in addition to functional one.

Our working hypothesis is that augmenting the information provided as input should improve the segmentation efficiency of the CNN, compared to using the PET data only. This motivates the addition of the CT data. More generally, this also justifies our strategy of computing region-based features from these data, in order to build feature maps involved as supplementary inputs.

3 Data

Our data are composed of 52 PET/CT full-body scans coming from 52 patients with different breast cancer stages. These data may have different resolutions. For instance, 4 of them were obtained with a scanner purchased before 2015, thus resulting in lower resolutions compared to the other images. In order to homogeneize the data, all exams were preprocessed in order to obtain isotropic resolution of 1 mm^3. Since they are acquired during the same exam, the PET and CT volumes were assumed registered, so that one voxel in the CT volume should correspond to the same voxel in the PET volume.

After normalization, each exam contains millions of voxels (e.g. $500 \times 500 \times 800$ volumes). This implies a high cost in terms of both memory and computational resources. In order to reduce this cost, thresholding-based lung segmentation was performed in order to extract the axillary region of interest (ROI). Beyond cost reduction, this also allows one to avoid false positives that may be generated within other parts of the body. Figure 2 illustrates this extraction of the axillary ROI.

We trained our model with 320 lymph nodes of CT full body scans (from neck to coccyx) from these 52 patients' images. Initially, we considered to work with both pathological and

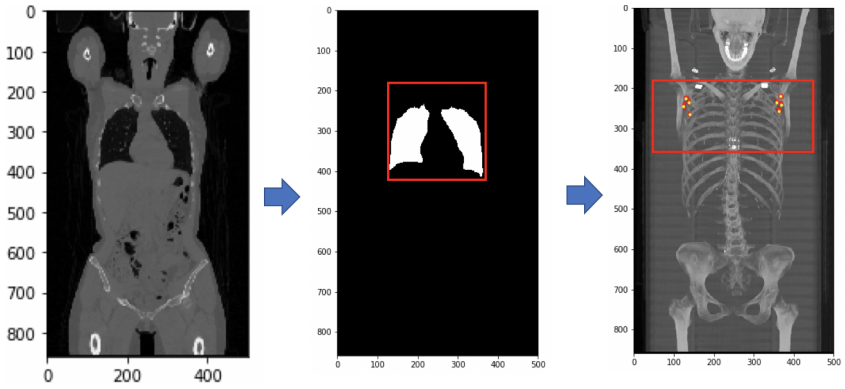


FIG. 2 – *From left to right: original volume; mask of the segmented lungs; extraction of the axilary ROI (coronal view).*

non-pathological lymph nodes. As a consequence, the morphology of these nodes may be different. Indeed, most lymph nodes that contain cells with tumors tend to have a larger volume, and an ellipsoidal morphologically. By contrast, healthy nodes tend to present a "bean" shape, and they are smaller. Figure 3 illustrates the difference between a node that contains tumour cells versus a node that does not.

4 Component-tree-based feature extraction

The component-tree is a lossless hierarchical model dedicated to grey-level images. Basically, a component-tree is a rooted, connected, acyclic graph (i.e. a tree) where each node corresponds to a connected component of a binary level-set of the image. These nodes / connected components are organized with respect to the inclusion relation.

When considering the $\leq$ relation on grey-level values of the image, the component-tree is also called max-tree. In such case, the root of the tree corresponds to the level-set at the lowest value (i.e. 0) where the unique connected component is the whole image support. At the other side of the max-tree, i.e. at the extremities of the branches, the leaves correspond to the flat zones of locally maximal values. In the case of PET images, these regions correspond to high-metabolism areas.

The max-tree can be used as a data-structure allowing to compute features for each node / connected component (Breen and Jones, 1996). More precisely, our purpose is to compute for each relevant node of the max-tree of the PET image, some feature values that gather high-level information that are not directly available to CNNs.

4.1 Node feature extraction

Let us assume that we have computed the max-tree of the considered PET image. For each node, we aim to compute two specific features: (1) the volume; and (2) the compacity

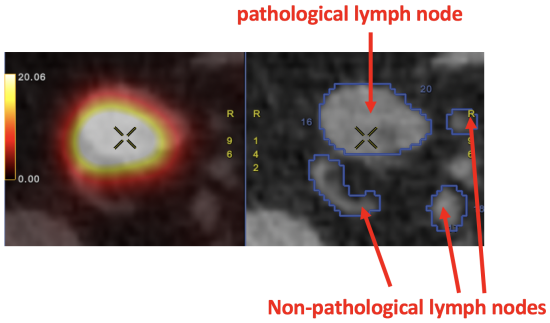


FIG. 3 – *A pathological lymph node next to 3 non-pathological ones (axial view).*

(Grossiord et al., 2015). Indeed, the lesions are of low volume (between 5 and 40 mm^3) and rather compact (round shape).

The volume (V) is proportional to the number of voxels in the connected component of the processed node.

The compacity (C) is more complex to compute, in particular in a Cartesian grid of dimension 3, where the size of a digital boundary of dimension 2 is not defined in a well-posed way. We chose to approximate the compacity by computing the bounding box of the connected component, and by considering as compact an object with a bounding box close to a cube (i.e. with its height, width and depth globally equal). This measure, although not exact, constitutes a fair approximation of compacity, in particular when dealing with spherical targets.

For each node, the two feature values V and C are computed. For the nodes of relevant volume V (i.e. between 5 and 40 mm^3) we associate the node to its actual compacity. The other, over/undersized, nodes are assigned a negative value that means that the feature / the node is non-relevant.

4.2 Feature map construction

Each node of the max-tree of the PET image is then associated with two feature values V and C. However, CNNs consist of performing convolutions with 3D kernels acting on 3D data. This is not the case of a max-tree, which is organized as a dimension-less graph structure. It is then mandatory to embed the feature values computed at the nodes in a feature map defined the same way as the PET image, i.e. in a 3D Cartesian grid.

A voxel of the PET image belongs, in general, to many nodes of the max-tree. Consequently, for a given voxel, many feature values are available for V and C. In particular, it is required to choose, for each voxel, the "most relevant" feature values between the candidate nodes where it lies.

Then, for any voxel x, we study the evolution of the value V for all the nodes from the node of maximal value containing x, until the root node. We search the first strong gradient on V between two successive nodes. It occurs when the current branch is fused with others, and/or when other regions of different semantics merge with the current region containing x.

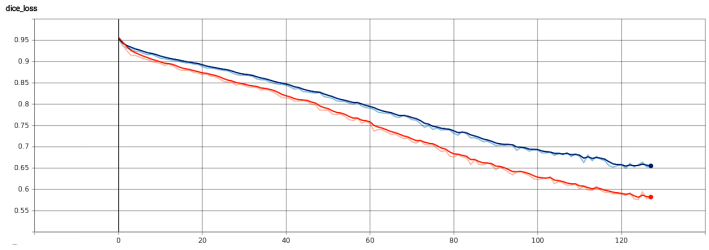


FIG. 4 – *Loss function after 125 epochs. The orange slope corresponds to the training set loss function whereas the blue slope corresponds to the validation set loss function.*

When such gradient is detected, we choose as feature values for x the values V and C of the last node before the gradient occurs.

These feature maps finally computed can then be considered as input of the CNN, for both learning and segmentation.

5 Segmentation results

We trained on 640 patches (320 patches containing lymph nodes and 320 patches containing other structures within the axillary region). The initial learning rate was set to 0.0001 and the number of epochs was 125.

Figure 4 shows the loss function evolution during the training. The loss function value on the validation set converges near 0.6, whereas the loss function value on the training set converges near 0.5.

Figure 5 illustrates segmentation results on 6 different exams from the validation set. One can observe in cases (d–f) that our CNN model is able to distinguish the region that is below the armpits from the rest of the axillary region. However, segmentation results are still not sufficiently accurate when it comes to predicting the contours of the lymph nodes. A possible reason to this fact is that lymph nodes are attached to the lymphatic vessels, and both have the same intensity. It is then difficult, even for medical experts, to correctly delineate the boundary between a lymph node and the lymphatic vessel to which it is connected.

In cases (a–c) we can observe an over-segmentation; larger structures near the heart are segmented in addition to lymph nodes. This over-segmentation influences the Dice score. These segmentation errors can be explained by the fact that the region near the heart tends to be hyperfixating in PET images. In addition, small, round structures around the heart which have a morphology similar to that of the lymph nodes appear to be inflamed or contain cancer cells in several exams. In such cases, using max-tree information could be valuable. Indeed, descriptive features such as lymph node area or lymph node morphology, obtained from the max-tree, could help to discriminate connected components corresponding to such false positives.

On the other hand, our model only relies on 360 examples of lymph nodes. In order to improve the results, it will be mandatory to increase our training set. Data augmentation techniques may contribute to such enrichment of the pool of information available for learning purpose. Nevertheless, additional real examples are also of crucial importance in order to enrich the variability of the training set.

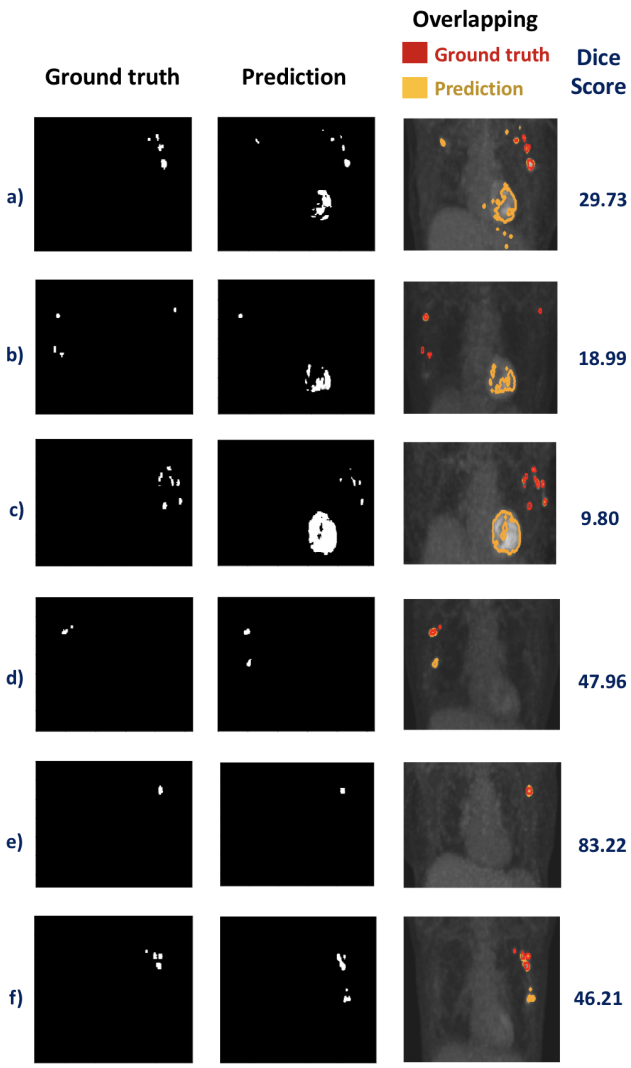


FIG. 5 – *Predictions results on 6 different patients (maximum intensity projection, coronal view, axillary region). First column: ground-truth. Second column: lymph node segmentation results. Third column: comparison of ground-truth contours (in red) and segmentation contours (in yellow). Fourth column: Dice score (over 100) of each segmentation result.*

6 Discussion

6.1 Lymph node clusters

In rare cases, we can observe that patients may contain several lymph nodes visually connected together, due to their small size and close relative positions, see Figure 6. In such cases,

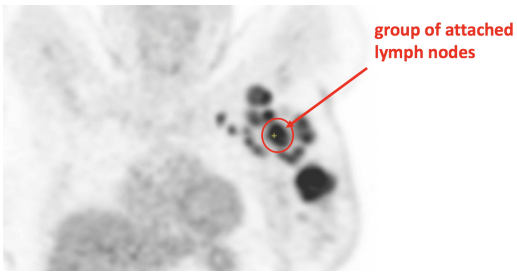


FIG. 6 – *Lymph nodes, forming a cluster that visually appears as a large, single node. PET image, coronal view.*

medical experts consider these clusters as single nodes, since it is impossible for them to discriminate the lymph nodes composition within the clusters. In particular, it is then impossible to determine exactly how many lymph nodes are connected.

Considering these clusters in our training set, and assimilating them as single nodes, may lead to introduce inconsistency into the training process. Indeed these clusters are significantly different from single nodes, both in terms of size and morphology. These cases are then challenging to predict, since they appear as outliers (both in training and testing), compared to usual lymph nodes.

6.2 Registration issues between PET and CT volumes

Both CT and PET images are computed, slice by slice, during a same acquisition. However, the time required for both is quite different, from a few seconds for the CT, to several minutes for the PET. In such conditions, two kinds of movements can alter the PET image acquisition (whereas CT remains robust to such issues). First, breathing has an impact on the position of the axillary lymph nodes, which are located close to the lungs. An acquisition of several minutes will be altered by these physiological artifacts, leading to partial volume effects and/or displacement of the putative position of the nodes in the image. On the other hand, the patient is asked to have his/her arms stretched upwards. This position can rapidly become uncomfortable, leading the patient to slightly move. In such case, the spatial coherence of the lymph nodes between the beginning and the end of the PET image acquisition may be altered.

As a consequence, a same lymph node in the CT image may be located a few millimeters away (and blurred) in the PET image, as illustrated in Figure 7.

In our initial approach, the ground-truths for the lymph nodes contours were defined from the CT image, taking advantage of the accurate morphological information provided by these images. Unfortunately, the movement-based registration issues make this strategy non-valid, since we have no sufficient guarantee of spatial coherence between the lymph node position in the CT and the PET.

A potential solution to tackle this issue may consist of considering multimodal, non-rigid registration procedures. However, such approaches are designed to compute 3D continuous mappings between two images, whereas in our case, the spatial incoherence is related to complex 3D + time movements that cannot be modeled by such kinds of transformations. As a conclusion, it will then be mandatory to define further ground-truth directly from the PET im-

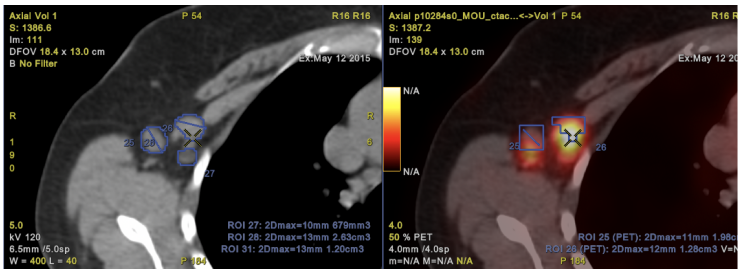


FIG. 7 – *Left: contouring of 2 lymph nodes in CT axial view. Right: Contouring of the same 2 lymph nodes in PET axial view. (See text.)*

ages (and no longer from CT ones). This will induce another difficulty, that lies in the fact that such functional images do not provide crisp frontiers between the nodes and their neighbourhood.

7 Perspective works

Segmentation of lymph nodes in PET/CT images is challenging, especially because these structures are small, sparse and located near the lungs. In our future work, we aim at segmenting tumors using two types of Neural Network architectures using different types of inputs: (1) only gray-level information from PET; (2) only gray-level information from PET/CT; (3) gray-level information from the PET/CT combined with max tree descriptors.

It is important to keep in mind that PET and CT modalities can both bring information, which are indeed complementary. As a consequence, during training using both PET and CT information, it will be important to provide patches in PET and CT, whereas ground-truth mask will correspond to the PET contours only.

In addition, it will be also mandatory to increase both the precision of the segmentation, but also to determine a set of ground-truths sufficiently large, involving at least 1,000 different lymph nodes, in order to reach a sufficient amount of data for further avoiding over-fitting.

Acknowledgements The research leading to these results has been supported by the French *Association Nationale Recherche Technologie* (ANRT).

References

Alvarez Padilla, F., B. Romaniuk, B. Naegel, S. Servagi-Vernat, D. Morland, D. Papathanassiou, and N. Passat (2018). Hierarchical forest attributes for multimodal tumor segmentation on FDG-PET/contrast-enhanced CT. In *ISBI, Procs.*, pp. 163–167.

Breen, E. J. and R. Jones (1996). Attribute openings, thinnings, and granulometries. *Computer Vision and Image Understanding 64*, 377–389.

Conze, P.-H., V. Noblet, F. Rousseau, F. Heitz, V. de Blasi, R. Memeo, and P. Pessaux (2017). Scale-adaptive supervoxel-based random forests for liver tumor segmentation in dynamic

contrast-enhanced CT scans. *International Journal for Computer Assisted Radiology and Surgery 12*, 223–233.

Dozat, T. (2016). Incorporating Nesterov momentum into Adam. In *ICLR Workshop, Procs.*, pp. 2013–2016.

Ehteshami Bejnordi, B., M. Veta, P. Johannes van Diest, B. van Ginneken, N. Karssemeijer, G. Litjens, J. A. W. M. van der Laak, and the CAMELYON16 Consortium (2017). Diagnostic assessment of deep learning algorithms for detection of lymph node metastases in women with breast cancer. *Journal of the American Medical Association 318*, 2199–2210.

Groheux, D., A. Cochet, O. Humbert, J.-L. Alberini, E. Hindié, and D. Mankoff (2016). 18F-FDG PET/CT for staging and restaging of breast cancer. *Journal of Nuclear Medicine 57(Supplement 1)*, 17S–26S.

Grossiord, É., H. Talbot, N. Passat, M. Meignan, and L. Najman (2017). Automated 3D lymphoma lesion segmentation from PET/CT characteristics. In *ISBI, Procs.*, pp. 174–178.

Grossiord, É., H. Talbot, N. Passat, M. Meignan, P. Terve, and L. Najman (2015). Hierarchies and shape-space for PET image segmentation. In *ISBI, Procs.*, pp. 1118–1121.

Kaseda, K., K. Watanabe, K. Asakura, A. Kazama, and Y. Ozawa (2016). Identification of false-negative and false-positive diagnoses of lymph node metastases in non-small cell lung cancer patients staged by integrated 18F-FDG-positron emission tomography/computed tomography: A retrospective cohort study. *Thoracic Cancer 7*, 473–480.

Krammer, J., A. Schnitzer, C. Kaiser, K. Buesing, E. Sperk, J. Brade, S. Wasgindt, M. Suetterlin, S. Schoenberg, E. Sutton, and K. Wasser (2015). 18 F-FDG PET/CT for initial staging in breast cancer patients–Is there a relevant impact on treatment planning compared to conventional staging modalities? *European Radiology 25*, 2460–2469.

Long, J., E. Shelhamer, and T. Darrell (2015). Fully convolutional networks for semantic segmentation. In *CVPR, Procs.*, pp. 3431–3440.

Machairas, V., T. Baldeweck, T. Walter, and E. Decencière (2016). New general features based on superpixels for image segmentation learning. In *ISBI, Procs.*, pp. 1409–1413.

Piva, R., F. Ticconi, V. Ceriani, F. Scalorbi, F. Fiz, S. Capitanio, M. Bauckneht, G. Cittadini, G. Sambuceti, and S. Morbelli (2017). Comparative diagnostic accuracy of 18F-FDG PET/CT for breast cancer recurrence. *Breast Cancer: Targets and Therapy 9*, 461.

Ronneberger, O., P. Fischer, and T. Brox (2015). U-Net: Convolutional networks for biomedical image segmentation. In *MICCAI, Procs. III*, pp. 234–241.

Salembier, P., A. Oliveras, and L. Garrido (1998). Antiextensive connected operators for image and sequence processing. *IEEE Transactions on Image Processing 7*, 555–570.

Vercher-Conejero, J., L. Pelegrí-Martinez, D. Lopez-Azna, and M. del Puig Cózar-Santiago (2015). Positron emission tomography in breast cancer. *Diagnostics 5*, 61–83.

Wang, H., Z. Zhou, Y. Li, Z. Chen, P. Lu, W. Wang, W. Liu, and L. Yu (2017). Comparison of machine learning methods for classifying mediastinal lymph node metastasis of non-small cell lung cancer from 18 F-FDG PET/CT images. *EJNMMI Research 7*, 11.

Zhong, Z., Y. Kim, L. Zhou, K. Plichta, B. Allen, J. Buatti, and X. Wu (2018). 3D fully convolutional networks for co-segmentation of tumors on PET-CT images. In *ISBI, Procs.*, pp. 228–231.

Clustering contraint par apprentissage profond appliqué aux séries temporelles d'images satellites

Baptiste Lafabregue[*,**] Jonathan Weber[*]
Pierre Gançarski[**], Germain Forestier[*]

[*]IRIMAS, University of Haute-Alsace, Mulhouse, France
<prenom>.<nom>@uha.fr,
https://www.irimas.uha.fr
[**]ICube, University of Strasbourg, Strasbourg, France
<nom>@unistra.fr
https://icube.unistra.fr

Résumé. Les avancées dans l'imagerie satellitaire ont généré un nombre sans précédent d'images de télédétection. Les derniers satellites fournissent des images avec une haute fréquence de revisite et très facilement accessibles. Les séries d'images ainsi acquises, sur une même région, peuvent être vues comme des séries temporelles. L'analyse de telles données permet de faire une observation continue et à grande échelle de la terre, avec des applications très diverses, allant de l'occupation du sol en agriculture, au suivi de catastrophe environnementales. Cependant, le manque d'une large quantité de données labellisée empêche d'utiliser directement des méthodes supervisées. A l'opposé, les méthodes non-supervisées ne requièrent aucune connaissance de l'expert, mais donnent souvent des résultats mitigés, ou qui ne correspondent pas aux attentes de l'expert. Dans ce contexte, le clustering contraint, qui est une forme d'algorithme d'apprentissage semi-supervisé, est une alternative offrant un compromis intéressant. Dans cet article, nous explorons l'utilisation de contraintes au sein de méthodes de clustering basées sur l'apprentissage profond appliquées aux séries temporelles d'images satellites. Notre étude expérimentale repose sur la méthode Deep Embedded Clustering et son adaptation qui intègre des contraintes par paires (must-link et cannot-link). Les tests conduits sur un jeu de données composé de 11 images satellites montrent des résultats encourageants et ouvrent de nombreuses perspectives dans l'application aux séries temporelles d'images satellites du clustering contraint basé sur de l'apprentissage profond.

1 Introduction

Les méthodes d'apprentissage profond sont largement utilisées dans un grand nombre de domaines, et font l'objet d'un intérêt grandissant dans la communauté de la télédétection Zhu et al. (2017). Ces méthodes donnent de très bons résultats, mais elles sont fortement dépendantes de la quantité de données disponibles, et plus particulièrement de données labellisées.

La télédétection produit un grand nombre de données, qui sont cependant rarement annotées. Ce manque d'annotation est encore plus marqué pour les séries d'images satellites. Des images satellites peuvent être librement obtenues tous les 5 jours, cependant, de par la complexité des données et le manque de typologie bien définie, il est difficile d'utiliser des approches supervisées. De ce fait, de nombreuses méthodes de clustering sont appliquées dans ce domaine Khiali et al. (2019); Rey et al. (2019). Cependant, même en l'absence de données annotées, l'expert a très souvent une forte connaissance thématique qu'il peut mettre à disposition. Les contraintes permettent d'intégrer une partie de cette connaissance dans le processus d'apprentissage, nos travaux se concentrent uniquement sur les contraintes par paires, les must-link (ML) et les cannot-link (CL). Ces contraintes indiquent que deux instances doivent être assignées au même cluster (must-link) ou à des clusters différents (cannot-link). Les contraintes par paires sont largement utilisées et ont déjà fait l'objet de nombreuses études Basu et al. (2008), ce qui nous permet de facilement obtenir une base de comparaison, ce type de contraintes étant supportées par de nombreuses méthodes.

Dans cet article, nous voulons étudier si on peut tirer avantage des avancées en apprentissage profond à travers une approche basée contrainte qui semble plus appropriée dans le domaine de la télédétection. Après avoir présenté les travaux antérieurs en clustering sur les séries temporelles et le clustering contraint, respectivement dans les sections 2.1 et 2.2, nous présenterons la méthode de clustering contraint d'apprentissage profond utilisée et son adaptation aux séries temporelles 3, puis nous comparerons les résultats à d'autres méthodes classiques de clustering contraint sur des données de télédétection dans la section 4. Enfin, dans la section 5 nous commenterons les résultats obtenus et les perspectives qui en résultent.

2 État de l'art

2.1 Le clustering pour les séries temporelles

Différentes approches de clustering pour les séries temporelles ont été proposées dans la littérature, essentiellement basées sur des méthodes de représentation tel que la transformée en ondelettes discrète Chan et Fu (1999) ou des mesures de similarité tel que Dynamic Time Warping Sakoe et Chiba (1978). Ces méthodes sont ensuite généralement intégrées comme prétraitement, le résultat étant alors utilisé pour alimenter une méthode standard de clustering, tel que les familles de méthodes k-means, k-medoid, clustering spectral ou encore clustering hiérarchique, comme illustré dans Aghabozorgi et al. (2015). Dans ce domaine, de nouvelles propositions sont faites, e.g., la méthode k-shape Paparrizos et Gravano (2015), qui est basée sur une procédure de raffinement itératif qui utilise une version normalisée de la corrélation croisée. Une des difficultés, lors de l'utilisation de séries temporelles, est l'hétérogénéité des sujets abordés et du type des données traitées, allant du nombre d'attributs ou de la longueur de la séquence, au type de corrélation entre éléments, basée sur la forme ou la structure, avec à chaque fois des amplitudes différentes. Cette problématique est largement traitée en apprentissage supervisé par l'utilisation de l'apprentissage de représentation à travers des réseaux de neurones profonds. Récemment, des approches de clustering utilisant ces méthodes d'apprentissage ont été proposées. Elles sont essentiellement basées sur des architectures *end-to-end* qui apprennent simultanément une représentation des données et un clustering, en utilisant un autoencodeur et une couche de clustering attachée à la sortie de l'encodeur Xie et al. (2016);

Guo et al. (2017). Une architecture dérivée de cette dernière a été développée pour les séries temporelles, elle utilise une couche convolutive 1-D suivie par un Bi-LSTM comme autoencodeur, afin de préserver la dimension temporelle dans l'encodage de la représentation, un clustering est alors appliqué sur cette représentation en utilisant une métrique de similarité comme couche de clustering Madiraju et al. (2018).

2.2 Clustering contraint

De nombreuses méthodes ont été proposées pour l'intégration de contraintes en clustering. La plupart d'entre elles consistent en l'extension d'algorithmes standards de clustering tel que k-means Wagstaff et al. (2001) ou du spectral clustering Li et al. (2009), mais il existe aussi des méthodes dédiées, comme par exemple la programmation par contrainte Duong et al. (2017). Une étude comparative a d'ailleurs été menée sur ce sujet Lampert et al. (2018). Dans le domaine de l'apprentissage profond, la plupart des méthodes semi-supervisées font référence à des méthodes d'auto-apprentissage ou à d'autres moyens d'intégrer des connaissances dans une tâche supervisée. Mais récemment, des méthodes ont été proposées pour inclure des contraintes par paires dans du clustering basé sur l'apprentissage profond Zhang et al. (2019); Ren et al. (2019). Ces deux articles utilisent les contraintes au niveau de leur fonction de coût qui va alors maximiser la similarité entre les encodages d'instances d'une contrainte must-link et respectivement la minimiser pour une contrainte cannot-link. Notre travail se base sur les travaux de Zhang et al. (2019). Nos contributions consistent en l'adaptation de cette approche aux séries temporelles et à l'étude des résultats sur des séries temporelles d'images satellites. La méthode proposée par Zhang et al. (2019) supporte plusieurs types de contraintes, mais, comme précisé précédemment, nos travaux se concentrent sur les contraintes ML et CL.

3 La méthode Deep Constrained Clustering et son adaptation aux séries temporelles

La méthode Deep Constrained Clustering (DCC) proposée par Zhang et al. (2019) est basée sur une méthode de clustering par apprentissage profond, Deep Embedded Clustering (DEC) (Xie et al., 2016) et sa version améliorée (IDEC) (Guo et al., 2017). Dans un premier temps, nous présenterons la méthode IDEC, puis son extension pour l'intégration de contraintes par la méthode DCC et finalement les adaptations apportées pour l'utilisation sur des séries temporelles.

3.1 Improved Deep Embedded Clustering

Deep Embedded Clustering (DEC), lors de la phase initiale, entraîne un autoencodeur ($x_i = g(f(x_i))$) puis supprime le décodeur. Le réseau ainsi obtenu, qui consiste en l'encodeur ($z_i = f(x_i)$), est affiné en optimisant la divergence de Kullback-Leiber entre deux distributions Q et P. Q est un *soft cluster assignment* , où pour chaque instance i on calcule un vecteur q_i de longueur k, k étant le nombre de clusters souhaités, où q_{ij} est le degré de confiance que l'instance i appartienne au cluster j. P est la distribution cible qui est définie en fonction de Q pour renforcer la prédiction faite pour chaque cluster, comme défini ci-dessous. Nous obtenons

donc la fonction de coût de clustering L_c :

$$L_c = KL(P|Q) = \sum_i \sum_j p_{ij} log \frac{p_{ij}}{q_{ij}} \tag{1}$$

où q_{ij} est la similarité entre l'encodage de x_i, z_i et le centroïde du cluster j, μ_j, mesurée par une t-distribution de Student (Maaten et Hinton, 2008) :

$$q_{ij} = \frac{(1 + ||z_i - \mu_j||^2)^{-1}}{\sum_j (1 + ||z_i - \mu_j||^2)^{-1}} \tag{2}$$

et p_{ij} est la distribution cible, tel que :

$$p_{ij} = \frac{q_{ij}^2 / \sum_i q_{ij}}{\sum_j (q_{ij}^2 / \sum_i q_{ij})} \tag{3}$$

L'ensemble des centroïdes μ est initialisé avec les centres d'un k-means exécuté sur la représentation z. L'amélioration apportée par IDEC est de garder le décodeur et la fonction de coût de reconstruction de l'autoencodeur L_r même après la phase initiale. L'intuition derrière cela est que la fonction de coût du clustering, en distordant l'espace de représentation, peut altérer la représentativité de l'encodage généré et de ce fait la performance du clustering. Ainsi, la fonction de coût de clustering augmente la séparabilité des clusters, tandis que la fonction de coût de reconstruction maintient la représentativité de l'encodage appris par l'autoencodeur. La fonction de coût de reconstruction correspond à l'erreur quadratique moyenne entre l'instance en entrée et sa reconstruction faite par l'autoencodeur. On a donc la fonction de coût global qui est définie comme suit :

$$L = L_r + \gamma * L_c \tag{4}$$

où $\gamma > 0$ est le coefficient qui contrôle le degré de distorsion de l'espace de représentation.

3.2 Intégration des contraintes

L'extension de DEC pour incorporer les contraintes est basée sur la méthode Deep Constrained (DCC). Ils proposent quatre types de contraintes, mais nous n'avons pris en considération que les contraintes par paires, car elles sont supportées par de nombreux autres méthodes de clustering contraint.

La fonction de coût utilisée pour l'ensemble ML des contraintes must-link est définie par :

$$l_{ML} = L_r - \gamma_{ML} * \sum_{(a,b) \in ML} log \sum_j q_{aj} * q_{bj} \tag{5}$$

De manière équivalente, la fonction de coût pour l'ensemble CL des contraintes cannot-link est définie par :

$$l_{CL} = - \sum_{(a,b) \in CL} log(1 - \sum_j q_{aj} * q_{bj}) \tag{6}$$

De manière intuitive, la fonction de coût pour les ML va favoriser des instances avec le même *soft assignment* et celle pour les CL qui ont le *soft assignment* opposé. La fonction de coût pour les ML est régularisée par l'ajout de la fonction de coût de reconstruction L_r pondérée par un facteur $\gamma_{ML} > 0$, de manière identique à L_c, afin d'éviter de tomber dans une solution triviale qui serait d'assigner toutes les instances concernées à un seul cluster.

3.3 Application aux séries temporelles d'images satellites

L'objectif principal de ces expériences est d'évaluer si ce nouveau type de clustering contraint est pertinent sur des séries temporelles d'images satellites, et de le comparer aux méthodes de l'état de l'art. Nous avons testé la version originale de DCC, qui est composée de couches totalement connectées. Pour cette version, l'architecture est identique mais les séries temporelles multivariées sont réduites à une dimension. Nous proposons également une version modifiée de DCC avec des convolutions 1D, ces dernières ayant montré leur efficacité pour les séries temporelles en classification supervisée Fawaz et al. (2019). Dans cette nouvelle version, nous gardons la dimension originale des séries temporelles et le réseau est composé uniquement de couches convolutionnelles 1D suivies à chaque fois d'une couche de *batch-normalisation*, une couche de *global average pooling* est placée à la fin juste avant la couche finale d'encodage. Cette dernière reste une couche totalement connectée.

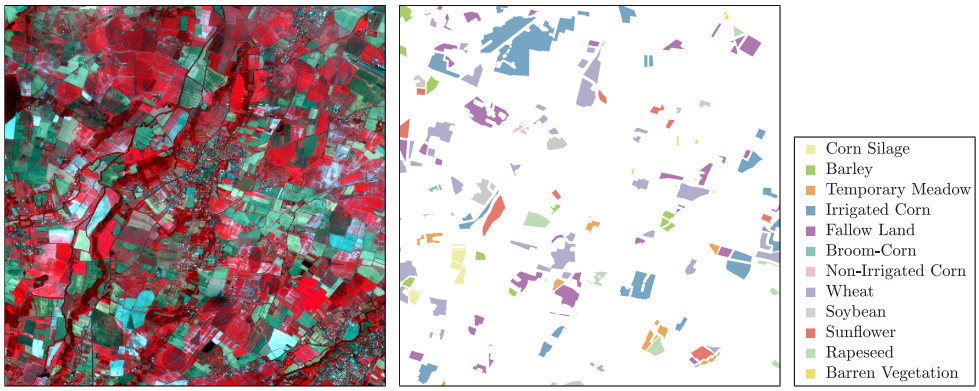
4 Expériences et résultats

Pour ces expérimentations, nous avons appliqué ces méthodes sur un problème de classification de cultures agricoles qui est un champ de recherche important en télédétection et qui fait l'objet de nombreuses études (Sicre et al., 2014; Garnot et al., 2019).

4.1 Jeu de données et paramètres expérimentaux

Le jeu de données est composé de 12 classes de cultures agricoles (blé, maïs irrigué, etc. voir Fig. 1c), situées près de Toulouse (Sud-Est de la France). Les images d'origine [1] sont composées de 11 images multi-spectrales (vert, rouge, proche-infrarouge) de 1000×1000 pixels réparties de manière non-uniforme entre le 15/02/07 et le 20/10/07 et acquises par le satellite Formosat-2. Une des images est présentée dans la Fig. 1a. Le jeu de données est composé de pixels sélectionnés aléatoirement dans les régions annotées (voir Fig. 1b) qui ont été ensuite répartis en jeu d'entraînement et de test, composés de respectivement 1974 et 9869 série temporelles de pixels. Les algorithmes de clustering sont entraînés et évalués uniquement sur le jeu de test. Le jeu d'entraînement est seulement utilisé pour fixer les hyperparamètres des méthodes si besoin. Les contraintes sont générées depuis le jeu de test en sélectionnant aléatoirement des paires de pixels et en créant une contrainte ML ou CL en fonction de leurs labels. La donnée de référence est basée sur la déclaration des agriculteurs à l'AEE pour la Politique Agricole Commune. Pour tester la sensibilité des méthodes au nombre de contraintes, nous avons défini trois niveaux de taille de jeu de contraintes : 5%, 15% et 50% de la cardinalité du jeu de test $N = 9869$ (une très petite fraction de l'ensemble des contraintes possibles, $\frac{1}{2}N[N-1]$). Pour l'évaluation nous avons utilisé la mesure d'Adjusted Rand Index (ARI) et le taux de satisfaction de contraintes par le résultat du clustering (Sat.) moyennés sur 10 exécutions. Pour chaque niveau de contraintes, 10 ensembles ont été générés aléatoirement, un par exécution. Les mêmes 10 ensembles sont utilisés pour chaque méthode, pour s'assurer que chaque méthode bénéficie des mêmes contraintes.

1. Mises à disposition par le *Centre d'Études Spatiales de la Biosphère (CESBIO) Unité Mixte de Recherche CNES-CNRS-IRD-UPS*, Toulouse, France.



(a) Image (b) Donnée de référence (c) Noms des classes

FIG. 1 – *Une image de la série temporelle :* 12 *classes, et* 11 *dates (*t_4 *affiché).*

4.2 Méthodes de comparaisons et paramétrisation

Afin d'avoir une base de comparaison, nous avons ajouté à DCC et DCC-Conv, quatre méthodes standards de clustering par contraintes. Nous avons utilisé une variation de k-means contraint (COP-KMeans) Wagstaff et al. (2001), un algorithme de clustering spectral contraint (Spec) Li et al. (2009), une méthode déclarative de programmation par contraintes (CPClustering) Duong et al. (2017) et une méthode de clustering collaboratif contraint (SAMARAH) Forestier et al. (2010) (utilisant 3 k-means). Pour illustrer la variabilité induite par le choix de la métrique sélectionnée, nous avons utilisé la distance Euclidienne et la métrique DTW Sakoe et Chiba (1978) [2]. Mis à part CPClustering, qui ne requiert aucun paramètre, les autres méthodes nécessitent au moins le nombre de clusters, sinon les paramètres par défaut ont été utilisés. La seule exception est la méthode Spec qui nécessite l'apprentissage d'hyperparamètres appris par grid search sur le jeu d'entraînement. Pour les méthodes de clustering par apprentissage profond, nous avons suivi le paramétrage proposé dans DEC et IDEC mais comme les résultats n'étaient pas stables (voir section 4.3 pour plus de détails) nous avons fait des modifications mineures. Pour la dimension de la couche d'encodage nous l'avons fixé à 2 au lieu de 10, car cela semblait donner plus de stabilité lors de l'apprentissage. Pour DCC, l'encodeur est fixé aux dimensions $d-500-500-2000-2$, où $d = l*f$, l étant la longueur de la série en entrée et f le nombre d'attributs de la série. Pour DCC-Conv les dimensions sont $l*t-128-256-128-2$, avec respectivement des filtres 1D de dimension $8-5-3$, suivant ainsi les recommandations dans Wang et al. (2017). Pour les deux versions, les dimensions des décodeurs sont en miroir de celles de l'encodeur. Pour les deux, la fonction d'optimisation utilisée est SGD avec un *momentum* de 0.9 et une valeur de *decay* de $1e-6$, pour compenser la variabilité mentionnée précédemment. γ et γ_{ML} sont tous deux fixés à 0.1 comme décrit dans les articles d'origine. [3]

2. Les méthodes utilisées pour la comparaison sont disponibles sur `https://icube-forge.unistra.fr/lampert/TSCC`

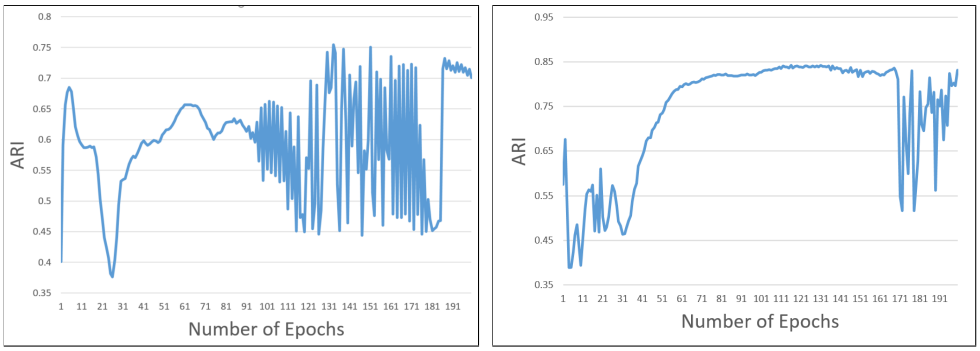
3. Le code utilisé pour cet article peut être trouvé sur : `https://github.com/blafabregue/DeepConstrainedClustering`

TAB. 1 – *ARI et satisfaction des contraintes, avec et sans contraintes. La meilleur performance par pourcentage de contraintes et métrique est mise en gras. La Sat., dans le cas sans contrainte, est mesurée par une moyenne sur les ensembles de contraintes à 50%.*

Méthode	Distance	Sans-contraintes		5%		15%		50%	
		ARI	Sat.	ARI	Sat.	ARI	Sat.	ARI	Sat.
COP-KMeans	DTW	0.426	0.812	0.416	**1.00**	0.407	**1.00**	0.436	**1.00**
Wagstaff et al. (2001)	Eucl.	0.420	0.807	0.406	**1.00**	0.443	**1.00**	0.369	**1.00**
Spec	DTW	**0.531**	**0.840**	**0.683**	0.867	**0.725**	0.888	0.786	0.911
Li et al. (2009)	Eucl.	**0.737**	**0.885**	0.671	0.854	0.702	0.875	0.781	0.916
CPClustering	DTW	0.437	0.803	0.469	**1.00**	0.510	**1.00**	0.589	**1.00**
Duong et al. (2017)	Eucl.	0.681	0.413	0.650	**1.00**	0.542	**1.00**	0.510	**1.00**
SAMARAH	DTW	0.406	0.802	0.597	0.870	0.637	0.867	0.681	0.878
Forestier et al. (2010)	Eucl.	0.463	**0.817**	**0.691**	0.884	**0.714**	0.890	0.702	0.885
DCC Zhang et al. (2019)		0.703	0.885	0.550	0.852	0.448	0.816	0.615	0.862
DCC-Conv		0.508	0.833	0.497	0.844	0.491	0.819	**0.820**	0.936

4.3 Résultats

Les résultats, avec et sans contraintes, sont présentés dans la Table 1. Spec donne globalement les meilleurs résultats, mais cela doit être relativisé, comme mentionné précédemment, par le fait que cette méthode nécessite l'apprentissage d'hyperparamètres (sans contraintes, le résultat moyen est de 0.367 d'ARI, contre 0.737 pour les paramètres retenus). On peut aussi remarquer que les résultats varient fortement selon la métrique utilisée. C'est également le cas pour quasiment toutes les autres méthodes basées métrique. Les versions par apprentissage profond ne font mieux que dans le cas où le nombre de contraintes est très élevé et uniquement pour la version convolutionnelle. Mais le plus surprenant est le comportement quasiment opposé des deux versions. DCC donne de relativement bons résultats sans-contraintes, mais les contraintes ont un fort effet négatif. Cet effet négatif a déjà pu être étudié en clustering contraint et il peut être observé également pour les autres méthodes, à l'exception de SAMARAH. Il a été observé dans Lampert et al. (2018) que si une méthode capture déjà bien la structure de la donnée sans contrainte, elle ne va pas bénéficier de l'ajout de contraintes, ceci peut être mesuré par la Sat. sur l'exécution sans-contrainte. Cela semble être le cas ici, ce qui va en opposition des observations faites par Zhang et al. (2019), qui concluaient en l'absence de cet effet. Dans notre cas, cela peut s'expliquer par la forte présence de bruit dans les contraintes (route à travers les champs, pixels en frontière des champs, ...). DCC-Conv, de son côté, obtient de bons résultats avec les contraintes, mais seulement quand leur nombre est assez élevé. En effet, la manière dont les contraintes sont utilisées lors de la descente de gradient, ne force pas l'algorithme à respecter les contraintes, il semble cependant qu'un grand nombre de contraintes permet de bien faire redescendre l'information dans les poids du réseau. Dans Lampert et al. (2018), il a été observé que les méthodes ne bénéficient pas de manière significative d'un nombre croissant de contraintes, mais plutôt de contraintes plus informatives et cohérentes. Dans le cas de DCC-Conv, nous pouvons légitimement nous demander si le réseau ne commence pas à apprendre le jeu de données lui-même et non la structure de la donnée, l'algorithme étant entraîné et testé sur le même jeu de données (celui de test). Cependant en



(a) DCC avec 50% de contraintes (b) DCC-COnv avec 50% de contraintes

FIG. 2 – *Évolution de l'ARI au cours du processus d'apprentissage de DCC et DCC-Conv.*

utilisant le modèle appris sur le jeu d'entraînement nous avons remarqué que nous obtenons un score très proche (0.81 d'ARI contre 0.82 sur le test), ce qui relativise cette explication.

Deux autres points doivent être abordés, qui ne sont pas directement visibles dans le tableau 1. Tout d'abord l'écart type augmente fortement avec l'ajout de contraintes et tend à diminuer quand le nombre de contraintes augmente, que ce soit pour DCC ou DCC-Conv (i.e. pour DCC-Conv, l'écart type est respectivement, sans contrainte, avec 5%, 15%, 50% de 0.005, 0.069, 0.010 et 0.015). L'effet des contraintes peut donc être une forte source de perturbation, cela peut s'expliquer par le bruit dans notre cas, mais quand ce nombre augmente cela régularise cet effet. Le second point est que le réseau n'est pas stable durant l'apprentissage, c'est le cas pour les deux versions, mais d'une manière plus amplifiée pour DCC. Même si nous avons pu réduire cette variabilité en intégrant un *decay* dans la fonction d'optimisation, celle-ci reste importante (i.e. pour DCC, l'ARI varie entre 0.55 et 0.74 avec un cas extrême à 0.45). Une illustration de cette instabilité peut être vu sur la figure 2. Cela se produit essentiellement quand des contraintes sont ajoutées, mais aussi sans-contrainte, mais dans une plus faible amplitude. Cela semble venir en partie du fait que la distribution cible est mise à jour à chaque époque, le réseau optimise donc vers une cible qui elle-même peut potentiellement fortement varier.

5 Conclusion

Le clustering par apprentissage profond démontre qu'il peut obtenir de bons résultats sur des séries temporelles en télédétection, avec ou sans contraintes. De plus, ces méthodes permettent de s'affranchir du choix d'une représentation ou d'une métrique, celle-ci étant apprise par l'autoencodeur, ce qui rend la tâche de l'expert du domaine plus simple. Cependant les résultats montrent que le choix des hyperparamètres (i.e. dimensions des couches, plus particulièrement celle de l'encodage, choix de la fonction d'optimisation) est important et requiert de plus amples investigations, ce qui vient contrebalancer en partie cet avantage. Il y a également deux points importants que nous souhaitons approfondir. Tout d'abord, DCC n'est pas aussi robuste que ne le laissaient penser les expériences précédentes. Second point, l'instabilité lors de la phase d'apprentissage et l'impact des contraintes dégradent fortement la moyenne

des résultats. Nous planifions d'étudier les facteurs qui entraînent ces problèmes dans notre cas, ainsi que l'effet du bruit dans les contraintes ou voir si cela se généralise à d'autres types de séries temporelles (domaine différent, taille du jeu de donnée, longueur des séries, ...).

Remerciements

Ces travaux ont été financés dans le cadre de l'ANR TIMES (financement ANR-17-CE23-0015) de l'Agence Nationale de la Recherche.

Références

Aghabozorgi, S., A. S. Shirkhorshidi, et T. Y. Wah (2015). Time-series clustering–a decade review. *Information Systems 53*, 16–38.

Basu, S., I. Davidson, et K. Wagstaff (2008). *Constrained clustering : Advances in algorithms, theory, and applications*. CRC Press.

Chan, K.-P. et A. W.-C. Fu (1999). Efficient time series matching by wavelets. In *Proceedings 15th International Conference on Data Engineering (Cat. No. 99CB36337)*, pp. 126–133. IEEE.

Duong, K.-C., C. Vrain, et al. (2017). Constrained clustering by constraint programming. *Artificial Intelligence 244*, 70–94.

Fawaz, H. I., G. Forestier, J. Weber, L. Idoumghar, et P.-A. Muller (2019). Deep learning for time series classification : a review. *Data Mining and Knowledge Discovery*, 1–47.

Forestier, G., P. Gançarski, et C. Wemmert (2010). Collaborative clustering with background knowledge. *Data & Knowledge Engineering 69*(2), 211–228.

Garnot, V. S. F., L. Landrieu, S. Giordano, et N. Chehata (2019). Time-space tradeoff in deep learning models for crop classification on satellite multi-spectral image time series. *arXiv preprint arXiv :1901.10503*.

Guo, X., L. Gao, X. Liu, et J. Yin (2017). Improved deep embedded clustering with local structure preservation. In *IJCAI*, pp. 1753–1759.

Khiali, L., M. Ndiath, S. Alleaume, D. Ienco, K. Ose, et M. Teisseire (2019). Detection of spatio-temporal evolutions on multi-annual satellite image time series : A clustering based approach. *International Journal of Applied Earth Observation and Geoinformation 74*, 103–119.

Lampert, T., B. Lafabregue, N. Serrette, G. Forestier, B. Crémilleux, C. Vrain, P. Gancarski, et al. (2018). Constrained distance based clustering for time-series : a comparative and experimental study. *Data Mining and Knowledge Discovery 32*(6), 1663–1707.

Li, Z., J. Liu, et X. Tang (2009). Constrained clustering via spectral regularization. In *2009 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 421–428. IEEE.

Maaten, L. v. d. et G. Hinton (2008). Visualizing data using t-sne. *Journal of machine learning research 9*(Nov), 2579–2605.

Madiraju, N. S., S. M. Sadat, D. Fisher, et H. Karimabadi (2018). Deep temporal clustering : Fully unsupervised learning of time-domain features. *arXiv preprint arXiv :1802.01059*.

Paparrizos, J. et L. Gravano (2015). k-shape : Efficient and accurate clustering of time series. In *Proceedings of the 2015 ACM SIGMOD International Conference on Management of Data*, pp. 1855–1870. ACM.

Ren, Y., K. Hu, X. Dai, L. Pan, S. C. Hoi, et Z. Xu (2019). Semi-supervised deep embedded clustering. *Neurocomputing 325*, 121–130.

Rey, D. M., M. Walvoord, B. Minsley, J. Rover, et K. Singha (2019). Investigating lake-area dynamics across a permafrost-thaw spectrum using airborne electromagnetic surveys and remote sensing time-series data in yukon flats, alaska. *Environmental Research Letters 14*(2), 025001.

Sakoe, H. et S. Chiba (1978). Dynamic programming algorithm optimization for spoken word recognition. *IEEE transactions on acoustics, speech, and signal processing 26*(1), 43–49.

Sicre, C. M., F. Baup, et R. Fieuzal (2014). Determination of the crop row orientations from formosat-2 multi-temporal and panchromatic images. *ISPRS journal of photogrammetry and remote sensing 94*, 127–142.

Wagstaff, K., C. Cardie, S. Rogers, S. Schrödl, et al. (2001). Constrained k-means clustering with background knowledge. In *Icml*, Volume 1, pp. 577–584.

Wang, Z., W. Yan, et T. Oates (2017). Time series classification from scratch with deep neural networks : A strong baseline. In *2017 International joint conference on neural networks (IJCNN)*, pp. 1578–1585. IEEE.

Xie, J., R. Girshick, et A. Farhadi (2016). Unsupervised deep embedding for clustering analysis. In *International conference on machine learning*, pp. 478–487.

Zhang, H., S. Basu, et I. Davidson (2019). Deep constrained clustering-algorithms and advances. *arXiv preprint arXiv :1901.10061*.

Zhu, X. X., D. Tuia, L. Mou, G.-S. Xia, L. Zhang, F. Xu, et F. Fraundorfer (2017). Deep learning in remote sensing : A comprehensive review and list of resources. *IEEE Geoscience and Remote Sensing Magazine 5*(4), 8–36.

Summary

The advent of satellite imagery is generating an unprecedented amount of remote sensing images. Current satellites now achieve frequent revisits and high mission availability and provide series of images of the Earth captured at different dates that can be seen as time series. Analyzing satellite image time series allows to perform continuous wide range Earth observation with applications in agricultural mapping, environmental disaster monitoring, etc. However, the lack of large quantity of labeled data generally prevents from easily applying supervised methods. On the contrary, unsupervised methods do not require expert knowledge but sometimes provide poor results. In this context, constrained clustering, which is a class of semi-supervised learning algorithms, is an alternative and offers a good trade-off of supervision. In this paper, we explore the use of constraints with deep clustering approaches to process satellite image time series. Our experimental study relies on deep embedded clustering and the deep constrained framework using pairwise constraints (must-link and cannot-link). Experiments on a real dataset composed of 11 satellite images show promising results and open many perspectives for applying deep constrained clustering to satellite image time series.

Liste des auteurs