Improving Semantic Segmentation with Graph-Based Structural Knowledge

Jérémy Chopin[1](✉), Jean-Baptiste Fasquel[1], Harold Mouchère[2], Rozenn Dahyot[3], and Isabelle Bloch[4]

[1] LARIS, Université d'Angers, Angers, France
{jeremy.chopin,jean-baptiste.fasquel}@univ-angers.fr
[2] LS2N, Universite de Nantes, CNRS UMR 6004, Nantes, France
harold.mouchere@univ-nantes.fr
[3] Department of Computer Science, Maynooth University, Maynooth, Ireland
Rozenn.Dahyot@mu.ie
[4] Sorbonne Université, CNRS, LIP6, Paris, France
isabelle.bloch@sorbonne-universite.fr

Abstract. Deep learning based pipelines for semantic segmentation often ignore structural information available on annotated images used for training. We propose a novel post-processing module enforcing structural knowledge about the objects of interest to improve segmentation results provided by deep learning. This module corresponds to a "many-to-one-or-none" inexact graph matching approach, and is formulated as a quadratic assignment problem. Using two standard measures for evaluation, we show experimentally that our pipeline for segmentation of 3D MRI data of the brain outperforms the baseline CNN (U-Net) used alone. In addition, our approach is shown to be resilient to small training datasets that often limit the performance of deep learning.

Keywords: Graph matching · Deep learning · Image segmentation · Volume segmentation · Quadratic assignment problem

1 Introduction

Deep learning approaches are now widely used in computer vision [11], and in particular for semantic image segmentation [10]. Through a set of convolution layers, semantic segmentation with Convolutional Neural Networks (CNNs) is intrinsically based on information embedded at low-level, *i.e.* at pixel and its neighborhood levels. CNNs do not explicitly model the structural information available at a higher semantic level, for instance the relationships between annotated regions that are present in the training dataset. High-level structural

This research was conducted in the framework of the regional program Atlanstic 2020, Research, Education and Innovation in Pays de la Loire, supported by the French Region Pays de la Loire and the European Regional Development Fund.

M. El Yacoubi et al. (Eds.): ICPRAI 2022, LNCS 13363, pp. 173–184, 2022.
https://doi.org/10.1007/978-3-031-09037-0_15

information may include spatial relationships between different regions (e.g. distances, relative directional position) [2] or relationships between their properties (e.g. relative brightness, difference of colorimetry) [8,9].

This type of high-level structural information is very promising [2,6,8,9,19] and it has found applications in medical image understanding [4,7,18] but also in document analysis (e.g. [5,12] for handwriting recognition) or in scene understanding (e.g. [13] for robotic). In some domains, the relations between objects have to be identified to recognize the image content [12] but in other domains these relations help the recognition of a global scene as a complementary knowledge [5,8,9,13]. Our work falls in this second category. This high-level information is commonly represented using graphs, where vertices correspond to regions, and edges carry the structural information. The semantic segmentation problem turns then into a region or node labeling problem, often formulated as a graph matching problem [8,9,16]. In this paper, we propose a new approach involving a graph-matching-based semantic segmentation applied to the probability map produced by CNNs for semantic segmentation, in order to take into account explicitly this high-level structural information observed in the training dataset but intrinsically ignored by convolutional layers. Our proposal aims at improving the semantic segmentation of images, in particular when the size of the training dataset is low. As such, our work also addresses, to some extent, one key limitation of deep learning: the requirement of a large and representative dataset for training purposes, this being often addressed by generating more training data (data augmentation) [21] or by considering a transfer learning technique [23]. By focusing on the high level global structure of a scene, our approach is expected to be less sensitive to the lack of diversity and representativity of the training dataset.

This paper extends [3] by combining the high level structural information observed in the training dataset with the output of the semantic segmentation produced by a deep neural network. It uses a graph matching approach formulated as a quadratic assignment problem (QAP) [17,24,25]. We deploy two types of relationships for capturing structural information and our approach is shown experimentally to perform well for segmenting 3D volumetric data (cf. Fig. 1)[1].

2 Proposed Method

Structural information, such as spatial relationships, is encoded in a graph model G_m that captures the observed relationships between regions in an annotated training dataset. Vertices and edges correspond respectively to regions of the annotated dataset and spatial relationships between them. A hypothesis graph G_r is similarly created from the semantic segmentation map of a query image using the same label taxonomy as the training set. Graph matching (GM) of G_r onto G_m allows matching the vertices (and thus the underlying regions of the query image) with those of the model. Correspondences between G_r and G_m

[1] The open-source code and data are to be shared with the community https://github.com/Jeremy-Chopin/APACoSI/.

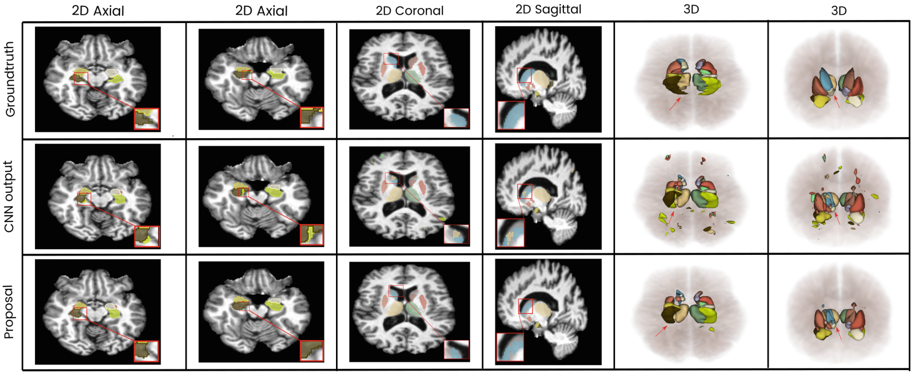


Fig. 1. Example of semantic segmentation of a brain (slices and 3D view) performed by the expert (reference segmentation - top), by the CNN (middle) and by our method (bottom). 100% of the training dataset is considered. Surrounded boxes and red arrows indicate segmentation errors that are corrected by our method. (Color figure online)

computed with GM provide a relabelling of some of the regions (vertices) in G_r hence providing a enhanced semantic segmentation map of the query image with additional high-level structural information.

Semantic Segmentation. A query image or volume is segmented providing a tensor $S \in \mathbb{R}^{P \times N}$ with P the dimensions of the query ($P = I \times J$ pixels for 2D images, or $P = I \times J \times K$ voxels in 3D volumes) and N is the total number of classes considered for segmentation. At each pixel or voxel location p, the value $S(p, n) \in [0, 1]$ is the probability of belonging to class n with the constraints:

$$(\forall n \in \{1, \ldots, N\},\ 0 \leq S(p, n) \leq 1) \wedge \left(\sum_{n=1}^{N} S(p, n) = 1\right)$$

The segmentation map $\mathcal{L}^*$ selects the label n of the class with the highest probability. Note that in practice semantic segmentation of a query image can be performed using deep neural networks such as, for instance, U-Net [21] or segNet [1].

2.1 Graph Definitions

From the segmentation map $\mathcal{L}^*$, a set R of all resulting connected components is defined. Additionally, to constrain graph matching (described in Sect. 2.2), we define a set $R^* = \{R_1^*, \ldots, R_N^*\}$, where, for each class $n \in \{1, \cdots, N\}$, R_n^* is a set of regions corresponding to the connected components belonging to class n. From the set R, the graph $G_r = (V_r, E_r, A, D)$ is defined, where V_r is the set of vertices, E_r the set of edges, A a vertex attribute assignment function and D an edge attribute assignment function. Each vertex $v \in V_r$ is associated with a region $R_v \in R$ with an attribute provided by the function A which is the average membership probability vector over the set of pixels $p \in R_v$, therefore computed on the initial tensor S:

$$\forall v \in V_r, \forall n \in \{1, \ldots, N\}, A(v)[n] = \frac{1}{|R_v|} \sum_{p \in R_v} S(p, n) \tag{1}$$

We consider a complete graph where each edge $e = (i, j) \in E_r$ has an attribute defined by the function D, associated with a relation between the regions R_i and R_j. Two functions D have been tested in our experiments. They are capturing the relative directional position or the trade-off between the minimal and maximal distances found between two regions. The choice of the function D is an hyperparameter in our method that can be tuned to improve performance for the considered application (cf. Sect. 2.3).

The model graph $G_m = (V_m, E_m, A, D)$ is composed of N vertices (one vertex per class) and is constructed from the annotated images of the training set. The attribute of a vertex is a vector of dimension N with only one non-zero component (with value equal to 1), associated with the index of the corresponding class. The edges are obtained by calculating the average spatial relationships (in the training set) between the regions (according to the relation D considered).

2.2 Graph Matching

We propose to identify the regions by associating each of the vertices of G_r to a vertex of the model graph G_m. The most likely situation encountered is when more regions are found in the image associated with G_r than in the model (i.e. $|V_r| \geq |V_m|$). To solve this, we propose here to extend the many-to-one inexact graph matching strategy [3,16] to a many-to-one-or-none matching. The "none" term allows some vertices in G_r to be matched with none of the vertices of the model graph G_m, which corresponds to removing the underlying image region (e.g. merged with the background). Graph matching is here formulated as a quadratic assignment problem (QAP) [25]. The matrix $X \in \{0, 1\}^{|V_r| \times |V_m|}$ is defined such that $X_{ij} = 1$ means that vertex $i \in V_r$ is matched with vertex $j \in V_m$. The objective is to estimate the best matching X^* as follows:

$$X^* = \arg\min_X \left\{ \mathrm{vec}(X)^T K \, \mathrm{vec}(X) \right\} \tag{2}$$

where vec(X) is the column vector representation of X and T denotes the transposition operator. This optimal matching is associated with the optimal matching cost $C^* = \mathrm{vec}(X^*)^T K \, \mathrm{vec}(X^*)$.

The matrix K embeds the dissimilarity measures between the two graphs G_r and G_m, at vertices (diagonal elements) and edges (non-diagonal elements):

$$K = \alpha \; K_v + (1 - \alpha) \; \frac{K_e(D)}{\max K_e(D)} \tag{3}$$

where K_v embeds dissimilarities between vertices (e.g. L2 Euclidean distance between class membership probability vectors) - more details for computing K

can be found in [25]. The matrix $K_e(D)$ is related to dissimilarities between edges, and depends on the considered relation D. K_e terms are related to distances between regions (normalized in the final K matrix). The α parameter ($\alpha \in [0, 1]$) allows weighting the relative contribution of vertex and edge dissimilarities: K_v terms range between 0 and 1, and K_e is also normalized in Eq. 3. Due to the combinatorial nature of this optimization problem [25] (i.e. set of possible X candidates in Eq. 2), we propose a two-steps procedure:

1. Search for an initial one-to-one matching.
2. Refinement by matching remaining vertices, finally leading to a many-to-one-or-none matching.

Initial Matching: One-to-One. One searches for the optimal solution to Eq. 2 by imposing the following three constraints on X, thus reducing the search space for eligible candidates:

1. $\sum_{j=1}^{|V_m|} X_{ij} \leq 1$: some vertices i of G_r may not be matched.
2. $\sum_{i=1}^{|V_r|} X_{ij} = 1$: each vertex j of G_m must be matched with only one vertex of G_r.
3. $X_{ij} = 1 \Rightarrow R_i \in R_j^*$: vertex $i \in V_r$ can be matched with vertex $j \in V_m$ if the associated R_i region was initially considered by the neural network to most likely belong to class j (i.e. $R_i \in R_j^*$).

The first two constraints ensure to search for a one-to-one matching thanks to the third constraint, one reduces the search space by relying on the neural network: one assumes that it has correctly, at least to some extent, identified the target regions, even if artifacts may still have been produced as well (to be managed by refining the matching). This step allows us to retrieve the general structure of the regions (thus verifying the prior structure modeled by G_m) with a cost $C^I = \text{vec}(X^I)^T K \, \text{vec}(X^I)$ related to the optimal initial matching X^I (I stands for "initial").

Refinement: Many-to-One-or-None. Unmatched nodes are integrated into the optimal matching X^I or removed (i.e. assigned to a "background" or "none" node) through a refinement step leading to X^* considered in Eq. 2. This many-to-one-or-none matching is performed through an iterative procedure over the set of unlabeled nodes $U = \{k \in V_r \mid \sum_{j=1}^{|V_m|} X_{kj}^I = 0\}$. For each node $k \in U$, one searches for the best assignment, among all possible ones, related to the set of already labeled nodes $L = \{k \in V_r \mid \sum_{j=1}^{|V_m|} X_{kj}^I = 1\}$. Mathematically, the best label candidate for a given node $k \in U$ is:

$$l_k^* = \arg\min_{l \in L} \{\text{vec}(X^I)^t K_{k \to l} \, \text{vec}(X^I)\} \tag{4}$$

where $K_{k \to l}$ corresponds to the matrix K after having merged both underlying regions (i.e. $R_l = R_l \cup R_k$) and updated relations (leading to the graph $G_r^{'}$, where both k and l vertices are merged). The cost related to the merging of k to l_k^* is $C_{k \rightarrow l_k^*}$. Figure 2 illustrates this iterative procedure.

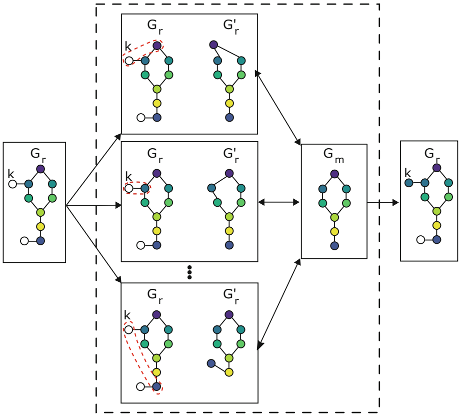


Fig. 2. Refinement: finding the best matching for a given unlabeled node $k \in U$ (white node). Only three possible matchings are reported for clarity (dashed surrounded nodes). The one in the middle is finally kept (smallest deformation of $G_r^{'}$ with respect to the model G_m).

The best candidate is retained if the related cost is smaller than a chosen threshold T, otherwise the related node k is discarded (i.e. $k \rightarrow \emptyset$, $\emptyset$ corresponding to the "none" vertex, meaning that the underlying image region is merged with the background). The optimal matching is updated according to the condition:

$$\begin{cases} X_{kl}^* = 1, & \text{if } C_{k \rightarrow l_k^*} < T \\ X_{kl}^* = 0, & \text{otherwise} \end{cases}$$

This enables to manage the removal of regions to be considered as artifacts and this was not managed in our earlier work [3].

Algorithm 1 provides an implementation of the proposed refinement. For each unlabeled vertex $k \in U$, the optimal cost is initially set to infinity (Line 2). Then, for each candidate $l \in L$, one creates an image region (temporary variable R_l') corresponding to the union of both unlabeled and merging candidate regions (Line 4). We update the dissimilarity matrix (leading to the temporary variable $K_{k \rightarrow l}$ - Line 5), and then compute the cost of this union (Line 6). If this union decreases the matching cost, the merging candidate is considered as the best one (Lines 8 and 9). After having evaluated the cost of the matching with the best candidate $l \in L$, we finally accept the resulting best matching, if the value of the associated cost is lower than the predefined threshold T (Lines 12 to 16). If the cost is higher, the vertex $k \in U$ is discarded (and the image region is removed).

Algorithm 1. Refinement algorithm

```
Require: $U, L, T, X^I$
for $k \in U$ do
    $C^*_{k \to l} \leftarrow \infty$
    for $l \in L$ do
        $R'_l \leftarrow R_l \cup R_k$
        $K_{k \to l} \leftarrow$ Update-K($R'_l$)
        $C_{k \to l} \leftarrow vec(X^I)^t \; K_{k \to l} \; vec(X^I)$
        if $C_{k \to l} < C^*_{k \to l}$ then
            $l^*_k \leftarrow l$
            $C^*_{k \to l^*_k} \leftarrow C_{k \to l}$
        end if
    end for
    if $C_{k \to l^*_k} < T$ then
        $k \to l^*_k$ {$k$ is assigned}
        $R_{l^*_k} \leftarrow R_{l^*_k} \cup R_k$
    else
        $k \to \emptyset$ {$k$ is discarded}
    end if
end for
```

2.3 Modelling Spatial Relationships

Two types of spatial relationships are considered (cf. Fig. 3), each being associated to a specific dissimilarity function D (used to compute the term $K_e(D)$ in Eq. 3). The first spatial relationship involves two distances (leading to two components on an edge attribute), corresponding to the minimal and maximum distances between two regions R_i and R_j (cf. Fig. 3-left):

$$d_{\min}^{(i,j)} = \min_{p \in R_i, q \in R_j} (|p - q|) \tag{5}$$

$$d_{\max}^{(i,j)} = \max_{p \in R_i, q \in R_j} (|p - q|) \tag{6}$$

Based on these relationships, the considered dissimilarity function is defined as:

$$D_1{}^{(k,l)}_{(i,j)} = \frac{\lambda}{C_s} \left(|d_{\min}^{(i,j)} - d_{\min}^{(k,l)}| \right) + \frac{(1-\lambda)}{C_s} \left(|d_{\max}^{(i,j)} - d_{\max}^{(k,l)}| \right) \tag{7}$$

where λ is a parameter balancing the influence of the dissimilarities on both distances. C_s corresponds to the largest distance observed in an image, ensuring that values range within $[0, 1]$.

The second spatial relationship is the relative directional position of the centroids of two regions, as in [20]. For two regions R_i and R_j, the relative position is defined by the vector $\vec{v_{ij}} = \overline{R}_j - \overline{R}_i$ (edge attribute), where $\overline{R}$ denotes the coordinates of the center of mass of region R. Based on this relationship, the considered dissimilarity function is:

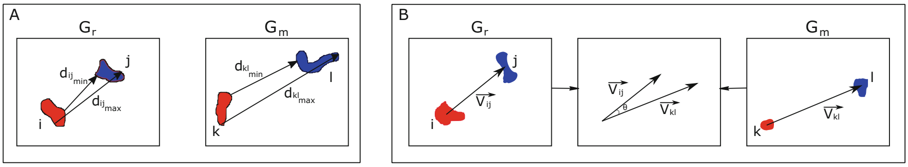


Fig. 3. Spatial relationships considered in experiments. A: Relationship based on distances (corresponding to the D_1 dissimilarity function). B: Relationship based on relative directional positions (corresponding to the D_2 dissimilarity function).

$$D_2{}_{(i,j)}^{(k,l)} = \lambda \frac{|\cos\theta - 1|}{2} + (1-\lambda)\frac{||\vec{v_{ij}}| - |\vec{v_{kl}}||}{C_s} \tag{8}$$

where θ is the angle between them $\vec{v_{ij}}$ and $\vec{v_{kl}}$ vectors, computed using a scalar product (Eq. 9):

$$\cos(\theta) = \frac{\vec{v_{ij}}.\vec{v_{kl}}}{|\vec{v_{ij}}|.|\vec{v_{kl}}|} \tag{9}$$

As for the first spatial relationship, the C_s term is the maximum distance value observed in an image, ensuring that values range within $[0, 1]$. The term $\lambda \in [0, 1]$ is a parameter balancing the influence of the difference in terms of distance and orientation.

Concerning the complexity, the computation time is mainly affected by the refinement step involving many relabelling (cf. Fig. 2). In Algorithm 1, the complexity of this second step of the matching linearly depends on the cardinalities of both U and L entities as well as on the complexity of the cost computation (i.e. union of regions, Update-K(R'_l) and $vec(X)^T K vec(X)$ reported in lines 4–6 of Algorithm 1).

3 Application to Segmentation of 3D MRI

IBSR Dataset: The IBSR[2] public dataset provides 18 3D MRI of the brain, together with the manual segmentation of 32 regions. In our experiments, similarly to the work by Kushibar et al. [14], only 14 classes (i.e. 14 regions) of the annotated dataset are considered: thalamus (left and right), caudate (left and right), putamen (left and right), pallidum (left and right), hippocampus (left and right), amygdala (left and right) and accumbens (left and right).

CNN Backbone: 3D U-Net neural network is used for creating three instances of a trained CNN for segmentation using training sets of different sizes:

- *100% (10/18):* 10 images are used for training (training set) out of the 18 available, an additional 4 are used validation (validation set) and the last 4 are used for testing (test set).

[2] The IBSR annotated public dataset can be downloaded at the following address: https://www.nitrc.org/projects/ibsr.

- *75% (8/18):* In this case, out of the 10 images available in the original training set, only 8 are used. Results reported correspond to an average over several CNNs trained with randomly selecting 8 images amongst the 10 in the training set. Validation and test sets remain the same.
- *50% (5/18):* out of the 10 images available in the original training set, only 5 are used. Results reported correspond to an average over several CNNs trained with randomly selecting 5 images amongst the 10 in the original training set. Validation and test sets remain the same.

50 epochs are used for training the network and an early stopping politic is applied to prevent over-fitting. The training process was terminated if there was no improvement on the loss (using cross entropy loss function) for 8 consecutive epochs. We used a 3D patch-based approach [15] since classes are highly unbalanced (i.e. small size of target regions with respect to other brain tissues and background). Patches are volumes of size 48^3 voxels, that have been extracted around the centroid of each label (random selection) using the *Torchio* library [22]. 150 patches are selected for each MRI volume, with a frequency that is proportional to the inverse prior probability of the corresponding class.

Measures for Assessment: The Hausdorff distance (HD) is widely used in this application domain [14] ($HD = 0$ corresponding to a perfect segmentation). The pixel-wise Dice index (DSC) is also reported and it is ranging within $[0, 1]$ where 1 corresponds to a perfect segmentation. The hyperparameters are chosen empirically without optimisation: $\alpha = 0.5$ and $\lambda = 0.5$.

Quantitative Results: Table 1 compares performances for both spatial relationships D_1 and D_2. Our pipeline improves the results of the CNN used alone either in terms of Dice index (best DSC with D_2) or in terms of Hausdorff distance (best HD with D_1). Structural information modelled with either D_1 or D_2 in our pipeline allows us to improve segmentation results.

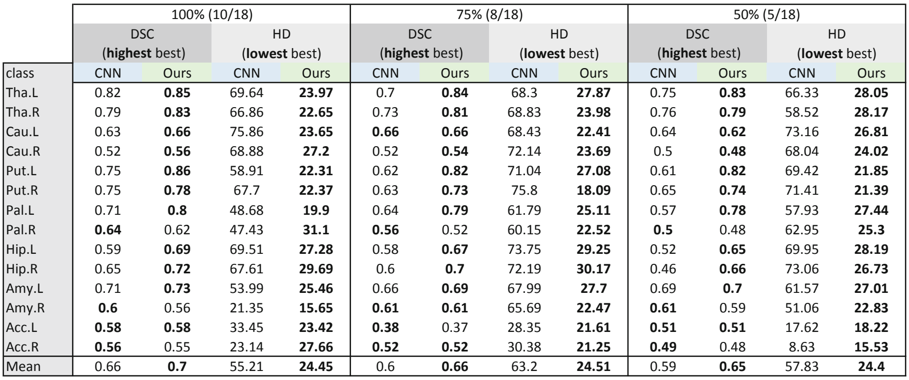
Table 1. Comparing dissimilarity functions D_1 and D_2 for modelling spatial relationships. The evaluation measures are the pixel-wise Dice index (DSC) and the Hausdorff distance (HD).

Method	CNN		Ours (D_1)		Ours (D_2)	
Tr.dataset (%)	DSC↑	HD↓	DSC↑	HD↓	DSC↑	HD↓
100%	0.66	55.21	0.67	**7.52**	**0.7**	24.45
75%	0.6	63.2	0.63	**9.58**	**0.66**	24.51
50%	0.59	57.83	0.64	**9.38**	**0.65**	24.4

Table 2 details the results for each class using D_2 that significantly improves the Dice index while also significantly reducing the Hausdorff distance. For

DSC, the improvement fluctuates between 4% (Tr. dataset 100%) and 6% (Tr. dataset 50%). The improvement is significant for large regions (e.g. “Tha.L” and “Put.L”). In terms of Hausdorff distance, the improvement is significant (58% on average) for most considered classes and size of the training dataset used.

Table 2. Comparison of segmentations provided by the CNN and by our proposal, for the second spatial relationships (D_2 dissimilarity function), considering the Dice index related to pixelwise precision and Hausdorff distance compared to the manual segmentation. Results are provided as average and for each class: Tha.L(left thalamus), Tha.R(right thalamus), Cau.L(left caudate), Cau.R(right caudate), Put.L(left putamen), Put.R(right putamen), Pal.L(left pallidum), Pal.R(right pallidum), Hip.L(left hippocampus), Hip.R(right hippocamus), Amy.L(left amygdala), Amy.R(right amygdala), Acc.L(left accumbens), Acc.R(right accumbens). Results are also provided for different sizes of the training/validation sets.



	100% (10/18)				75% (8/18)				50% (5/18)			
	DSC (**highest** best)		HD (**lowest** best)		DSC (**highest** best)		HD (**lowest** best)		DSC (**highest** best)		HD (**lowest** best)	
class	CNN	Ours	CNN	Ours	CNN	Ours	CNN	Ours	CNN	Ours	CNN	Ours
Tha.L	0.82	**0.85**	69.64	**23.97**	0.7	**0.84**	68.3	**27.87**	0.75	**0.83**	66.33	**28.05**
Tha.R	0.79	**0.83**	66.86	**22.65**	0.73	**0.81**	68.83	**23.98**	0.76	**0.79**	58.52	**28.17**
Cau.L	0.63	**0.66**	75.86	**23.65**	**0.66**	**0.66**	68.43	**22.41**	0.64	**0.62**	73.16	**26.81**
Cau.R	0.52	**0.56**	68.88	**27.2**	0.52	**0.54**	72.14	**23.69**	0.5	**0.48**	68.04	**24.02**
Put.L	0.75	**0.86**	58.91	**22.31**	0.62	**0.82**	71.04	**27.08**	0.61	**0.82**	69.42	**21.85**
Put.R	0.75	**0.78**	67.7	**22.37**	0.63	**0.73**	75.8	**18.09**	0.65	**0.74**	71.41	**21.39**
Pal.L	0.71	**0.8**	48.68	**19.9**	0.64	**0.79**	61.79	**25.11**	0.57	**0.78**	57.93	**27.44**
Pal.R	**0.64**	0.62	47.43	**31.1**	**0.56**	0.52	60.15	**22.52**	**0.5**	0.48	62.95	**25.3**
Hip.L	0.59	**0.69**	69.51	**27.28**	0.58	**0.67**	73.75	**29.25**	0.52	**0.65**	69.95	**28.19**
Hip.R	0.65	**0.72**	67.61	**29.69**	0.6	**0.7**	72.19	**30.17**	0.46	**0.66**	73.06	**26.73**
Amy.L	0.71	**0.73**	53.99	**25.46**	0.66	**0.69**	67.99	**27.7**	0.69	**0.7**	61.57	**27.01**
Amy.R	**0.6**	0.56	21.35	**15.65**	**0.61**	**0.61**	65.69	**22.47**	**0.61**	0.59	51.06	**22.83**
Acc.L	**0.58**	**0.58**	33.45	**23.42**	**0.38**	0.37	28.35	**21.61**	**0.51**	**0.51**	17.62	**18.22**
Acc.R	**0.56**	0.55	23.14	**27.66**	**0.52**	**0.52**	30.38	**21.25**	**0.49**	0.48	8.63	**15.53**
Mean	0.66	**0.7**	55.21	**24.45**	0.6	**0.66**	63.2	**24.51**	0.59	**0.65**	57.83	**24.4**

Qualitative Results: Figure 1 provides an example of a 3D image processed by the CNN only and by our pipeline. The CNN (Fig. 1-CNN Output) provides a visually acceptable semantic segmentation: at the exception of many surrounding artefacts (particularly visible on 3D views), most target structures are globally recovered. Despite these surrounding artefacts, segmentation errors occur in parts of the target structures that need to be relabelled (see 2D slices, bounding boxes and arrows in 3D views). Our pipeline succeeds in correcting most segmentation errors: many parts of the structures of interest are correctly relabeled and most surrounding artefacts are removed. Note that artefacts removal corresponds to the matching with the class “none” in our “many-to-one-or-none” graph matching strategy, and it is managed using the threshold T (cf. Algorithm 1) that needs to be correctly tuned as it affects computation of HD.

4 Conclusion

We have proposed a post-processing technique for improving segmentation results using a graph matching procedure encoding structural relationships between regions. This correction of deep learning segmentation with the exploitation of structural patterns is performed thanks to inexact graph matching formulated as a two-steps Quadratic Assignment Problem (QAP). We validated our approach with experiments on 3D volumetric data, and we have shown significant improvements can be observed. When training the neural network on a limited dataset, our approach provides a very clear advantage by outperforming the baseline. Future work will investigate how to reduce the high computational time resulting from the complexity of operations (segmentation, graph matching and refinement) of our approach that may hinder real time applications.

References

1. Badrinarayanan, V., Kendall, A., Cipolla, R.: SegNet: a deep convolutional encoder-decoder architecture for image segmentation. IEEE Trans. Pattern Anal. Mach. Intell. **39**(12), 2481–2495 (2017). https://doi.org/10.1109/TPAMI.2016.2644615
2. Bloch, I.: Fuzzy sets for image processing and understanding. Fuzzy Sets Syst. **281**, 280–291 (2015). https://doi.org/10.1016/j.fss.2015.06.017
3. Chopin, J., Fasquel, J.B., Mouchère, H., Dahyot, R., Bloch, I.: Semantic image segmentation based on spatial relationships and inexact graph matching. In: 2020 Tenth International Conference on Image Processing Theory, Tools and Applications (IPTA), pp. 1–6 (2020). https://doi.org/10.1109/IPTA50016.2020.9286611
4. Colliot, O., Camara, O., Bloch, I.: Integration of fuzzy spatial relations in deformable models - application to brain MRI segmentation. Pattern Recogn. **39**, 1401–1414 (2006). https://doi.org/10.1016/j.patcog.2006.02.022
5. Delaye, A., Anquetil, E.: Fuzzy relative positioning templates for symbol recognition. In: International Conference on Document Analysis and Recognition, Beijing, China, September 2011. https://doi.org/10.1109/ICDAR.2011.246
6. Deruyver, A., Hodé, Y.: Qualitative spatial relationships for image interpretation by using a conceptual graph. Image Vis. Comput. **27**(7), 876–886 (2009). https://doi.org/10.1016/j.imavis.2008.10.002, 7th IAPR-TC15 Workshop on Graph-based Representations (GbR 2007)
7. Fasquel, J.B., Agnus, V., Moreau, J., Soler, L., Marescaux, J.: An interactive medical image segmentation system based on the optimal management of regions of interest using topological medical knowledge. Comput. Methods Programs Biomed. **82**, 216–230 (2006). https://doi.org/10.1016/j.cmpb.2006.04.004
8. Fasquel, J.B., Delanoue, N.: An approach for sequential image interpretation using a priori binary perceptual topological and photometric knowledge and k-means based segmentation. J. Opt. Soc. Am. A **35**(6), 936–945 (2018). https://doi.org/10.1364/JOSAA.35.000936
9. Fasquel, J.B., Delanoue, N.: A graph based image interpretation method using a priori qualitative inclusion and photometric relationships. IEEE Trans. Pattern Anal. Mach. Intell. **41**(5), 1043–1055 (2019). https://doi.org/10.1109/TPAMI.2018.2827939

10. Garcia-Garcia, A., Orts-Escolano, S., Oprea, S., Villena-Martinez, V., Martinez-Gonzalez, P., Garcia-Rodriguez, J.: A survey on deep learning techniques for image and video semantic segmentation. Appl. Soft Comput. **70**, 41–65 (2018). https://doi.org/10.1016/j.asoc.2018.05.018
11. Goodfellow, I.J., Bengio, Y., Courville, A.: Deep Learning. MIT Press, Cambridge (2016)
12. Julca-Aguilar, F., Mouchère, H., Viard-Gaudin, C., Hirata, N.S.T.: A general framework for the recognition of online handwritten graphics. Int. J. Doc. Anal. Recogn. (IJDAR) **23**(2), 143–160 (2020). https://doi.org/10.1007/s10032-019-00349-6
13. Kunze, L., et al.: Combining top-down spatial reasoning and bottom-up object class recognition for scene understanding. In: 2014 IEEE/RSJ International Conference on Intelligent Robots and Systems, pp. 2910–2915. IEEE (2014). https://doi.org/10.1109/IROS.2014.6942963
14. Kushibar, K., et al.: Automated sub-cortical brain structure segmentation combining spatial and deep convolutional features. Med. Image Anal. **48**, 177–186 (2018). https://doi.org/10.1016/j.media.2018.06.006
15. Lee, B., Yamanakkanavar, N., Choi, J.Y.: Automatic segmentation of brain MRI using a novel patch-wise U-Net deep architecture. PLOS ONE **15**(8), 1–20 (2020). https://doi.org/10.1371/journal.pone.0236493
16. Lezoray, O., Leo, L.: Image Processing and Analysis with Graphs: Theory and Practice. CRC Press, Cambridge (2012)
17. Maciel, J., Costeira, J.P.: A global solution to sparse correspondence problems. IEEE Trans. Pattern Anal. Mach. Intell. **25**(2), 187–199 (2003). https://doi.org/10.1109/TPAMI.2003.1177151
18. Moreno, A., Takemura, C., Colliot, O., Camara, O., Bloch, I.: Using anatomical knowledge expressed as fuzzy constraints to segment the heart in CT images. Pattern Recogn. **41**(8), 2525–2540 (2008). https://doi.org/10.1016/j.patcog.2008.01.020
19. Nempont, O., Atif, J., Bloch, I.: A constraint propagation approach to structural model based image segmentation and recognition. Inf. Sci. **246**, 1–27 (2013). https://doi.org/10.1016/j.ins.2013.05.030
20. Noma, A., Graciano, A.B., Cesar, R.M., Jr., Consularo, L.A., Bloch, I.: Interactive image segmentation by matching attributed relational graphs. Pattern Recogn. **45**(3), 1159–1179 (2012). https://doi.org/10.1016/j.patcog.2011.08.017
21. Ronneberger, O., Fischer, P., Brox, T.: U-Net: convolutional networks for biomedical image segmentation. In: Navab, N., Hornegger, J., Wells, W.M., Frangi, A.F. (eds.) MICCAI 2015. LNCS, vol. 9351, pp. 234–241. Springer, Cham (2015). https://doi.org/10.1007/978-3-319-24574-4_28
22. Pérez-García, F., Sparks, R., Ourselin, S.: TorchIO: a Python library for efficient loading, preprocessing, augmentation and patch-based sampling of medical images in deep learning. arXiv:2003.04696 [cs, eess, stat], March 2020
23. Weiss, K., Khoshgoftaar, T.M., Wang, D.D.: A survey of transfer learning. J. Big Data **3**(1), 1–40 (2016). https://doi.org/10.1186/s40537-016-0043-6
24. Zanfir, A., Sminchisescu, C.: Deep learning of graph matching. In: 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 2684–2693 (2018). https://doi.org/10.1109/CVPR.2018.00284
25. Zhou, F., De la Torre, F.: Factorized graph matching. IEEE Trans. Pattern Anal. Mach. Intell. **38**(9), 1774–1789 (2016). https://doi.org/10.1109/TPAMI.2015.2501802