On the relevance of edge-conditioned convolution for GNN-based semantic image segmentation using spatial relationships

Patty Coupeau, Jean-Baptiste Fasquel, Mickael Dinomais

On the relevance of edge-conditioned convolution for GNN-based semantic image segmentation using spatial relationships

P. Coupeau
Université d'Angers, LARIS
SFR MATHSTIC
F-49000 Angers, France
ORCID: 0000-0003-0164-5540

J.-B. Fasquel
Université d'Angers, LARIS
SFR MATHSTIC
F-49000 Angers, France
ORCID: 0000-0001-9183-0365

M. Dinomais
Université d'Angers, LARIS, SFR MATHSTIC
Département de médecine physique et de réadaptation
Centre Hospitalier Universitaire d'Angers, France
ORCID: 0000-0002-6980-1178

***Abstract*—This paper addresses the fundamental task of semantic image segmentation by exploiting structural information (spatial relationships between image regions). To perform such task, we propose to combine a deep neural network (CNN) with inexact "many-to-one-or-none" graph matching where graphs encode efficiently class probabilities and structural information related to regions segmented by the CNN. In order to achieve node classification, a basic 2-layer graph neural network (GNN) based on the edge-conditioned convolution operator (ECConv), managing both node and edge attributes, is considered. Preliminary experiments are performed on both a synthetic dataset and a public dataset of face images (FASSEG). Our approach is shown to be resilient to small training datasets that often limit the performance of deep learning thanks to a preprocessing task of graph coarsening. Results show that the proposal reaches a perfect accuracy on synthetic dataset and improves performance of the CNN by 6% (bounding box dice index) on FASSEG. Moreover, it enhances by 27% the initial Hausdorff distance (i.e. with CNN only) using the entire training dataset and by 41% with only 75% of training samples.**

***Index Terms*—image segmentation, structural information, inexact graph matching, graph neural network, edge-conditioned convolution**

I. Introduction

Semantic image segmentation is a fundamental task in computer vision usually managed using modern convolutional-neural-network-based (CNN) deep learning approaches [1]. CNNs do not explicitly model the structural information at a higher semantic level (relationships between annotated regions of the training dataset or qualitative description of the scene content [2]). For segmentation, such information can be exploited using inexact-graph-matching-based techniques (i.e. matching regions produced by a CNN with the ones of the model built from annotations or from a qualitative description). This is classically formulated as a quadratic assignment problem (QAP) [3], [4], unfortunately limited by its intrinsic highly combinatorial nature [5]. To overcome this limitation, one considers graph neural networks (GNN), constituting an emerging and active field in the context of deep learning, as recently underlined [6], [7]. In computer vision, most related works operate, to mention but a few recent studies, on point clouds [8] or bounding boxes of detected objects [9]. In our context, a crucial constraint is that the matching must integrate both nodes (region information - membership probabilities provided by the CNN) and edge attributes (weighted relationships), by considering an appropriate message passing strategy and, in particular, the appropriate neighborhood aggregation operator, in charge of combining weighted node and edge information [6] for finally identifying nodes (regions). Note that, there are already some GNN-based works proposed for semantic image segmentation but they do not exploit such combination of information [10], [11].

The originality and main contribution regards the proposal of a GNN-based method for inexact many-to-one-or-none graph matching in this context of a CNN-based semantic image segmentation using structural information. An important part of this contribution concerns the study of the relevance of a particular neighborhood aggregation operator, namely the edge-conditioned convolution operator (ECConv) [8], allowing to manage both attributed node and edge information on arbitrary graphs. This convolution operator has been recently proposed and evaluated for graph classification only and not for node classification [8]. In our sense, this work will, besides, contribute in reducing the gap between the research community focusing on semantic image segmentation and the one interested in other GNN-centered computer vision applications.

The proposal is described in Section II. Preliminary experiments are detailed in Section III and discussed in Section IV, before concluding.

II. Method

Figure 1 provides an overview of the proposed approach. A deep neural network (CNN) is trained for semantic image segmentation using an annotated dataset. A graph neural network (GNN) is also trained to match nodes of the graph built from the segmented image produced by the CNN with nodes of the graph built from the annotated dataset. When analysing a new image (Figure 1-Semantic segmentation), the trained CNN first provides a segmentation proposal from which a

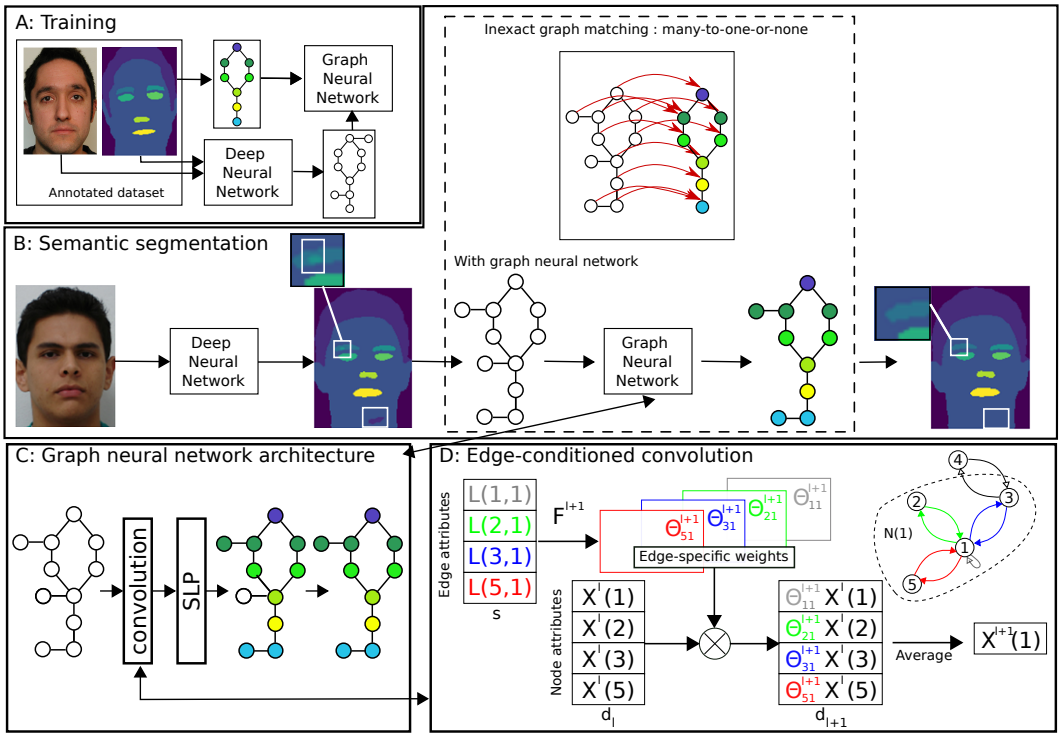

Fig. 1. Overview of the proposed approach.

graph is built before using the graph neural network to identify nodes (node classification): this steps corresponds to a many-to-one-or-none inexact graph matching. In the example related to Figure 1, 11 regions are provided by the CNN. The 10 nodes corresponding to 10 of them are matched with 8 nodes (many-to-one as the CNN produces an oversegmentation) while the last node is matched with none node (region artefact to be removed). According to this matching, regions are finally identified.

A. Images and graphs

When segmenting an image, the CNN provides a segmentation map being a tensor $S \in \mathbb{R}^{P \times C}$ with P the dimensions of the image (e.g. $P = I \times J$ pixels for 2D images) and C is the total number of considered classes. At each pixel location p, the value $S(p, c) \in [0, 1]$ is the probability of belonging to class c. According to probabilities, one builds a set R of all resulting connected components (i.e. set of connected pixels a priori belonging to the same class according to $S(p, c)$). From this, one finally constructs a complete graph $G = (V, E, X, L)$, where V is the set of nodes (each $v \in V$ corresponds to a region $R_v \in R$) and E the set of edges. X, a node attribute assignment function: $X : V \rightarrow \Re^C$ regards the average membership probability vector over the set of pixels $p \in R_v$. L is an edge attribute assignment function: $L : E \rightarrow \Re^s$ and depends on the considered relationships, being an hyperparameter of the method.

B. Graph neural network

As illustrated by Figure 1 - C, the proposed GNN contains only two layers : the first one focuses on convolution and the second assigns a membership probability vector to each node. One faces many-to-one-or-none matching (or node classification) since the number of nodes to be matched is a priori unknown (i.e. number of regions candidates built by a CNN) and thus graphs are of arbitrary sizes. Consequently, one considers a spatial-based approach rather than a spectral one (spectral graph theory) [6].

The first layer consists in a convolution aiming at aggregating neighborhood information related to each node (general notion of message passing [6]). One considers the edge-conditioned convolution operator [8] (ECConv), recently proposed for graph classification [8], that we consider for node classification in our context of semantic image analysis. For a given node $i \in V$, such a layer computes a new attribute at layer $l + 1$ (leading to $X^{l+1}(i)$), by combining different information from layer l : the attributes of the set $N(i)$ of nodes ($N(i) = \{j | (j, i) \in E\} \cup \{i\}$) and the attributes of the set of related edges (i.e. set $\{L(j, i) | j \in N(i)\}$). According to [8], it can be formalized as follows:

$$\begin{aligned} X^{l+1}(i) &= \frac{1}{|N(i)|} \sum_{j \in N(i)} F^{l+1}(L(j,i)) X^l(j) + b^{l+1} \\ &= \frac{1}{|N(i)|} \sum_{j \in N(i)} \Theta^{l+1}_{ji} X^l(j) + b^{l+1} \end{aligned} \quad (1)$$

where b^{l+1} is a bias and F^{l+1} is a differentiable function (a

multi-layer perceptron in our case). Both entities are learned by training. X^{l+1} is computed from neighboring nodes (and related edges) using the average operator, being a permutation invariant operator (required in such context [6]). The probably most important entity is the mapping function $F^l : \Re^s \rightarrow \Re^{d_l \times d_{l-1}}$, where d_l is the dimension of the node attributes at layer l (i.e. $\forall i \in V,\ X^l(i) \in \Re^{d_l}$). This edge-conditioned function manages the combination of information embedded by nodes (weighted region property) with the one embedded by edges (weighted relationships between regions), through the product $\Theta_{ji}^{l+1} X^l(j)$. The parameters of this function are optimized, over the training dataset, for maximizing the node classification rate. Except for the input of the first layer ($d_0 = C$ as $X^0 = X$), dimensions of node attributes are hyperparameters (i.e. $d_l \mid l \neq 0$). In this work, only one convolution layer is considered to study the relevance of ECConv with d_1 empirically defined in experiment. Note that several convolution layers could be cascaded (as in [8]).

The second layer is a single layer perceptron SLP : $\Re^{d_{l+1}} \rightarrow \Re^C$ providing a class membership probability vector to each node of the graph. Note that one of the classes corresponds to the background (none class). Dimension d_{l+1} is the one of node attributes at the output of the convolution operator.

III. Experiments

A. Datasets

The datasets considered for our experiments are a synthetic dataset and the FASSEG-Instances[1] public dataset created for these experiments (based on the FASSEG).

Synthetic dataset: To construct synthetic images, the reference image, composed of 5 classes (one regarding the background or none class), illustrated in Figure 2 is used. Figure 2 (left part) gives an example of the related graph G built from the reference image. Each region corresponds to a node whose attribute is a probability vector of belonging to each of the 5 classes (mimics the CNN output). Spatial relationships are carried by the edges. In fact, each edge indicates the distance between the barycenter of regions R_i and R_j corresponding to the connected nodes.
100 altered images are generated from the reference one through different processing stages:

- modifying regions (number, location) and the corresponding node attributes to address the issue of many-to-one classification. A random number of regions (between 0 and 2) are added for each class. Their location is fixed with a random shift around the initial position to simulate variations that can occur between any two regions in realistic images. To simulate the uncertainty in the CNN output, membership probabilities are slightly modified. For each node, the probability of its real class is reduced by a random value $a \in [0, 0.4]$ and probabilities of the other classes are randomly increased so that the sum of all probabilities is equal to 1.
- incorporating artefacts (0, 1 or 2) belonging to a new class (white regions in Figure 2) to address the issue of none classification (when CNN detects artefacts corresponding, for instance, to the background). The new nodes of artefacts have a small probability of belonging to the 4 other classes (probability defined randomly in $[0, 0.03]$).

Examples of altered images are given in Figure 2. The number of regions (and nodes) varies from one image to another in order to deal with the issue of arbitrary graph sizes.

FASSEG-Instances: This public dataset is based on the public FASSEG[2] dataset containing 70 human face images with the associated segmentation (hair, eyes, nose, etc.). Some modifications were applied to the original dataset in order to subdivide original labels (e.g. right-eye and left-eye instead of eyes), leading to 9 classes (including the background). For sake of simplicity, the term FASSEG is used in the rest of the paper.

B. Evaluation protocol

All the experiments are carried out in a Python environment on 64-bit Windows with an Intel Core i7 @ 2.70 GHz CPU with 32GB of RAM and an Nvidia Quadro RTX 3000 GPU. In all cases, the GNN model is trained with Adam (Adaptive Moment Estimation). A strategy of reduction of the learning rate on plateau is used with an initial learning rate lr_0 = 0.01 and a reduction factor σ = $5e-4$. The network parameters are adapted to minimize the negative log likelihood loss function:

$$\text{Loss}(Y, \hat{y}) = -\sum_{n=1}^{N} \sum_{c=1}^{C} Y_{n,c} \times \log(\hat{y}_{n,c}) \qquad (2)$$

where N is the total number of nodes, C the number of classes, $Y_{n,c}$ the real class of node n (1 if node n belongs to class c, 0 otherwise) and $\hat{y}_{n,c}$ the probability of node n of belonging to class c with $\hat{y}_n = \text{Softmax}(y_n)$.

In order to study the interest of considering both node and edge attributes, we compare the use of ECConv with the convolution operator GCNConv [6], efficient for semi-supervised node classification [12], which relies only on node attributes and local graph structure (related to incident edges of a node) without considering edge attributes.

To see the impact of the size of the neighborhood [6], especially when the number of nodes (and thus of edges) becomes important, we compare the use of complete graphs and coarsened ones. Coarsening a graph G to $G_c = (V, E_c, X, L)$, where $E_c \subseteq E$, allows to remove edges in order to reduce the number of neighbors of each node. Coarsening is based on edge properties $L(i, j)$ between regions R_i and R_j. In our case, a radius ρ is considered so that for each region R_i ($i \in V$) only the nodes corresponding to the regions R_j at a distance (computed from $L(i, j)$) lower than ρ are connected to the node of R_i.

[1] https://github.com/Jeremy-Chopin/FASSEG-instances

[2] FASSEG: https://github.com/massimomauro/FASSEG-repository

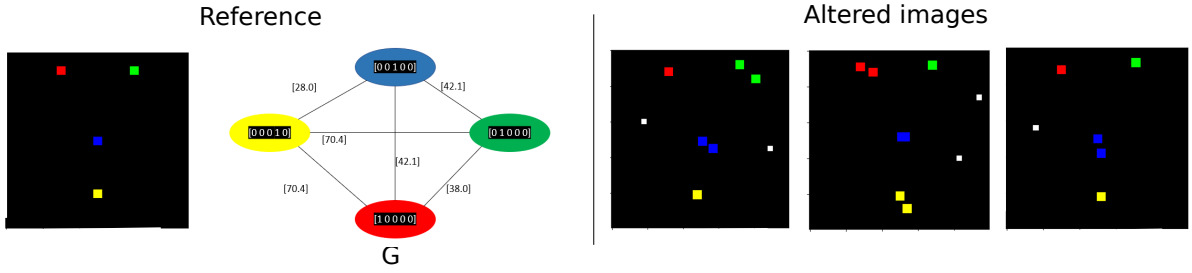

Fig. 2. Synthetic dataset. From a reference image (left part), altered ones are randomly created by modifying the regions (location and number) associated to each class and by integrating outliers. Graphs similar to graph G are created from the images.

Synthetic dataset The GNN is trained using 70 images and tested over the remaining 30 images. The model is trained on 250 epochs. The output dimension of the ECConv operator d_1 is set to 6. Different configurations of graphs are considered:

- Complete graphs with node (vector probability) and edge (distance between barycenters) attributes.
- Complete graphs without node attributes.
- Coarsened graphs considering a radius $\rho = 50$ pixels and $L(i,j) = |b_i - b_j|$, b_i is the barycenter of region R_i.

Performance of classification is measured with the accuracy.

FASSEG U-Net [13] is considered for the preliminary CNN-based segmentation. We split the dataset as follows: 20 images are used for training, 10 for the validation and 40 for the test. 50 epochs are used for training the U-Net and an early stopping strategy is applied, based on the dice loss, to prevent over-fitting. We apply a median filter to remove tiny artefacts from the segmentation map provided. To study the influence of the reduction of the training dataset, smaller datasets (75% of the size of the reference training one i.e. 15 images) are used. In such a case, results are averaged over 20 random selections of such smaller datasets.
To extract connected components from the segmentation map of the CNN, a 8-connectivity is considered. To reduce the number of nodes, only components larger than 30 pixels are associated to a node (smaller components being assigned to the class given by the U-Net). For each image, a graph G is created as detailed in Section II-A. To deal with the irregular shape of the regions, especially the hair whose barycenter can be in the middle of the face, spatial relationships considered for edge attributes correspond to the minimum ($d_{min}^{R_i,R_j} = \min_{a\in R_i, b\in R_j} |a-b|$) and maximum distance ($d_{max}^{R_i,R_j} = \max_{a\in R_i, b\in R_j} |a-b|$) between the connected regions R_i and R_j ($L(i,j) = [d_{min}^{R_i,R_j}, d_{max}^{R_i,R_j}]$).

The GNN is trained using the graphs constructed from the training and validation images and tested over the 40 test ones. The model is trained on 600 epochs and the output dimension of ECConv d_1 is set to 7. We compare our proposal to the segmentation obtained at the output of the U-Net and also to coarsened graphs considering a radius $\rho = 100$ pixels (with respect to mean($d_{min}^{R_i,R_j}, d_{max}^{R_i,R_j}$)). Figure 3 illustrates the coarsening for some examples.

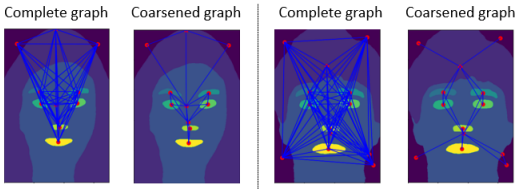

Fig. 3. Examples of coarsening on FASSEG using a 100 pixel radius

To assess the quality of the segmentation obtained, we calculate the Dice (DSC) [14], the Hausdorff distance (HD) [15] and the bounding box dice index (B-DSC), being the dice index between the bounding box of the segmented region and the one of the annotated region. HD and B-DSC are considered to measure the spatial spread of the segmentation (maximum distance between a point of the segmented region and the closest point of the annotated one with HD, spatial coherence with B-DSC).

C. Results

Table I reports involved graph sizes for all experiments, in our many-to-one-or-none classification context, where the number of classes is smaller that the number of nodes. We see that we face arbitrary graphs size. Thus, for the synthetic dataset, the number of nodes goes from 4 to 14 while for FASSEG from 9 to 26 (100%) or 86 (75%).

TABLE I
GRAPHS PARAMETERS FOR SYNTHETIC DATASET AND FASSEG. VALUES INDICATED ARE A MEAN OVER ALL IMAGES OF THE TEST DATASET. NUMBER OF CLASSES (C), OF NODES ($|V|$) AND OF EDGES ($|E|$ AND $|E_c|$), WHERE $|E_c|$ IS THE NUMBER OF EDGES AFTER COARSENING

Dataset	C	$\|V\|$	$\|E\|$	$\|E_c\|$
Synthetic	5	7 (max: 14)	44 (max: 90)	9 (max: 12)
FASSEG 100%	9	12 (max: 26)	172 (max: 650)	33 (max: 134)
FASSEG 75%	9	17 (max: 86)	378 (max: 3867)	99 (max: 728)

Synthetic images
Table II compares the performance of classification of the different graph configurations considered. The very low accuracy (0.20) when ignoring the node features confirms the importance of these attributes for the ECConv convolution

operator in the classification. Edge attributes, providing information on spatial relationships, seem to play a key role in observing the poor performance obtained with GCNConv (0.21) which does not take them into account. The reduction of the number of neighbors by considering a radius of 50 pixels simplifies considerably the graphs by dividing by 5 the number of edges according to Table I while providing perfect classification results (accuracy of 1). Note that coarsening brings additional spatial information to GCNConv (indication of nearby nodes) improving its results (0.59 with coarsening vs 0.21 with complete graphs).

TABLE II
RESULTS OF CLASSIFICATION OF SYNTHETIC DATA WITH DIFFERENT CONFIGURATIONS OF GRAPHS AND CONVOLUTION OPERATORS.

Method	Accuracy
GCNConv	0.21
GCNConv (coarsening)	0.59
ECConv (no node attributes)	0.20
ECConv	0.98
ECConv (coarsening)	**1.00**

FASSEG

Table III shows that the use of a GNN with ECConv improves the results of the U-Net : +0.024 for B-DSC and -7.44 for HD (i.e. improvement of about 27%). In particular, it improves the results of classification for all classes (except the left eyebrow) as illustrated in Table IV. The very poor efficiency of GCNConv proves the importance of edge attributes for classification. We can also see that coarsening with a radius $\rho = 100$ pixels divides by 5 the number of edges (33 against 172 for complete graphs). In this situation, coarsening does not improve the results obtained with ECConv but it improves considerably performance of GCNConv (which remains less efficient than ECConv and the U-Net alone). These findings can also be seen in the first line of Figure 4 where ECConv manages to correct the artefact at the top of the image while ECConv considering coarsening fails and GCNConv produces new segmentation errors (disappearance of the face, two eyes with the same label).

The positive impact of coarsening is shown by the results based on the smaller datasets (75% of the training dataset), with a significant improvement of 41% in terms of HD. Small dataset size leads to degraded CNN performance (DSC of 0.798 vs 0.845 at 100%) and therefore to many more connected components (i.e. 17 nodes against 12 on average) and thus of edges (378 against 172). Coarsening reduces the average number of edges to 99 and improves the performance of ECConv according to Table III. Table IV shows that our solution significantly improves the performance of the U-Net for all considered classes (up to 10% for B-DSC, 3% for DSC, 75% for HD considering the nose). This is also confirmed by the last three lines of Figure 4, for which ECConv with coarsening most accurately corrects CNN errors. With 75% of the training dataset, GCNConv with coarsening also produces new segmentation errors (lines 2 and 4 of Figure 4).

TABLE III
SEGMENTATION RESULTS ON FASSEG WITH CNN ONLY AND CNN FOLLOWED BY GNN (USING ECCONV OR GCNCONV). COMPLETE GRAPHS AND COARSENED ONES (100 PIXEL RADIUS: G_c) ARE COMPARED.

Method	75%			100%		
	DSC	***B-DSC***	***HD***	***DSC***	***B-DSC***	***HD***
CNN	0.798	0.675	54.40	0.845	0.745	27.20
ECConv	0.798	0.728	33.53	0.845	**0.769**	**19.76**
ECConv (G_c)	**0.804**	**0.731**	**32.00**	0.845	0.759	22.80
GCNConv	0.011	0.017	295.74	0.025	0.029	294.45
GCNConv (G_c)	0.537	0.470	124.87	0.599	0.516	100.95

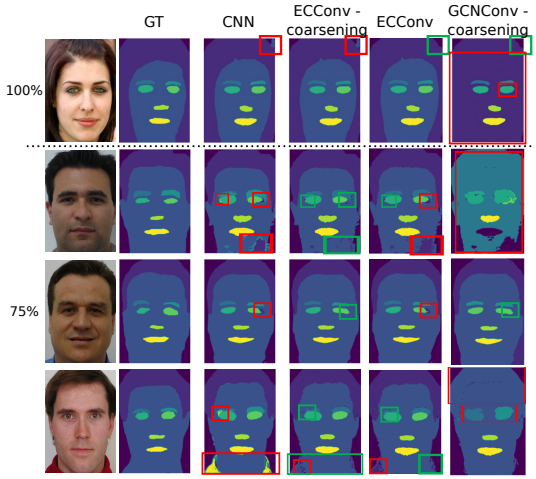

Fig. 4. Examples of segmentation results with the different configurations studied. Bounding boxes highlight regions with significant improvements.

IV. DISCUSSION

This preliminary work illustrates the ability of a basic GNN, exploiting CNN prediction and spatial relationships between regions, to improve deep neural network semantic image segmentation. Promising results of the original approach, combining both node and edge attributes in the message passing strategy and outperforming more common neighborhood aggregator like GCNConv, demonstrates the relevance of using the edge-conditioned convolution (ECConv). Furthermore, the simplicity of the proposed network (only 2 layers) makes it more understandable and reduces the "black box" effect that some GNN can have [6] while avoiding the highly combinatorial nature of QAP approaches [4], [5]. In fact, for instance, the inference time for each face (graph construction and inexact graph matching) is less than 20 seconds.

Moreover, our proposal is robust to small datasets which often lead to larger graphs (many connected components detected by deep learning). This efficiency seems to be helped by graph coarsening as illustrated in Table III and Figure 4. Viewing the benefit of coarsening, influenced by the value of the hyperparameter (neighborhood radius), it appears relevant for future studies to combine in the final SLP the convolution output of several graph configurations with more or less edges depending on the considered radius. This multi-coarsening-

TABLE IV

SEGMENTATION RESULTS PROVIDED BY THE CNN ONLY AND OUR PROPOSAL. RESULTS ARE PROVIDED FOR EACH CLASS (NOT THE BACKGROUND): HR (HAIR), FC (FACE), L-BR (LEFT EYEBROW), R-BR (RIGHT EYEBROW), L-EYE (LEFT EYE), R-EYE (RIGHT EYE), NOSE AND MOUTH.

	75%						**100%**					
Method	**CNN**			**Proposal**			**CNN**			**Proposal**		
Class	***DSC***	***B-DSC***	***HD***	***DSC***	***B-DSC***	***HD***	***DSC***	***B-DSC***	***HD***	***DSC***	***B-DSC***	***HD***
Hr	0.924	0.773	126.26	**0.925**	**0.841**	**86.15**	0.941	0.825	85.18	0.941	**0.838**	**73.54**
Fc	0.948	0.917	48.29	**0.949**	**0.960**	**25.06**	0.957	0.955	24.38	0.956	**0.965**	**19.17**
L-br	0.681	0.547	65.33	**0.686**	**0.617**	**30.19**	0.751	0.679	11.41	0.751	0.678	11.41
R-br	0.667	0.537	65.77	0.652	**0.599**	**42.44**	0.744	0.584	42.50	0.745	**0.653**	**21.10**
L-eye	0.783	0.670	36.47	**0.804**	**0.707**	**23.06**	0.865	0.740	19.88	0.865	**0.782**	**10.11**
R-eye	0.783	0.643	36.97	0.783	**0.681**	**29.30**	0.837	0.718	14.29	0.837	**0.750**	**8.27**
Nose	0.742	0.559	41.41	**0.771**	**0.662**	**10.14**	0.797	0.684	8.47	0.797	**0.697**	**7.18**
Mouth	0.859	0.752	14.69	0.858	**0.779**	**9.42**	0.867	0.770	11.46	0.867	**0.791**	**7.31**

based GNN architecture would allow to combine complementary neighborhood information from different graph structures and thus enrich the predictions for each node. Note that other works dealing with multi-scale GNN exist but do not consider both node and edge attributes [16].

Although our approach is promising, certain limitations may be pointed out. First, we only compare our results with a U-Net network but we should in future works consider other more recent CNN-based method [17], [18]. Then, performance is evaluated on short datasets. It is sufficient to show the relevance of the method but additional studies will evaluate our method on other applications, such as medical ones [19], with larger datasets.

V. CONCLUSION

We propose a GNN-based technique to improve deep neural network image segmentation using an inexact graph matching procedure. Considered approach exploits vector probability from the CNN output as node attributes and spatial relations between regions as edge attributes. The simple architecture of the GNN considered, composed of a edge-conditioned convolution and a single-layer perceptron demonstrates the relevance of ECConv to perform node classification in a context of semantic image segmentation. The speed and simplicity of the proposed model are major advantages over traditional QAP approaches. Proposal is also robust to small datasets thanks to a preprocessing graph coarsening. Preliminary experiments on both a synthetic dataset and FASSEG are promising as they show that our approach significantly improves segmentation provided by a CNN.

Future works will evaluate our method on other applications with larger datasets and compare it with different CNN-based segmentation method. The multi-coarsening GNN path remains to be exploited to refine segmentation performance.

REFERENCES

[1] A. Garcia-Garcia, S. Orts-Escolano, S. Oprea, V. Villena-Martinez, P. Martinez-Gonzalez, and J. Garcia-Rodriguez, "A survey on deep learning techniques for image and video semantic segmentation," *Applied Soft Computing*, vol. 70, pp. 41 – 65, 2018.

[2] J.-B. Fasquel and N. Delanoue, "A graph based image interpretation method using a priori qualitative inclusion and photometric relationships," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 41, pp. 1043–1055, 2019.

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

[4] J. Maciel and J.P.Costeira, "A global solution to sparse correspondence problems," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 25, pp. 187–199, 2003.

[5] A. Zanfir and C. Sminchisescu, "Deep learning of graph matching," in *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2018, pp. 2684–2693.

[6] D. Bacciu, F. Errica, A. Micheli, and M. Podda, "A gentle introduction to deep learning for graphs," *Neural Networks*, vol. 129, pp. 203–221, 2020.

[7] Z. Zhang, P. Cui, and W. Zhu, "Deep learning on graphs: A survey," *IEEE Transactions on Knowledge and Data Engineering*, vol. 34, pp. 249–270, 2020.

[8] M. Simonovsky and N. Komodakis, "Dynamic edge-conditioned filters in convolutional neural networks on graphs," in *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR 2017)*, 2017.

[9] A. S. Nassar, S. D'Aronco, S. Lefevre, and J. D. Wegner, "Geograph: Graph-based multi-view object detection with geometric cues end-to-end," in *Computer Vision - ECCV 2020 - 16th European Conference, Glasgow, UK, Proceedings, Part VII*, ser. Lecture Notes in Computer Science, vol. 12352. Springer, 2020, pp. 488–504.

[10] S. Ouyang and Y. Li, "Combining deep semantic segmentation network and graph convolutional neural network for semantic segmentation of remote sensing imagery," *Remote Sensing*, vol. 13, 2021.

[11] Q. Diao, Y. Dai, C. Zhang, Y. Wu, X. Feng, and F. Pan, "Superpixel-based attention graph neural network for semantic segmentation in aerial images," *Remote Sensing*, vol. 14, p. 305, 2022.

[12] T. Kipf and M. Welling, "Semi-supervised classification with graph convolutional networks," in *International Conference on Learning Representations*, 2017.

[13] O. Ronneberger, P. Fischer, and T.Brox, "U-net: Convolutional networks for biomedical image segmentation," *Medical Image Computing and Computer-Assisted Intervention*, vol. 9351, pp. 234–241, 2015.

[14] A. Zijdenbos, B. Dawant, R. Margolin, and A. Palmer, "Morphometric analysis of white matter lesions in mr images: method and validation," *IEEE Transactions on Medical Imaging*, vol. 13, pp. 716–724, 1994.

[15] M. Beauchemin, K. Thomson, and G. Edwards, "On the hausdorff distance used for the evaluation of segmentation results," *Canadian Journal of Remote Sensing*, vol. 24, pp. 3–8, 1998.

[16] G. Hongyang and J. Shuiwang, "Graph U-Nets," in *Proceedings of the 36th International Conference on Machine Learning*, ser. Proceedings of Machine Learning Research, vol. 97. PMLR, 2019, pp. 2083–2092.

[17] M. Lou, J. Meng, Y. Qi, X. Li, and Y. Ma, "MCRNet: Multi-level context refinement network for semantic segmentation in breast ultrasound imaging," *Neurocomputing*, vol. 470, pp. 154–169, 2022.

[18] S. Chen, Z. S. Gamechi, F. Dubost, G. van Tulder, and M. de Bruijne, "An end-to-end approach to segmentation in medical images with cnn and posterior-crf," *Medical Image Analysis*, vol. 76, 2022.

[19] C. Oyarzun Laura, S. Wesarg, and G. Sakas, "Graph matching survey for medical imaging: On the way to deep learning," *Methods*, 2021.