QAP Optimisation with Reinforcement Learning for Faster Graph Matching in Sequential Semantic Image Analysis

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, Nantes 6004, 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. The paper addresses the fundamental task of semantic image analysis by exploiting structural information (spatial relationships between image regions). We propose to combine a deep neural network (CNN) with graph matching where graphs encode efficiently structural information related to regions segmented by the CNN. Our novel approach solves the quadratic assignment problem (QAP) sequentially for matching graphs. The optimal sequence for graph matching is conveniently defined using reinforcement-learning (RL) based on the region membership probabilities produced by the CNN and their structural relationships. Our RL based strategy for solving QAP sequentially allows us to significantly reduce the combinatorial complexity for graph matching. Preliminary experiments are performed on both a synthetic dataset and a public dataset dedicated to the semantic segmentation of face images. Results show that the proposed RL-based ordering significantly outperforms random ordering, and that our strategy is about 386 times faster than a global QAP-based approach, while preserving similar segmentation accuracy.

Keywords: Semantic image analysis · Structural information · Graph matching · Quadratic assignment problem · Reinforcement learning

1 Introduction

Semantic segmentation is a fundamental but challenging task in computer vision, often managed using deep neural networks such as U-Net [9]. Structural infor-

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. 47–58, 2022.
https://doi.org/10.1007/978-3-031-09037-0_5

mation [2,5] such as spatial relationships is not explicitly used in such networks, although some recent works aim at exploiting it, e.g. CRF-based approaches [7] and CNN based semantic segmentation followed by inexact graph matching [3].

In this paper, we focus likewise on graph-based approaches exploiting relationships observed at high semantic level in annotated training images or provided by qualitative descriptions of the scene content [5]. In this context, graph vertices and edges encode regions and spatial relationships produced by a segmentation network and observed in annotated training images, leading to an inexact graph matching problem, expressed classically as a quadratic assignment problem (QAP) [16]. Note that some recent approaches solve graph matching with machine learning (e.g. graph neural networks [1]). Although promising for many application domains [17], large and representative training datasets of annotated graphs are required. Another difficulty is the definition of the appropriate architecture, and the management of both vertex and edge information, while edge features (related to relationships between regions) are often ignored [17].

One of the main drawbacks of QAP-based graph matching lies in its highly combinatorial nature [16]. In this context, our proposal is to solve it in a sequential manner, where vertices are progressively matched in order to reduce the complexity. This means that the semantic image analysis is done progressively: first identified regions are used to discover next ones [4,6] (this is closed to the notion of seeded graph matching [8]). The difficulty is to learn the optimal segmentation/graph matching order, to ensure that all regions are finally recovered. In this paper, we propose to solve this problem by reinforcement learning [12,14]. Note that, to our knowledge, such an approach has never been considered for graph-based semantic image segmentation, although it has been recently studied for graph matching (but in a different context [8]). Recent related works in computer vision focus on other tasks such as, for instance, object detection [10], object tracking [15], landmark detection [11] or control of regions of interest in video analysis [11].

This work is an extension of a recently proposed approach involving QAP-based graph matching that ignores this sequential alternative and therefore suffers from a high complexity [3]. The originality and contribution of this paper rely on challenging image understanding tasks by combining, on top of deep-learning-based segmentation, high-level structural information, inexact graph matching and a reinforcement-learning-based sequential strategy. Section 2 describes the proposed method while Sect. 3 presents experiments and results demonstrating the performance of our approach. We finally conclude in Sect. 4.

2 Reinforcement Learning for Sequential Graph Matching

Figure 1 provides an overview of the approach. A Convolutional Neural Network (CNN) is trained for image semantic segmentation using an annotated dataset (Fig. 1-Training). To correct segmentation errors (Fig. 1-Inference), we propose to use spatial relationships observed between identified regions of the annotated

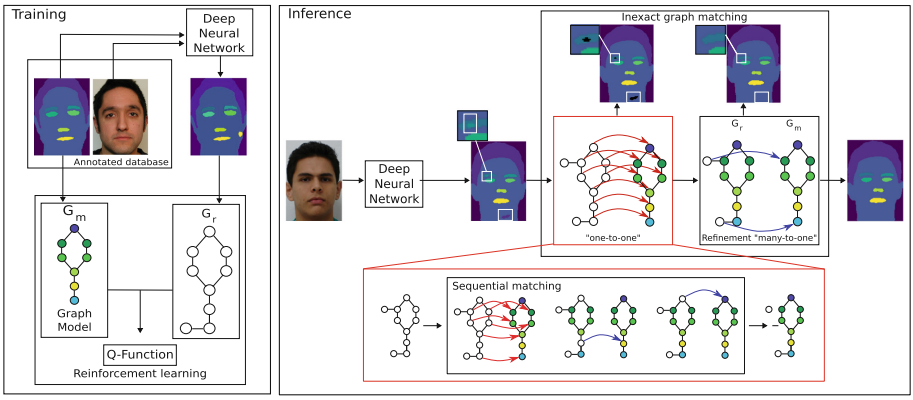

Fig. 1. Overview of the proposed approach. **Training**: Annotated data are used to train a CNN and learn the model graph. Segmentations over the training data are used with the graph model to learn the Q-function. **Inference**: Segmentation produced by the CNN is used to create image graph. A sequential one-to-one matching is done with a sequential refinement to improve the semantic segmentation.

training dataset, leading to an inexact-graph-matching procedure, between G_m (built from the training dataset) and G_r (built from the CNN output). When analysing an unknown image (Fig. 1-right), a hypothesis graph G_r is built from the initial CNN segmentation result. To identify regions, G_r is matched with G_m, which is an inexact graph matching problem, as there are more regions in G_r than in G_m due to artifacts. We propose to do this sequentially in two steps. First, an initial "one-to-one" matching is performed to recover one region candidate (vertex of G_r) per class (one vertex of G_m). This is done sequentially according to the ordering learned by reinforcement (based on a Q-Function resulting from a preliminary training - Fig. 1-Training). The second step (refinement) focuses on matching remaining artifacts, this being also done sequentially in any order. We hereafter detail each of these steps.

2.1 Neural Network and Graphs

When analysing an image, the neural network provides a tensor $S \in \mathbb{R}^{P \times N}$ with P the dimensions of the query image (e.g. $P = I \times J$ pixels for 2D images) and N is the total number of classes considered for segmentation. At each pixel location p, the value $S(p, n) \in [0, 1]$ is the probability of belonging to class n. The segmentation map $\mathcal{L}^*$ selects the label n of the class with the highest probability:

$$\forall p \in \{1, \ldots, P\},\ \mathcal{L}^*(p) = \underset{n \in \{1,\ldots,N\}}{\arg\max}\ S(p, n). \tag{1}$$

From $\mathcal{L}^*$, we define a set R of all resulting connected components, and finally the graph $G_r = (V_r, E_r, A, D)$, 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:

$$\forall v \in V_r, \forall n \in \{1, \ldots, N\},\ A(v)[n] = \frac{1}{|R_v|} \sum_{p \in R_v} S(p, n). \tag{2}$$

We consider a complete graph where each edge $e = (i, j) \in E_r$ has an attribute defined by the function D (hyperparameter in our method, detailed in experiments), associated with a relation between the regions R_i and R_j.

The model graph $G_m = (V_m, E_m, A, D)$ is built from the training set and is composed of N vertices (one vertex per class). 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 relationships (in the training set) between the regions (according to the relation D considered).

2.2 Sequential One-to-one Matching by Reinforcement Learning

The proposed sequential one-to-one matching between $G_r = (V_r, E_r, A, D)$ and $G_m = (V_m, E_m, A, D)$ is formulated as a QAP to be solved sequentially by Q-learning, for finally finding the best assignment X^*:

$$X^* = \arg\min_X \left\{ \mathrm{vec}(X)^T K\, \mathrm{vec}(X) \right\}, \tag{3}$$

where $X \in \{0, 1\}^{|V_r| \times |V_m|}$, X_{ij} means that the vertex $i \in V_r$ is matched to the vertex $j \in V_m$, $vec(X)$ is the column vector representation of X, and T denotes the transposition operation. The matrix K is defined by:

$$K = \alpha\ K_v + (1 - \alpha)\ K_e, \tag{4}$$

and embeds the dissimilarities between the two graphs: K_v embeds the dissimilarities between V_r and V_m (diagonal elements) and K_e embeds the dissimilarities between E_r and E_m (non-diagonal elements). The parameter $\alpha \in [0, 1]$ allows weighting the relative contributions of vertex and edge dissimilarities.

For a sequential graph matching, one learns, by reinforcement, from interactions between the agent and the environment [12]. From a given state s_t (set of already matched nodes, at step t of the sequential matching procedure), the agent (the algorithm) selects and triggers an action (i.e. trying to match a new vertex of $|V_r|$ with a new one of $|V_m|$, or a new subset of vertices). The environment (encompassing image, semantic segmentation, graphs and graph matching computations) performs this action, and gives back to the agent the resulting new state s_{t+1} (matching result) together with a reward.

In this work, the considered reinforcement learning (RL) method is based on Q-learning using a Q-function defined by a Q-Table, that appeared appropriate

for preliminary experiments. As underlined in [12], it is widely accepted that such a value-based RL algorithm is appropriate for a discrete RL scenario, which is the case of our graph matching problem (discrete decision making problem).

The design of the agent for our graph matching problem is detailed hereafter in terms of state, action and reward.

State. As in [8], the state $s_t \in \mathcal{S}$ (at the step t of the episode or matching procedure) is the subset $V_{r,t} \subseteq V_r$ of vertices matched with a subset $V_{m,t} \subseteq V_m$, where $|V_{r,t}| = |V_{m,t}|$, and $\mathcal{S}$ represents all possible partial matchings. The related bijective assignment matrix is $X_t \in \{0,1\}^{|V_{r,t}| \times |V_{m,t}|}$, so that $\forall p \in V_{m,t}, (\sum_{i=1}^{|V_{m,t}|} X_{pi} = 1) \wedge (\sum_{i=1}^{|V_{m,t}|} X_{ip} = 1)$. The matching procedure (episode) goes from $t = 0$ ($V_{r,0} = V_{m,0} = \emptyset$) to $t = \infty$ ($|V_{r,\infty}| = |V_m|$ and $V_{m,\infty} = V_m$). We observed experimentally that only a limited number of steps is needed.

Action. The action $a_t \in \mathcal{A}_t$, achieved by the agent at step t, consists in selecting a set of vertices of V_m not in $V_{m,t}$ (i.e. $V_m \setminus V_{m,t}$) and finding the corresponding ones in $V_r \setminus V_{r,t}$. $\mathcal{A}_t$ is the set of possible sets of vertices, and depends on t (i.e. already matched vertices at step t are ignored). In our case, at $t = 0$, sets of size larger than one element are considered, while, for $t > 0$, single nodes are investigated. The motivation is to begin by finding a small subgraph matching (seeded graph matching [8]) and then to consider only single nodes to ensure a low complexity. At each step, a QAP optimization is achieved to find the new matching(s), according to Eq. 3, where the assignment matrix is initialized according to X_t.

Reward. When learning, the agent receives a reward r, based on the quality of the resulting matching. Compared to [8], the reward is not based on the cost related to Eq. 3 but on the quality of the resulting semantic segmentation, similarly to [11], involving a similarity measurement between the recovered region(s) and the expected one(s). The motivation is to favor the matching with the most similar regions, as several regions (over-segmentation) of the image being analyzed can be associated (and therefore matched) with the same region of the reference segmentation. The reward, depending on both the state s_t and the selected action a_t, is the one considered in [11]:

$$r(a_t, s_t) = \begin{cases} \text{DC} + 1 & \text{if DC} > 0.1, \\ -1 & \text{otherwise} \end{cases} \tag{5}$$

where DC is the Dice index between the region(s) associated with the newly matched vertex (or vertices) and the expected one(s).

Sequential Matching. After the learning procedure leading to the Q function, the matching ordering (i.e. optimal action a_t to be selected at step t) is defined, at each step $t \in [0, \infty]$, by:

$$a_t = \underset{a \in \mathcal{A}_t}{\arg\max}(Q(s_t, a)). \tag{6}$$

where Q ($Q : \mathcal{S} \times \mathcal{A} \rightarrow \Re$) is the learned Q-Table [12], representing the maximum expected future rewards for actions at each state ($\mathcal{A}_t \subseteq \mathcal{A}$ in Eq. 6). Q is learned [12] over several episodes achieved on the training dataset, using previously defined notions of state, action and reward. Applying this policy leads to the one-to-one matching X^I.

Complexity. The complexity is directly related to the number of evaluated assignments according to Eq. 3, depending on the number of vertices involved and the related set of possible matchings (i.e. set of $X \in \{0,1\}^{|V_r| \times |V_m|}$). Without considering the proposed sequential approach, the number of evaluations $\mathrm{NE}_{\mathrm{QAP}}$ equals the following number of $|V_m|$-permutations of $|V_r|$ (without repetitions), or arrangements (i.e. vertex sets from V_r, of size $|V_m|$, to be matched with the V_m vertices):

$$\mathrm{NE}_{\mathrm{QAP}} = P_{|V_m|}^{|V_r|} = \frac{|V_r|!}{(|V_r| - |V_m|)!} \tag{7}$$

With the sequential approach, the number of evaluations $\mathrm{NE}_{\text{QAP-RL}}$ is:

$$\mathrm{NE}_{\text{QAP-RL}} = P_{|S|}^{|V_r|} + \sum_{i=0}^{|V_m|-|S|} |V_r| - |S| - i \tag{8}$$

where $S \subseteq V_m$ is the set of vertices involved in the first step of graph matching procedure. Each following step involves only one vertex (right term of Eq. 8). Because $|S| \leq |V_m|$, the number of evaluations can be significantly reduced by minimizing $|S|$ (i.e. $|S| \ll |V_m|$).

2.3 Sequential Refinement: Many-to-one-or-none Matching

The unmatched remaining nodes are then matched sequentially but in a random manner. For each node $k \in V_r \setminus V_{r,\infty}$, one searches for the best assignment (element of V_m), minimizing the matching cost according to Eq. 3. In terms of complexity, this only involves the evaluation of $|V_m|$ assignment matrices per remaining $k \in V_r \setminus V_{r,\infty}$: $\mathrm{NE}_{\mathrm{Refinement}} = |V_r \setminus V_{r,\infty}| \times |V_m|$ (to be added to the complexity related to Eq. 7 or 8).

3 Experiments

3.1 Datasets

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

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

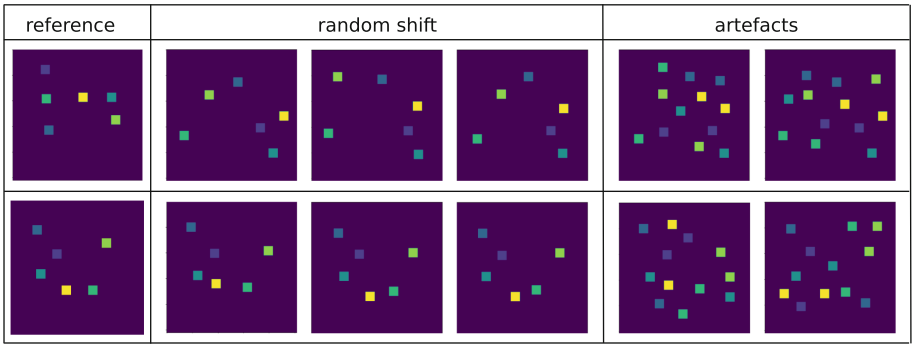

Fig. 2. Synthetic dataset. From reference images, altered ones are randomly created by first applying a random shift on region positions and by then integrating artifacts.

Synthetic dataset. Ten types of synthetic images are used (two are reported in Fig. 2). For each one, a reference image, composed of 6 regions/classes, is considered (from which G_m is built) together with 100 altered versions (from which 100 G_r are built). Altered images are generated from reference ones, by modifying region location (random shift around the initial position) and by incorporating randomly placed artifacts (one artifact per class). Note that the random shift for some regions is larger than for others, in order to simulate relationships variations that can differ between any two regions in realistic images. The considered relationships (D assignment function) are the distances between region barycenters: $\forall (i,j) \in V_{\cdot}^2,\ D((i,j)) = \|\bar{R}_j - \bar{R}_i\|$, where $\bar{R}_i$ is the barycenter of the region associated with vertex i. The dissimilarity between two edges ($(i,j) \in E_m$ and $(k,l) \in E_r$) is computed as the difference between $D((i,j))$ and $D((k,l))$ (used to compute K in Eq. 3): ${D_1}_{(i,j)}^{(k,l)} = \frac{D((k,l)) - D((i,j))}{C_s}$, where C_s is the length of the diagonal of the image, so as to keep the value in the interval $[0,1]$. To mimic the CNN output, each region is associated with an attributed vector representing a membership probability vector (A assignment function). For altered images, a probability is set randomly in $[0.7, 0.9]$ and assigned to the reference region/class, while the remaining quantity is randomly divided among the other classes.

FASSEG-Instances. This public dataset is based on the public FASSEG[2] dataset containing 60 human face images with the associated expert segmentation of face regions (eyes, nose, mouth...). We applied some modifications to the original FASSEG dataset in order to subdivide original labels (e.g. right-eye and left-eye instead of only eyes, i.e. two distinct instances of eyes), leading to 9 classes. Note that, although FASSEG includes faces in multiple poses, one considers frontal ones only because the considered graph matching technique may not be robust to face pose changes [3], except if spatial relations are defined in an intrinsic frame and not absolutely, this aspect being out of the scope of our study focusing

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

on QAP optimization. For the sake of simplicity, the term FASSEG is used in the rest of the paper. The considered relationships are based on both minimum and maximum distances between regions, leading to the following assignment function $D((i,j)) = [d_{\min}^{(i,j)}, d_{\max}^{(i,j)}]$, where:

$$d_{\min}^{(i,j)} = \min_{p \in R_i, q \in R_j} (|p-q|) \text{ and } d_{\max}^{(i,j)} = \max_{p \in R_i, q \in R_j} (|p-q|) \tag{9}$$

The resulting dissimilarity between two edges $(i,j) \in E_m$ and $(k,l) \in E_r$ is:

$$D_{2}{}_{(i,j)}^{(k,l)} = \frac{\lambda}{C_s} \left(|d_{\min}^{(i,j)} - d_{\min}^{(k,l)}| \right) + \frac{(1-\lambda)}{C_s} \left(|d_{\max}^{(i,j)} - d_{\max}^{(k,l)}| \right) \tag{10}$$

where $\lambda \in [0,1]$ is a parameter (set to 0.5 is our experiments) to balance the influence of the distances, and C_s the maximum diagonal length of the image.

3.2 Evaluation Protocol

On the synthetic dataset, for each reference image, the Q function is learned using 60 images, with 50 episodes per image, therefore leading to 3000 episodes. The remaining 40 images are used for testing purposes. Results are averaged over the 10 reference images. On this dataset, we considered two sub-experiments: one with attributes on edges only (Synthetic 1, with $\alpha = 0$ in Eq. 4 to favor structural information only) and one on both vertices and edges (Synthetic 2, with $\alpha = 0.5$ in Eq. 4). On the FASSEG dataset, 20 images are used for training both the U-Net [9] (used for the initial segmentation) and the Q function (with 50 episodes per image, i.e. 1000 episodes for this dataset). In both cases, a seed of 3 vertices is considered for the first step of the sequential graph matching (the seed composition being learned by reinforcement), while the next steps involve only single vertices.

Our sequential RL-based approach is compared to a random ordering (averaged over 100 random orderings for each of the synthetic test images and for FASSEG test ones). When possible, our approach is compared to the standard QAP. Due to the huge number of permutations, QAP may not be applied in some cases, in particular on FASSEG. We therefore consider a constrained QAP [3], by reducing the number of investigated assignment matrices (in Eq. 3): for a given vertex (region) $i \in G_m$, the considered candidates in G_r are not all possible vertices (regions) but only those with the highest membership probability, according to the U-Net, of being associated with class/vertex i.

Evaluation measures include the number of permutations (i.e. number of assignment matrices), and, when possible, the runtime (Intel i7-8850H CPU). One also measures, when possible, the segmentation accuracy (Dice index).

3.3 Results

Table 1 reports results on the segmentation accuracy for both synthetic and FASSEG datasets. The reinforcement learning significantly outperforms random

Table 1. Segmentation results (Dice index) obtained by the sequential approach (RL-based ordering and random ordering), the QAP, and the constrained QAP. FASSEG-R corresponds to segmentation results after the refinement step, while FASSEG concerns only the one-to-one matching. Some results are not available (too high a computation time).

Method	Synthetic 1	Synthetic 2	FASSEG	FASSEG-R
Reinforcement	0.6 ± 0.38	0.96 ± 0.08	0.82 ± 0.13	0.82 ± 0.13
Random ordering	0.21 ± 0.25	0.92 ± 0.14	0.71 ± 0.12	0.78 ± 0.1
QAP	0.72 ± 0.31	0.98 ± 0.05	NA	NA
Constrained QAP		0.99 ± 0.03	0.83 ± 0.06	0.84 ± 0.04

ordering, demonstrating the relevance of the proposed sequential approach. Compared to a global QAP-based matching, our approach is significantly less efficient on Synthetic 1, while only slightly less efficient on Synthetic 2. This illustrates that, even with an optimized ordering, considering few nodes (only 3 at the beginning compared to the global QAP that directly searches for the 6 ones) is not sufficient when only the relationships are considered (i.e. Synthetic 1 ignores vertex attributes), because one fails identifying the relevant matching among the large set of possible sub-graph matchings.

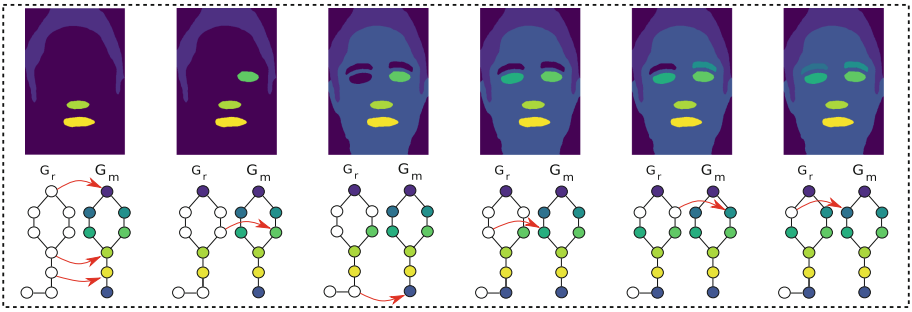

Fig. 3. Learned optimized ordering on FASSEG: starting with the nose, mouth and hair (initial seeded graph matching), before continuing with one eye, the skin, the second eye and finally eyebrows.

On FASSEG, the efficiency of our proposal is highly similar to the one of the constrained QAP (and significantly higher than the random ordering), although more classes (9) are involved, illustrating the relevance of our proposal. Figure 3 illustrates the learned optimized ordering on FASSEG, while Fig. 4 reports some examples of results on different faces (both our approach and constrained QAP fail for the second face), where $s_\infty = s_6$ and $V_{r,\infty} = V_{r,6}$. These examples qualitatively illustrate, in particular, the relevance of our approach compared to a random ordering.

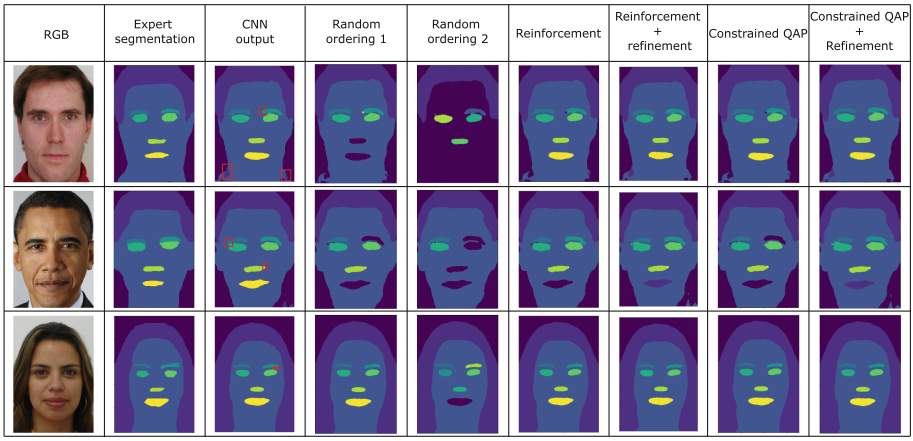

Fig. 4. Examples of segmentations obtained on FASSEG, using our approach (sequential matching before and after the refinement), the constrained QAP (before and after refinement) and random ordering (without considering the refinement). One also reports expert segmentation and CNN output (red boxes surround some initially misclassified regions).

Table 2 provides the required number of evaluated assignment matrices (computed according to Eqs. 7 and 8, respectively for QAP and reinforcement, and measured for the constrained QAP), as well as measured computation times (except for QAP on FASSEG because it is too time consuming). The QAP involves significantly much more evaluations than our proposal (values are averaged over test images on FASSEG): the sequential approach depicts a significantly smaller complexity than QAP.

Table 2. Number of evaluated assignment matrices and measured runtime (in seconds in brackets).

Method	Synthetic 1	Synthetic 2	FASSEG
Reinforcement	1344 (0.035s)		3570 (0.25s)
QAP	665280 (13.5s)		$8.89\,10^9$ (NA)
Constrained QAP	665280 (13.5s)	64 (0.001s)	81 (0.57s)

This is confirmed by measured computation times: on the synthetic dataset, our proposal (0.035 s) is about 386 times faster that QAP (13.5 s). In our experiments, the constrained QAP is used to provide segmentations on FASSEG (too much time consuming for the QAP), by considering a global matching, to be compared with our sequential one. The counterpart is that we assume that the final identity/label of a region (final matching) initially corresponds to a label

associated with the highest membership probability (CNN output), which may be the case in practice (e.g. when the CNN hesitates between two labels). Moreover, such a constrained QAP does not apply if vertex attributes do not embed membership probabilities (e.g. non CNN-based over-segmentation, such as for Synthetic 1 where the number of evaluated assignment is the same for both QAP and constrained QAP). In such a restrictive and less generic context, we measured that the number of permutations ranges from 1 (case of perfect CNN-based segmentation) to 1600 (many region candidates per class), with a mean value of 81 (see Table 2). Note that the measured computation time is, in average, equal to 0.57 s (mainly due to the application of the constraint, i.e. finding the list of region candidates) compared to 0.25 s with our sequential approach, although more assignment matrices are evaluated (3570).

4 Conclusion

We propose a reinforcement-learning-based framework for the sequential semantic analysis of image content by exploiting structural information formulated as a QAP-based inexact graph matching problem. Preliminary experiments on both a synthetic dataset and the FASSEG dataset are promising as they show that our approach dramatically reduces the complexity of this QAP-based inexact graph matching problem, while preserving the efficiency of the analysis.

Future works and additional studies will first evaluate our method on other applications with larger datasets. An important point to be studied, and ignored in this preliminary evaluation, is the influence of the size of the initial seed in the sequential approach based on reinforcement learning, as well as the ability to automatically learn its optimal size. Another aspect to be studied is the extension of this framework so that the ordering can be dynamically adapted, involving, for instance, the ability to integrate revocable actions [8]. Using a Dueling Deep Q-Networks approach [13] would allow adapting the strategy to the current image. Finally, the final refinement step, possibly involving outliers/artifacts, is managed by considering a random ordering. It would be interesting to investigate whether it could benefit from an optimized ordering, again based on reinforcement learning.

References

1. Bacciu, D., Errica, F., Micheli, A., Podda, M.: A gentle introduction to deep learning for graphs. Neural Netw. **129**, 203–221 (2020). https://doi.org/10.1016/j.neunet.2020.06.006
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. Fasquel, J.B., Delanoue, N.: 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
5. 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
6. Fouquier, G., Atif, J., Bloch, I.: Sequential model-based segmentation and recognition of image structures driven by visual features and spatial relations. Comput. Vis. Image Underst. **116**(1), 146–165 (2012). https://doi.org/10.1016/j.cviu.2011.09.004
7. Kamnitsas, K., et al.: Efficient multi-scale 3D CNN with fully connected CRF for accurate brain lesion segmentation. Med. Image Anal. **36**, 61–78 (2017). https://doi.org/10.1016/j.media.2016.10.004
8. Liu, C., Wang, R., Jiang, Z., Yan, J., Huang, L., Lu, P.: Revocable deep reinforcement learning with affinity regularization for outlier-robust graph matching. CoRR (2020). https://arxiv.org/abs/2012.08950
9. 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
10. Pirinen, A., Sminchisescu, C.: Deep reinforcement learning of region proposal networks for object detection. In: 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 6945–6954 (2018). https://doi.org/10.1109/CVPR.2018.00726
11. Sun, M., Xiao, J., Lim, E.G., Xie, Y., Feng, J.: Adaptive ROI generation for video object segmentation using reinforcement learning. Pattern Recogn. **106**, 107465 (2020). https://doi.org/10.1016/j.patcog.2020.107465
12. Sutton, R.S., Barto, A.G.: Reinforcement Learning: An Introduction, 2nd edn. The MIT Press (2018). http://incompleteideas.net/book/the-book-2nd.html
13. Wang, Z., Schaul, T., Hessel, M., Van Hasselt, H., Lanctot, M., De Freitas, N.: Dueling network architectures for deep reinforcement learning. In: 33rd International Conference on International Conference on Machine Learning, vol. 48, pp. 1995–2003 (2016)
14. Yang, Y., Whinston, A.: A survey on reinforcement learning for combinatorial optimization (2020). https://arxiv.org/abs/2008.12248
15. Yun, S., Choi, J., Yoo, Y., Yun, K., Choi, J.Y.: Action-driven visual object tracking with deep reinforcement learning. IEEE Trans. Neural Netw. Learn. Syst. **29**(6), 2239–2252 (2018). https://doi.org/10.1109/TNNLS.2018.2801826
16. 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
17. Ziyao, L., Liang, Z., Guojie, S.: GCN-LASE: towards adequately incorporating link attributes in graph convolutional networks. In: 28th International Joint Conference on Artificial Intelligence (IJCAI), pp. 2959–2965 (2019). https://doi.org/10.24963/ijcai.2019/410