Semantic image segmentation based on spatial relationships and inexact graph matching
Jérémy Chopin, Jean-Baptiste Fasquel, Harold Mouchère, R. Dayot, I. Bloch

To cite this version:

HAL Id: hal-02916165
https://hal.archives-ouvertes.fr/hal-02916165
Submitted on 16 Dec 2020

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L’archive ouverte pluridisciplinaire HAL, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d’enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.
Semantic image segmentation based on spatial relationships and inexact graph matching

Jérémie Chopin
LARIS, Université d’Angers
Jean-Baptiste Fasquel
LARIS, Université d’Angers
Harold Mouchère
LS2N, Université de Nantes, CNRS UMR 6004
Rozenn Dahyot
School of Computer Science & Statistics, Trinity College Dublin
Isabelle Bloch
LTCI, Télécom Paris, Institut Polytechnique de Paris

Abstract—We propose a method for semantic image segmentation, combining a deep neural network and spatial relationships between image regions, encoded in a graph representation of the scene. Our proposal is based on inexact graph matching, formulated as a quadratic assignment problem applied to the output of the neural network. The proposed method is evaluated on a public dataset used for segmentation of images of faces, and compared to the U-Net deep neural network that is widely used for semantic segmentation. Preliminary results show that our approach is promising. In terms of Intersection-over-Union, the improvement is of 2.4% in average, compared to U-Net, and up to 24.4% for some regions. Further improvements are observed when reducing the size of the training dataset (up to 8.5% in average).

Index Terms—Computer vision, Deep learning, Inexact graph matching, Quadratic assignment problem.

I. INTRODUCTION

Deep learning has shown its efficiency in many fields [1], in particular for semantic segmentation of images in computer vision [2]. One limitation of deep neural network approaches concerns the requirement of a large and representative training dataset, including its annotations, as recently highlighted in [3]. Moreover, deep learning is intrinsically based on embedded information at pixel level, which is then processed and combined through different layers involving multiple parameters to be optimized.

This type of approach ignores the structural information that can be observed at a high level. This structural information can for example concern spatial relationships between different spatial entities, as illustrated in Figure 1 with the relative positions between the main regions of the face observed in the annotated image. This type of high-level structural information is often ignored in semantic image analysis, despite its potential illustrated by related works, integrating spatial, inclusion or even photometric relations [4]–[7], often applied to medical imaging [8]–[10]. This information is commonly represented using graphs, where vertices correspond to regions, and edges carry the structural information. Then, recognition turns into a general graph matching problem [5], [6], [11].

Fig. 1. Semantic segmentation and spatial relationships. Images are extracted from the public FASSEG database [12], [13]. a) Initial image. b) Semantic segmentation where some distinct regions belong to the same class (e.g. eyes). c) Semantic segmentation where each region belong to a specific class (e.g. left eye, right eye). d) Spatial relationships modeled by a graph where each vertex corresponds to a specific region of c: background, hair, left/right eyebrow, left/right eye, nose, mouth. Edges carry relationships, corresponding to distances between regions in our case. For the sake of clarity, only some of the edges of the complete graph are displayed.

In this context, we propose to combine two approaches: deep neural networks that have proven very efficient but often require large training datasets, and graphs for encoding high level relational representations in the visual scenes. Of note the use of deep neural networks for graph matching is a topic that is currently attracting a lot of interest from the scientific community, for problems other than computer vision such as biology, social sciences, linguistics [14]–[16].

The originality of the proposed method is that it allows for such a combination, formulated as an inexact graph matching problem applied to the output of deep neural networks. It enables to correct the semantic segmentation resulting from the single use of the probability map produced by the neural networks, by taking into account spatial relationships observed in the annotated database. The use of the global spatial structure
of the scene also enables to be less sensitive to the diversity (and therefore the size) of the training dataset used to train the neural network. In order to manage spatial relationships in an explicit manner, we consider a graph matching approach formulated as a quadratic assignment problem (QAP) instead of a neural-networks-based approach.

The proposed method is detailed in Section II. Preliminary experiments illustrating the potential of this combination are presented in Section III. Section IV concludes the paper with a discussion.

II. COMBINING DEEP LEARNING AND STRUCTURAL KNOWLEDGE

Figure 2 gives an overview of the proposed method, using, for illustration purposes, images considered in the experiments. Using an annotated training dataset, the deep neural network is trained to perform semantic segmentation. Moreover, using annotated images only, spatial relationships between the different regions are measured (e.g. average of measured distances), leading to the model graph $G_m$: vertices and edges correspond to annotated regions and spatial relationships, respectively.

When processing an unknown image, the neural network provides a segmentation proposal, from which a hypothesis graph $G_r$ is constructed (see $G_r$ in Figure 2). This hypothesis graph is then matched with the model graph (two steps inexact graph matching, as illustrated in Figure 2). The purpose is to match the vertices (and thus the underlying regions) produced by the neural network with those of the model, involving the relabelling of some regions (many-to-one matching in Figure 2). This produces a final semantic segmentation corresponding to the high-level relations observed in the training dataset.

We detail hereafter the step of construction of the hypothesis graph from the output of the deep neural network (Section II-A) and then its matching with the model graph (Section II-B).

A. Graph construction

The input image is processed by the neural network which produces, as output, a tensor $S \in \mathbb{R}^{I \times J \times C}$ with $I$ the width (in pixels) of the image, $J$ the height (in pixels) of the image and $C$ the total number of classes. At pixel location $(i, j)$, the value $S(i, j, c) \in [0, 1]$ is the probability of belonging to each class considered in the segmentation, with the constraints:

$$\forall c = 1, \ldots, C, \ 0 \leq S(i, j, c) \leq 1$$

subject to:

$$\sum_{c=1}^{C} S(i, j, c) = 1$$

The segmentation map $\mathcal{L}^*$ selects the label $c$ of the class with the highest probability:

$$\forall (i, j) \in \{1, \ldots, I\} \times \{1, \ldots, J\},$$

$$\mathcal{L}^*(i,j) = \arg \max_{c \in \{1, \ldots, C\}} S(i, j, c) \quad (1)$$

From this segmentation map, we define a set $R$ of all resulting connected components (see Figure 3, where $R = \{R_1, \cdots, R_4\}$). We also define a set $R^* = \{R_1^*, \cdots, R_C^*\}$, where, for each class $c \in \{1, \cdots, C\}$, $R_c^*$ is a set of regions corresponding to the connected components belonging to the class $c$ according to the neural network (see Figure 3, where
$R^* = \{R_1^*, \ldots, R_3^*\}$. This set $R^*$ is used for constraining the graph matching as described in Section II-B1.

From the set $R$, a structural representation is built and modeled by 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 interpreter and $D$ an edge interpreter. Each vertex $v \in V_r$ is associated to 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 $p = (i, j)$ composing $R_v$, therefore computed on the initial $S$ tensor (see Figure 3):

$$\forall v \in V_r, c \in \{1, \ldots, C\}, A(v)[c] = \frac{1}{|R_v|} \sum_{(i,j) \in R_v} S(i,j,c)$$

(2)

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 $R_j$ (see Figure 3). In our experiments, we choose the minimum distance between the two regions:

$$\forall e = (i, j) \in E_r, D(e) = \min_{p \in R_i, q \in R_j} |p - q|$$

(3)

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

### B. Matching with the model graph

In order to identify the regions, the purpose is to associate each of the vertices of $G_r$ to a vertex of the model graph $G_m$. According to the realistic assumption of having more regions in the image associated with $G_r$ than in the model (i.e. $|V_r| \geq |V_m|$), we face a problem of inexact graph matching, namely many-to-one matching [11]. We propose to formulate this problem in the general form of a quadratic assignment problem (QAP), as recently considered in [17].

In our case, the concept of matching is represented by a matrix $X \in \{0, 1\}^{|V_r| \times |V_m|}$, where $X_{ij} = 1$ means that vertex $i \in V_r$ is matched with vertex $j \in V_m$. This is illustrated in Figure 4-left in two cases (“one-to-one” and “many-to-one” matchings). The goal is to determine the best matching ($X^*$), minimizing the following cost:

$$X^* = \arg \min_{X} \{ \text{vec}(X)^T K \text{vec}(X) \}$$

(4)

where vec($X$) is the column vector representation of $X$ and $T$ denotes the transposition operator. The $K$ matrix, not detailed here for the sake of brevity (see [17] for details), embeds the dissimilarity measures between the two graphs $G_r$ and $G_m$, at vertices (diagonal elements) and edges (non-diagonal elements):

$$K = \alpha K_e + (1 - \alpha) - \frac{K_e}{\max K_e}$$

(5)

where $K_e$ embeds dissimilarities between vertices (Euclidian distance between class membership probability vectors). In the example in Figure 4-right, $K_e[1,1] = 0.4$ (row and column named 2a) represents the dissimilarity, in terms of probability vectors, between vertices 2 of $G_r$ and a of $G_m$, if one would match these two vertices. The matrix $K_e$ is related to dissimilarities between edges. For instance, in Figure 4-right, $K_e[6,1] = 9$ (row and column respectively named 3b and 2a) corresponds to the dissimilarity between the edges $(2,3) \in E_r$ (scalar attribute whose value is 16) and $(a,b) \in E_m$ (scalar attribute whose value is 7), if we would simultaneously match vertex 2 with vertex $a$ and vertex 3 with vertex $b$. In such a case, $K_e[6,1]$ is computed using edge

---

Fig. 3. Graph construction from the tensor $S$ (output of the U-Net) and the resulting $L^*$ entity. Each point of the left image is associated to the probability vector represented by a blurring effect. $R^*_i$ is the set of regions (regions $R_1$ and $R_2$) that belong to class 1 (according to probabilities). Edge attributes of the graph are computed from spatial relationships between regions $R_i$. Vertex attributes are mean probability vectors, computed over related regions $R_i$.

Fig. 4. Graph matching formulated as a quadratic assignment problem (illustration inspired by [17]). Left: our proposal consists of two steps, where the $X$ matrix models a matching between $G_r$ and $G_m$ graphs. The first step (top-left) focuses on a one-to-one matching (each vertex of $V_m$ is associated with only one vertex of $V_r$). The second step (bottom-left) aims at matching remaining vertices of $G_r$, leading to the final matching (many-to-one matching). Right: For finding the optimal one-to-one matching, a $K$ matrix is used, combining both $K_e$ and $K_v$ matrices, which respectively measure dissimilarities between vertices and edges. For the sake of clarity, only two edge attributes are reported.
attributes: \( K_r[6,1] = 16 - 7 = 9 \). \( K_r \) terms are related to distances between regions (normalized in the final \( K \) matrix).

The \( \alpha \) parameter (\( \alpha \in [0,1] \)) allows weighting the relative contribution of vertex and edge dissimilarities (\( K_r \) terms range between 0 and 1, and \( K_r \) is normalized in Equation 5).

Due to the combinatorial nature of this optimization problem [17] (i.e. set of possible \( X \) candidates in Equation 4), we propose a two-steps procedure, relying on the initial semantic segmentation provided by the neural network:

1) Search for an initial one-to-one matching (Figure 4-top-left).
2) Refinement by matching remaining vertices, finally leading to a many-to-one matching (Figure 4-bottom-left).

**1) Initial matching: one-to-one:** One searches for the optimal solution to Equation 4 by imposing the following three constraints on \( X \), thus reducing the search space for eligible candidates:

\[
\sum_{i=1}^{V_m} X_{ij} \leq 1: \text{some } i \text{ vertices of } G_r \text{ may not be matched.}
\]

\[
\sum_{j=1}^{V_m} X_{ij} = 1: \text{each } j \text{ vertex of } G_m \text{ must be matched with only one vertex of } G_r.
\]

\[
X_{ij} = 1 \Rightarrow R_i \in R_j^r: \text{vertex } i \in V_r \text{ can be matched with vertex } j \in V_m \text{ if the associated } R_i \text{ region was initially considered by the neural network to most likely belong to class } j (i.e. R_i \in R_j^r). \text{ For instance, in the case of Figure 3, only vertices related to regions } R_1 \text{ and } R_2 \text{ would be considered as candidates for class 1} (R_1^r).
\]

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 \( G_m \)).

2) Refinement: many-to-one: We assign each remaining vertex \( k \) (i.e. \( k \in V_r \mid \sum_{j=1}^{V_m} X_{kj} = 0 \)) to the vertex \( i^* \in V_r \) by considering the following cost function between two vertices \( i \) and \( j \) of \( G_r \):

\[
\text{cost}(i,j) = \alpha |A(j) - A(i)| + (1-\alpha) \frac{D(j) - D(i)}{\max_{a \in E} D(a)}. \quad (6)
\]

For each remaining vertex \( k \), the best matching vertex \( i^* \), among already matched vertices with \( V_m \), minimizes the cost function:

\[
i^* = \arg\min_{i \in V_r \mid \sum_{j=1}^{V_m} X_{ij} = 1} \text{cost}(i,k). \quad (7)
\]

According to this formulation, it appears that remaining vertices are matched to vertices of \( G_m \) by indirectly searching for correspondences with already matched vertices of \( G_r \). Therefore, one focuses on similarities within the current image and not with the model. In Figure 4-bottom-left, this corresponds to finding the matching between vertex 1 and vertex \( b \) by indirectly studying the relevance of the matching of vertex 1 with 3 (vertex 3 being already matched with vertex \( b \)).

The formulation related to Equation 7 and concerning one vertex only is similar to the matrix formulation related to Equations 4 and 5 (and concerning simultaneously several vertices). The only difference is that we consider vertex and edge dissimilarities within \( G_r \) graph instead of considering dissimilarities between \( G_r \) and \( G_m \).

### III. Experiments

We present, hereafter, the dataset used in the experiments, then the evaluation protocol, and finally the results.

**A. Data**

We consider the FASSEG\(^1\) public data set. This dataset focuses on the multi-class semantic segmentation of the face [12] (see Figure 1) as well as the estimation of its pose [13]. For this preliminary study, we consider a subset of this dataset corresponding to a specific pose (the front view).

This subset contains 70 images. FASSEG is a dataset of images with semantic segmentation rich in structural information that does not however distinguish certain regions (i.e. left eye and right eye, left eyebrow and right eyebrow). We have therefore refined the annotations in order to give a unique label to these regions. Note that we have also refined some semantic label maps for the creation of our model graph (see Figure 1).

**B. Evaluation protocol**

For these experiments, we consider the U-Net neural network [3] that copes well with a training set with a small number of samples. For these experiments, we split our dataset as follows: 20 images are used for training (reference training set), 10 for the validation (reference validation set), and 40 for the test (reference test set). 100 epochs are used for training the network.

The model graph is constructed by calculating the average distances (Equation 3) between the different annotated regions of the training set. The \( \alpha \) parameter is empirically selected, based on observations on some images, and set to 0.4 for experiments, yet without any optimization.

As this was experimentally investigated, a simple distance measurement between the centers of gravity of the regions appeared inappropriate, due to the variability of the shape of some regions (e.g. hair).

We evaluate the difference between the quality of semantic segmentation at the output of neural networks and that after matching, i.e., with integration of structural information. We consider the Intersection over Union (IoU or Jaccard index) to evaluate the quality of our results against our manual annotation used as ground truth. This evaluation measure is used to compare regions at the pixel level and also at the bounding box level. This comparison of bounding boxes allows us to quantify segmentation errors corresponding to a correct main region but with errors related to one or more minor sub-regions far from the main region (and not very significant in terms of number of pixels). These measurements

---

\(^1\)The FASSEG annotated public dataset can be downloaded at the following address: https://github.com/massimomauro/FASSEG-dataset.
TABLE I
COMPARISON OF SEGMENTATIONS PROVIDED BY THE U-Net AND OUR APPROACH ADDING GRAPH MATCHING (U-Net+GM) CONSIDERING THE IOU INDEX RELATED TO REGION BOUNDING BOXES AND PIXELWISE PRECISION COMPARED TO THE MANUAL SEGMENTATION. RESULTS ARE PROVIDED AS AVERAGE AND FOR EACH CLASS: BG (BACKGROUND), HR (HAIR), FC (FACE), L-br (LEFT EYEBROW), R-br (RIGHT EYEBROW), L-eye (LEFT EYE), R-eye (RIGHT EYE), NOSE AND MOUTH. RESULTS ARE ALSO PROVIDED FOR DIFFERENT SIZES OF THE TRAINING/VALIDATION SETS.

<table>
<thead>
<tr>
<th>Training dataset (%)</th>
<th>Approach</th>
<th>Pixelwise Clases</th>
<th>Bounding box Clases</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>Average</td>
<td>Rg</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>100</td>
<td>U-Net</td>
<td>75.3</td>
<td>88.0</td>
</tr>
<tr>
<td></td>
<td>U-Net + GM</td>
<td>75.4</td>
<td>88.0</td>
</tr>
<tr>
<td>75</td>
<td>U-Net</td>
<td>74.0</td>
<td>88.5</td>
</tr>
<tr>
<td></td>
<td>U-Net + GM</td>
<td>74.3</td>
<td>88.3</td>
</tr>
<tr>
<td>50</td>
<td>U-Net</td>
<td>73.7</td>
<td>86.7</td>
</tr>
<tr>
<td></td>
<td>U-Net + GM</td>
<td>73.0</td>
<td>86.4</td>
</tr>
<tr>
<td>25</td>
<td>U-Net</td>
<td>39.8</td>
<td>83.8</td>
</tr>
<tr>
<td></td>
<td>U-Net + GM</td>
<td>38.2</td>
<td>84.5</td>
</tr>
</tbody>
</table>

Table I also reports the influence of reducing the training dataset. In such cases, graph matching fails because of missing class candidates, this case being not yet managed by our approach. For this reason, results reported in Table I for 25% ignore such cases, and reported performances are only averaged on segmented test images providing at least one region per class (48.9% of images are not considered).

C. Quantitative Segmentation Results

Overall, the semantic segmentation is improved thanks to graph matching as can be seen on average over all classes (training sets size between 50% to 100%) in Table I providing results in terms of pixelwise IoU index and IoU index of bounding boxes.

The degree of improvement is more significant when using IoU on region bounding boxes in comparison to pixel based IoU. This is due to the fact that, for a given class, misclassified pixels represent a small proportion of the ground truth area (benefits appear higher for small areas such as eyes and eyebrow). With IoU based on bounding boxes, one favors the analysis of the spatial distribution of misclassified areas.

In this case, improvement appears particularly significant: 24.4% of improvement for the class right eyebrow (R-Br) with 50% training set size (Table I). The use of spatial relationships allows avoiding misclassified regions that are spatially incoherent (e.g. piece of hair between the nose and the mouth).

Table I also reports the influence of reducing the training data set. It appears that the smaller the dataset is, the higher are the benefits of using graph matching. Our modeling capturing spatial relationships combined with graph matching allows then improving segmentation results, compensating for the lack of representativity of a small training dataset.

Note that with only 25% of the training dataset, depending on randomly selected images, some trained networks appeared unable to propose region candidates for some classes on some test images. This is due to the poor representativity in the training dataset. In such cases, graph matching fails because of missing class candidates, this case being not yet managed by our approach. For this reason, results reported in Table I for 25% ignore such cases, and reported performances are only averaged on segmented test images providing at least one region per class (48.9% of images are not considered).

D. Qualitative Segmentation Results

Figure 5 gives some examples of semantic segmentations with both U-Net and our approach. As visually observed, improvements are significant in many cases (e.g. part of the eye on the middle image, neck), in particular when reducing the size of the training dataset (i.e. partial compensation of an under-trained model).

IV. DISCUSSION AND PERSPECTIVES

The proposed approach appears efficient to improve the semantic segmentation achieved by the U-Net deep neural network. The improvement appears particularly significant.
for some classes (up to 24.4% for the Right-eyebrow label, compared to the average improvement of 8.5% considering the bounding box IoU with a training percentage of 50%). This is due to the nature of the considered information, which appears highly complementary to the low-level (pixel-level) information considered by the U-Net that intrinsically ignores high-level spatial relationships. As observed, an additional strength of our approach is its ability to compensate for segmentation errors resulting from the reduction of the training database. This is an important aspect as the size of the learning database in deep learning is a strong constraint, as underlined in the introduction.

While promising, our approach suffers from a few limitations. First, our approach is invariant in translation and rotation, but not yet in scale. This could be managed by introducing a scale factor as part of the matching process (in particular regarding the computation of the $K$ matrix reported in Equation 4). Moreover, several parameters (e.g. $\alpha$) need to be set empirically.

Secondly, partial occlusions in images that could affect the computation of graph $G_r$ are cases not yet considered in our current formulation. Indeed, each vertex (region) is assumed to be necessarily matched to a region of the model. Handling occlusions could be managed by relaxing hypotheses pertaining to matching the vertex of the model.

Thirdly, our modeling may not to be robust to face pose changes that alter spatial relationships. This issue can be addressed by choosing a more generic and representative model. Note that other domains of applications for our technique such as 3D medical images segmentation may not present the same challenges in practice as face segmentation presented here as illustration for our technique. To support high variation in spatial relationships, an alternative could be to consider a set of representative graph models instead of only one, with the underlying difficulty of selecting the appropriate one when segmenting an image. Our future work will also investigate how deep learning on graphs [18] can help in improving our approach.

Finally, small computation time may be crucial in some applications. In particular, the high complexity of the first step (formulated as a quadratic assignment problem) may involve a dramatic increase of the computation time, if the number of classes and regions grows. Nevertheless, raw estimates provided in the paper are encouraging and further improvements could be sought using dedicated hardware (e.g. GPU).

V. Conclusion

In this paper, we have proposed an original method combining spatial relationships with deep learning for semantic segmentation. Preliminary results show how this approach improves significantly segmentation, with the additional benefit of using a limited number of training samples. The best obtained improvement over the U-Net neural network used alone for segmentation is of 8.5% IoU on average, trained with only 50% of the available training dataset.

Future work will focus on the scale invariance, a finer evaluation of the computational complexity and on experiments in other application domains. Another aspect to be studied is the ability to manage graph matching using neural networks (i.e. neural networks on graphs and not only on images [14], [16]).

REFERENCES