Segment My Object: A pipeline to extract segmented objects in images based on labels or bounding boxes

Robin Deléarde, Camille Kurtz, Philippe Dejean, Laurent Wendling

Segment my object: A pipeline to extract segmented objects in images based on labels or bounding boxes

Robin Deléarde[1,2][a], Camille Kurtz[1][b], Philippe Dejean[2], and Laurent Wendling[1][c]

[1]*LIPADE, Université de Paris, France*
[2]*Magellium, Artal Group, Toulouse, France*
{firstname.lastname}@{u-paris|magellium}.fr

Abstract: We propose a pipeline (SegMyO – Segment my object) to automatically extract segmented objects in images based on given labels and / or bounding boxes. When providing the expected label, our system looks for the closest label in the list of outputs, using a measure of semantic similarity. And when providing the bounding box, it looks for the output object with the best coverage, based on several geometric criteria. Associated with a semantic segmentation model trained on a similar dataset, or a good region proposal algorithm, this pipeline provides a simple solution to segment efficiently a dataset without requiring specific training, but also to the problem of weakly-supervised segmentation. This is particularly useful to segment public datasets available with weak object annotations (*e.g.*, bounding boxes and labels from a detection, labels from a caption) coming from an algorithm or from manual annotation. An experimental study conducted on the PASCAL VOC 2012 dataset shows that these simple criteria embedded in SegMyO allow to select the proposal with the best IoU score in most cases, and so to get the best of the pre-segmentation.

1 INTRODUCTION

Segmentation is a challenging task in image analysis for decades, from image processing with solutions based on contours detection and pixel regions, to machine learning with (deep) models trained on large sets of annotated images. In fact, segmented images are useful for many high-level computer vision tasks, since they provide a precise delineation of the objects appearing in the scene, or of the structural parts of an object. This may further allow to compute relevant descriptors for these objects or parts, describing their texture, shape, pose or even spatial configuration.

Originally, *image segmentation* was the task of partitioning the image content into homogeneous regions, by assigning a region to each pixel. Now it is often used as *semantic* segmentation (also known as "image parsing"), aiming to find the best label for the region or the pixel. The outputs are called "segments" and are made of a label and a set of pixels of the image, which can be represented as a "segmentation mask". It differs from classical segmentation

and region proposal where the output is only a set of regions. A variant is instance segmentation, where different instances of the same class are expected to be output into different segments.

In parallel, *object segmentation* is the task of extracting an object from its background. It can be seen as the specific case of an image containing only one object and background, and the same methods can be used. But specific approaches also exist for this task, assuming for example that there is only one object in the image. Other object priors can be exploited, like its position (*e.g.*, at the center), extent (*e.g.*, its bounding box), etc. Solutions relying on bounding box and labels are detailed in the following.

Semantic segmentation appeared with the rise of (deep) machine learning, with CNN models like R-CNN or FCN. It relies on supervision with annotations at the pixel level, which is really tedious to obtain on large datasets. Thus, several datasets have been made available, like the well-known PASCAL VOC or COCO, but they are still limited to some specific applications. Beside that, "collecting bounding boxes around each object in the image is 15 times faster than labeling images at the pixel level" (Lin et al., 2014). In this context, weakly-supervised solutions appeared as a cheap alternative, using only

---

[a] https://orcid.org/0000-0001-6628-7778
[b] https://orcid.org/0000-0001-9254-7537
[c] https://orcid.org/0000-0003-1091-5995

weak annotations like bounding boxes or image captions for the training. And when they are available at the inference step, these annotations can be used to help the segmentation, what we call "segmentation with test clues". This results in tasks with variable difficulty and specific solutions. Finally, it can be noticed that segmentation is closely related to region proposal, which consists in extracting coherent regions in the image.

Benefiting from the development of huge panoptic datasets, it is now possible to find numerous pre-trained segmentation models and to use them directly on new data with good performance, for many applications. The output can be a segmented image with different levels, or several candidate masks with possible overlap. In both cases, it might be interesting to automatically extract a particular object among all the output segments, given some clues on it, so as to integrate this step into a larger process. This is the problem we consider here. Weakly-supervised segmentation is a natural extension of this problem, since a simple solution consists in using segment proposal and selection to generate pixel-level supervision from weakly-annotated images.

In this article, we propose complementary criteria to extract a particular output among several proposals, based on its bounding box and / or its label. Concerning the bounding box, we use a mix of geometric criteria on the covering of the targeted bounding box by each proposal. And concerning the label, we use semantic information to find the closest to the researched one. We propose to use this selection in association with a segmentation model to generate the proposals, leading to a complete pipeline to segment datasets with weak object annotations. Moreover, it can be easily integrated as the first step of a weakly-supervised segmentation framework, so as to segment new data without annotation at inference step, using the weak annotations for the training.

Thus, the main contributions of this paper are:

- a proposition of geometric and semantic criteria to select an object in the image content, given its label and / or bounding box;
- an integration of these criteria into a pipeline (SegMyO – Segment my object) aiming to segment any dataset from weak annotations (labels and / or bounding boxes);
- an insight of how to use these criteria for weakly-supervised segmentation.

The article is organized as it follows. Section 2 reviews some related works. Section 3 presents our methodological contribution. In Section 4, we propose an experimental study. Finally, conclusions and perspectives of this work are provided in Section 5.

2 RELATED WORKS

2.1 Segmentation transfer

Modern segmentation models, whatever their supervision level, are trained on representative data, making them data/task-dependent, with variable generalization capacity. Using such a model for another dataset or another task is possible, but it requires some adaptations to make it efficient, by exploiting the knowledge available on the new test case. This general solution called "transfer learning" is known as "domain adaptation" when the task is modified, and "fine-tuning" when the model is just adapted to new classes (by modifying the last layer) and re-trained on new data. Fine-tuning is much used for object detection and recognition, but much less for semantic segmentation, which requires dense annotations.

In this context, (Hong et al., 2016) proposes to use transfer learning for weakly-supervised semantic segmentation, by using a encoder-decoder with a visual attention model to transfer knowledge from categories with strong annotations to unseen categories with weak annotations. (Sun et al., 2019) proposes a solution to benefit both from real and synthetic data, by using jointly with the segmentation network a second network dedicated to learn the similarity of synthetic pixels to real ones. Furthermore, (Pascal et al., 2019) proposes to use semantic similarity, by re-using the last layer of the original network when the labels are semantically close. It is only evaluated on a classification task, but the idea can be used for segmentation too. We consider this approach here, but only as a post-processing since we do not have access to pixel-level annotations to train a new model.

2.2 Segmentation with test clues

We do not use the expression of "weakly-supervised segmentation" for this task because it usually refers to the training step only, with solutions based on weakly-supervised learning, while here we consider that weak annotations are available at the inference step. Actually this is often the first step of weakly-supervised segmentation solutions, which exploit weak annotations to generate a pixel-level ground truth that is used to train a common segmentation model, in the same way as self-supervised methods.

Two main categories of approaches co-exist: those exploiting the bounding box of the object, and those exploiting image-level labels. Typically, these clues come from a first detection or image captioning step, or from manual annotations. Few solutions exploit a combination of image-level labels and bound-

ing boxes: (Papandreou et al., 2015) can use both, in function of what is available, while (Li et al., 2018) uses bounding-boxes for objects of interest and image-level labels for the background (or "stuff" objects). But to our knowledge and surprisingly, no solution exploits the combination of a bounding-box directly with the label of the considered object.

Object segmentation with a bounding box: Knowing the object bounding box is obviously an interesting clue to infer its real extent at the pixel level, even more if it is tight around the object, *i.e.*, if it contains the entire object and no more other stuff than necessary. The most simple case is object / background segmentation, where the object is alone in the image or the bounding box. But in more difficult cases, even a tight bounding box may contain several objects instead of simple background, and choosing the best one is tricky.

While former solutions used to rely on higher user interaction, like selecting points or scribbles inside and outside the object, GrabCut (Rother et al., 2004) was the first solution to rely only on a bounding box. It is based on two appearance models, for the background and the foreground, and on iterative graph-cut to get the final segmentation. Its efficiency made it popular, and it is still used in several state-of-the-art solutions, with some variations like (Lempitsky et al., 2009) that exploits the tightness prior by adding constraints into the global energy minimization.

Another common solution is to generate multiple hypothesis segments, either with an over-segmentation model or region proposal, and to select or generate the final output with a voting or combination scheme. In the literature, most solutions are based on region proposal, especially in weakly-supervised frameworks where it is a first step used to generate pixel-level ground truth. For instance, Box-Sup (Dai et al., 2015) is based on Multiscale Combinatorial Grouping (MCG) (Arbeláez et al., 2014) and evaluates several other solutions. (Khoreva et al., 2017) is based on MCG too, but it combines it with GrabCut, and allows to handle several instances of the same object. As for over-segmentation, it is used in (Chen et al., 2012) solution for object / background segmentation, with an adaptive Mean-Shift algorithm. In our solution, we suggest to use an over-segmentation made of a full segmentation model trained on a similar dataset, using for instance the efficient Mask R-CNN (He et al., 2017).

Other tracks were also explored, particularly for weakly-supervised segmentation. (Papandreou et al., 2015) constrains its CRF to consider the center area of the bounding box (a percentage of pixels within the box) as foreground, and the pixels outside the bounding box as background. As for it, (Hsu et al., 2019) uses a Multiple Instance Learning (MIL) framework, and integrates the tightness prior on the bounding box. More recently, BB-UNet (Jurdi et al., 2020), which is based on U-Net, exploits shape priors by introducing a novel convolution layer, to segment medical images.

Image segmentation with image-level labels: Many solutions exploiting image-level labels exist in the literature, particularly for weakly-supervised segmentation (Kolesnikov and Lampert, 2016; Zhou et al., 2018; Ahn et al., 2019; Wang et al., 2020). These solutions are usually based on class-activation maps (CAM) (Zhou et al., 2016) and visual attention to generate the pixel-level ground-truth.

The approach of (Guillaumin et al., 2014) can be mentioned here too, since its goal is to segment a complete dataset containing weak annotations like us (ImageNet in their case), but with a totally different idea. It is based on an original greedy approach, by progressively segment the objects whose labels are semantically close to the already segmented ones, using two appearance models for foreground and background pixels. The model is initialized with PASCAL VOC pixel-level annotations, and also exploits bounding boxes annotations when they are available.

3 SEGMYO PIPELINE

3.1 Workflow description

We propose a solution to the task of object segmentation based on a bounding box and / or label test clue, on an image containing several objects and not only "stuff" background. It relies on a two-stage pipeline called SegMyO (Segment My Object): first a set of regions is extracted with a classical (instance) image segmentation solution, or with region proposal, and then the best output is selected for the given bounding box, and label if using semantic segmentation, according to the criteria described in Section 3.2.

Concerning the segmentation step, using a pre-trained model supposes that the latter was trained on objects and labels similar to the target ones. Yet, there now exists good segmentation models pre-trained on huge and varied datasets, for many applications. Moreover, so as to extend the training to other similar labels, we propose to use semantic similarity to find a similar label in the model ones. This can also be seen as a transfer of a model trained on a set of labels to another set not containing exactly the same labels, as it is done in (Pascal et al., 2019).
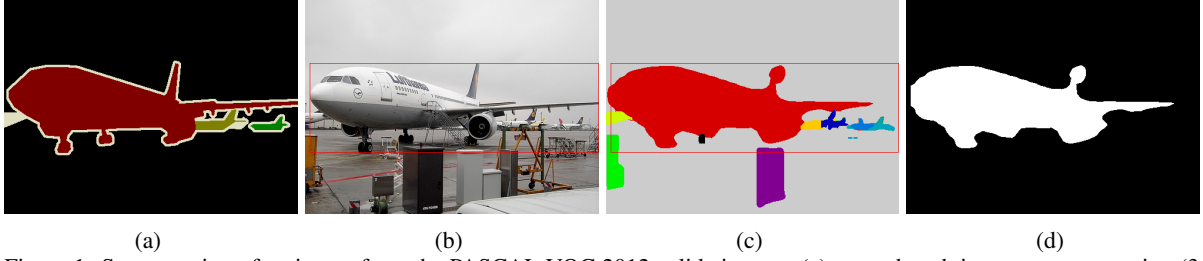
| (a) | (b) | (c) | (d) |

Figure 1: Segmentation of an image from the PASCAL VOC 2012 validation set: (a) ground truth instance segmentation (3 instances of the class "aeroplane"), (b) input image and bounding box, given with the label "aeroplane", (c) output masks from Mask R-CNN, restricted to those overlapping to the bounding box (12 objects), (d) mask selected by SegMyO for the given bounding box and label (recognized as "airplane").

Our system takes as input:

- an image I of dimension $W \times H$, with W the width and H the height of the image;

- a bounding box B around a specific object, defined by the spatial coordinates of its upper left corner (x_1, y_1) and bottom right corner (x_2, y_2), with (x_1, y_1) and $(x_2, y_2) \in [1, W] \times [1, H]$;

- a semantic label $l \in L^{target}$ of an object appearing in the image (*e.g.*, $L^{target} = \{\text{Man}, \text{Animal}, \dots\}$);

- a segmentation of I made of a set of regions $\{R_i\}_{i=1..N}$ possibly overlapping, detected with a confidence score r_i, each region potentially being endowed with a semantic label $l_i \in L^{init}$ (*e.g.*, $L^{init} = \{\text{Person}, \text{Automobile}, \dots\}$).

In the sequel, we note $R_{i,B} = R_i \cap B$ the restriction of the region R_i to the bounding box B, as it is illustrated on Figure 2.

For each candidate region R_i, a score is computed from its covering of the bounding box, and from the semantic similarity between the expected label l and the predicted label l_i if using semantic segmentation, thanks to the criteria defined in Section 3.2. Then the system outputs the region with the best score among all the candidates as the correct segmentation of the object, as shown on Figure 1. Another threshold can be used here to filter low scores, where it may be better to use another solution like GrabCut or just filling the bounding box (or a part of it).

This pipeline can be exploited for two other main use cases:

- in full image segmentation aided by bounding box and / or labels priors (with labels either for each bounding box or at image-level), by considering each object separately. This supposes to have access to those weak annotations for all objects of interest in the image, and requires to deal with possible overlap between objects.

- as the first step of a weakly-supervised segmentation method, by using the selected region to train a supervised segmentation model, as it is done in

state-of-the-art solutions. This approach is similar to BoxSup (Dai et al., 2015), that uses region proposal to extract ground truth masks and selects the best one according to a similar criterion in its objective function (see criterion C3, Section 3.2).

An implementation of the proposed pipeline is available at: https://github.com/RobinDelearde/SegMyO.

3.2 Selection criteria

Here different criteria are proposed to automatically select the best candidate segment: geometric criteria are dedicated to the bounding box clue, while a semantic criterion is dedicated to the label. All criteria are in the range $[0, 1]$.

3.2.1 Bounding box geometric criteria

First, two criteria aiming at exploiting the tightness prior on the bounding box are considered, as they are basic necessary and universal criteria. They rely on the hypothesis that the bounding box is tight around the object, *i.e.*, that it contains all the object and no more other stuff (*e.g.*, background) than necessary. Thus, the tighter the bounding box is, the better the segmentation will be. This tightness prior is used in (Lempitsky et al., 2009; Hsu et al., 2019) and can also be derived from the background and object extent clues of (Khoreva et al., 2017).

The measures that we propose are based on:

C1. the maximum relative distance of the region to the edges of the bounding box, as it should be close to zero for all 4 edges:

$$c_1(R_i, B) = 1 - \max\left[\frac{d_x(R_{i,B}, B)}{W}, \frac{d_y(R_{i,B}, B)}{H}\right]$$

C2. the relative extent of the region, *i.e.*, the part of the region that is in the bounding box, assuming that all the object must be in the bounding box:

$$c_2(R_i, B) = \frac{area(R_{i,B})}{area(R_i)}$$

Other optional criteria can be defined to refine the segmentation step, depending on the a priori knowledge about the shape of the objects, like:

C3. the area covering of the bounding box, which intends to give more weight to large objects. This is the criterion used in BoxSup (Dai et al., 2015):

$$c_3(R_i, B) = \frac{area(R_{i,B})}{area(B)}$$

C4. the presence at the center of the bounding box, for "barycentric" objects or to avoid objects present only on the edges;

C5. the covering of all x and y of the bounding box, to avoid disjoint objects.
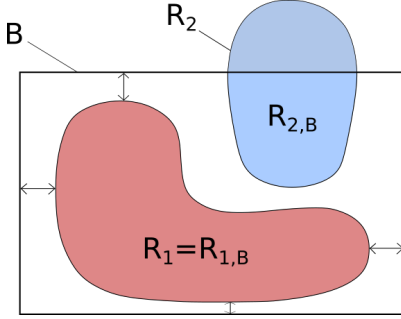

Figure 2: Two candidate regions in a same bounding box.

3.2.2 Semantic criterion

When using semantic segmentation, each segment is assigned a label from the set of labels learned by the model. So knowing the label of the item we look for is useful, but it may not be part of this set. In fact, a major problem lies in the specificity and the low number of classes in most of the academic standard datasets published on the Internet. The segmentation models pre-trained on these datasets are thus limited to the segmentation of existing object classes in these datasets. For example, if the "cat" class is learned in the initial dataset, it will be difficult to provide as an input label "bobcat" in the target dataset. However the "cat" segmentation model might be used to segment a "bobcat" since these objects are visually close and semantically related. This is why we propose to use semantic label matching to extend the vocabulary to labels outside the set.

To do so, we use natural language processing tools to compute a semantic similarity c_{sem} between the predicted label l_i and the expected label l, as:

$$c_{sem}(l_i, l) = semantic\_similarity(l_i, l)$$

Two main approaches can be considered: similarities between words in a taxonomy like WordNet, or similarities between words embeddings learned on a corpus, with a model like Word2Vec. However, this requires that both labels are present in the vocabulary, so it is necessary to take care of that when choosing the model, or to perform manual corrections.

Several methods are available to compute the semantic similarity in a taxonomy, like Rada, Resnik or Li similarities, etc. Most of them are based either on structural measures between concepts in the taxonomy (*e.g.*, path length and depth), or on the Information Content (IC). We suggest to use the *wpath* similarity (Zhu and Iglesias, 2016), which combines both approaches by using IC to weight the shortest path length between concepts, and usually demonstrates the best performance. This measure is defined as:

$$c_{sem,wpath}(l_i, l) = \frac{1}{1 + length(l_i, l) * k^{IC(lcs(l_i, l))}}$$

with lcs the least common subsumer, and k a parameter indicating the contribution of the lcs's IC.

3.3 Criteria combination

A global criterion can be computed from all the previous criteria and from the segmentation confidence score r_i. It can be made simply with a weighted sum of the different scores. As mentioned before, the criteria C1 and C2 are particularly important, so we suggest to give a more important weight to them, whereas the criteria C3, C4 and C5 are more questionable, so they should have a smaller weight unless we have some prior knowledge about the shape of objects in the bounding boxes. The semantic criterion is also particularly significant, so we suggest to give an important weight to it also.

In our experiments, we used the following score:

$$score_1 = \begin{cases} (r_i + 2*c_1 + 2*c_2 + c_3 + 4*c_{sem})/10 \\ \quad \text{when } l_i \in L^{init} \text{ and } l \in L^{target} \\ (r_i + 2*c_1 + 2*c_2 + c_3)/6 \\ \quad \text{otherwise} \end{cases}$$

To go even further, the criteria C1 and C2 are necessary conditions, so we suggest to keep only the minimal value of C1, C2 and the global score, with:

$$score_2 = min(c_1, c_2, score_1)$$

Another solution would be to learn the weights to get the best fusion of the different criteria. However we experimentally found that the proposed combination was already satisfactory (see Section 4).

This is worth mentioning that all these criteria might be impacted if several instances of the object

are present in the image and the segmentation is not able to separate them. So they should be used only with instance segmentation, or with datasets including only one instance of each object. It is also interesting to note that these criteria can be computed on the masks with "fuzzy" raw scores (in $[0,1]$) rather than on binary masks, so as to take into account the score for each pixel in the computation. This is what we did for the criteria C2 and C3 in our experiments.

4 EXPERIMENTS

4.1 Data and experimental protocol

We evaluate our pipeline on the PASCAL VOC 2012 segmentation dataset (Everingham et al., 2010), using the validation set. It is made of 1 449 images, containing 3 427 objects from 20 classes in the segmentation masks. As a pre-segmentation model to generate the candidate regions, we used Mask R-CNN ResNet-50 FPN *torchvision* model[1] (He et al., 2017), which was trained on COCO 2017 (Lin et al., 2014). This other dataset is made of 80 classes, including the 20 classes from PASCAL VOC, but sometimes with a different name, like "couch" in COCO which is "sofa" in PASCAL VOC, or "tv" vs. "tv monitor".

The output of the pre-segmentation is a list of objects proposals made of a label, a segmentation score and a mask with values in $[0,1]$. We only keep the proposals with a segmentation score above a threshold (set to 0.25). We take this list as input of our selection process, and return the best proposal for each object, given its bounding box and its label, according to our criteria. We also test a region proposal solution with Multiscale Combinatorial Grouping (MCG) (Arbeláez et al., 2014) to see if it can be efficient, by using the pre-computed proposals available online[2].

We evaluate each criterion individually and the two combined scores to compare their performance. For the semantic similarity, we used WordNet taxonomy and *wpath* method (Zhu and Iglesias, 2016), with *sematch* python framework[3]. As a segmentation evaluation metric, we compute the intersection-over-union (IoU) score for each object by using the instance segmentation annotations provided for the validation set, and after a binarization of the image (we used a threshold of 0.3). We repeat this process for each object with such annotations, *i.e.*, for the 3 427 objects of this dataset, and report the mean over all objects as $mIoU_{obj}$ in Table 1.

---

[1] https://pytorch.org/docs/stable/torchvision/models.html

[2] https://www2.eecs.berkeley.edu/Research/Projects/CS/vision/grouping/mcg/

[3] https://github.com/gsi-upm/sematch

So as to compare to other methods (weakly-supervised or fully-supervised), we also compute the mean IoU for each class of objects, after transforming our outputs to conform to image segmentation. To do so, we added a step of fusion of the individual selected segments, by inserting them in the output image from the largest to the smallest, so as to deal with overlapping segments. We report as *mIoU* the mean over all the classes, including the background class. We also compare to GrabCut (Rother et al., 2004) as a state-of-the-art unsupervised solution for our task.

4.2 Results and discussion

The results of our experiments on segmentation with test clues are reported in Table 1, for several solutions and various selection criteria, while Table 2 provides comparisons with some recent weakly and fully supervised models from the literature. For each solution, we indicate either the level of the test clues available at the inference step in the first case, or the level of annotations used to train the model in the second case. It can be "pixels" when pixel-level annotations are used, "caption" when image-level labels are considered, and "b.box" or "b.box+labels" for bounding box level annotations.

In our tests (Table 1), we distinguish between totally unsupervised solutions, *i.e.*, not using any data to elaborate the model, and solutions not using PASCAL VOC train data but another similar dataset, here COCO, in a transfer learning manner. As a low baseline for the bounding box test clues, we use the solution to fill this one, totally or partially. The scores reported (solution "filled b.box") are obtained by filling the bounding box at 90%, which gave a better score than 80% or 100%. Concerning solutions based on our pipeline with a pre-segmentation and output selection, an upper bound baseline can also be obtained for the output selection by using the IoU as criterion instead of ours, which supposes to have access to pixel-level annotations. Thus, this baseline allows to get the best score for the considered pre-segmentation model (solution $SegMyO_{mIoU}$).

These results show that totally unsupervised segmentation with test clues does not give satisfactory results, and that it cannot segment efficiently a dataset despite the bounding box information. On the contrary, using a model pre-trained on another dataset allows to reach a good performance, ranking between weakly-supervised and fully-supervised models, without training on the considered dataset. Then it is possible to segment a dataset without a specific training, by benefiting from pre-trained models and test clues, with an appropriate use of these clues.

method	test clues	mIoU	mIoU$_{obj}$
unsupervised			
filled b.box	b.box	56.50%	**53.16%**
GrabCut (Rother et al., 2004)	b.box	**58.82%**	44.29%
MCG + SegMyO$_{C3}$	b.box	47.64%	44.39%
MCG + SegMyO$_{mIoU}$	pixels	51.08%	47.50%
transfer learning (from COCO)			
MR-CNN + SegMyO$_{mIoU}$	pixels	**74.68%**	**70.67%**
MR-CNN + SegMyO$_{C1}$	b.box	70.55%	65.04%
MR-CNN + SegMyO$_{C2}$	b.box	53.19%	51.95%
MR-CNN + SegMyO$_{C3}$	b.box	71.02%	**66.85%**
MR-CNN + SegMyO$_{sem}$	caption	68.71%	58.19%
MR-CNN + SegMyO$_{C3*sem}$	b.box+label	73.16%	68.63%
MR-CNN + SegMyO$_{score1}$	b.box	73.08%	68.97%
MR-CNN + SegMyO$_{score2}$	b.box	73.04%	68.74%
MR-CNN + SegMyO$_{score1}$	b.box+label	**73.62%**	**69.35%**
MR-CNN + SegMyO$_{score2}$	b.box+label	73.30%	68.93%
MR-CNN+ SegMyO$_{score1}$ / filled b.box	b.box+label	**73.99%**	**69.88%**

Table 1: Segmentation scores on PASCAL VOC validation set, when testing with test clues, for unsupervised models and a model trained on another dataset (Mask R-CNN trained on COCO) (mIoU: average over the 21 classes, mIoU$_{obj}$: average over all the objects w/o background).

method	train data	mIoU$_{val}$	mIoU$_{test}$
weakly-supervised			
BoxSup (Dai et al., 2015)	b.box	62.0%	64.2%
SEAM (Wang et al., 2020)	caption	64.5%	65.7%
fully-supervised without additional data			
ResNet-38 (Wu et al., 2019)	pixels	n.c.	82.5%
DeepLabv3+ (Chen et al., 2018)	pixels	81.63%	n.c.
fully-supervised with additional data (COCO)			
ResNet-38 (Wu et al., 2019)	pixels	80.84%	84.9%
DeepLabv3+ (Chen et al., 2018)	pixels	84.56%	89.0%

Table 2: Segmentation scores from the literature on PASCAL VOC validation / test sets, models trained with PASCAL VOC train data (mIoU averaged over the 21 classes).

Concerning the different proposed criteria (see Section 3.2), they reach variable performance when taken individually, with a good behaviour of the criteria C3 and C1, and in a lower extent of the semantic criterion (which shows an important gap between the two computation modes of the mIoU). Combining several criteria allows to get closer to the upper bound given by the mIoU, for instance by combining the semantic criterion and criterion C3.

For the combined criteria $score_1$ and $score_2$, we evaluated both with the bounding box only and with the bounding box + label. All results are quite similar, but the differences allow to conclude on the interest of each variation. Thus, we can infer that $score_2$ is not better than $score_1$ on average, contrary to expectations, showing also that the weights used in $score_1$ are quite adapted. More importantly, it can be noticed that the gain provided by the label is not very high, so that it is much better to annotate datasets with bounding boxes only than with labels only. Finally, a last solution is given by using $score_1$ and filling the bounding box (at 90%) when the computed score is below a threshold (set to 0.5), which reaches the best performance, really close to the mIoU upper bound.

sail, water *street, bus* *chair, table*

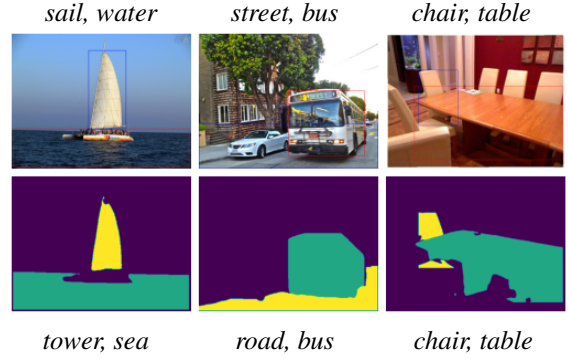*tower, sea* *road, bus* *chair, table*

Figure 3: Samples of segmentation outputs obtained on SpatialSense, with HRNet trained on ADE20K and selection with SegMyO: (First line) input images + prior clues (b.box+label); (Second line) selected segmented objects.

As an additional experiment, we used this pipeline to segment SpatialSense (Yang et al., 2019), a dataset containing annotations of spatial relations between objects depicted by their bounding boxes. This dataset is made of 2 162 images with persons, animals, everyday-life objects, but also stuff classes (like "sky", "ground", "wall"...). As a pre-segmentation model, we used a HRNet model trained on ADE20K[4], since this dataset contains similar classes, and HRNet is one of the best models for segmentation, although it doesn't achieve instance segmentation by default. Figure 3 presents some samples of the outputs obtained. Such results highlight the interest of our pipeline in a real case, since SpatialSense is provided without any segmentation annotation, while such segmented objects can be useful for various computer vision tasks, as the computation of relative position descriptors between objects (Deléarde et al., 2021).

5   CONCLUSION

We introduced a turnkey pipeline (SegMyO) to automatically extract segmented objects in images based on given labels and / or bounding boxes. Ir relies on simple criteria to select the best segment among several proposals for a given object, by exploiting the knowledge of its class or its bounding box, with semantic similarity for the label and several geometric measures for the bounding box. SegMyO allows to easily and automatically segment and select any object knowing its bounding box and / or its label, making it a useful solution to segment a dataset without requiring dense annotation or specific training. This is also a promising improvement for weakly-supervised segmentation frameworks.

---

[4] https://github.com/CSAILVision/semantic-segmentation-pytorch

As a perspective, we consider to push our experiments further in order to assess more precisely the complementarity of our criteria and optimize their aggregation (using for example voting or learning strategies), on other datasets of the literature, where the semantic differences between the classes could be more important. In this context, other criteria might also be evaluated to take into account the specificity of each dataset. Finally, we also plan to evaluate the impact of embedding our criteria in weakly-supervised segmentation schemes, where they can be easily integrated.

ACKNOWLEDGEMENTS

REFERENCES

Ahn, J., Cho, S., and Kwak, S. (2019). Weakly supervised learning of instance segmentation with inter-pixel relations. In *CVPR*, pages 2209–2218.

Arbeláez, P., Pont-Tuset, J., Barron, J. T., Marques, F., and Malik, J. (2014). Multiscale combinatorial grouping. In *CVPR*, pages 328–335.

Chen, L.-C., Zhu, Y., Papandreou, G., Schroff, F., and Adam, H. (2018). Encoder-decoder with atrous separable convolution for semantic image segmentation. In *ECCV*, pages 801–818.

Chen, Y., Chan, A. B., and Wang, G. (2012). Adaptive figure-ground classification. In *CVPR*, pages 654–661.

Dai, J., He, K., and Sun, J. (2015). Boxsup: Exploiting bounding boxes to supervise convolutional networks for semantic segmentation. In *ICCV*, pages 1635–1643.

Deléarde, R., Kurtz, C., Dejean, P., and Wendling, L. (2021). Force banner for the recognition of spatial relations. In *ICPR 2020*, pages XX–XX.

Everingham, M., Van Gool, L., Williams, C. K., Winn, J., and Zisserman, A. (2010). The PASCAL visual object classes (VOC) challenge. *Int J Comput Vis*, 88(2):303–338.

Guillaumin, M., Küttel, D., and Ferrari, V. (2014). Imagenet auto-annotation with segmentation propagation. *Int J Comput Vis*, 110(3):328–348.

He, K., Gkioxari, G., Dollár, P., and Girshick, R. (2017). Mask R-CNN. In *ICCV*, pages 2961–2969.

Hong, S., Oh, J., Lee, H., and Han, B. (2016). Learning transferable knowledge for semantic segmentation with deep convolutional neural network. In *CVPR*, pages 3204–3212.

Hsu, C.-C., Hsu, K.-J., Tsai, C.-C., Lin, Y.-Y., and Chuang, Y.-Y. (2019). Weakly supervised instance segmentation using the bounding box tightness prior. In *NIPS*, pages 6586–6597.

Jurdi, R. E., Petitjean, C., Honeine, P., and Abdallah, F. (2020). BB-UNet: U-Net with bounding box prior. *IEEE J Sel Top Signal Process*.

Khoreva, A., Benenson, R., Hosang, J., Hein, M., and Schiele, B. (2017). Simple does it: Weakly supervised instance and semantic segmentation. In *CVPR*, pages 876–885.

Kolesnikov, A. and Lampert, C. H. (2016). Seed, expand and constrain: Three principles for weakly-supervised image segmentation. In *ECCV*, pages 695–711.

Lempitsky, V., Kohli, P., Rother, C., and Sharp, T. (2009). Image segmentation with a bounding box prior. In *ICCV*, pages 277–284.

Li, Q., Arnab, A., and Torr, P. H. (2018). Weakly-and semi-supervised panoptic segmentation. In *ECCV*, pages 102–118.

Lin, T.-Y., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., Dollár, P., and Zitnick, C. L. (2014). Microsoft COCO: Common objects in context. In *ECCV*, pages 740–755.

Papandreou, G., Chen, L.-C., Murphy, K. P., and Yuille, A. L. (2015). Weakly-and semi-supervised learning of a deep convolutional network for semantic image segmentation. pages 1742–1750.

Pascal, L., Bost, X., and Huet, B. (2019). Semantic and visual similarities for efficient knowledge transfer in cnn training. In *CBMI*, pages 1–6.

Rother, C., Kolmogorov, V., and Blake, A. (2004). "Grab-Cut" interactive foreground extraction using iterated graph cuts. *ACM Trans Graph*, 23(3):309–314.

Sun, R., Zhu, X., Wu, C., Huang, C., Shi, J., and Ma, L. (2019). Not all areas are equal: Transfer learning for semantic segmentation via hierarchical region selection. In *CVPR*, pages 4360–4369.

Wang, Y., Zhang, J., Kan, M., Shan, S., and Chen, X. (2020). Self-supervised equivariant attention mechanism for weakly supervised semantic segmentation. In *CVPR*, pages 12275–12284.

Wu, Z., Shen, C., and Van Den Hengel, A. (2019). Wider or deeper: Revisiting the resnet model for visual recognition. *Pattern Recognit*, 90:119–133.

Yang, K., Russakovsky, O., and Deng, J. (2019). SpatialSense: An adversarially crowdsourced benchmark for spatial relation recognition. In *ICCV*.

Zhou, B., Khosla, A., Lapedriza, A., Oliva, A., and Torralba, A. (2016). Learning deep features for discriminative localization. In *CVPR*, pages 2921–2929.

Zhou, Y., Zhu, Y., Ye, Q., Qiu, Q., and Jiao, J. (2018). Weakly supervised instance segmentation using class peak response. In *CVPR*, pages 3791–3800.

Zhu, G. and Iglesias, C. A. (2016). Computing semantic similarity of concepts in knowledge graphs. *IEEE Trans Knowl Data Eng*, 29(1):72–85.