Supervised Learning of Hierarchical Image Segmentation

Raphael Lapertot, Giovanni Chierchia, Benjamin Perret

Supervised Learning of Hierarchical Image Segmentation *

Raphael Lapertot[1 [0009−0004−1208−8115]], Giovanni Chierchia[1 [0000−0001−5899−689X]], and Benjamin Perret[1 [0000−0003−0933−8342]]

LIGM, Univ Gustave Eiffel, CNRS, ESIEE Paris, F-77454 Marne-la-Vallée, France

Abstract. We study the problem of predicting hierarchical image segmentations using supervised deep learning. While deep learning methods are now widely used as contour detectors, the lack of image datasets with hierarchical annotations has prevented researchers from explicitly training models to predict hierarchical contours. Image segmentation has been widely studied, but it is limited by only proposing a segmentation at a single scale. Hierarchical image segmentation solves this problem by proposing segmentation at multiple scales, capturing objects and structures at different levels of detail. However, this area of research appears to be less explored and therefore no hierarchical image segmentation dataset exists. In this paper, we provide a hierarchical adaptation of the Pascal-Part dataset [2], and use it to train a neural network for hierarchical image segmentation prediction. We demonstrate the efficiency of the proposed method through three benchmarks: the precision-recall and F-score benchmarks for boundary location, the level recovery fraction for assessing hierarchy quality, and the false discovery fraction. We show that our method successfully learns hierarchical boundaries in the correct order, and achieves better performance than the state-of-the-art model trained on single-scale segmentations.

Keywords: Image Segmentation · Supervised Learning · Ultrametric · Hierarchy · Graph.

1 Introduction

Image segmentation is the process of partitioning an image into distinct regions, which simplifies the image by focusing on the structure of its objects. A characteristic of image segmentation (and of images in general) is the scale: in an image, the visible structure depends on the observation scale. The choice of the scale is crucial and strongly depends on the application. To overcome this limitation, one solution is to not choose a scale at all, by proposing several consistent segmentations at different scales (satisfying the principle of strong causality [7]), i.e., building a *hierarchy (of segmentations)*. In this case, the choice of the scale

* This work is supported by the French ANR grant ANR-20-CE23-0019, and was granted access to the HPC resources of IDRIS under the allocation 2023-AD011013101R1 made by GENCI.

is not made during the segmentation, but *after* the segmentation, if even needed. In a hierarchy, an image is represented as a sequence of coarse to fine segmentations. Hierarchical segmentation also provides a more versatile and informative structure than traditional segmentation. It naturally allows multi-scale analysis, but also provides object hierarchy by capturing the relationships and dependencies between different segments. It is more flexibility by enabling users to adapt the segmentation output to suit their needs or application requirements. Finally, it can be useful in interactive scenarios, where a user can interact with the segmentation hierarchy to refine or adjust the segmentation results.

Hierarchies have long been used in computer vision as an intermediate representation to perform segmentation [18,6,1,13,15,5], or object detection and proposal [20,15]. Several works have been done to improve hierarchies using supervised learning techniques. In [16] the authors trained a cascade of edge classifiers based on classical human-designed features. Maninis *et al.* [9] trained a deep contour detector, the output of which is transformed into a hierarchy during post-processing ; they do not explicitly train their neural network for hierarchical segmentation, as they use classical image segmentation datasets with single scale annotations. More recently, Tao *et al.* [19] proposed a way to fuse segmentations at different scales using attention masks, but they do not predict a hierarchical segmentation. In general, while a variety of labeled datasets exist for image segmentation, this is not the case for hierarchical image segmentation, which is obviously a major problem for achieving supervised learning of hierarchical image segmentation.

The aim of this work is to train a neural network for hierarchical image segmentation. The contributions are threefold: (i) we build hierarchical segmentation ground truths for the Pascal-Part dataset, (ii) we propose a pipeline for supervised learning of a neural network that predicts hierarchies, and (iii) we define a benchmark to assess the quality of hierarchies.

Definitions. A *hierarchy* on an image is a sequence of partitions $P_1, ..., P_\ell$ of the image pixels, such that P_i is a *refinement* of P_{i-1}. Another possible representation of a hierarchy is the *ultrametric dissimilarity grid*, where the vertices are the pixels of the image, the edges represent the 4-adjacency relation between pixels, and the edge weights are a measure of dissimilarity satisfying the ultrametric property (a large dissimilarity means that the boundary represented by that edge persists along large scales). An ultrametric dissimilarity grid can be visualized as an image called an Ultrametric Contour Map (UCM), where interpixels are added to the original image to represent the grid edges; the size of an UCM is thus twice the size of the original image. The values of the interpixels are determined by the weights of the edges they represent (see Figure 1).

2 Ultrametric dataset

Our first contribution is to create a hierarchical dataset by transforming the existing annotations of the Pascal-Part dataset [2] into UCMs that enforce the principle of strong causality between objects and parts.

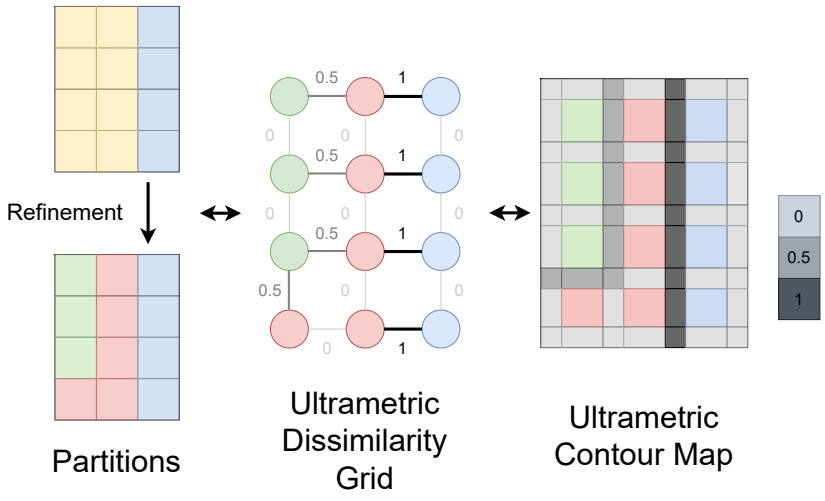


Fig. 1. A hierarchy of two partitions (left), ultrametric dissimilarity grid (middle) or Ultrametric Contour Map (right).

The Pascal-Part dataset extends the Pascal VOC 2010 dataset [4], which consists of 10103 natural images of different sizes, and annotations for different challenges such as classification, segmentation, and object detection. This dataset is a widely used dataset for supervised image segmentation learning and is challenging due to the complexity and diversity of its images. Pascal-Part provides an additional set of annotations for the Pascal VOC 2010 images, with segmentation masks for each instance of 20 classes of objects in each image, and segmentation masks for parts of these objects.

However, the segmentation masks of the Pascal-Part dataset have several limitations. First, some parts overlap with each other, and some parts sometimes completely cover other parts. Note that the objects, on the other hand, do not intersect with each other. Secondly, sometimes the contours of the parts do not match the contours of their object, being slightly off in the inner side of the object. Thirdly, there are parts of some objects that are behind non-annotated objects: sometimes the annotator *imagined* the continuation of the object behind. In the third sample of Figure 2, a leg of the chair is imagined, even though it is hidden by the non-annotated stool. A consequence of these observations is that Pascal-Part annotations do not respect the principle of strong causality: they do not form hierarchies.

We now describe our method for constructing an ultrametric dataset from the Pascal-Part dataset. First, we build a high-level segmentation, the *instance* segmentation, by stacking the object masks. We then build a low-level segmentation, the *part* segmentation, by successively stacking the part masks on top of the instance segmentation. The parts are processed in an order that ensures that smaller parts are not covered by the larger ones. This results in a hierarchical

segmentation, with a high-level segmentation (the instance map), and a low-level segmentation (the part map). At this stage, the misalignment of object and part boundaries creates a lot of spurious regions in the hierarchies. We mitigate this problem by filtering out the small regions (size less than 30 pixels) from the hierarchy using the method described in [12]. The entire processing pipeline was developed using the Higra library [11]. By performing this computation on each sample of the Pascal-Part dataset, we obtain a hierarchical segmentation dataset with an ultrametric dissimilarity grid for 10103 natural images. Three samples of our ultrametric dataset are shown in Figure 2. The dataset is publicly available here.

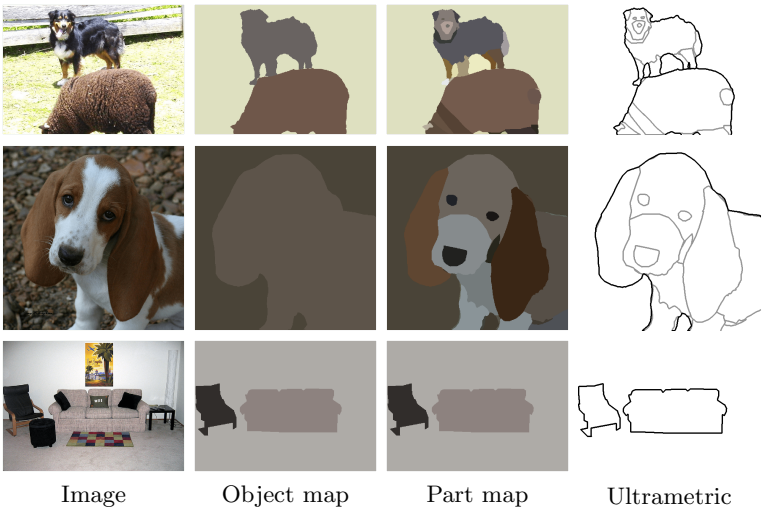


Fig. 2. Three samples from our Ultrametric Pascal-Part dataset, each consisting of an image, an object segmentation, a part segmentation, and the corresponding UCM.

The ultrametric dissimilarity grid will be the annotations that we will use in our method. Since the images are the same as the PASCAL Context [10], we can use the same dataset splits as used in COB [9]: *VOC train* refers to the official PASCAL Context train set, while the official PASCAL Context validation set is divided in two to create *VOC val* and *VOC test*. We have verified that this split maintains acceptable object class proportions. In the worst case (for the sofa class), there are still 19.2% of the total number of sofas in the *val* split, where we would expect 25%.

3 Model

Our second contribution is to train a neural network for hierarchical segmentation in a supervised manner, using the ultrametric dataset described in Section

2. Specifically, we approach this as a classification problem on the edges of the 4-adjacency grid of the input image: each edge is classified as being a low, mid, or high-level edge in accordance with the ultrametric dataset annotations.

The central part of our pipeline is a U-Net [17] that outputs three dissimilarity grids, one for each level of the hierarchy. To predict such dissimilarity grids, we first predict in the pixel domain, and then compute the mean of neighboring pixels to obtain dissimilarity grid weights. In the last layer, a softmax activation is used so that the neural network predicts for each edge the probability of belonging to each of the three levels. To train the U-Net, we were inspired by the loss function used in [21,9]. Let Θ be the U-Net parameters, I the input image, and w the dissimilarity weights of the corresponding target ultrametric dissimilarity grid. Furthermore, let $\Lambda(w)$ be the set of unique values in the targets, which are 0 (low), 0.5 (medium), or 1 (high-level). Our balanced cross-entropy loss function is defined as

$$\mathcal{L}(\Theta, I, w) = \sum_{\lambda \in \Lambda(w)} -\beta_\lambda \sum_{e \in E_\lambda(w)} \log \mathbb{P}\big(e \mapsto \lambda \mid \Theta, I\big). \tag{1}$$

In this equation, $E_\lambda(w) = \{e \in E \mid w(e) = \lambda\}$ is the set of edges whose value is λ in the target w, where E denotes the set of all edges in the target. The parameter $\beta_\lambda = 1 - |E_\lambda(w)|/|E|$ mitigates the class imbalance. Finally, $\mathbb{P}(e \mapsto \lambda \mid \Theta, I)$ is the predicted probability that the edge e has value λ in the ultrametric dissimilarity grid, according to the U-Net with parameters Θ for the input image I. Note that this loss function has no hyperparameter.

Once training is complete, image segmentation requires the conversion of the predicted edge class probabilities into hierarchical regions. This is done by post-processing the network predictions: First, we compute a dissimilarity grid from the edge class probabilities:

$$(\forall e \in E) \quad \mathsf{predict}(I, e) = \sum_{\lambda \in \Lambda(w)} \lambda \, \mathbb{P}(e \mapsto \lambda \mid \Theta, I). \tag{2}$$

Second, we compute superpixels with a watershed cut on the dissimilarity grid. Then, we construct a hierarchy of superpixels with average linkage. Finally, we filter out the small regions (smaller than 30 pixels) from the hierarchy [12]. Our complete pipeline is shown in Figure 3. It allows us to derive hierarchical segmentations for any input image.

4 Evaluation metrics

Our third contribution is a benchmark for evaluating the quality of hierarchical segmentation using two evaluation metrics. First, we adapted the *Boundary Recovery Order by Hierarchy Level* benchmark proposed in HSA [8]. It originally reports the proportion of boundaries from each level of the target UCM that were recovered in the predicted UCM, as a function of the overall recall. We have adapted it by making it a function of the segmentation threshold $t \in [0, 1]$

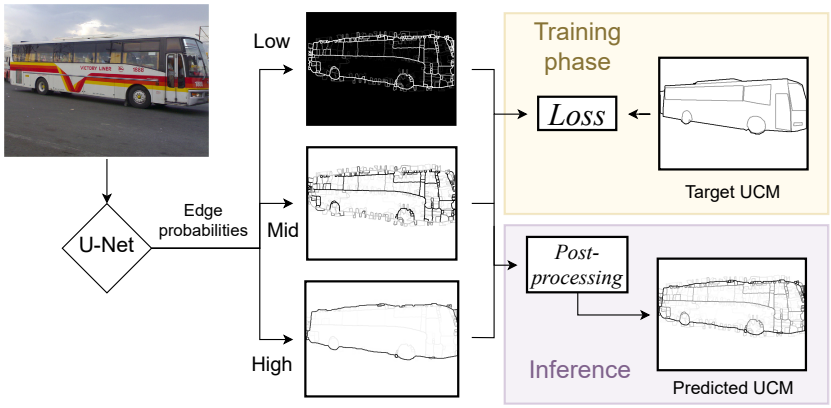


Fig. 3. Description of our method, for training and inference phase.

to see at what threshold the boundaries of each level are the most recovered. More formally, for a given predicted UCM U_{pred} and a given target UCM U_{tar}, we threshold U_{pred} at multiple thresholds t: for each edge e, $U_{pred_t}(e) = 0$ if $U_{pred}(e) < t$, and 1 otherwise. We then match U_{pred_t} with U_{tar}, and compute the Level Recovery Fraction (LRF) of each level $\lambda \in \Lambda(w)$:

$$LRF(\lambda, t) = \frac{|\{e \mid \text{match}_t(e) \text{ and } U_{tar}(e) = \lambda\}|}{|\{e \mid U_{tar}(e) = \lambda\}|}. \tag{3}$$

Where $\text{match}_t(e)$ is true if the edge e of U_{tar} matches an edge of U_{pred_t}. With a proper segmentation, the high-level boundaries will be recovered at a high threshold, and the mid-level boundaries will be recovered at a medium threshold. If the boundaries of each level are recovered simultaneously (i.e. the lines in the figure are similar), it means that the order of the target hierarchy is not reflected in the prediction.

Second, the False Discovery Fraction (FDF) is calculated as follows:

$$FDF(t) = \frac{|\{e \mid \text{not-match}_t(e) \text{ and } U_{pred_t}(e) = 1\}|}{|\{e \mid U_{pred_t}(e) = 1\}|}. \tag{4}$$

Where $\text{not-match}_t(e)$ is true if the edge e of U_{pred_t} does not match with any edge of U_{tar}. This measure, which is close to the false positives, calculates the proportion of boundaries that were detected but did not match with any level of the target UCM, at any segmentation threshold. With a good segmentation, this measure should be 0 for every threshold: this would mean that every boundary of the predicted UCM matched with a level of the target UCM.

Finally, we compute classical precision-recall curves for boundaries [1] and associated F-scores for the finest segmentation of the ground truth (this measure thus ignores the hierarchical nature of the ground-truth). We compute these three evaluation metrics on the test set *VOC test*.

5 Experiments

We evaluate the efficiency of our method through four questions: (i) Is it possible to infer the hierarchical levels of an image in the correct order? (ii) Are the boundaries placed correctly? (iii) Is the performance good for all classes? (iv) How do the models perform on a non-hierarchical dataset?

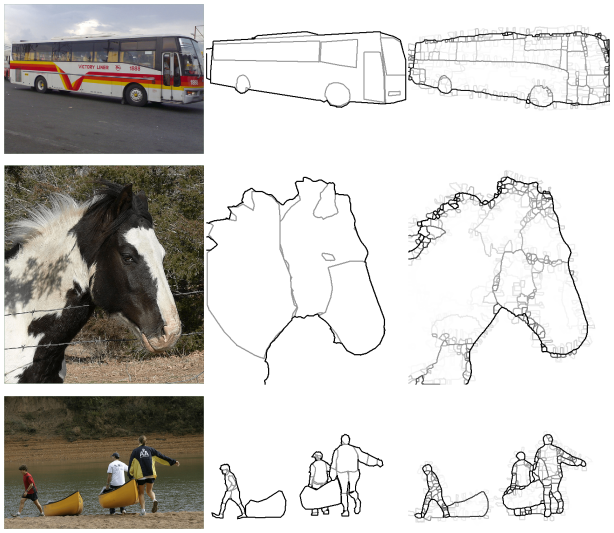
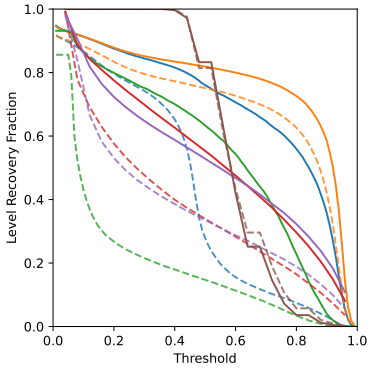


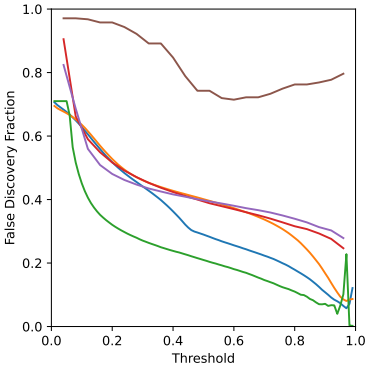
Fig. 4. Predictions of our method *HGM* on *VOC test*.

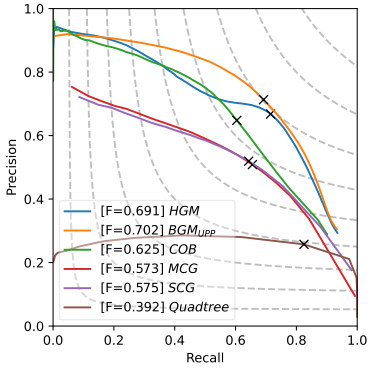
To answer these questions, we trained several neural networks. For each of them, we use a U-Net with a ResNet50 backbone pre-trained on ImageNet. We also augment our training data with simple spatial and texture transformations such as slight rotations, Gaussian noise, and optical distortions. We also perform fine-tuning by freezing the encoder weights of our U-Net except for the last layer of the encoder, training it with a learning rate of $1e^{-4}$ for 30 epochs, then unfreezing the neural network completely, and training it with a learning rate of $1e^{-5}$ for 20 epochs. Finally, we use a learning rate scheduler that divides the learning rate by 3 if the loss on the validation set (*VOC val*) does not decrease for more than 5 epochs, with a batch size of 64. Since the images in the dataset have different dimensions, we had to use mini-batches of 1 and backpropagate every 64 samples. To optimize the neural networks, we used the loss function described in the previous section. We train a first neural network *Binary Grid-weight Model (Ultrametric Pascal-Part)* (BGM_{UPP}), which classifies edges as low-level or high-level boundaries, on the part segmentation of our ultrametric dataset. We train a second neural network *Hierarchical Grid-weight Model* (*HGM*), which classifies edges as low-, mid-, or high-level boundaries, on our

ultrametric dataset. Some of its predictions are shown in Figure 4. Finally, we train a neural network (*Binary Grid-weight Model (PASCAL Context)* BGM_{PC}) that classifies edges as low or high-level boundaries, on the PASCAL Context dataset.

We now answer the first two questions by comparing BGM_{UPP}, HGM, COB [9], MCG [14], SCG [14] and $Quadtree$ with the three metrics. Since the other methods' neural networks were trained on PASCAL Context and not Pascal-Part, it would be unfair to compare their predictions directly with ours. To mitigate this problem, we remove the edges *that are far from the objects of interest*, both for our predictions (HGM and BGM_{UPP}) and for the predictions of COB, MCG, SCG and $Quadtree$. To do this, for each image, we merge the object masks provided by Pascal-Part, dilate the resulting mask by 20 pixels, remove contour parts outside this mask, and then remove non-closed contours. This leaves only the edges that are around and inside the objects of interest. The results are shown in Figure 5, along with the quantitative results. Let's in-







	ODS	OIS	AP
HGM	0.690818	0.718872	0.691637
BGM_{UPP}	**0.702478**	**0.735238**	**0.724585**
COB [9]	0.625332	0.690773	0.639541
MCG [14]	0.572893	0.643585	0.492475
SCG [14]	0.574702	0.636831	0.474807
Quadtree	0.392296	0.430570	0.262522

Fig. 5. Benchmarks our methods HGM (blue) and BGM_{UPP} (orange), on COB [9] (green), MCG [14] (red), SCG [14] (purple) and $Quadtree$ (brown) on our ultrametric dataset, with Level Recovery Fraction (top-left), solid lines for high-level, and dashed lines for mid-level), False Discovery Fraction (top-right), Precision-Recall Curves (bottom-left), and quantitative results (bottom-right).

terpret this figure, starting with the Level Recovery Fraction on the top-left. As expected, for BGM_{UPP} (in orange), both high-level (solid lines) and mid-level (dashed lines) boundaries are recovered at the same high threshold (around 1). This is not the case for HGM (in blue) for which the mid-level boundaries are recovered at a medium threshold (around 0.5). This demonstrates the effectiveness of our method for learning hierarchical image segmentation in a supervised manner. COB, on the other hand, recovers mid-level boundaries at a low threshold, which is normal since their neural network has not been trained to detect them. They, as well as MCG and SCG, detect high-level boundaries almost linearly, whereas we would expect them to be mostly detected at a high threshold. *Quadtree* recovers both mid-level and high-level boundaries indifferently and linearly at a medium threshold. Let's now focus on the False Discovery Fraction, on the top-right. Here, COB has the lowest amount of false detections, and the other methods except *Quadtree* seem even. Finally, let's look at the Precision-Recall curves. BGM_{UPP} has the best F-score, followed closely by HGM, and then COB. Note the curve of HGM which, compared to BGM_{UPP}, has a plateau around medium threshold. This is due to the order of level recovery: high-level boundaries are recovered first, and when the medium level boundaries are finally recovered, they are accurately detected. All in all, HGM effectively recovers the levels in the right order while maintaining very good F-scores and FDF. The other methods do not recover the mid-level boundaries at the right threshold, and BGM_{UPP} achieves the best F-score on our ultrametric dataset.

We address the third question by comparing HGM to COB on our hierarchical dataset, class by class. To do this, for each class, we have cropped the predictions with a small margin around each instance of the class. We do the same on the target UCMs, as well as removing the boundaries of other classes, and compute the three evaluation metrics on them. Some results are shown in Figure 6, and the full quantitative results are available at the end of the article (Figure 8). Let's start with the Level Recovery Fraction: for most classes, the hierarchical order is reflected in HGM, except for *bicycles*, *bottles*, *plants*, and *trains* where it is more ambiguous. Regarding the False Discovery Fraction, there is basically no difference with the first experiment: COB has a lower False Discovery Fraction for every class. Finally, the Precision-Recall curves and the F-scores change significantly from class to class. HGM has better F-scores for some classes (*bus*, *car*, *cat*, *person*), COB is better for some other classes (*boat*, *bottle*, *chair*, *pottedplant*), and the F-scores are even for the remaining classes.

Finally, we answer the last question: how do we perform on a non-hierarchical dataset such as PASCAL Context, compared to other methods? To do this, we compare BGM_{PC}, COB, MCG, SCG and *Quadtree* with the Precision-Recall, but not the Level Recovery Order as PASCAL Context is not hierarchical, nor the False Discovery Fraction as in a single-scale segmentation environment it is the false positive rate that is already reflected in the Precision-Recall curves. The results are shown in Figure 7. Although COB still leads in terms of F-score, our simple neural network method remains competitive. This also proves that

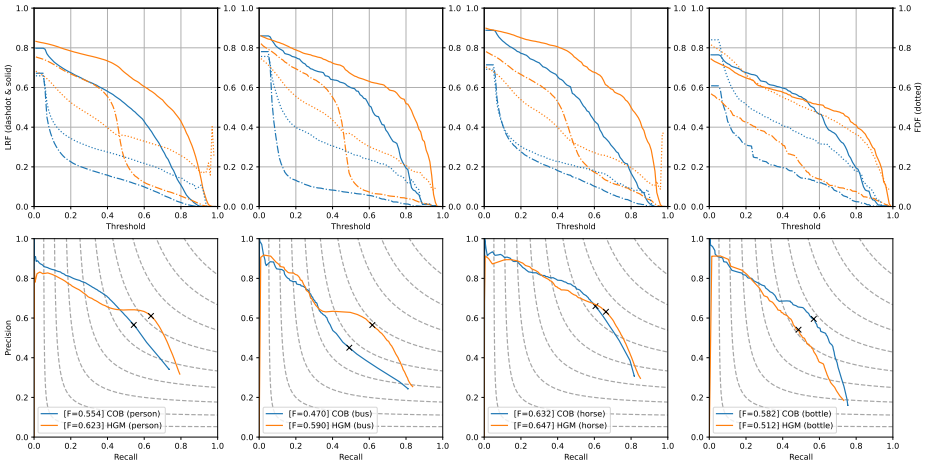
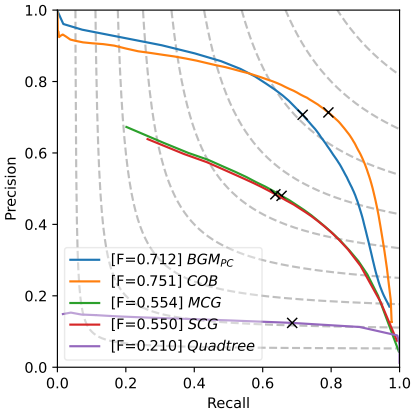


Fig. 6. Benchmarks on *COB* [9] (blue) and our method *HGM* (orange) on our ultrametric dataset per class (person on the left, bus in the middle-left, horse in the middle-right, bottle on the right), with Level Recovery Fraction (top, solid and dash-dot lines), False Discovery Fraction (top, dotted lines) and Precision-Recall Curves (bottom).



	ODS	OIS	AP
BGM_{PC}	0.711773	0.750888	0.749766
COB [9]	**0.750677**	**0.785024**	**0.773474**
MCG [14]	0.554028	0.614356	0.375386
SCG [14]	0.550186	0.608747	0.330745
Quadtree	0.210233	0.212235	0.126505

Fig. 7. Benchmark on the second experiment on PASCAL Context, with Precision-Recall Curves, and quantitative results.

the performance improvement in the previous experiments is not due to the backbone models, but rather to the hierarchical structure.

6 Conclusion

Hierarchical image segmentation offers interesting advantages over traditional image segmentation methods by allowing for multi-scale analysis, capturing objects and structures at different levels of detail. In this paper, we described a comprehensive pipeline for supervised learning of hierarchical image segmentation. We performed an ultrametric adaptation of the Pascal-Part dataset, built a neural network to predict ultrametric dissimilarity grids, and trained it on the latter dataset. We also evaluated its performance, in terms of boundary localization, hierarchy order, and false discovery fraction, and demonstrated the effectiveness of our method for learning hierarchical image segmentation. We showed that the results vary significantly from class to class. We have shown that our method, although using a simple neural network, remains competitive for non-hierarchical image segmentation compared to more complex neural network architectures such as COB. In future work, we plan to incorporate continuous hierarchy optimization methods [3] to obtain an end-to-end supervised hierarchical segmentation method. Another interesting question would be the prediction of semantic information in the hierarchy of contours.

References

1. Arbelaez, P., Maire, M., Fowlkes, C., Malik, J.: Contour detection and hierarchical image segmentation. IEEE transactions on pattern analysis and machine intelligence **33**(5), 898–916 (2010)
2. Chen, X., Mottaghi, R., Liu, X., Fidler, S., Urtasun, R., Yuille, A.: Detect what you can: Detecting and representing objects using holistic models and body parts. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (June 2014)
3. Chierchia, G., Perret, B.: Ultrametric fitting by gradient descent. In: Wallach, H., Larochelle, H., Beygelzimer, A., d'Alché-Buc, F., Fox, E., Garnett, R. (eds.) Advances in Neural Information Processing Systems. vol. 32. Curran Associates, Inc. (2019), `https://proceedings.neurips.cc/paper/2019/file/b865367fc4c0845c0682bd466e6ebf4c-Paper.pdf`
4. Everingham, M., Van Gool, L., Williams, C.K.I., Winn, J., Zisserman, A.: The pascal visual object classes (voc) challenge. International Journal of Computer Vision **88**(2), 303–338 (Jun 2010)
5. Funke, J., Tschopp, F., Grisaitis, W., Sheridan, A., Singh, C., Saalfeld, S., Turaga, S.C.: Large scale image segmentation with structured loss based deep learning for connectome reconstruction. IEEE transactions on pattern analysis and machine intelligence **41**(7), 1669–1680 (2018)
6. Guigues, L., Cocquerez, J.P., Le Men, H.: Scale-sets image analysis. International Journal of Computer Vision **68**, 289–317 (2006)
7. Koenderink, J.J.: The structure of images. Biological cybernetics **50**(5), 363–370 (1984)

8. Maire, M., Yu, S.X., Perona, P.: Hierarchical scene annotation. In: British Machine Vision Conference (BMVC) (2013)
9. Maninis, K.K., Pont-Tuset, J., Arbeláez, P., Van Gool, L.: Convolutional oriented boundaries: From image segmentation to high-level tasks. IEEE transactions on pattern analysis and machine intelligence **40**(4), 819–833 (2017)
10. Mottaghi, R., Chen, X., Liu, X., Cho, N.G., Lee, S.W., Fidler, S., Urtasun, R., Yuille, A.: The role of context for object detection and semantic segmentation in the wild. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 891–898 (2014)
11. Perret, B., Chierchia, G., Cousty, J., Guimarães, S.J.F., Kenmochi, Y., Najman, L.: Higra: Hierarchical graph analysis. SoftwareX **10**, 100335 (2019)
12. Perret, B., Cousty, J., Guimarães, S.J.F., Kenmochi, Y., Najman, L.: Removing non-significant regions in hierarchical clustering and segmentation. Pattern Recognition Letters **128**, 433–439 (2019). https://doi.org/10.1016/j.patrec.2019.10.008
13. Perret, B., Cousty, J., Tankyevych, O., Talbot, H., Passat, N.: Directed connected operators: Asymmetric hierarchies for image filtering and segmentation. IEEE transactions on pattern analysis and machine intelligence **37**(6), 1162–1176 (2014)
14. Pont-Tuset, J., Arbeláez, P., Barron, J., Marques, F., Malik, J.: Multiscale combinatorial grouping for image segmentation and object proposal generation. In: arXiv:1503.00848 (March 2015)
15. Pont-Tuset, J., Arbelaez, P., Barron, J.T., Marques, F., Malik, J.: Multiscale combinatorial grouping for image segmentation and object proposal generation. IEEE transactions on pattern analysis and machine intelligence **39**(1), 128–140 (2016)
16. Ren, Z., Shakhnarovich, G.: Image segmentation by cascaded region agglomeration. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 2011–2018 (2013)
17. Ronneberger, O., Fischer, P., Brox, T.: U-net: Convolutional networks for biomedical image segmentation. In: Medical Image Computing and Computer-Assisted Intervention–MICCAI 2015: 18th International Conference, Munich, Germany, October 5-9, 2015, Proceedings, Part III 18. pp. 234–241. Springer (2015)
18. Salembier, P., Garrido, L.: Binary partition tree as an efficient representation for image processing, segmentation, and information retrieval. IEEE transactions on Image Processing **9**(4), 561–576 (2000)
19. Tao, A., Sapra, K., Catanzaro, B.: Hierarchical multi-scale attention for semantic segmentation (2020). https://doi.org/10.48550/ARXIV.2005.10821, `https://arxiv.org/abs/2005.10821`
20. Uijlings, J.R., Van De Sande, K.E., Gevers, T., Smeulders, A.W.: Selective search for object recognition. International journal of computer vision **104**, 154–171 (2013)
21. Xie, S., Tu, Z.: Holistically-nested edge detection. In: Proceedings of the IEEE international conference on computer vision. pp. 1395–1403 (2015)

	ODS	OIS	AP
COB (aeroplane)	0.680047	0.657341	0.686952
HGM (aeroplane)	0.658955	0.621932	0.670039
COB (bicycle)	0.640781	0.603702	0.561952
HGM (bicycle)	0.602852	0.575719	0.585744
COB (bird)	0.648532	0.560844	0.619821
HGM (bird)	0.621336	0.522969	0.586066
COB (boat)	0.482006	0.409506	0.359728
HGM (boat)	0.421522	0.379770	0.285375
COB (bottle)	0.581943	0.509849	0.514802
HGM (bottle)	0.512207	0.451764	0.454253
COB (bus)	0.469884	0.491188	0.462006
HGM (bus)	0.589534	0.562339	0.538801
COB (car)	0.448132	0.413176	0.386884
HGM (car)	0.550439	0.446858	0.455647
COB (cat)	0.574996	0.600807	0.582450
HGM (cat)	0.614230	0.623224	0.604625
COB (chair)	0.508438	0.507779	0.374837
HGM (chair)	0.444168	0.460078	0.325109
COB (cow)	0.623693	0.594259	0.607716
HGM (cow)	0.646403	0.588841	0.606057
COB (dog)	0.580688	0.592093	0.582632
HGM (dog)	0.618172	0.610517	0.589478
COB (horse)	0.631965	0.601856	0.606264
HGM (horse)	0.647019	0.606201	0.616302
COB (motorbike)	0.586627	0.570394	0.550539
HGM (motorbike)	0.545491	0.542184	0.525384
COB (person)	0.553968	0.518099	0.500073
HGM (person)	0.623382	0.547710	0.531034
COB (pottedplant)	0.541500	0.489925	0.462293
HGM (pottedplant)	0.456295	0.407224	0.340515
COB (sheep)	0.583776	0.515271	0.514862
HGM (sheep)	0.583537	0.500565	0.489362
COB (sofa)	0.421639	0.527803	0.279333
HGM (sofa)	0.440106	0.506891	0.340161
COB (train)	0.479617	0.519992	0.405943
HGM (train)	0.474488	0.495916	0.414802
COB (tvmonitor)	0.538754	0.572695	0.444587
HGM (tvmonitor)	0.507300	0.499316	0.379238

Fig. 8. Quantitative results on Ultrametric Pascal-Part, class by class