Maturity-Aware Active Learning for Semantic Segmentation with Hierarchically-Adaptive Sample Assessment

Amirsaeed Yazdani
yazdaniamir38@yahoo.com
Xuelu Li
xueluli122@gmail.com
Vishal Monga
vum4@psu.edu

Department of Electrical Engineering
The Pennsylvania State University
University Park, USA

Abstract

Active Learning (AL) for semantic segmentation is challenging due to heavy class imbalance and different ways of defining "sample" (pixels, areas, etc.), leaving the interpretation of the data distribution ambiguous. We propose "Maturity-Aware Distribution Breakdown-based Active Learning" (MADBAL), an AL method that benefits from a hierarchical approach to define a multiview data distribution, which takes into account the different "sample" definitions jointly, hence able to select the most impactful segmentation pixels with comprehensive understanding. MADBAL also features a novel uncertainty formulation, where AL supporting modules are included to sense the features' maturity whose weighted influence continuously contributes to the uncertainty detection. In this way, MADBAL makes significant performance leaps even in the early AL stage, hence reducing the training burden significantly. It outperforms state-of-the-art methods on Cityscapes and PASCAL VOC datasets as verified in our extensive experiments.

1 Introduction

Neural networks in the past decade have been dominant solutions for a wide majority of computer vision problems [8, 25, 28, 30, 45]; however, these solutions often suffer from being data-eccentric, which means a burden in both the data collection and annotation. While this burden exists for almost every computer vision task, it becomes more costly and laborious for tasks that need fine-grained annotations, such as image segmentation.

Active learning (AL) methods have been proposed to overcome this bottleneck by incrementally selecting the samples for improving the performance of the current model, which has been trained on a limited training set. AL offers some criteria based on which the samples in an unlabeled pool are assessed, ranked, selected, and then added to the current training pool. The majority of existing AL methods [1, 2, 12, 31] rely on a combination of the criteria such as model uncertainty and the diversity in the labeled pool. Although these criteria seem intuitive and are well-defined for tasks such as classification, for segmentation their definition becomes ambiguous, hence challenging to quantify. This is due to the fact that

© 2023. The copyright of this document resides with its authors.
It may be distributed unchanged freely in print or electronic forms.

in segmentation, "sample" does not have a concrete definition based on which the formulation can uniquely rank the samples like classification. In segmentation, a "sample" can be perceived either as a pixel, a patch, or an image. Moreover, the informativeness of the samples for a specific definition is correlated to the information of surrounding samples. This leaves the existing segmentation AL methods [4, 5, 38], focusing on only one interpretation of samples, with a narrow insight into the data distribution.

In this work, we propose a systematic and inclusive AL strategy as a natural evolution of existing works with different sample considerations, tied with backbone-agnostic, AL supporting network components. Specifically, our key contributions are:

- **Distribution Breakdown**: We propose a **hierarchical approach** to estimate the data distribution based on different definitions of "sample," which allows for a multilevel assessment of the data. We traverse this hierarchy level by level, while at each level breaking down the distribution of the data according to the corresponding "sample" definition (see Fig. 2). This means that the representativeness of the selected data is checked across multiple views making the training set as insightful as possible.

- **Maturity-Awareness**: We propose a set of **backbone-agnostic, AL supporting modules** associated with **carefully devised uncertainty terms** which together are capable of detecting the most impactful samples for network performance improvement. AL supporting modules help monitor the flow of information through different layers with different features' maturity level (see Fig. 1). This flow is interpreted via our proposed uncertainty formulation which evaluates the model maturity for different samples.

- Integration of the aforementioned algorithmic pieces results in a model referred to as "**M**aturity-**A**ware **D**istribution **B**reakdown-based **A**ctive **L**earning" (**MADBAL**). We evaluate the performance of our model on Cityscapes [6] and Pascal VOC 2012 [10] datasets and prove that not only does MADBAL outperform state of the art w.r.t different metrics, but also exhibits immediate performance leaps unlike state of the art where the improvements are more gradual. This makes MADBAL a preferred AL solution for reducing the training burden overhead from two standpoints: 1) **Lower number of AL steps** needed for achieving acceptable results. 2) **No requirement for a rich, carefully-selected initial labeled pool**.

2 Related Works

Selection criteria in the AL literature consist mainly of two types of nature: **uncertainty** and **diversity**. Uncertainty-based criteria [12] focus on how certain the model is in its prediction for a possible candidate and select ones with more uncertainty. They can be mathematically formulated in a variety of ways such as posterior probability of the predicted class [23] or the margin between the posterior probabilities of the predicted class and the the class that received the second highest predicted probability [19]. This complements diversity-based criteria [31] with the main objective to help the training set maintain a representation as close as possible to the whole distribution of the data. This would lead to detection and addition of the samples so that the distance between the training set and the unlabeled pool is minimized (Core-set AL) [36], or the most representative subset of the unlabeled pool is constructed [9]. AL methods consider one or a hybrid combination of these two criteria for their selection. For example, BALD [16] uses a Bayesian framework to select samples

based on the uncertainty of sampled networks. Later, BatchBALD [20] was proposed as a modification of BALD to take the diversity into consideration. Besides these criteria, **expected model change** is used in a few works [11, 21, 34, 37] as a criterion to select samples that cause the greatest change in the current model or its output. For example, Freytag *et al.* [11] use the current model to predict the output changes, while Settles *et al.* [37] rely on the predicted gradient length to select samples.

AL methods for semantic segmentation are categorized into image-based and region-based methods: **Image-based methods** [14, 41, 43, 44] are often faster with lower computational complexity owing to their definition of "sample," which gives them smaller search space at the expense of adding redundant classes at every AL step. This in turn leads to less budget-efficiency of these approaches. As an image-based method for medical image segmentation, Yang *et al.* [44] propose a CNN architecture and a heuristic method to find the most representative samples among top k with highest uncertainty. Within the same domain, [14] leverages drop-out to represent the Monte Carlo sampling at test time for melanoma segmentation. [41] leverages the min-max game between the adversarial network and the variational autoencoder (VAE) to discriminate between challenging and easy samples based on the features in the latent space. Inspired by the work of Yoo *et al.* [46] for dedicating network components for loss prediction, Xie *et al.* [43] develop a difficulty-aware network to generate difficulty heatmaps using the missclassified/correctly classified pixels in the labeled pool. **Region-based methods** [4, 29, 33, 38, 39], unlike image-based methods, show higher performance with significantly lower budget as they are able to select only the regions with the most helpful classes for annotation, hence no need for annotating useless regions. This has led to emergence of more region-based methods recently. CEREALS [29] estimates the cost of annotating regions and finds a trade-off between the informativeness and annotation cost of the candidates. Golestaneh *et al.* [13] utilize the fact that the most uncertain regions show high uncertainty under equivariant transformations. Recently, Cai *et al.* [4] was one of the pioneers in estimating the data distribution by using the trained model at the current step to find the dominant labels across superpixels and select the most uncertain superpixels whose dominant labels belong to less frequent classes, which inspires our uncertainty formulation at the superpixel level; nevertheless we extend [4] by introducing AL supporting modules at other levels. Focusing on pixels, PixelPick [38] in each round of AL selects an equal number of pixels with highest uncertainty from each image. Recently, [33] deploys a regional Gaussian attention module to select regions and leverages contextual guidance to extend the regional annotations to unlabeled regions, while borrowing the idea of the loss prediction module from [43]. The proved benefits of loss prediction module in [33, 43, 46] motivates us in including it in our AL supporting modules; however, as it will be elaborated, ours benefits from a more effective training protocol (separate training phases), allocating boundary-aware output channels, and more effective ground truth formulation.

3 Methods

Our method includes two main components: 1) AL supporting modules, components of which reflect the **information flow needed for maturity-awareness**, and 2) selection strategy, which reflects our **hierarchical distribution breakdown scheme** integrated with our **custom uncertainty formulation**.

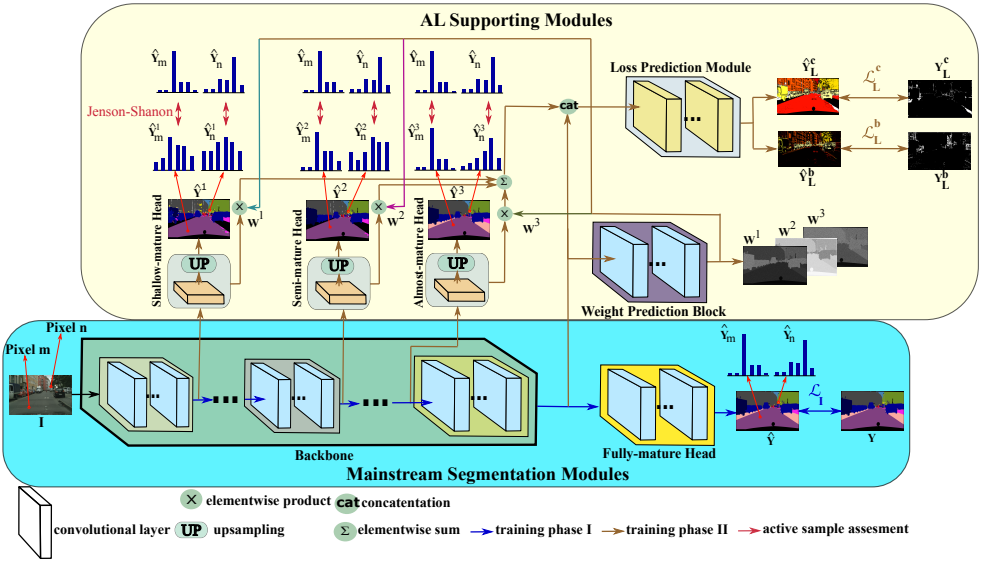

Figure 1: The proposed backbone-agnostic architecture. We use pixels m and n as examples of easy and challenging samples to understand sample assessment mechanism. For the easy sample, the predicted probability distributions ($\hat{Y}_m^i$) show high similarity to the final distribution ($\hat{Y}_m$) (measured by Jenson-Shannon divergence). For the challenging sample, the predicted probabilities ($\hat{Y}_n^i$) show a confusing trend at shallow stages, taking longer to show a consistent trend and high similarity to the final distribution ($\hat{Y}_n$). For the details of the modules and their training schemes refer to 3.1.

3.1 AL Supporting Modules

Our network consists of conventional modules essential for carrying out the main segmentation task (Mainstream Segmentation modules – see Fig. 1) which makes it backbone-agnostic. These modules are trained through a preliminary phase of training (training phase I) with cross-entropy (CE) loss. Once these modules are trained, the training of AL supporting modules, whose purpose is critical at the time of sample selection, starts based on their designated goal.

Varied-Maturity Heads besides the main segmentation head (Fully-mature head – see Fig. 1), include three heads with access to different depths of the backbone layers. Indeed, as the depth increases, the maturity of features provided to these heads increases. Starting from the shallowest, we denote them with **Shallow-mature Head**, **Semi-mature Head**, and **Almost-mature Head**. Each head in training phase II is trained for the segmentation task and assigned a loss term ($\mathcal{L}^{\mathrm{k}}_{seg}$) which is CE loss defined on segmentation outputs.
$k\in\{1,2,3\}$

Loss Prediction Module is in charge of predicting the probability of each pixel's error-proneness for the segmentation task. This module makes use of a different version of the feature maps provided to the varied-maturity heads and the endpoint features of the backbone (see Fig. 1). These features are weighted by the weight maps provided by the **Weight Prediction Block** to help the model reweight the features based on their importance. We define the ground truth for the loss prediction task by considering the class-specific average

loss across the labeled pool as a threshold for determining the loss labels:

$$Y_L^i = \begin{cases} 1, & \text{if } L_I^i \geq \tau_{Y_i} \\ 0, & \text{otherwise} \end{cases} \tag{1}$$

Where Y_L^i, τ_c, and L_I^i are the loss label for pixel i, the phase I mean loss (CE) across all the pixels belonging to class c in the labeled pool, and phase I loss for pixel i, respectively. Since each pixel is labeled based on how it compares to other members of its class, the model acquires a more insightful loss prediction capability specialized for each class. Moreover, as the model is already trained for the main task, the ground truth does not change during the training of this module unlike existing works [33, 43]. Next, we follow the training for the loss prediction task separately for the pixels lying on the boundary and center regions via assigning two output channels. This aids the module for a better focus on different levels of error as it is known that the segmentation error is generally higher on the boundary of the objects [27, 47]. Thus, the loss for loss prediction module would be:

$$\mathcal{L}_L^{m\in\{c,\,b\}}(\theta_{\,seg\ heads}, \theta_L, \theta_W) = \frac{1}{|\{x : x \in m\}|} \sum_{\{i:x_i\in m\}} \Big(-Y_i^L log(\sigma(\hat{Y}_{L-m}^i)) - (1-Y_i^L) log(1-\sigma(\hat{Y}_{L-m}^i))\Big) \tag{2}$$

Where $\theta_{seg\ heads}$, θ_L, and θ_W denote the parameters of varied-maturity segmentation heads, loss prediction module, and weight prediction block, respectively. Additionally, $|.|$, $\sigma(.)$, and $\hat{Y}_{L-m}^i$ are the cardinality operator, the Sigmoid function, and the output of the channel m (boundary or center) of the loss prediction module, respectively.
Now, we have everything for the loss of training phase II in place:

$$\mathcal{L}_{II} = \lambda_0 \mathcal{L}_L^c + \lambda_1 \mathcal{L}_L^b + \sum_{k=1}^{3} \lambda_{k+1} \mathcal{L}_{seg}^k \tag{3}$$

Where λ_k's are the regularization constants chosen by cross-validation and parameter search.

3.2 Selection Strategy

We follow a hierarchical approach to breakdown the distribution of the data through which we measure the uncertainty of samples by starting to look from a low field of view (pixel level: top level of the hierarchy) going incrementally to the highest field of view (Image level). At each field of view, we assess the uncertainty level within different scopes to find regions needing attention for sampling.

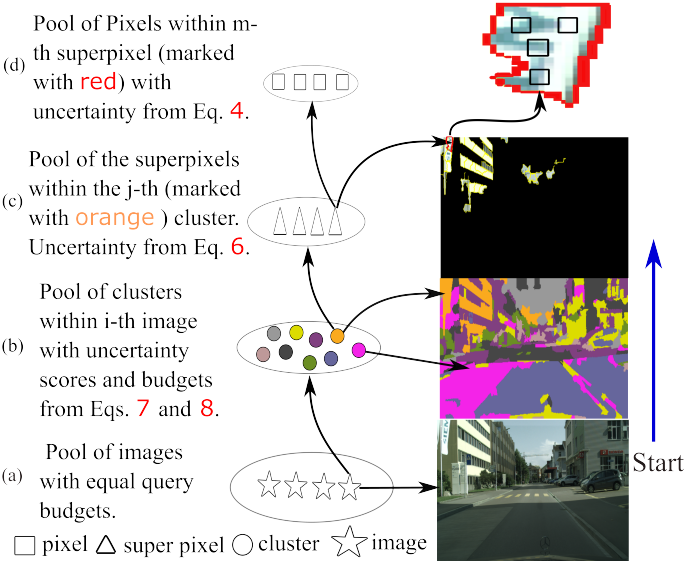

Figure 2: Our hierarchical distribution breakdown approach.

Pixel Level-Getting Aware of the Maturity: We start the uncertainty assessment by analyzing the pixels individually. At this level, we feed all the samples to the trained model and for the pixels in the unlabeled pool, measure the uncertainty based on:

$$u(x) = \Big(H(\hat{Y}_x) + W_x^1 JS(\hat{Y}_x, \hat{Y}_x^1) + W_x^2 JS(\hat{Y}_x, \hat{Y}_x^2) + W_x^3 JS(\hat{Y}_x, \hat{Y}_x^3)\Big) \Big[\Big(1-\delta(x)\Big)e^{\sigma(\hat{Y}_{L-b}^x)} + \delta(x)e^{\sigma(\hat{Y}_{L-c}^x)}\Big] \quad (4)$$

$$\delta(x) = \begin{cases} 1, & \text{if } x \in \textit{center} \\ 0, & \text{otherwise} \end{cases}$$

Here $H(P)$, $JS(P_1,P_2)$, W_x^k, and $\hat{Y}_x^k$ denote entropy of probability distribution P, Jenson-Shannon divergence between distributions P_1 and P_2, weight map predicted by the weight prediction block for the k-th head, and the output distribution by the k-th head for pixel x, respectively. Via Eq. 4, we measure the uncertainty of the model for pixel x by checking: i) the entropy of the final output distribution (reflected by the first term), ii) the similarity of the final output distribution to those of the varied-maturity heads (reflected by the second-fourth terms). The intuition is that the easier a pixel is to classify, the less depth is needed to produce an output similar to the final one. The importance of each term is determined by the weight map corresponding to the pixel and segmentation head (W_x^k). Lastly, iii) error-proneness of the pixel (reflected by the exponential terms), determined based on the score given for the pixel by its corresponding channel (center or boundary) of the loss prediction module.

Superpixel Level: Next, we zoom out and look through the superpixel level. By definition, a superpixel is a group of perceptually similar pixels. First, we assign each superpixel to its dominant label $Do(s)$ (the predicted class for the majority of the pixels within that superpixel [4]); however, unlike [4], we estimate the probability of the class C_i within cluster k (cl_k) by counting the superpixels with the dominant label of C_i:

$$P_{cl_k}(C_i) = \frac{|\{s : Do_s = C_i \& s \in cl_k\}|}{|\{s : s \in cl_k\}|} \quad (5)$$

Now, we assess the uncertainty of each superpixel by:

$$u(s) = \frac{\sum_{x \in s} u(x)}{|\{x | x \in s\}|} e^{-P_{cl_k}(Do(s))} \; s.t. \; s \in cl_k \quad (6)$$

Based on Eq. 6 the uncertainty of a superpixel in a cluster is proportional to the average uncertainty of its pixels and inversely proportional to the abundance of its dominant label.

Cluster Level: Having the uncertainty of the superpixels in each cluster, we now assess the uncertainty of each cluster:

$$u(cl_k) = \frac{\sum_{s \in cl_k} u(s)}{|\{s : s \in cl_k\}|} \quad (7)$$

The uncertainty of each cluster determines the budget it is assigned in the sample selection step. The more uncertain a cluster is, the larger budget it is assigned to:

$$B_{cl_k} = \left\lceil \frac{u(cl_k)}{\sum_{j=1}^{N_{clusters}} u(cl_j)} B_t \right\rceil \quad (8)$$

Where B_{cl_k}, B_t, and $\lceil . \rceil$ are the budget assigned to cluster k, the total budget, and the ceil function, respectively.

Image Level: Once the uncertainty scores of all the lower fields of view are figured out, we query pixels for each image based on: 1) considering budget dedicated to each cluster in the image, 2) finding superpixels with highest uncertainty within that cluster, and

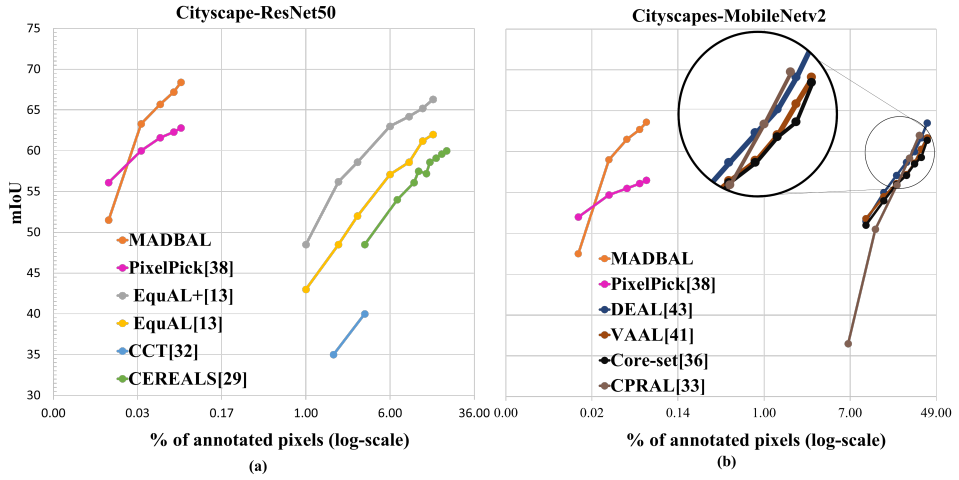

Figure 3: Comparison with SOTA on Cityscapes with two of the most popular backbones in AL methods (ResNet50 and MobileNetv2).

3) selecting pixels with highest uncertainty within these superpixels. It is worth mentioning that traversing from the top to the bottom of the hierarchy helps us achieve a global insight of the uncertainty across different regions of the image, while the trip back to the top aids with finely detecting and selecting a small, yet impactful number of samples for annotation.

4 Experiments [1]

Implementation Details: We evaluate MADBAL on Cityscapes [6] and Pascal VOC 2012 [10] datasets by training on samples from the training set and testing on the validation set. Our initial labeled pools have 10 and 20 randomly selected pixels per image for VOC and Cityscapes, respectively. The AL budget in our experiments is 10 and 20 pixels per image for VOC and Cityscapes, respectively. We use SEEDS algorithm [3] for superpixel extraction and set the number of superpixels per images for both datasets equal to the number of squares when the image is divided to the squares of size 16×16. we cluster these superpixels; however, superpixels are of irregular (not necessarily rectangular or vector) shapes, which is not acceptable by K-means. To address this, we first fit the superpixel at the center of a rectangular patch with minimum size and then resize that patch to a certain size (16×16). Consequently, we feed the resized patch to the backbone of a pretrained VGG 16 [40] and apply K-means to the extracted feature vectors. This way the clustering would be done based on the perceptual properties of the superpixels. For each dataset, we conduct our experiments three times with its most prevalent backbones in the literature: ResNet50 [15], MobileNetv2 [35], and MobileNetv3 [17] for Cityscapes and ResNet50 and MobileNetv3 for VOC.

During training, for Cityscapes, we acquire random crops of size 768×768 from the samples and for VOC random crops of size 256×256. Our models are deployed using Pytorch and we use stochastic gradient descent optimizer with an initial learning rate of 0.01, momentum of 0.9 and a poly learning rate scheduler decaying the learning rate from the initial value to zero linearly through 150 epochs (for phase I) and 30 epochs (for phase II). The hyperparameres in Eq. 3 are 1, 1, 0.05, 0.1, and 0.15 respectively, selected via cross-validation on initial labeled pools of Cityscapes.

[1] Find the numerical data for the plots and codes here: github/MADBAL

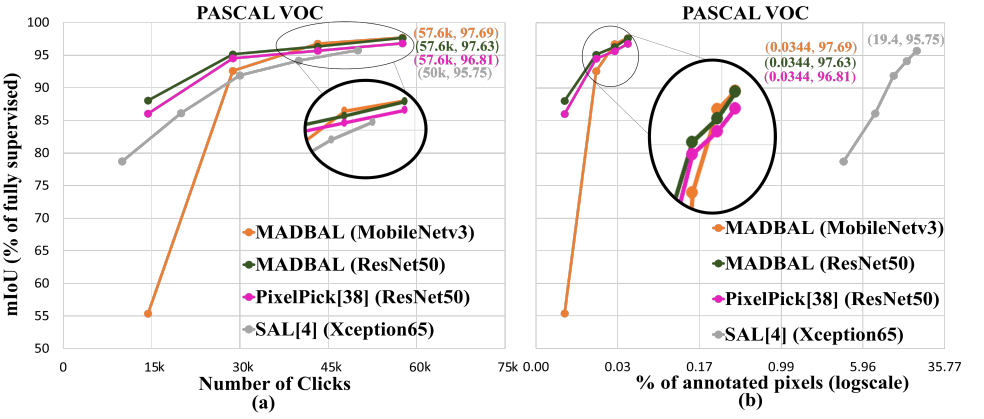

Figure 4: Comparison with SOTA on VOC. w.r.t. number of clicks **(a)** and percentage of annotated pixels **(b)**.

Comparison with State of the Art: Figs. 3 , 4, and 5a report the average *mean intersection over union* (mIoU) of three repetitions of our experiments for each dataset and backbone w.r.t. different budget measures. We can observe that our annotation cost is two orders of magnitude lower than the majority of the SOTA (regardless of the backbone) w.r.t. the percentage of annotated pixels, while outperforming SOTA with a significant margin w.r.t. number of clicks. Moreover, MADBAL starts with a lower performance than Shin *et al.*'s [38], which is mostly depending on the richness of the initial labeled pool, and makes considerably large leaps and outperforms their method quickly. This implies MADBAL's effectiveness in selecting the most important samples early on. To get a qualitative sense of these leaps, Fig. 5b visualizes the performance of MADBAL through the first two AL steps on a validation sample from each dataset. Finally, Tab. 1 compares various weakly-supervised and interactive weak supervision methods on VOC, confirming the benefits of MADBAL trained with only 20 pixels per image.

Table 1: Comparison with Weakly-supervised methods and PixelPick [38] on VOC.

Method	Backbone	Train set (anno. type)	mIoU
Weakly-supervised methods			
GAIN [24]	VGG16	10.k imgs (classes)	55.3
MDC [42]	VGG16	10.k imgs (classes)	60.4
DSRG [18]	ResNet101	10.5k imgs (classes)	61.4
FickleNet [22]	ResNet101	10.5k imgs (classes)	64.9
BoxSup [7]	VGG16	10.5k imgs (boxes)	62.0
ScribbleSup [26]	VGG16	10.5k imgs (scribbles)	63.1
Interactive weak supervision			
PixelPick [38]	ResNet50	1.5k imgs (20 pixels per image)	65.6
MADBAL	ResNet50	1.5k imgs (20 pixels per image)	**72.4**

Ablation Study: We validate our design of MADBAL by conducting experiments devised to show how presence and absence of various components affect the performance. For these experiments, we incorporate MobileNetV3 [17] backbone and Cityscapes [6] dataset. **i) Effect of maturity-awareness and loss prediction** in this set of experiments is focused through four different AL scenarios: **1)** AL with MADBAL. **2)** AL with a modified MAD-

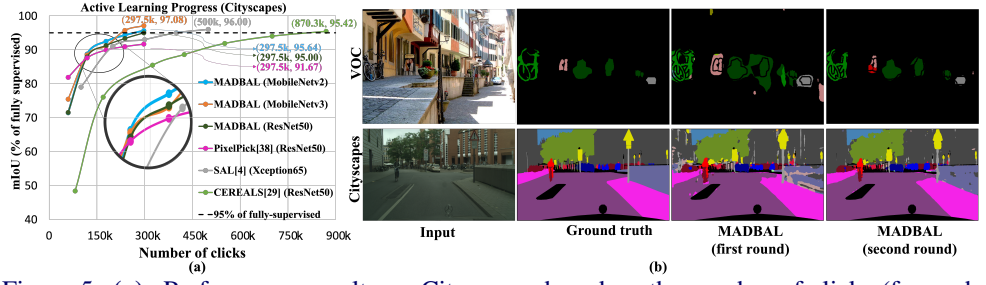

Figure 5: **(a)**: Performance results on Cityscapes based on the number of clicks (for each method the exact values of the last AL step are shown with its matching color). **(b)**: Visualization results on both datasets. First round of AL is completed with 20 and 10 pixels per image, and second round with 40 and 20 for Cityscapes and VOC, respectively.

BAL in which weight prediction block is dropped and averaging is used instead (i.e. $\frac{1}{3}$ is used instead of weight maps). This is to show the essence of giving different importance to different intermediate features and its benefits for recognition of impactful pixels (denoted with "Averaging"). **3)** AL with a modified MADBAL in which the loss prediction module only accesses the backbone features (i.e. no inputs from the varied-maturity heads) and the uncertainty score calculation (Eq. 4) does not have Jenson-Shannon divergence terms. This helps observe the effect of maturity-awareness directly by removing the corresponding terms in uncertainty score formulation (denoted with "No maturity-awareness"). **4)** AL with vanilla backbone (no loss prediction module, weight prediction block, and varied-maturity heads) to analyze the performance of MADBAL solely relying on distribution breakdown (denoted with "Vanilla").

ii) Effect of distribution breakdown, as another important piece of novelty during the sample selection stage, is studied via 3 AL scenarios: **1)** AL with MADBAL to show the benefits of the distribution breakdown, **2)** AL with modified MADBAL in which superpixels are assigned to random clusters while keeping the number of clusters the same. This is to observe how clustering superpixels based on their perception plays a role in detecting the most uncertain samples while keeping the diversity (denoted with "Random dist-breakdown"). **3)** AL with modified MADBAL which does not benefit from distribution breakdown at all. In this scenario N pixels with highest uncertainty scores in each image are queried for annotation (denoted with "No dist-breakdown").

We continue each AL progress until 90% performance of fully-supervised model ($0.9 \times 68.5 = 65.1\%$) is achieved. Fig. 6 depicts the results. As expected, MADBAL, with its fully extended features, achieves 90% performance with only 50 pixels per image owing to all the devised components. For the first ablation study, the second-best performance belongs to "Averaging," which matches the intuition as the algorithm benefits from the varied-maturity heads both for loss prediction and uncertainty score calculation; however, removing learnable weight maps adversely affects its performance compared to MADBAL. "Vanilla" and "No maturity-awareness" show the worst performances due to missing the critical components. Between the two, "Vanilla" is inferior as it does not benefit from the maturity-awareness nor from the loss prediction module. "No maturity-awareness," on the other hand, shows better performance owing to loss prediction module helping with better assessment of samples' uncertainty. For the second ablation, it is worth noticing that "Random dist-breakdown" is still showing a better performance compared to "No dist-breakdown." This

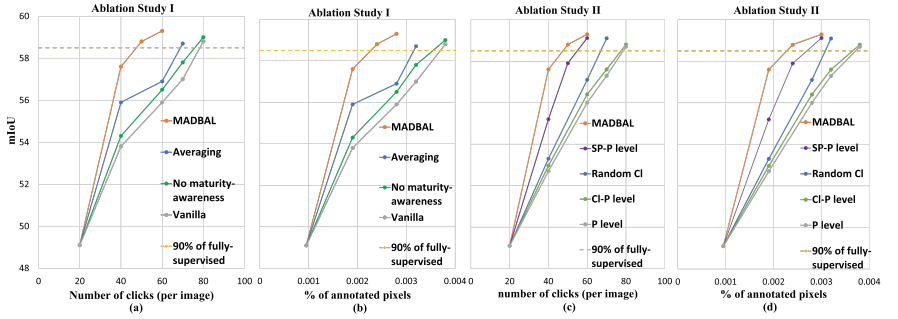

Figure 6: Ablation study on the effect of proposed components. (**a**), (**b**): the effect of maturity-awareness and loss prediction. (**c**), (**d**): the effect of distribution breakdown. When all the components are put into work the highest annotation efficiency is achieved. The more components are dropped, the more degradation on the efficiency is resulted.

can be attributed to the inevitable diversity the clustering (whether it be a perception-based clustering algorithm or random clustering) brings to the pulled samples in each round of sample selection. In other words, by grouping the superpixels, each of which corresponds roughly to an object class, we prevent "over-selection" of pixels belonging to the same object category in each step. Hence, despite its lower performance, "Random dist-breakdown" is still able to achieve higher performance than "No dist-breakdown."

5 Conclusion

In this work we proposed an active learning framework for semantic segmentation by integrating maturity-awareness and distribution breakdown. Maturity-awareness helps develop an effective understanding and recognition of the most critical pixels for performance improvement, while distribution breakdown provides a hierarchical approach to have an inclusive insight of the data distribution across different fields of view. Combined with a novel uncertainty formulation, the proposed MADBAL is shown to outperform many state of the art methods with significant margin. MADABAL can significantly reduce training burdens and also be impactful for tasks where annotation is expensive and not readily available.

References

[1] Hamed H Aghdam, Abel Gonzalez-Garcia, Joost van de Weijer, and Antonio M López. Active learning for deep detection neural networks. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 3672–3680, 2019.

[2] Jordan T Ash, Chicheng Zhang, Akshay Krishnamurthy, John Langford, and Alekh Agarwal. Deep batch active learning by diverse, uncertain gradient lower bounds. *arXiv preprint arXiv:1906.03671*, 2019.

[3] Michael Van den Bergh, Xavier Boix, Gemma Roig, Benjamin de Capitani, and Luc Van Gool. Seeds: Superpixels extracted via energy-driven sampling. In *European conference on computer vision*, pages 13–26. Springer, 2012.

[4] Lile Cai, Xun Xu, Jun Hao Liew, and Chuan Sheng Foo. Revisiting superpixels for active learning in semantic segmentation with realistic annotation costs. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10988–10997, 2021.

[5] Arantxa Casanova, Pedro O. Pinheiro, Negar Rostamzadeh, and Christopher J. Pal. Reinforced active learning for image segmentation. In *International Conference on Learning Representations*, 2020. URL https://openreview.net/forum?id=SkgC6TNFvr.

[6] Marius Cordts, Mohamed Omran, Sebastian Ramos, Timo Rehfeld, Markus Enzweiler, Rodrigo Benenson, Uwe Franke, Stefan Roth, and Bernt Schiele. The cityscapes dataset for semantic urban scene understanding. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3213–3223, 2016.

[7] Jifeng Dai, Kaiming He, and Jian Sun. Boxsup: Exploiting bounding boxes to supervise convolutional networks for semantic segmentation. In *Proceedings of the IEEE international conference on computer vision*, pages 1635–1643, 2015.

[8] Majed El Helou, Ruofan Zhou, Sabine Susstrunk, and Radu Timofte. Ntire 2021 depth guided image relighting challenge. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, pages 566–577, June 2021.

[9] Ehsan Elhamifar, Guillermo Sapiro, Allen Yang, and S Shankar Sasrty. A convex optimization framework for active learning. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 209–216, 2013.

[10] Mark Everingham, SM Eslami, Luc Van Gool, Christopher KI Williams, John Winn, and Andrew Zisserman. The pascal visual object classes challenge: A retrospective. *International journal of computer vision*, 111(1):98–136, 2015.

[11] Alexander Freytag, Erik Rodner, and Joachim Denzler. Selecting influential examples: Active learning with expected model output changes. In David Fleet, Tomas Pajdla, Bernt Schiele, and Tinne Tuytelaars, editors, *Computer Vision – ECCV 2014*, pages 562–577, Cham, 2014. Springer International Publishing. ISBN 978-3-319-10593-2.

[12] Yarin Gal, Riashat Islam, and Zoubin Ghahramani. Deep bayesian active learning with image data. In *International Conference on Machine Learning*, pages 1183–1192. PMLR, 2017.

[13] S Alireza Golestaneh and Kris M Kitani. Importance of self-consistency in active learning for semantic segmentation. *arXiv preprint arXiv:2008.01860*, 2020.

[14] Marc Gorriz, Axel Carlier, Emmanuel Faure, and Xavier Giro-i Nieto. Cost-effective active learning for melanoma segmentation. *arXiv preprint arXiv:1711.09168*, 2017.

[15] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[16] Neil Houlsby, Ferenc Huszár, Zoubin Ghahramani, and Máté Lengyel. Bayesian active learning for classification and preference learning. *arXiv preprint arXiv:1112.5745*, 2011.

[17] Andrew Howard, Mark Sandler, Grace Chu, Liang-Chieh Chen, Bo Chen, Mingxing Tan, Weijun Wang, Yukun Zhu, Ruoming Pang, Vijay Vasudevan, et al. Searching for mobilenetv3. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 1314–1324, 2019.

[18] Zilong Huang, Xinggang Wang, Jiasi Wang, Wenyu Liu, and Jingdong Wang. Weakly-supervised semantic segmentation network with deep seeded region growing. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 7014–7023, 2018.

[19] Ajay J. Joshi, Fatih Porikli, and Nikolaos Papanikolopoulos. Multi-class active learning for image classification. In *2009 IEEE Conference on Computer Vision and Pattern Recognition*, pages 2372–2379, 2009. doi: 10.1109/CVPR.2009.5206627.

[20] Andreas Kirsch, Joost van Amersfoort, and Yarin Gal. Batchbald: Efficient and diverse batch acquisition for deep bayesian active learning. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019. URL https://proceedings.neurips.cc/paper/2019/file/95323660ed2124450caaac2c46b5ed90-Paper.pdf.

[21] Ksenia Konyushkova, Raphael Sznitman, and Pascal Fua. Learning active learning from data. *Advances in neural information processing systems*, 30, 2017.

[22] Jungbeom Lee, Eunji Kim, Sungmin Lee, Jangho Lee, and Sungroh Yoon. Ficklenet: Weakly and semi-supervised semantic image segmentation using stochastic inference. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5267–5276, 2019.

[23] David D Lewis. A sequential algorithm for training text classifiers: Corrigendum and additional data. In *Acm Sigir Forum*, volume 29, pages 13–19. ACM New York, NY, USA, 1995.

[24] Kunpeng Li, Ziyan Wu, Kuan-Chuan Peng, Jan Ernst, and Yun Fu. Guided attention inference network. *IEEE transactions on pattern analysis and machine intelligence*, 42 (12):2996–3010, 2019.

[25] Xuelu Li and Vishal Monga. Group based deep shared feature learning for fine-grained image classification. *arXiv preprint arXiv:2004.01817*, 2020.

[26] Di Lin, Jifeng Dai, Jiaya Jia, Kaiming He, and Jian Sun. Scribblesup: Scribble-supervised convolutional networks for semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3159–3167, 2016.

[27] Wenrui Liu, Zongqing Lu, and He Xu. Auxiliary edge detection for semantic image segmentation. In *Proceedings of the 2020 6th International Conference on Computing and Artificial Intelligence*, ICCAI '20, 2020.

[28] Dengsheng Lu and Qihao Weng. A survey of image classification methods and techniques for improving classification performance. *International journal of Remote sensing*, 28(5):823–870, 2007.

[29] Radek Mackowiak, Philip Lenz, Omair Ghori, Ferran Diego, Oliver Lange, and Carsten Rother. Cereals-cost-effective region-based active learning for semantic segmentation. *arXiv preprint arXiv:1810.09726*, 2018.

[30] Shervin Minaee, Yuri Y Boykov, Fatih Porikli, Antonio J Plaza, Nasser Kehtarnavaz, and Demetri Terzopoulos. Image segmentation using deep learning: A survey. *IEEE transactions on pattern analysis and machine intelligence*, 2021.

[31] Hieu T Nguyen and Arnold Smeulders. Active learning using pre-clustering. In *Proceedings of the twenty-first international conference on Machine learning*, page 79, 2004.

[32] Yassine Ouali, Céline Hudelot, and Myriam Tami. Semi-supervised semantic segmentation with cross-consistency training. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12674–12684, 2020.

[33] Yu Qiao, Jincheng Zhu, Chengjiang Long, Zeyao Zhang, Yuxin Wang, Zhenjun Du, and Xin Yang. Cpral: Collaborative panoptic-regional active learning for semantic segmentation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, pages 2108–2116, 2022.

[34] Nicholas Roy and Andrew McCallum. Toward optimal active learning through monte carlo estimation of error reduction. *ICML, Williamstown*, 2:441–448, 2001.

[35] Mark Sandler, Andrew Howard, Menglong Zhu, Andrey Zhmoginov, and Liang-Chieh Chen. Mobilenetv2: Inverted residuals and linear bottlenecks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018.

[36] Ozan Sener and Silvio Savarese. Active learning for convolutional neural networks: A core-set approach. *arXiv preprint arXiv:1708.00489*, 2017.

[37] Burr Settles, Mark Craven, and Soumya Ray. Multiple-instance active learning. *Advances in neural information processing systems*, 20, 2007.

[38] Gyungin Shin, Weidi Xie, and Samuel Albanie. All you need are a few pixels: semantic segmentation with pixelpick. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 1687–1697, 2021.

[39] Yawar Siddiqui, Julien Valentin, and Matthias Nießner. Viewal: Active learning with viewpoint entropy for semantic segmentation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 9433–9443, 2020.

[40] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.

[41] Samarth Sinha, Sayna Ebrahimi, and Trevor Darrell. Variational adversarial active learning. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 5972–5981, 2019.

[42] Yunchao Wei, Huaxin Xiao, Honghui Shi, Zequn Jie, Jiashi Feng, and Thomas S Huang. Revisiting dilated convolution: A simple approach for weakly-and semi-supervised semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 7268–7277, 2018.

[43] Shuai Xie, Zunlei Feng, Ying Chen, Songtao Sun, Chao Ma, and Mingli Song. Deal: Difficulty-aware active learning for semantic segmentation. In *Proceedings of the Asian Conference on Computer Vision*, 2020.

[44] Lin Yang, Yizhe Zhang, Jianxu Chen, Siyuan Zhang, and Danny Z Chen. Suggestive annotation: A deep active learning framework for biomedical image segmentation. In *International conference on medical image computing and computer-assisted intervention*, pages 399–407. Springer, 2017.

[45] Amirsaeed Yazdani, Sumit Agrawal, Kerrick Johnstonbaugh, Sri-Rajasekhar Kothapalli, and Vishal Monga. Simultaneous denoising and localization network for photoacoustic target localization. *IEEE Transactions on Medical Imaging*, 40(9):2367–2379, 2021. doi: 10.1109/TMI.2021.3077187.

[46] Donggeun Yoo and In So Kweon. Learning loss for active learning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 93–102, 2019.

[47] Nan Zou, Zhiyu Xiang, Yiman Chen, Shuya Chen, and Chengyu Qiao. Boundary-aware cnn for semantic segmentation. *IEEE Access*, 7:114520–114528, 2019. doi: 10.1109/ACCESS.2019.2935816.