All you need are a few pixels: semantic segmentation with PIXELPICK Supplementary Material

Gyungin Shin[1] Weidi Xie[1] Samuel Albanie[1,2]
[1] Visual Geometry Group, University of Oxford, UK
[2] Department of Engineering, University of Cambridge, UK
{gyungin, weidi, albanie}@robots.ox.ac.uk
https://www.robots.ox.ac.uk/~vgg/research/pixelpick

A. Overview

In this supplementary material, we first give a more detailed description of the dataset used in this work (Sec. A.1) and training setting (Sec. A.2). Next, we present three additional studies: (i) an evaluation into the effect of varying the number of pixel coordinates sampled in each round of training (Sec. B); (ii) the influence of our proposed diversity heuristic (Sec. C), and (iii) the effectiveness of a human at selecting pixel coordinates in comparison to using model uncertainty (Sec. D). Finally, we present formal definitions of the employed acquisition functions (Sec. E), further experimental comparisons on PASCAL VOC 2012 (Sec. F) and detailed descriptions of methods we compared to in the main paper, which were omitted due to space constraints (Sec. G).

A.1. Datasets

CAMVID [2] is an urban scene segmentation dataset composed of 11 categories and containing 367, 101, and 233 images of 360×480 resolution for training, validation, and testing, respectively.

CITYSCAPES [4] is a dataset collected for the purpose of autonomous driving consisting of 2975 training, 500 validation and 1525 test high-resolution images (1024×2048) with 19 classes. During training, we resize the images to 256×512 pixels to make the training time manageable, and perform inference on images of 512×1024 pixels.

PASCAL VOC 2012 [7] (abbreviated to VOC12) contains 1464, 1449, and 1456 images for training, validation and testing respectively. Each pixel is labelled as one of the 20 semantic categories or background. Since images in this dataset have different sizes, during training we resize the larger image dimension to 400 and randomly crop a 320×320 patch as input, and use the original size for inference, following [17].

A.2. Optimisation and data augmentation

Optimisation. We use Adam [10] with a learning rate of 5×10^{-4} for the CAMVID and CITYSCAPES datasets, and SGD with momentum 0.9 and a learning rate of 10^{-2} for the PASCAL VOC 2012 dataset. For CAMVID, we train for 50 epochs and decay the learning rate at 20 and 40 epochs by a factor of 10. On CITYSCAPES and PASCAL VOC 2012, we use the poly learning rate schedule as in [17, 23, 3, 16].
Data augmentation. We largely follow [17], and use random scaling between [0.5, 2.0] and random horizontal flipping. In addition, we apply photometric transformations such as colour jittering, random grayscaling and Gaussian blurring.

B. Effect of the number of queried pixel coordinates per round

To understand how the number of labelled pixels added at each round affects the model's performance, we train MobileNetv2-based DeepLabv3+ models on PASCAL VOC 2012. Each model queries $n \in \{1, 2, 5, 10\}$ pixel(s) per image per round and the maximum budget is set to 30 pixels per image (in the notation employed in Sec. 3 of the paper, $n = B/N$ with $N = 1464$ for the PASCAL VOC 2012 dataset). All models are given random 1 pixel per image at the beginning of training. As shown in Fig. 1 (left), we note that when the annotation budget is very low (e.g., $\leq$ 10 pixels per image), a model with a lower n value shows a higher mIoU. However, when more annotations are allowed (e.g. ≥ 20 pixels per image), performance is similar across the models.

On the other hand, as the number of query rounds required to reach the max budget is inversely proportional to n, we also measure the GPU time for the models to complete the whole training process (Fig. 1, right).[1] We observe that, there is a trade-off between training time and n.

[1] We measure timings on a NVIDIA RTX2080ti GPU card.

For instance, to reach about 0.5 mIoU, the model has to be re-trained 6 times (corresponding to an annotation budget of 6 pixels per image) when $n = 1$, whereas one would only need to query once (corresponding to an annotation budget of 11 pixels per image), if $n = 10$, reducing the overall training time by a factor of 5.

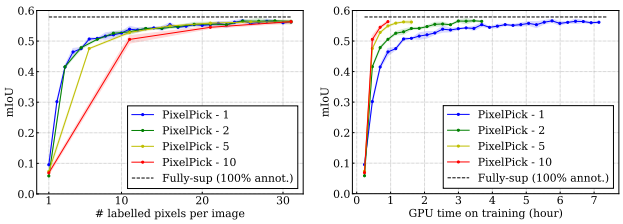

Figure 1: **Effect of the number of queried pixel coordinates per round on VOC12.** PIXELPICK-n denotes our model which samples n pixels per image per query round. Left: given a highly limited annotation budget (e.g., ≤ 10 pixels per image), we observe that it is beneficial to pick fewer pixels at each round to achieve a better label efficiency in terms of performance. Right: we show a trade-off between the number of queried pixels per round and total GPU training time taken to reach a certain level of performance.

C. Diversity heuristic

As noted in [1, 24], simply selecting samples with the highest uncertainty can result in poor performance due to a lack of diversity among samples. In our PIXELPICK framework, this manifests as querying pixels from a limited set of spatial regions, which is likely to incur redundant queries, and in turn degrades the labelling efficiency.

To alleviate this effect, [24] sub-sampled the unlabelled pool and chose the n-most uncertain samples from the resulting subset. We experiment with this approach by uniformly sampling 5% pixel coordinates within an image and then taking as queries the 10 most uncertain pixels amongst them at each query stage. Specifically, we train DeepLabv3+ models on CAMVID for 10 rounds, with 10 random labelled pixels per image given at the beginning of training. However, as shown in Fig. 2 (left, denoted by {MS, LC, ENT}-A), this heuristic does not show promising results compared to the random baseline (RAND) and the performance varies significantly depending on the sampling strategies. For example, choosing entropy (ENT-A) as the acquisition function yields a lower mIoU than RAND, whereas using margin sampling (MS-A) allows a better performance. We conjecture that this is because directly selecting n-most uncertain pixels from the uniformly sub-sampled unlabelled pixels still tends to collect from a few restricted regions (i.e. less diversity).

Instead, to gather queried pixels from more diverse objects, we propose in the paper to first sample 5% unlabelled pixels with highest uncertainty and uniformly select 10 pixels from the this subset (denoted as {MS, LC, ENT}-B in Fig. 2). Put differently, we swap the order of the uniform and uncertainty sampling processes. As can be seen in Fig. 2, the proposed approach brings better results and is robust to the choice of a uncertainty strategy in the pixel-level active learning setting.

To provide evidence for our hypothesis on diversity of the queried pixels, we compute the average number of unique categories for queried pixels within an image as an approximate diversity measure. As can be seen in Fig. 2 (right), ENT-A and LC-A, which show worse performance than the uniform sampling (RAND) at the end of AL, queried pixels from less diverse classes than RAND. On the other hand, methods with a higher mIoU queried from objects with greater category diversity than RAND, underpinning our hypothesis. We therefore use the proposed diversity heuristic throughout our experiments in the main paper.

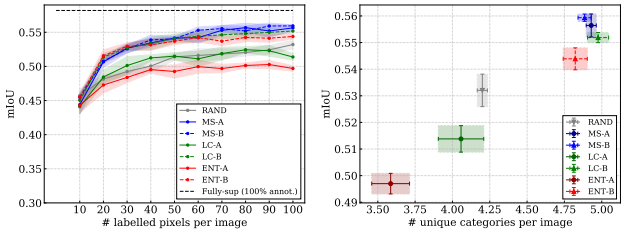

Figure 2: **Effect of diversity heuristic on CAMVID.** Left: we observe that directly selecting n-most uncertain pixels from randomly sub-sampled regions as in [24] within an image is sensitive to the choice of an acquisition function (denoted as {MS, LC, ENT}-A). In contrast, uniformly choosing n pixels per image from M% pixels with highest uncertainty is robust to the acquisition functions and shows better performance (denoted as {MS, LC, ENT}-B). Right: we show that the average class diversity per image covered by the queried pixel locations plays an important role in performance.

D. Human labelling oracle

To show it is beneficial to query labels from the model's perspective rather than a human annotator, we compare models trained with labelled pixels selected by one of the uncertainty sampling strategies and by a human annotator. For this, we train a MobileNetv2-based DeepLabv3+ on CAMVID, given 10 labelled pixels per image queried based on a sampling method and 10 random pixels per image initially offered at the beginning of AL (i.e. retrain after one query round). For human-picking, we ask one annotator to pick 10 pixels per image on CAMVID from the regions where the model makes wrong predictions, assuming humans can well recognise the groundtruth annotation from an image, and thus are able to easily validate the errors from

Sampling method	mean IoU (%)
Random	48.1 ± 0.5
Entropy	51.6 ± 0.9
Least Confidence	51.4 ± 0.5
Margin Sampling	50.8 ± 0.2
Human annotator	46.5 ± 0.4

Table 1: **Performance comparison between human-picked and uncertainty-based pixels on CamVid.**

the model prediction. The annotator was encouraged to pick pixel coordinates that they believe most useful for boosting segmentation performance from the annotator's view.

Interestingly, as shown in Tab. 1, we found the performance of the model trained on human-picked pixels is worse than any other uncertain-based strategies, even lower than the random baseline by 1.6 mIoU (%). We found this result surprising—our hypothesis is that human annotators tend to treat each image independently, and consequently tend not to take account of the differing degrees of visual variety present in each class (for example, "sky" pixels often look similar, but the "building" class can vary significantly in appearance and therefore requires more labels) whereas the model can determine this information readily (via its uncertainty) across the full training set. The result highlights a potential discrepancy between what really helps the model and what human annotators think useful for solving the task. A better understanding of the nuances underpinning this effect would be useful future work.

Method	Backbone	Training set imgs (anno. type)	mIoU
Semi-supervised methods			
WSSL [18]	VGG16	1.5K (dense) & 9K (classes)	64.6
GAIN [14]	VGG16	1.5K (dense) & 9K (classes)	60.5
MDC [22]	VGG16	1.5K (dense) & 9K (classes)	65.7
DSRG [9]	VGG16	1.5K (dense) & 9K (classes)	64.3
FickleNet [12]	VGG16	1.5K (dense) & 9K (classes)	65.8
BoxSup [6]	VGG16	1.5K (dense) & 9K (boxes)	63.5
CCT [17]	ResNet50	1.5K (dense) & 9K (classes)	69.4
Interactive weak supervision			
PixelPick (Ours)	ResNet50	1.5K (sparse pixels)	65.6

Table 2: **Comparison to semi-supervised methods on VOC12 validation set.** In the third column, we denote the number of training images with different annotation levels, *e.g.* classes, boxes, and dense represent image-, box-level, and per-pixel-level annotation, respectively.

E. Acquisition functions

Here we provide the formal definitions of the acquisition functions employed in the main paper. The notation below uses the variables introduced in Sec. 3.

The *Least Confidence* acquisition strategy [13, 5] draws, at each iteration, the pixel coordinate for which the model has *least confidence* in its *most likely* class label:

$$u^*_{LC} = \operatorname*{argmin}_{u \in \Omega} \operatorname*{argmax}_{c \in \{1,\dots,C\}} \hat{y}_u(c). \tag{1}$$

The *Margin Sampling* strategy [19] looks for samples that exhibit the smallest difference (i.e. lowest "margin") between the first and second most probable labels:

$$u^*_{MS} = \operatorname*{argmin}_{u \in \Omega} \Big(\operatorname*{argmax}_{c_1 \in \{1,\dots,C\}} \hat{y}_u(c_1) - \operatorname*{argmax2}_{c_2 \in \{1,\dots,C\}} \hat{y}_u(c_2) \Big), \tag{2}$$

where the notation $\operatorname{argmax} 2$ denotes the argument with the second largest value. Intuitively, pixel coordinates with small margins are ambiguous for the classifier, while those with large margins represent samples for which the classifier has greater confidence in its correctness.

The *Entropy Sampling* strategy aims to select the pixel coordinate with the greatest conditional entropy [21] under the current model:

$$u^*_{ENT} = \operatorname*{argmax}_{u \in \Omega} -\sum_{c=1}^{C} \hat{y}_u(c) \log \hat{y}_u(c). \tag{3}$$

F. Comparison to semi-supervised methods

In addition to comparing to weakly-supervised methods in Tab. 1 of the main paper, we also compare our work to semi-supervised models in Tab. 2. We train PixelPick using 1.5K sparsely labelled images (20 pixel labels per image) and compare against semi-supervised methods that train with 1.5K densely labelled images (i.e., 1.2×10^5 pixel labels per image, considering the average spatial resolution of 308×381) and 9K weak labels (classes or boxes). Despite using vastly fewer annotations, PixelPick performs competitively.

G. Methods description

To help readers understand the difference in the methods used for the comparison on Pascal VOC 2012 validation set in our paper, we categorise them according to annotation level they use (i.e., image-, box-, or scribble-level) and briefly summarise each method. We also describe CCT [17], which primarily addresses semi-supervised learning. All weakly-supervised methods train on VOC12 augmented by SBD [8] (10.5K images). When they consider semi-supervised setting jointly with their

weakly-supervised approach, they use the original VOC12 1.5K pixel-level annotations for full-supervision and the remaining 9K images for weak-supervision. By contrast, our PIXELPICK framework leverages sparse weak-supervision on the 1.5K VOC12 images.

- **Image-level annotation**
  - **WSSL** [18] adopts an EM-approach in which they estimate segmentation masks given observed image values and image-level labels in the E-step and optimise model parameters on the estimated segmentation in the M-step.
  - **GAIN** [14] proposes to use attention maps to enable a better quality of localisation maps for training a segmentation model. To this end, they train an image classification model with an additional attention mining loss to enforce the model to guide itself where to look. To validate their approach, they evaluate another weakly supervised segmentation model, SEC [11] trained on pseudo-segmentation masks generated by hard-thresholding their attention maps.
  - **MDC** [22] leverages image-level labels to produce pseudo segmentation masks. In particular, they propose to use a convolutional block with multiple dilated rates in order to transfer the discriminative object region to other parts of the object.
  - **DSRG** [9] uses image-level labels and a deep network pretrained on image classification to produce seed cues which a segmentation network is trained on. The seed cues are further extended to unlabelled pixels by the proposed region growing algorithm in an iterative manner.
  - **FickleNet** [12] generates localisation maps with a pretrained image classification network by saliency, which are further used as pseudo-labels to train a segmentation network. For this, they aggregate a variety of localisation maps, which of each is produced from a single image by applying stochastic hidden unit selection and Grad-CAM [20] and highlights different parts of objects present in the image.
- **Box-level annotation**
  - **BoxSup** [6] exploits bounding box annotations, which are much easier to obtain than dense pixelwise annotations, at a cost of offering weaker supervision. For this, they iteratively generate semantic masks by forming candidate segments with a unsupervised region proposal method and assigning a semantic label of a groundtruth box to the most overlapped segment and train deep networks on the estimated semantic masks.
- **Scribble-level annotation**
  - **ScribbleSup** [15] proposes to use scribble annotations and iterate over propagating them to unmarked regions by optimising a graphical model and training a segmentation model on the generated masks.
- **Semi-supervised approach**
  - **CCT** [17] utilises cross-consistency loss to take advantage of unlabelled data under the cluster assumption. For this, they enforce invariance between outputs of auxiliary decoders and main decoder, where the former takes a perturbed embedding from the encoder, and the latter receives clean features from the encoder. They train on VOC12 for the fully-supervised pixel-wise cross-entropy loss and on the images from [8] for the cross-consistency loss.

References

[1] William H. Beluch, Tim Genewein, A. Nürnberger, and J. Köhler. The power of ensembles for active learning in image classification. In *Proc. CVPR*, 2018. 2

[2] Gabriel J Brostow, Julien Fauqueur, and Roberto Cipolla. Semantic object classes in video: A high-definition ground truth database. *Pattern Recognition Letters*, 30(2):88–97, 2009. 1

[3] Liang-Chieh Chen, Yukun Zhu, George Papandreou, Florian Schroff, and Hartwig Adam. Encoder-decoder with atrous separable convolution for semantic image segmentation. In *ECCV*, 2018. 1

[4] 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. *CoRR*, abs/1604.01685, 2016. 1

[5] Aron Culotta and Andrew McCallum. Reducing labeling effort for structured prediction tasks. In *AAAI*, 2005. 3

[6] Jifeng Dai, Kaiming He, and Jian Sun. Boxsup: Exploiting bounding boxes to supervise convolutional networks for semantic segmentation. In *Proc. ICCV*, 2015. 3, 4

[7] Mark Everingham, S. M. Ali Eslami, Luc Van Gool, Christopher K. I. Williams, John M. Winn, and Andrew Zisserman. The pascal visual object classes challenge: A retrospective. *International Journal of Computer Vision*, 111(1):98–136, 2015. 1

[8] Bharath Hariharan, Pablo Arbeláez, Lubomir Bourdev, Subhransu Maji, and Jitendra Malik. Semantic contours from inverse detectors. In *Proc. ICCV*, 2011. 3, 4

[9] Zilong Huang, Xinggang Wang, Jiasi Wang, Wenyu Liu, and Jingdong Wang. Weakly-supervised semantic segmentation network with deep seeded region growing. In *Proc. CVPR*, 2018. 3, 4

[10] Diederik P. Kingma and Jimmy Ba. Adam: A Method for Stochastic Optimization. *arXiv e-prints*, page arXiv:1412.6980, Dec. 2014. 1

[11] Alexander Kolesnikov and Christoph H. Lampert. Seed, expand and constrain: Three principles for weakly-supervised image segmentation. In *Proc. ECCV*, 2016. 4

[12] Jungbeom Lee, Eunji Kim, Sungmin Lee, Jangho Lee, and Sungroh Yoon. Ficklenet: Weakly and semi-supervised semantic image segmentation using stochastic inference. In *Proc. CVPR*, 2019. 3, 4

[13] David D Lewis and Jason Catlett. Heterogeneous uncertainty sampling for supervised learning. In *Machine Learning Proceedings*. 1994. 3

[14] Kunpeng Li, Ziyan Wu, Kuan-Chuan Peng, Jan Ernst, and Yun Fu. Tell me where to look: Guided attention inference network. In *Proc. CVPR*, 2018. 3, 4

[15] Di Lin, Jifeng Dai, Jiaya Jia, Kaiming He, and Jian Sun. Scribblesup: Scribble-supervised convolutional networks for semantic segmentation. In *Proc. CVPR*, 2016. 4

[16] Wei Liu, Andrew Rabinovich, and Alexander C. Berg. ParseNet: Looking Wider to See Better. *arXiv e-prints*, page arXiv:1506.04579, June 2015. 1

[17] Yassine Ouali, Celine Hudelot, and Myriam Tami. Semi-supervised semantic segmentation with cross-consistency training. In *Proc. CVPR*, 2020. 1, 3, 4

[18] George Papandreou, Liang-Chieh Chen, Kevin P. Murphy, and Alan L. Yuille. Weakly- and semi-supervised learning of a deep convolutional network for semantic image segmentation. In *Proc. ICCV*, 2015. 3, 4

[19] T. Scheffer, Christian Decomain, and S. Wrobel. Active hidden markov models for information extraction. In *IDA*, 2001. 3

[20] Ramprasaath R. Selvaraju, Michael Cogswell, Abhishek Das, Ramakrishna Vedantam, Devi Parikh, and Dhruv Batra. Grad-cam: Visual explanations from deep networks via gradient-based localization. In *Proc. ICCV*, 2017. 4

[21] C. Shannon. A mathematical theory of communication. *Bell Syst. Tech. J.*, 27:379–423, 1948. 3

[22] 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 *Proc. CVPR*, 2018. 3, 4

[23] Shuai Xie, Zunlei Feng, Y. Chen, Songtao Sun, Chao Ma, and Ming-Li Song. Deal: Difficulty-aware active learning for semantic segmentation. In *ACCV*, 2020. 1

[24] Donggeun Yoo and I. Kweon. Learning loss for active learning. *Proc. CVPR*, 2019. 2