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

Gyungin Shin¹
Weidi Xie¹
Samuel Albanie¹²

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

Abstract

A central challenge for the task of semantic segmentation is the prohibitive cost of obtaining dense pixel-level annotations to supervise model training. In this work, we show that in order to achieve a good level of segmentation performance, all you need are a few well-chosen pixel labels.

We make the following contributions: (i) We investigate the semantic segmentation setting in which labels are supplied only at sparse pixel locations, and show that deep neural networks can use a handful of such labels to good effect; (ii) We demonstrate how to exploit this phenomenon within an active learning framework, termed PIXELPICK, to radically reduce labelling cost, and propose an efficient “mouse-free” annotation strategy to implement our approach; (iii) We conduct extensive experiments to study the influence of annotation diversity under a fixed budget, model pretraining, model capacity and the sampling mechanism for picking pixels in this low annotation regime; (iv) We provide comparisons to the existing state of the art in semantic segmentation with active learning and demonstrate comparable performance with up to two orders of magnitude fewer pixel annotations on the CAMVID, CITYSCAPES and PASCAL VOC 2012 benchmarks; (v) Finally, we evaluate the efficiency of our annotation pipeline and its sensitivity to annotator error to demonstrate its practicality.

1. Introduction

The coupling of deep neural networks and large-scale labelled datasets has yielded significant progress on a host of core machine perception tasks. A key challenge of training these models is their need for considerable quantities of annotation, which can be prohibitively expensive to collect for applications that require either specialised annotators such as medical image diagnostics [1, 23, 56, 62], or fine-grained labels, such as for detection and segmentation [36].

Semantic segmentation, in particular, has proven valuable for decision making in a variety of applications such as digital pathology [59], remote sensing [64] and autonomous driving [67]. However, its requirement of per-pixel annotations raises significant scalability challenges—on average more than 1.5 hours of annotation and quality control was required for each image in the CITYSCAPES segmentation dataset [13].

The objective of this work is to propose a simple yet effective approach for training a good semantic segmentation model at minimal annotation cost. Our approach is motivated by three observations: (1) Within a given image, pixels exhibit significant spatial mutual information; (2) Deep neural networks possess a strong inductive bias that renders them appropriate for modelling these spatial dependencies [60]; (3) Collecting mask, scribble or click annotations requires annotators to “localise and classify” using a mouse or trackpad. By contrast, assigning a class to a pixel pro-
posal can be “mouse-free”, requiring instead only a “classify” task without a localisation component (and which can be performed via a single key-press). The first two factors imply that densely labelling all pixels in images may be highly redundant, while the third suggests the possibility of designing an efficient sparse pixel labelling strategy. Several questions then arise: how many sparse pixel labels are needed to achieve good performance? how should those pixel locations be selected? and how can the selected pixels be annotated efficiently?

In this paper, we address these questions through the lens of active learning [2, 54]. In contrast to passive supervised learning (where the model is tasked to learn the mapping from a fixed set of input-output pairs), active learning considers a dynamic scenario in which a model can interactively request labels for the samples that it believes will be most useful for solving a given task. Our proposed PIXELPICK framework adopts this paradigm, learning a model for semantic segmentation by alternating between training on previously labelled pixels and requesting new labels.

We make the following contributions: (i) We study the problem setting in which labels are supplied at the level of sparse pixels and show that with only a small collection of such labels, modern deep neural networks can achieve good performance; (ii) We show how this phenomenon can be exploited with an efficient and practical “mouse-free” annotation strategy as part of a proposed PIXELPICK active learning framework; (iii) We perform a series of experiments into factors that affect model performance in the low-annotation regime: annotation diversity, architectural choices and the design of the sampling mechanisms for selecting most useful pixels; (iv) We compare with other state of the art active learning approaches on standard segmentation benchmarks: CAMVID, CITYSCAPES and PASCAL VOC 2012, where we demonstrate comparable segmentation performance with significantly lower annotation budget (Fig. 1); (v) Lastly, we assess PIXELPICK from the perspective of practical deployment, assessing its annotation efficiency and robustness.

2. Related work

Our work is related to several themes of research that have sought to minimise labelling costs for semantic segmentation, as discussed next.

Weakly-supervised semantic segmentation. Many weak supervisory signals have been explored in the literature as a pragmatic compromise between fully supervised [37] and fully unsupervised approaches to semantic segmentation [26]. These cues include scribbles [35], eye tracking [43], object pointing [3, 45, 12], web-queried samples [27], bounding boxes [15, 28, 58], extreme clicks for objects [44, 39] and image-level labels [70, 63, 17]. Differently from these approaches, we gather labels at sparse pixel locations proposed by the model itself, rather than at locations selected by the annotator, and show that very few such annotations are needed for good performance.

Interactive annotation. There is rich body of computer vision literature considering the related problem of accelerating interactive annotation. The seminal work of [6] demonstrated how to exploit scribbles to indicate the foreground/background appearance model and leverage graph-cuts for segmentation [5]. This was later extended to the use of multiple scribbles on both object and background, applied to annotating objects in videos [41]. [48] exploited 2D bounding boxes provided by the annotator and performed pixel-wise foreground/background labelling using EM. Recent work [10] tasks a model with sequentially producing the vertices of a polygon outlining an object, given an appropriate crop. As with the weakly-supervised signals described above, these methods are passive in the sense that the labelling process is driven by the human annotator, rather than the model.

Semi-supervised semantic segmentation. Inspired by classical self-labelling approaches which aim to leverage unlabelled data to improve a classifier [51, 68], a number of semi-supervised approaches have been developed to make use of pseudo-labelling algorithms [30] for semantic segmentation in a low-annotation regime. Consistency-based pseudo-labelling methods have recently demonstrated promising results, highlighting the important role of aggressive data augmentations when only a small number of densely annotated images or regions are available [42, 18].

Our approach differs from theirs in several ways: (i) our model is trained from sparse pixel annotations, rather than a small number of densely labelled images, (ii) we employ active learning (samples are dynamically selected and queried for annotation by the model), which, as we show through experiments, brings additional improvements. We compare our approach quantitatively with theirs in Sec. 4.3.

Active learning for semantic segmentation. At its core, active learning is a set selection problem; the aim being to determine the most informative subset of samples to acquire labels for, given a labelling budget [2, 54, 32, 19, 69]. In this case the maximally informative labelled-pixel subset is the one which yields the lowest generalisation error when used to train a supervised semantic segmentation model. Prior work targeting segmentation has investigated strategies to select superpixels that induce the maximum label change for a CRF on the training set by using weak (image-level category) supervision [61], incorporate geometric constraints [29, 40] and propagate foreground masks to large-scale image collections [25]. For foreground segmentation of medical imagery, FCNs [37] have been coupled with bootstrapping [66], and U-Nets [47] with dropout-
based Monte Carlo estimates of uncertainty [22] to drive label acquisition via uncertainty sampling. The strategy of learning an estimator for difficult regions [69] has proven effective as a basis for selecting which images should be densely labelled for semantic segmentation [65]. Adversarial learning has also been explored to query labels that align distributions of labelled and unlabelled images [57].

More closely related to our work, prior studies have considered region-based sampling strategies for semantic segmentation, employing reinforcement learning [9], equivariance constraints [21] and learned estimators of labelling cost [38]. In contrast to these lines of research, our work aims to introduce a more efficient paradigm of active learning for segmentation, which is to train models by only querying sparse pixel annotations (removing the localisation component of the annotation task). Concurrent to our work, Cai et al. [8] explore the use of super-pixels with active learning. We compare our work with theirs and with other region-based sampling strategies noted above in Sec. 4.

3. Method

In this section, we describe the problem formulation and introduce our framework for pixel-level active learning semantic segmentation in Sec. 3.1. We then detail our mouse-free annotation tool to efficiently implement the framework in Sec. 3.2.

3.1. PIXELPick framework

We seek to train a model for semantic segmentation with pool-based active learning [53], in which we alternate between training a model on available annotation and requesting labels for unlabelled samples from an oracle (see Fig. 2).

More formally, let $\mathcal{X} \subset \mathbb{R}^{H \times W \times 3}$ denote the space of colour images and let $\Phi(\cdot; \Theta) : \mathcal{X} \rightarrow \mathcal{Y}^{H \times W}$ represent a ConvNet with parameters $\Theta$ that maps a given image to a grid of elements in a $C$-class semantic label space (here $\mathcal{Y}$ denotes the $(C - 1)$-simplex, i.e. $\mathcal{Y} = \{(p_1, \ldots, p_C) \in [0, 1]^C : \sum_{i=1}^{C} p_i = 1, p_i \geq 0\}$). We assume access to an initial unlabelled pool of $N$ images, $\mathcal{D}_U$, indexed by the $H \times W \times N$ pixel coordinate lattice, $\Omega$, and an annotation database, $\mathcal{D}^0_L$, initialised to an empty state.

At the $k^{th}$ round of learning, a batch of $B \in \mathbb{N}$ pixel coordinates, $\omega_k \subset \Omega$, are sampled by an annotation function, $A$, using the predictions of the model trained in the previous round, $\Phi(\cdot; \Theta_{k-1})$, on the unlabelled pool $\mathcal{D}_U$, i.e. $A(\mathcal{D}_U, \Phi(\cdot; \Theta_{k-1})) = \omega_k \subset \Omega$. The sampled pixel coordinates $\omega_k$ are then sent to an oracle for labelling to produce a corresponding set of one-hot labels $\{y_{u} \in \mathcal{Y} : u \in \omega_k\}$ that are added to the latest version of the annotation database, $\mathcal{D}^{k-1}_L$. Finally, the model is retrained on this expanded database, $\mathcal{D}^k_L = \bigcup_{i=1}^k \{(u, y_u) : u \in \omega_i\}$ (comprising all annotations gathered so far), to produce a new model, $\Phi(\cdot; \Theta_k)$, and the process is repeated. We term this framework PIXELPick due to its emphasis on selecting appropriate pixels for annotation. The two components of the framework, namely retraining the segmentation model and sampling new pixel coordinates, are discussed next.

Retraining the segmentation model. At round $k$ of the active learning algorithm, we solve for parameters $\Theta_k$ by minimising a cross-entropy loss at each labelled pixel coordinate present in the current annotation database $\mathcal{D}^k_L$:

$$\Theta_k = \arg\min_{\Theta} \mathcal{L}(\Theta, \mathcal{D}^k_L)$$

where

$$\mathcal{L}(\Theta, \mathcal{D}^k_L) = -\frac{1}{|\mathcal{D}^k_L|} \sum_{(u, y_u) \in \mathcal{D}^k_L} \sum_{c=1}^C y_{u}(c) \cdot \log(y_{u}(c)).$$

In the expression above, $y_{u}(c)$ and and $\hat{y}_{u}(c)$ denote the $c^{th}$ channel of the oracle-provided label and corresponding model prediction at pixel coordinate $u$, respectively.

Sampling new pixel coordinates for labelling. The objective of the acquisition function, $A$, is to sample the $B$ pixel locations at round $k$ that maximise improvement in segmentation performance for the current model $\Phi(\cdot; \Theta_{k-1})$. Functionally, it acts by examining the predictions of $\Phi(\cdot; \Theta_{k-1})$ across all candidate pixel coordinates among the unlabelled pool $\mathcal{D}_u$ and sampling $B$ such coordinates according to a specified criterion.

Discussion. The distinction between sampling contiguous spatial patches for annotation (e.g. grids of 128x128 pixels or larger as considered in prior work [38, 21, 9]), and sampling individual pixel coordinates, as proposed within the PIXELPick framework, is a subtle but important one. It has two key benefits. The first, as noted in Sec. 1, is that it allows us to leverage the powerful inductive biases provided by deep neural network architectures that render them well suited to modelling spatial dependencies in natural images [60]. The second is a practical one: by providing annotators with pixel coordinate proposals, the labelling process is transformed from a “localise and classify” task (required when segmenting semantic regions and typically performed with a mouse or trackpad), into simply a “classify” task in which a class label is assigned to a coordinate proposal, and which can often be performed with a single key-press. We validate both claims through experiments in Sec. 4, where we show that (i) deep neural networks achieve strong segmentation performance at extremely sparse annotation levels, (ii) “mouse-free” annotation can be performed very efficiently.

Acquisition functions. The design of the specific criteria employed by the acquisition function has been the subject of considerable attention in the active learning literature (see [53] and [46] for surveys of classical and recent
approaches, respectively). Since the focus of our work is not the design of another criterion, but rather on the effectiveness of individual pixels as the base unit for annotation, we consider several existing approaches based on the framework of uncertainty sampling [32] that have been noted as effective in the literature, discussed next.

The Least Confidence acquisition strategy [33, 14] draws, at each iteration, the pixel coordinate for which the model has least confidence in its most likely class label. The Margin Sampling strategy [50] looks for samples that exhibit the smallest difference (i.e. lowest “margin”) between the first and second most probable labels. Finally, the Entropy Sampling strategy aims to select the pixel coordinate with the greatest conditional entropy [55] under the current model. Formal mathematical descriptions of each method are given in the supplementary.

As noted in prior work [4, 69], these strategies can suffer from a lack of diversity if applied naively, but can be readily adapted to minimise this effect by first sub-sampling the unlabelled pool and then employing the acquisition function to choose only from this restricted subset. A variation of this diversity heuristic worked well on our task: We first rank all pixels using the acquisition function, then uniformly sampling $B/N$ pixel coordinates from the top $M\%$ ranked locations in each image, where $M$ is a hyperparameter and $N$ denotes the number of images we distribute our budget $B$ amongst. We note that while more sophisticated strategies (e.g. [52]) could also be considered within our framework, a simple Margin Sampling strategy coupled with the modification described above proved effective (shown through experiments in Sec. 4), and thus we adopt it in this work.

Sampling batches. The number of pixel coordinates sampled in each round, $B$, is set as a hyperparameter. A larger value of $B$ corresponds to fewer rounds of annotation (and therefore a potentially faster deployment cycle), at some cost in performance. A detailed study of the effects of $B$ is provided in the supplementary.

3.2. PIXELPick Annotation tool

To demonstrate the practical utility of the PIXELPick framework, we created an annotation tool to support the labelling process (Fig. 3). The tool is simple: for each image, the annotator is presented with a few pixels that were selected by the PIXELPick acquisition function (described in Sec. 3.1). They are also shown a mapping from keyboard keys to semantic labels (Fig. 3, right hand side). The tool iterates over the pixel locations, highlighting the current pixel in red and the annotator simply presses the appropriate key to classify it. The tool then moves on to the next pixel proposal, and the procedure repeats until all proposals in the image are exhausted, when a new image is shown.

We note that an important difference between this annotation technique and those considered in prior work (e.g.
In this section, we explore the effect of four factors that affect performance in the PIXELPICK framework, with a particular focus on annotation diversity (with the goal of finding the most effective way to spend an annotation budget); encoder depth (varying the capacity of the encoder); encoder initialisation (self-supervised vs supervised pretraining); and acquisition function (determining the best way to select pixels). Note that, while investigating the first three factors, all pixels are selected via simple uniform random sampling, with the goal of validating the effectiveness of inductive bias in modern ConvNets. We simulate the active learning process, following standard practice [21, 65, 9], i.e. to label the queried pixels, we simply reveal labels by querying the ground truth annotations at their spatial coordinates.

**Annotation diversity.** Given a fixed pixel labelling budget (B pixels), a natural question arises: *is it better to label a small number of images densely or a large number of images sparsely?* To address this question we design a simple experiment, where a fixed annotation budget of n pixels is to be distributed over a dataset of $N_{\text{total}}$ images. We define the annotation diversity ratio, $\eta = \frac{N_{\text{img}}}{N_{\text{total}}}$, where $N_{\text{img}}$ refers to the number of images that have had at least one pixel labelled (for simplicity, we assume the labelling budget is evenly distributed over the selected set of images). Therefore, $\eta \to 1$ refers to a budget uniformly distributed over the full dataset (thereby forming a sparse, but diverse, label set), $\eta \to 0$ denotes the case where the budget is only spent on a few images (yielding a densely annotated subset of images). We then train DeepLabv3+ models on CAMVID and CITYSCAPES, fixing $B$ so as to end up with 10 pixel labels per image when $\eta = 1$, and experiment with 5 different diversity ratios $\eta$ from 0.01 to 1.0. In Fig. 5(a), we observe that mean IoU increases monotonically with $\eta$. This indicates that, given a fixed budget, it is better to sparsely annotate as many images as possible, rather than a smaller number more densely, motivating our sparse PIXELPICK approach. In the remaining experiments, we likewise spend our annotation budget evenly across all images within a dataset (as described in Sec. 3.1), with each image being only sparsely labelled.

**Encoder depth.** We next investigate the effect of encoder capacity in the low annotation regime. Specifically, we...
experiment with a ResNet-based FPN by changing the number of layers in the encoder from 18 to 101 layers. All encoders are initialised with a model pretrained for classification on ImageNet [49]. We conduct experiments both on CAMVID (training each model with 1 to 100 randomly labelled pixel coordinates per image) and VOC12 (training each model with 1 to 1000 randomly sampled labelled pixel coordinates per image), reporting results in Fig. 4(a) and Fig. 4(b), respectively. We observe that deeper networks yield higher performance above a minimum number of labelled pixels (approximately 10) per image. This implies that, at the cost of greater computational complexity, the use of a deeper network may be a viable way to reduce annotation requirements in low annotation regimes (above some minimum labelling threshold).

**Encoder initialisation.** Next, we investigate whether supervised pretraining is necessary for good segmentation performance in a low annotation regime. Concretely, we compare the performance of an FPN-based architecture with a ResNet50 encoder that is initialised using either supervised (ImageNet classification) or self-supervised (MoCov2 [11]) pretraining. To study how performance differs with the number of labelled pixels, we vary the annotation budget from 1 to $10^4$ randomly sampled labelled pixels per image on CAMVID (Fig. 4(c)) and VOC12 (Fig. 4(d)). On CAMVID, we observe an interesting biphasic phenomenon: when the number of labelled pixels per image is fewer than 10, the model initialised with supervised ResNet50 shows superior performance. However, as the number of pixel labels increases, self-supervised pretraining gradually outperforms its supervised counterpart. This phenomenon is also observed on the VOC12 dataset, with a cross-over occurring at approximately $10^2$ labelled pixels per image. Thus, supervised pretraining may be an appropriate choice for low annotation budgets, when suitable pretraining annotations are readily available, but its advantage wanes the annotation budget grows. Given its superiority at low annotation levels, we adopt supervised pretraining for the remaining experiments.

**Acquisition function.** Thus far, we have only labelled pixels selected via simple uniform random sampling, showing that modern CNNs—with their strong inductive biases—can be trained for semantic segmentation with just a handful of pixel annotations per image. Here, we go one step further, investigating whether a better choice of acquisition function can further improve learning efficiency. To this end, we experiment on CAMVID with three popular uncertainty sampling methods (described in Sec. 3.1): Least Confidence (LC), Margin Sampling (MS) and Entropy Sampling (ENT). In addition, we also experiment with a Query-By-Committee (QBC) [54] approach that queries labels using model ensembles [53]. We implement this with dropout after each convolutional layer, repeating inference 20 times to obtain a Monte Carlo estimate following [20]. Due to the large number of models to be trained (i.e., different acquisition functions, each trained five times to estimate variance), we employ the lightweight MobileNetv2-based DeepLabv3+ model. We initialise training with 10 uniform randomly selected labelled pixels per image. Once training converges, we query 10 additional pixel labels with the given acquisition function. As described in Sec. 3.1, we first take top $M\%$ ranked pixels (here, $M = 5$) per image under the uncertainty estimation ranking and uniformly sample 10 pixels from these pixels. Fig. 5(b) shows the results. We see that all uncertainty-based methods outperform the random baseline in every round. Interestingly, dropout-based voting variants of LC, MS and ENT each show worse performance than their counterparts voting—a similar observation was also made in [9]. We note that in our problem setting, Margin Sampling (MS) outperforms other strategies, reaching about 96% of the performance of the fully supervised baseline with only 100 pixels per image (0.06% of the annotations). Therefore, we use MS as our sampling method for PIXELPICK to compare against...
Figure 5: Ablation studies. In (a), we observe that sparsely annotating a larger number of images (higher $\eta$ value) outperforms denser labelling of fewer images, with consistent trends on the CAMVID and CITYSCAPES datasets. In (b), we compare acquisition functions on CAMVID and find that Margin Sampling performs best. In (c), we investigate the sensitivity of the PIXELPICK framework to annotator errors by simulating a pixel classification user error (SUE) rate of 10%. We observe that performance is only marginally affected, indicating the practical robustness of the PIXELPICK framework.

Figure 6: Comparison to state-of-the-art and qualitative results. In (a) we observe that PIXELPICK performs favourably against existing state-of-the-art approaches for active learning and semi-supervised learning on CITYSCAPES. In (b) we show qualitative results. With only 10 labelled pixels per image, segmentation models trained with PIXELPICK achieve promising visual quality, which further improves to capture fine details (e.g. the cleanly segmented thin lamppost in the bottom right image) as further labelled pixels are used.

Discussion. To summarise, we can draw the following conclusions from the ablation studies: First, given a fixed pixel annotation budget, it is best to spread it over as many images as possible; Second, the inductive bias in modern ConvNets makes them well-suited to capture local correlations within an image, evidenced by the first three experiments, where models trained with randomly sampled pixels still perform well; Third, although it might be thought that deeper networks with greater capacity would suffer significantly from over-fitting in the low-annotation regime, we found that for many budget choices, deeper networks are the preferred option. Fourth, in terms of acquisition functions, active learning outperforms random sampling, and in particular, Margin Sampling performs best in our setting.

4.3. Comparison to state of the art methods

We next validate our framework by comparing against prior work in active/semi-supervised learning on CAMVID (Fig. 1) and CITYSCAPES (Fig. 6(a)) and in weakly-supervised learning on PASCAL VOC 2012 (Tab. 1). To strike a balance between computation complexity and performance, we adopt the FPN model with a ResNet50 backbone, and query additional samples each round with Margin Sampling, as suggested by the ablation study. We train for 10 query rounds, with each round adding 10 labelled pixels per image for CAMVID and CITYSCAPES and 5 pixels for VOC12.

Comparison to region-based active learning and semi-supervised methods. In Fig. 1 and Fig. 6(a), we observe that our approach performs favourably to existing semi-supervised and region-based active learning approaches in terms of label efficiency on both CAMVID
Table 1: Comparison to existing weakly-supervised methods on VOC12 validation set. PixelPick is competitive against existing methods, using a budget of 20 pixel annotations per image when trained on a much smaller number of images.

<table>
<thead>
<tr>
<th>Method</th>
<th>Backbone</th>
<th>Train set (anno. type)</th>
<th>mIoU</th>
</tr>
</thead>
<tbody>
<tr>
<td>GAIN [34]</td>
<td>VGG16</td>
<td>10.5K imgs (classes)</td>
<td>55.3</td>
</tr>
<tr>
<td>MDC [63]</td>
<td>VGG16</td>
<td>10.5K imgs (classes)</td>
<td>60.4</td>
</tr>
<tr>
<td>DSRG [24]</td>
<td>ResNet101</td>
<td>10.5K imgs (classes)</td>
<td>61.4</td>
</tr>
<tr>
<td>FickleNet [31]</td>
<td>ResNet101</td>
<td>10.5K imgs (classes)</td>
<td>64.9</td>
</tr>
<tr>
<td>BoxSup [15]</td>
<td>VGG16</td>
<td>10.5K imgs (boxes)</td>
<td>62.0</td>
</tr>
<tr>
<td>ScribbleSup [35]</td>
<td>VGG16</td>
<td>10.5K imgs (scribbles)</td>
<td>63.1</td>
</tr>
</tbody>
</table>

Interactive weak supervision

<table>
<thead>
<tr>
<th>Method</th>
<th>Train set (anno. type)</th>
<th>mIoU</th>
</tr>
</thead>
<tbody>
<tr>
<td>PixelPick (Ours, ResNet50)</td>
<td>1.5K imgs (sparse pixels)</td>
<td>65.6</td>
</tr>
</tbody>
</table>

Figure 7: A comparison based on estimated annotation cost. We compare PixelPick against the concurrent work of [8] in terms of mIoU ratio vs annotation cost approximated by the amount of clicks on (left) Cityscapes and (right) VOC12 datasets.

4.4. Practical deployment

Thus far, we have largely followed the common practice in previous active learning segmentation work, mimicking the labelling process by simply disclosing the corresponding labels from the fully-annotated dataset. In this section, we evaluate the efficiency of PixelPick (Fig. 3) and its sensitivity to annotator noise during model training.

In detail, we ask eleven annotators to label 100 images from VOC12 dataset, with 10 pixels per image, we measure the average time and accuracy (between annotator and the groundtruth from original dataset). As a result, with our simple unoptimised annotation tool, it takes 1.42 seconds on average to label the queried pixel (14s per image), with 87.7% average accuracy. To our knowledge, this annotation speed is significantly faster than drawing bounding boxes or scribbles [15, 35], and approximately twice as fast as picking extreme points according to times reported by [44]. Additionally, given the observed annotation error rate, we conduct an experiment to assess the influence of these noisy annotations, that is, we artificially jitter 10% of the groundtruth annotations to simulate errors during the annotation process and train a model on pixels containing this label noise. As shown in Fig. 5(c), the performance gap incurred from annotation noise is negligible, indicating that our framework is not only efficient with respect to annotation time but also robust to potential errors caused by annotators. This is an encouraging sign for the practical potential of the PixelPick framework for real-world deployment.

5. Conclusion

In this work we proposed PixelPick, a framework for semantic segmentation that employs a small number of sparsely annotated pixels to train effective segmentation models. We showed that PixelPick requires considerably fewer annotations than existing state-of-the-art to achieve comparable performance. Finally, we showed how annotation for pixel-level active learning can be obtained efficiently with a mouse-free labelling tool, facilitating real-world deployment. We hope that our work encourages further research into the promising use of sparse pixel-level annotation for image understanding.

Acknowledgements. GS is supported by AI Factory, Inc. in Korea. WX and SA are supported by Visual AI (EP/T028572/1). The authors would like to thank Tom Gunter for suggestions. SA would also like to thank Z. Novak and S. Carlson for support.
References


[53] Burr Settles. Active learning literature survey. 2009. 3, 6


