Full or Weak annotations? An adaptive strategy for budget-constrained annotation campaigns

Javier Gamazo Tejero[1], Martin S. Zinkernagel[2], Sebastian Wolf[2]
Raphael Sznitman[1], Pablo Márquez Neila[1]

[1]University of Bern, [2]Inselspital Bern, Switzerland

{javier.gamazo-tejero, raphael.sznitman, pablo.marquez}@unibe.ch

{martin.zinkernagel, sebastian.wolf}@insel.ch

Abstract

Annotating new datasets for machine learning tasks is tedious, time-consuming, and costly. For segmentation applications, the burden is particularly high as manual delineations of relevant image content are often extremely expensive or can only be done by experts with domain-specific knowledge. Thanks to developments in transfer learning and training with weak supervision, segmentation models can now also greatly benefit from annotations of different kinds. However, for any new domain application looking to use weak supervision, the dataset builder still needs to define a strategy to distribute full segmentation and other weak annotations. Doing so is challenging, however, as it is a priori unknown how to distribute an annotation budget for a given new dataset. To this end, we propose a novel approach to determine annotation strategies for segmentation datasets, whereby estimating what proportion of segmentation and classification annotations should be collected given a fixed budget. To do so, our method sequentially determines proportions of segmentation and classification annotations to collect for budget-fractions by modeling the expected improvement of the final segmentation model. We show in our experiments that our approach yields annotations that perform very close to the optimal for a number of different annotation budgets and datasets.

1. Introduction

Semantic segmentation is a fundamental computer vision task with applications in numerous domains such as autonomous driving [11, 43], scene understanding [45], surveillance [50] and medical diagnosis [9, 18]. As the advent of deep learning has significantly advanced the state-of-the-art, many new application areas have come to light and continue to do so too. This growth has brought and continues to bring exciting domain-specific datasets for segmentation tasks [6, 19, 29, 32, 52].

Today, the process of establishing machine learning-based segmentation models for any new application is relatively well understood and standard. Only once an image dataset is gathered and curated, can machine learning models be trained and validated. In contrast, building appropriate datasets is known to be difficult, time-consuming, and yet paramount. Beyond the fact that collecting images can be tedious, a far more challenging task is producing ground-truth segmentation annotations to subsequently train (semi) supervised machine learning models. This is mainly because producing segmentation annotations often remains a manual task. As reported in [4], generating segmentation annotations for a single PASCAL image [15] takes over 200 seconds on average. This implies over 250 hours of annotation time for a dataset containing a modest 5'000 images. What often further exacerbates the problem for domain-specific datasets is that only the dataset designer, or a small group of individuals, have enough expertise to produce the annotations (*e.g.*, doctors, experts, etc.), making crowd-sourcing ill-suited.

To overcome this challenge, different paradigms have been suggested over the years. Approaches such as Active Learning [7, 8, 26] aim to iteratively identify subsets of images to annotate so as to yield highly performing models. Transfer learning has also proved to be an important tool in reducing annotation tasks [13, 17, 24, 25, 30, 36]. For instance, [37] show that training segmentation models from scratch is often inferior to using pre-training models derived from large image classification datasets, even when the target application domain differs from the source domain. Finally, weakly-supervised methods [2, 40] combine pixel-wise annotations with other weak annotations that are faster to acquire, thereby reducing the annotation burden.

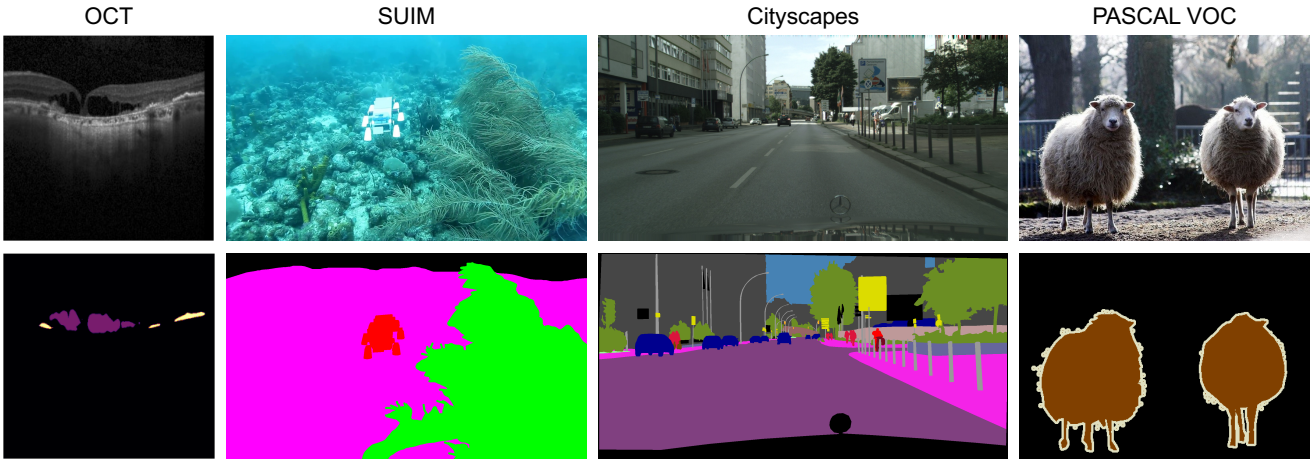

Figure 1. Illustration of different semantic segmentation applications; OCT: Pathologies of the eye in OCT images, SUIM: Underwater scene segmentation [19], Cityscape: street level scene segmentation [11], PASCAL VOC: natural object segmentation.

In particular, Papandreou *et al*. [40] showed that combinations of strong and weak annotations (*e.g*., bounding boxes, keypoints, or image-level tags) delivered competitive results with a reduced annotation effort. In this work, we rely on these observations and focus on the weakly supervised segmentation setting.

In the frame of designing annotation campaigns, weakly-supervised approaches present opportunities for efficiency as well. Instead of completely spending a budget on a few expensive annotations, weakly-supervised methods allow a proportion of the budget to be allocated to inexpensive, or weak, labels. That is, one could spend the entire annotation budget to manually segment available images, but would ultimately lead to relatively few annotations. Conversely, weak annotations such as image-level labels are roughly 100 times cheaper to gather than their segmentation counterparts [4]. Thus, a greater number of weakly-annotated images could be used to train segmentation models at an equal cost. In fact, under a fixed budget, allocating a proportion of the budget to inexpensive image-level class labels has been shown to yield superior performance compared to entirely allocating a budget to segmentation labels [4].

Yet, allocating how an annotation budget should be distributed among strong and weak annotations is challenging, and inappropriate allocations may severely impact the quality of the final segmentation model. For example, spending the entire budget on image-level annotations will clearly hurt the performance of a subsequent segmentation model. Instead, a naive solution would be to segment and classify a fixed proportion of each (*e.g*., say 80% - 20%). Knowing what proportion to use for a given dataset is unclear, however. Beyond this, there is no reason why the same fixed proportion would be appropriate across different datasets or application domains. That is, it would be highly unlikely that the datasets shown in Fig. 1 all require the same proportion of strong and weak annotations to yield optimal segmentation models.

Despite its importance, choosing the best proportion of annotation types remains a largely unexplored research question. Weakly-supervised and transfer-learning methods generally assume that the annotation campaign and the model training are independent and that all annotations are simply available at training time. While active learning methods do alternate between annotation and training, they focus on choosing optimal samples to annotate rather than choosing the right type of annotations. Moreover, most active learning methods ignore constraints imposed by an annotation budget. More notable, however, is the recent work of Mahmood *et. al.* [33, 34] which aims to determine what weak and strong annotation strategy is necessary to achieve a target performance level. While noteworthy, this objective differs from that here, whereby given a fixed budget, what strategy is best suited for a given new dataset?

To this end, we propose a novel method to find an optimal budget allocation strategy in an online manner. Using a collection of unlabeled images and a maximum budget, our approach selects strong and weak annotations, constrained by a given budget, that maximize the performance of the subsequent trained segmentation model. To do this, our method iteratively alternates between partial budget allocations, label acquisition, and model training. At each step, we use the annotations performed so far to train multiple models to estimate how different proportions of weak and strong annotations affect model performance. A Gaussian Process models these results and maps the number of weak and strong annotations to the expected model improvement. Computing the Pareto optima between expected improvement and costs, we choose a new sub-budget installment

and its associated allocation so to yield the maximum expected improvement. We show in our experiments that our approach is beneficial for a broad range of datasets, and illustrate that our dynamic strategy allows for high performances, close to optimal fixed strategies that cannot be determined beforehand.

2. Related work

2.1. Weak annotations for segmentation

Weakly supervised semantic segmentation (WSSS) relies on coarser annotations, such as bounding boxes [46], scribbles [31, 49] or image-level classification labels [1], to train a segmentation network. WSSS methods have often employed saliency maps as weak annotations for segmentation models, as these are typically obtained from CAM [55], which leverages image-level classification annotation. These methods then focus on refining the saliency maps with a variety of techniques [16, 28]. Others make use of attention to achieve coarse segmentations [20, 23]. Conversely, [54] combined annotations in the form of bounding boxes and image-level labels to accurately generate image graphs, to be used by a graph neural network to predict node values corresponding to pixel labels. In this context, the work in [33] and [34] are close to this one, whereby their objective is to determine what annotation strategy over annotation types is likely to yield a target performance level.

2.2. Transfer learning

Due to the limited availability of annotated image data in some domains, it is now common to use neural networks pre-trained on large image classification tasks [12] for subsequent target tasks. Specifically, in cases where the target task has limited data or annotations, this has been shown to be particularly advantageous. Among others, this practice is now widely used in medical imaging and has been linked to important performance gains after fine-tuning [13, 14, 24, 36, 48].

Efforts are now pivoting towards the use of in-domain pre-training, avoiding the leap of faith that is often taken with Imagenet [17, 30]. In [30], the model is pre-trained on ChestX-ray14 [51] to more accurately detect pneumonia in chest X-ray images from children. In [17], the authors show that joint classification and segmentation training, along with pre-training on other medical datasets that have domain similarity, increases segmentation performances with respect to the segmentation using Imagenet-based pre-training.

Alternatively, cross-task methods seek to transfer features learned on one task (*e.g.* classification, normal estimation, etc.) to another, usually more complex one. Along this line, Taskonomy [53] explored transfer learning capabilities among a number of semantic tasks and built a task similarity tree that provided a clustered view of how much information is available when transferring to other tasks. Similarly, [37] performed an extensive study of cross-task transfer capabilities for a variety of datasets, reaching the conclusion that Imagenet pre-training outperforms random initialization in all cases, but further training on related tasks or domains also brings additional benefits.

2.3. Active learning

In active learning, the goal is to train a model while querying an oracle to label new samples that are expected to improve the model's accuracy. In computer vision, it has been applied to image classification [22, 41] or semantic segmentation [3, 5, 44] among others. As a byproduct, Active learning has also been used as a way to reduce labeling time. For example, [27] describes a method that couples Reinforcement Learning and Active Learning to derive the shortest sequence of annotation actions that will lead to object detection within an image. Others have focused on speeding up this process via eye-tracking [38] or extreme clicking [39]. As such, Active Learning is related to the present work in the sense that our approach is adaptive but differs in that our method determines what annotations types should be collected under a constrained budget instead of predicting at each time step which samples should be added to the annotated set.

3. Method

Training segmentation models using a combination of expensive pixel-wise annotations and other types of cheaper annotations, such as image-wise labels or single-pixel annotations is known to be beneficial, as well as using cross-task transfer learning techniques [37]. This is motivated by empirical findings showing that, under a limited annotation budget, allocating a proportion of the budget to inexpensive image-level class labels led to superior performance compared to allocating the budget entirely to segmentation labels [4]. However, the optimal proportion of the budget to allocate per annotation type is a-priori unknown beforehand and data-dependent. Thus, the goal of our method is to find this data-specific optimal budget allocation in an online manner, as it is necessary for any dataset builder starting off.

We describe our method in the subsequent sections. For clarity, we focus on image segmentation and assume two kinds of annotations are possible: strong annotations as segmentation labels and weak annotations as image-level classification labels. Generalizing this formulation to other tasks or settings with more than two annotations types should follow directly.

3.1. Problem formulation

Let $p_{\text{data}}(\mathbf{x})$ be the distribution of training images for which we have no annotations initially. Each training im-

age $\mathbf{x}$ can be annotated with a pixel-wise segmentation labeling $(\mathbf{x}, \mathbf{y}) \sim p_{\text{data}}(\mathbf{x}) p_{\text{sgm}}(\mathbf{y} \mid \mathbf{x})$ or an image-wise classification annotation $(\mathbf{x}, c) \sim p_{\text{data}}(\mathbf{x}) p_{\text{cls}}(c \mid \mathbf{x})$Sampling from the distributions p_{cls} and p_{sgm} represents the task of manually annotating the image and has associated costs of $\alpha_{\text{c}} > 0$ and $\alpha_{\text{s}} > 0$, respectively. Supported by previous work [4, 33, 37], we will assume that $\alpha_{\text{s}} \gg \alpha_{\text{c}}$.

By sampling C classifications from p_{cls} and S segmentation from p_{sgm}, we can build an annotated training dataset $\mathcal{T} = (\mathcal{T}_c, \mathcal{T}_s) \sim (p_{\text{cls}}^C, p_{\text{sgm}}^S)$. The dataset $\mathcal{T}$ then has an annotation cost,

$$\alpha_{\text{c}} C + \alpha_{\text{s}} S, \tag{1}$$

which we assume to be bounded by an upper limit, or *budget*, B.

To annotate $\mathcal{T}$, however, we can choose different *allocation strategies*, or combinations of C and S, that have different costs and that yield different segmentation model performances. The utility u of an allocation strategy (C, S) is the expected performance of a model trained with datasets that follow that strategy,

$$u(C, S) = \mathbb{E}_{(\mathcal{T}_c, \mathcal{T}_s) \sim (p_{\text{cls}}^C, p_{\text{sgm}}^S)} \left[m(\mathcal{T}_c, \mathcal{T}_s) \right], \tag{2}$$

where $m(\mathcal{T}_c, \mathcal{T}_s)$ is the performance score (*e.g.*, Dice score, IoU) of a segmentation model trained with datasets $(\mathcal{T}_c, \mathcal{T}_s)$ and evaluated on a separate fixed test dataset. Note that in contrast to Active Learning, the utility is defined over the set of strategies (C, S) and not over the individual samples of a fixed training set. This is motivated by our aim to estimate the performance of the annotation strategy (C, S) and not the ensuing specific training dataset.

Our goal then is to find the annotation strategy that maximizes the expected performance constrained to a budget B,

$$\begin{aligned} \max_{(C,S) \in \mathbb{N}^2} \quad & u(C, S), \\ \text{s.t.} \quad & \alpha_c C + \alpha_s S \leq B. \end{aligned} \tag{3}$$

In the following, we describe how we optimize Eq. (3).

3.2. Utility model

As defined Eq. (2), the utility function, u, marginalizes over all possible training sets, which is intractable to compute in practice. To overcome this computational challenge, we approximate u with a collection $\mathcal{M}$ of discrete samples, where each sample $m \in \mathcal{M}$ is a tuple containing an allocation strategy (C, S) and the estimated score $m(\mathcal{T}_c, \mathcal{T}_s)$ obtained for a dataset sampled with that allocation strategy. To build $\mathcal{M}$, one could simply sample a random strategy (C', S'), annotate a dataset $(\mathcal{T}_c', \mathcal{T}_s') \sim (p_{\text{cls}}^{C'}, p_{\text{sgm}}^{S'})$, and measure its performance. However, this would imply annotating for different potential budgets and is thus infeasible in practice. Instead, a practical alternative is to leverage previously annotated data $(\mathcal{T}_c, \mathcal{T}_s)$. For each sampled strategy (C', S'), we build the corresponding dataset $(\mathcal{T}_c', \mathcal{T}_s')$ by taking random samples from the already annotated data according to the strategy. While this procedure, formalized in Alg. 1, leads to biased samples, we empirically found this bias to have a minor impact on the final strategies compared to estimations with unbiased sampling.

Algorithm 1 Build utility samples from annotated data

function BUILDUTILITYSAMPLES($\mathcal{T}_c, \mathcal{T}_s$)
  $C \leftarrow |\mathcal{T}_c|$, $S \leftarrow |\mathcal{T}_s|$
  $\mathcal{M} \leftarrow \{((C, S), m(\mathcal{T}_c, \mathcal{T}_s))\}$ ▷ Add sample with all the available data
  repeat $M - 1$ **times**
    Sample $(C', S') \in [0, C] \times [0, S]$
    $\mathcal{T}_c' \leftarrow \{C'$ elements sampled from $\mathcal{T}_c\}$
    $\mathcal{T}_s' \leftarrow \{S'$ elements sampled from $\mathcal{T}_s\}$
    $\mathcal{M} \leftarrow \mathcal{M} \cup ((C', S'), m(\mathcal{T}_c', \mathcal{T}_s'))$
  end repeat
end function

While $\mathcal{M}$ provides an estimation of u as a set of discrete locations, we generalize these estimations to the entire space of strategies by fitting a Gaussian process (GP) to the samples in $\mathcal{M}$. The Gaussian process, $\mathcal{GP}(\mu, k)$ is parameterized by a suitable mean function μ and covariance function k. In our case, we use the mean function,

$$\mu(C, S) = \gamma_c \log(\beta_c C + 1) + \gamma_s \log(\beta_s S + 1), \tag{4}$$

which accounts for the fact that the segmentation performance increases logarithmically with the volume of the training data [47] and that each annotation type has a different rate of performance growth. Similarly, the covariance k is a combination of two RBF kernels with different scales ℓ_c, ℓ_s for each annotation type,

$$k\left((C, S), (C', S')\right) = \sigma^2 e^{-\frac{(C-C')^2}{2\ell_c^2}} e^{-\frac{(S-S')^2}{2\ell_s^2}}. \tag{5}$$

The values γ_c, β_c, γ_s, β_s from the mean, the length scales ℓ_c, ℓ_s and the amplitude σ from the covariance are trainable parameters of the GP.

The trained GP models a distribution over utility functions, $u \sim \mathcal{GP}(\mu, k)$, that are plausible under the samples $\mathcal{M}$. This distribution represents not only the expected utility, but also its uncertainty in different areas of the strategy space. Sampling just a single u from the GP to solve Eq. (3) would thus be suboptimal. For this reason, we substitute the utility u in Eq. (3) by a surrogate function $\hat{u}$ that trades-off exploitation and exploration, thus incorporating uncertainty information into the optimization problem. Following a Bayesian optimization approach [21], we choose $\hat{u}$ to be the expected improvement (EI),

$$\hat{u}(C, S) = \mathbb{E}_{u \sim \mathcal{GP}_t}[\max\{u(C, S) - m^*, 0\}], \tag{6}$$

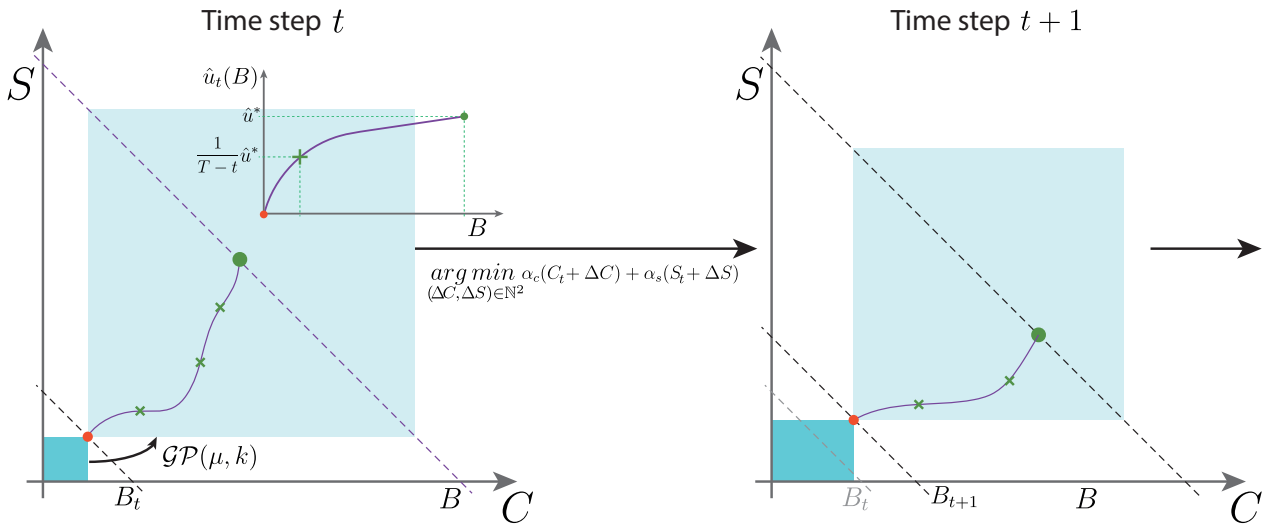

Figure 2. Illustration of proposed method. At a given iteration t, C_t and S_t classification and segmentation annotations have already been collected (blue region, left panel) with a budget of B_t. For the next annotation phase, the budget is increased to B_{t+1}. To determine how many new classification and segmentation annotations to collect, M combinations of different quantities $(C^{(i)}, S^{(i)})$ are gathered according to Alg. 1 to compute $m(C^{(i)}, S^{(i)})$. A Gaussian Process is then trained to estimate the utility of different combinations of annotation types (light blue area, left panel). From this, we infer ΔC and ΔS to select next by computing the combination that maximizes the expected improvement along the Pareto front given by the budget B_2 (red point, left panel). The next iteration starts then with the new proportions (red point, right panel) and follows the same steps (see text and Alg. 2 for details). For illustration purposes, the costs are set here to $\alpha_c = \alpha_s = 1$.

where m^* is the current maximum point.

3.3. Optimization

Training the GP requires annotated data to build the set $\mathcal{M}$, which in turn relies on an annotation strategy that we are trying to find, whereby implying a circular dependency. We address this circular dependency by optimizing Eq. (3) in an iterative manner.

Our algorithm shown in Alg. 2, allocates the available budget B in a fixed number of adaptive installments, alternating between data annotation with the current strategy, GP fitting, and strategy selection for the next budget installment. More specifically, our method starts with an initial strategy (C_0, S_0) with associated cost B_0. At each iteration t, new data is annotated according to the current strategy (C_t, S_t) so that the sets of annotated data $(\mathcal{T}_c, \mathcal{T}_s)$ contain C_t classification and S_t segmentation annotations, respectively. From the available annotated data $(\mathcal{T}_c, \mathcal{T}_s)$, we extract new samples for $\mathcal{M}$ and fit the GP, which defines the surrogate function $\hat{u}_t$. The corresponding current maximal point m_t^* is set to be the maximum performance found so far, (*i.e.*, the performance of the model trained with all the annotated data available at this iteration), $m_t^* = m(\mathcal{T}_c, \mathcal{T}_s)$. Finally, this surrogate function is used to estimate the next best strategy (C_{t+1}, S_{t+1}). We find a delta strategy $(\Delta C, \Delta S)$ that increases the expected improvement by a fixed fraction of its maximum possible value,

$$\begin{aligned} \underset{(\Delta C,\Delta S)\in\mathbb{N}^2}{\arg\min} \quad & \alpha_c(C_t+\Delta C)+\alpha_s(S_t+\Delta S), \\ \text{s.t.} \quad & \hat{u}_t(C_t+\Delta C, S_t+\Delta S) \geq \frac{1}{T-t}\hat{u}_t^*, \end{aligned} \tag{7}$$

where T is the desired maximum number of iterations of the algorithm and $\hat{u}_t^*$ is the maximum expected improvement that can be reached using the entire budget B for the current surrogate function $\hat{u}_t$ according to Eq. (3). The found delta strategy defines the new strategy $(C_{t+1}, S_{t+1}) = (C_t + \Delta C, S_t + \Delta S)$ for the next iteration. The process is depicted in Fig. 2.

Note that solving Eq. (7) requires finding $\hat{u}_t^*$, which in turn requires solving Eq. (3). While solving two optimization problems may seem unnecessary, the solutions of both problems are in the Pareto front of strategies (*i.e.*, the set of non-dominated strategies for which no other strategy has simultaneously smaller cost and larger or equal expected improvement). Given that the space of strategies is discrete, the elements of the Pareto front can be easily found in linear time by enumerating all possible strategies, computing their costs and expected improvements with $\hat{u}_t$, and discarding the dominated elements. Given the Pareto front, the strategy with the maximum EI u_t^* and the strategy of minimum budget with EI larger than $\frac{1}{T-t}\hat{u}_t^*$, which correspond to the solutions of Eq.(3) and Eq. (7), respectively, can be found in linear time.

4. Experimental setup

To validate our approach, we evaluated it on four different datasets, while comparing its performance to a set of typical fixed budget allocation strategies. In addition, we explore the impact of different hyper parameters on the overall performance of the method.

4.1. Datasets

We chose a collection of datasets with different image modalities, including a medical dataset as they often suffer from data and annotation scarcity. In this context, they represent a typical new application domain where our method could be particularly helpful. In each case, we enumerate the number of images for which classification or segmentation images can be sampled by a method:

Augmented PASCAL VOC 2012 [15]: 5'717 classification and 10'582 segmentation natural images with 21 classes for training. The validation sets contain 1'449 segmented images.

SUIM [19]: training set consists of 1'525 underwater images with annotations for 8 classes. For evaluation, we used a separate split of 110 additional images. The classification labels were estimated from the segmentation ground-truth as a multi-label problem by setting the class label to 1 if the segmentation map contained at least one pixel assigned to that class.

Cityscapes [11]: 2'975 annotated images for both classification and segmentation are available for training. We test on the official Cityscapes validation set, which contains 500 images.

OCT: 22'723 Optical Coherence Tomography (OCT) images with classification annotations and 1,002 images with pixel-wise annotations corresponding to 4 different types of retinal fluid for segmentation. We split the data into 902 training images and 100 test images.

Algorithm 2 Proposed approach

Input: Number of iterations T, initial labelling strategy (C_0, S_0)

$t \leftarrow 0$, $\Delta C \leftarrow C_0$, $\Delta S \leftarrow S_0$, $\mathcal{T}_c = \emptyset$, $\mathcal{T}_s = \emptyset$, $\mathcal{M} = \emptyset$
while $t < T$ **do**
  Annotate new data $(\Delta\mathcal{T}_c, \Delta\mathcal{T}_s) \sim (p_{\text{cls}}^{\Delta C}, p_{\text{sgm}}^{\Delta S})$
  $\mathcal{T}_c \leftarrow \mathcal{T}_c \cup \Delta\mathcal{T}_c$, $\mathcal{T}_s \leftarrow \mathcal{T}_s \cup \Delta\mathcal{T}_s$ ▷ Note that $|\mathcal{T}_c| = C_t$ and $|\mathcal{T}_s| = S_t$
  $\mathcal{M} \leftarrow \mathcal{M} \cup \text{BUILDUTILITYSAMPLES}(\mathcal{T}_c, \mathcal{T}_s)$
  Train GP with samples in $\mathcal{M}$
  Compute $(\Delta C, \Delta S)$ from Eq. (7)
  $C_{t+1} \leftarrow C_t + \Delta C$, $S_{t+1} \leftarrow S_t + \Delta S$
  $t \leftarrow t + 1$
end while

Output: (C_T, S_T)

4.2. Baseline strategies.

We compared our method to ten different *fixed* budget allocation strategies. Each of these randomly sample images for classification and segmentation annotations according to a specified and fixed proportion. We denote these policies by the percentage dedicated to segmentation annotations: B_0: $50\%, 55\%, \ldots, 95\%$ with increases in 5%. For fair comparison, the strategies are computed from the budget B_0.

In addition, we consider an *estimated-best-fixed* budget allocation strategy, whereby the method estimates what fixed budget should be used for a given dataset. This is done by using the initial budget B_0 to compute the best performing fixed strategy (mentioned above) and then using this fixed strategy for the annotation campaign until budget B is reached. This strategy represents an individual that chooses to explore all fixed strategies for an initial small budget and then exploit it.

4.3. Implementation details.

Weakly supervised segmentation model: To train a segmentation model that uses both segmentation and classifications, we first train the models with the weakly-annotated data $\mathcal{T}_c$ until convergence and then with the segmentation data $\mathcal{T}_s$. We use the U-Net segmentation model [42] for OCT, and the DeepLabv3 model [10] with a ResNet50 backbone on the SUIM, PASCAL, and Cityscapes. For the U-Net, a classification head is appended at the end of the encoding module for the classification task. For the DeepLab-like models, we train the entire backbone on the classification task and then add the ASPP head for segmentation. In all cases, we use the cross-entropy loss for classification and the average of the Dice loss and the cross-Entropy loss for segmentation. While we choose this training strategy for its simplicity, other cross-task or weakly supervised alternatives could have been used as well [2,40]. Additional details are provided in the supplementary materials.

Note that all models are randomly initialized to maximize the impact of classification labels, as Imagenet-pretraining shares a high resemblance to images in PASCAL and Cityscapes. Failing to do so would lead to classification training not adding significant information and may even hurt performance due to catastrophic forgetting [35].

Hyperparameters: We measured costs in terms of class-label equivalents setting $\alpha_c = 1$ and leaving only α_s as a hyperparameter of our method. We set $\alpha_s = 12$ for all datasets following previous studies on crowdsourced annotations [4]. We predict the first GP surface with 8% of

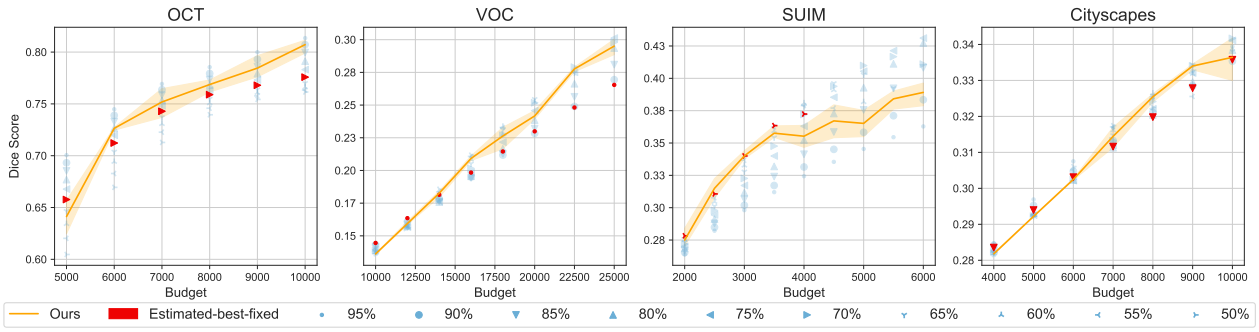

Figure 3. Performance of our method (orange line) on OCT, PASCAL VOC, SUIM and Cityscapes datasets. Shaded region is computed from three seeds. Fixed strategies are shown in blue. Red points show the *estimated-best-fixed* strategy with B_0. Labels expressed as percentage of the budget allocated to segmentation. Note that the first budget B fulfills $B \gg B_0$ in all cases.

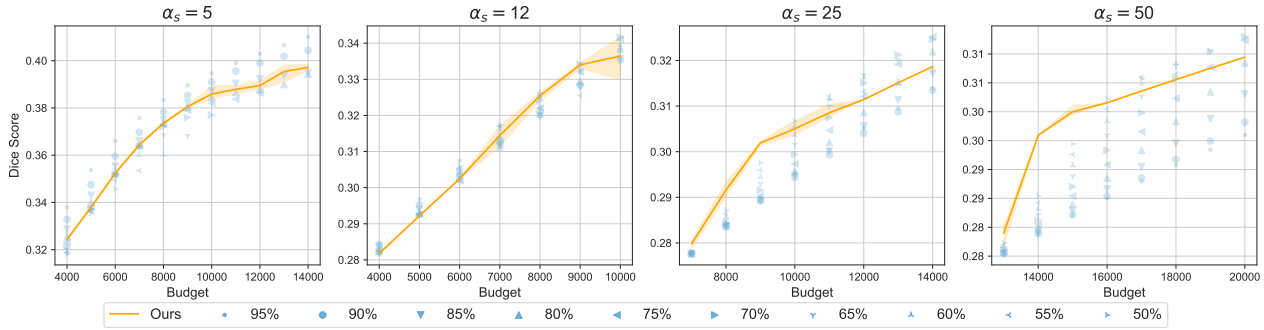

Figure 4. Mean of our method with $\alpha_s = \{5, 12, 25, 50\}$ on Cityscapes (orange, line). Shaded region is computed from three seeds. Fixed strategies are shown in blue. Labels expressed as percentage of the budget allocated to segmentation.

the dataset for both classification and segmentation. This quantity is reduced for OCT classification and VOC segmentation due to the high number of labels available. In all cases, we fixed the number of iterative steps to 8 and set the learning rate of the GP to 0.1.

5. Results

Main results: Figure 3 compares the performance achieved by our method against that of the different fixed strategies and the estimated best fixed strategy when using $\alpha_s = 12$ across the different datasets. From these results we can make a number of key observations.

First, we can observe that no single fixed strategy is performing optimally across the different datasets evaluated. This is coherent with our initial claims and with the literature. Indeed, for OCT the best strategy appears to be one that samples 90% of segmentations, while this same policy performs poorly on the SUIM dataset. This implies that blindly using a fixed policy would on average not be very effective.

Second, the estimated best-fixed strategy (in red) appears to do well initially and progressively loses competitiveness as the budget increases. This behaviour is expected as the estimated fixed strategy is that with B_0 (the lowest budget),

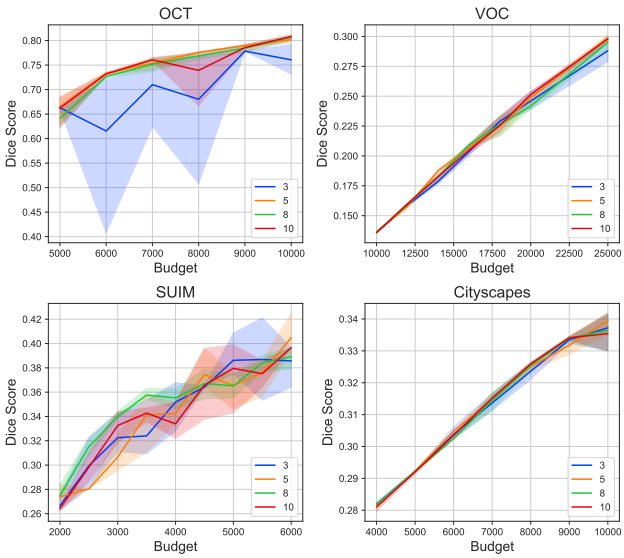

Figure 5. Mean of our method when using different numbers of iteration steps $\{3, 5, 8, 10\}$. Results shown with three seeds.

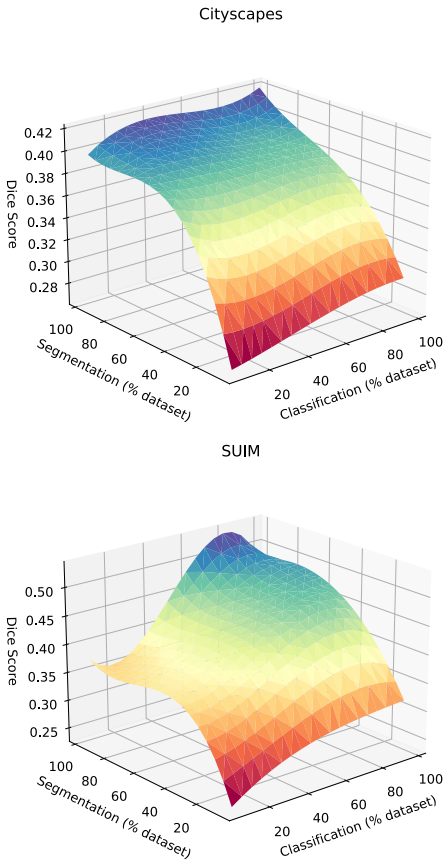

Figure 6. Cityscapes (top) and SUIM (bottom) ground truth budget-segmentation surfaces. We note that segmentation performance grows logarithmically with training set size on Cityscapes (as well as OCT and VOC, see the Supplementary materials). This trend is not observed on the SUIM dataset.

and becomes increasingly irrelevant as B grows. This is particularly clear on VOC where the best low-budget strategy allocates 95% of the budget to segmentation and still achieves superior performance up to $B = 12'000$. However, that strategy drops below average performance with budgets greater than $B = 25'000$. In the case of SUIM, the best-fixed strategy corresponds to 50% of the budget allocated to segmentation. Since the dataset contains only 1,525 segmentation samples, this strategy is not attainable with $B > 4000$.

Last, we can observe that our method is consistently able to produce good performances, across both different budget quantities and datasets. We can also clearly see that our strategy is not guaranteed to be the top performing strategy, but that on average it performs well in different cases.

At the same time, we notice that the performance of our approach on SUIM begins well and then drops after a 3'500 budget. This differs sharply from the other datasets. By observing the true budget-performance surface of SUIM and the other datasets (see Fig. 6), we can see that the SUIM surface does not grow logarithmically with the dataset size, while it does for Cityscapes (and the other too, see the Supplementary materials). This is relevant as our GP mean prior (4) assumes this relationship and explains why our approach fails when the true surface deviates from our GP mean form. While the use of adaptive, higher-level order priors would be beneficial to deal with such cases, we leave this as future work to be researched.

5.1. Sensitivity to α_s and T

Different types of annotations or domains may have different ratios of cost. While we have fixed α_s in our experiments across all datasets regardless of their domain, some datasets such as OCT and VOC require different expertise and domain knowledge to annotate and thus different α_s. In Fig. 4, we three additional values of $\alpha_s = \{5, 12, 25, 50\}$ and show the performance implication it has on our methods and the baselines. For Cityscapes, we see that the method is robust regardless of the value of α_s, showing above average performance especially for $\alpha_s = 25$ and $\alpha_s = 50$. This behavior is reproduced in all four datasets (see Supplementary materials).

Similarly, the number of steps T given to reach the final budget is a hyperparameter of our approach. While low T values could lead to poor solutions due to the unreliability of the GP far from the sampled region, higher T values (*i.e.*, therefore smaller steps) may exacerbate the intrinsic greedy nature of our method. We thus seek a trade-off between reliability and greediness. To study the sensitivity of the algorithm with respect to this variable, we show the behaviour of our method with different number of steps in Fig. 5. We see that lower T values greatly affect the reliability of the found strategy, especially for OCT and SUIM (blue line). However, as the number of steps increases, the variance of the strategy reduces sharply. We can therefore conclude that the method is robust to this hyperparameter as long as it is kept within reasonable ranges.

6. Conclusion

In this paper, we propose a novel approach to determine a dynamic annotation strategy for building segmentation datasets. We design an iterative process that identifies efficient dataset-specific combinations of weak annotations in the form of image-level labels and full segmentations. We show in our experiments that the best strategies are often dataset and budget-dependent, and therefore the trivial approaches do not always produce the best results. Our method however is capable of adapting to different image domains and finds combinations of annotations that reach high-performance levels. We show our method is robust to a number of hyperparameters and that it offers a good option for allocating annotation strategies.

References

[1] Jiwoon Ahn, Sunghyun Cho, and Suha Kwak. Weakly supervised learning of instance segmentation with inter-pixel relations. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 2209–2218, 2019. 3

[2] Jiwoon Ahn and Suha Kwak. Learning pixel-level semantic affinity with image-level supervision for weakly supervised semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4981–4990, 2018. 1, 6

[3] Mykhaylo Andriluka, Jasper R R Uijlings, and Vittorio Ferrari. Fluid annotation: a human-machine collaboration interface for full image annotation. In *Proceedings of the 26th ACM international conference on Multimedia*, pages 1957–1966, 2018. 3

[4] Amy Bearman, Olga Russakovsky, Vittorio Ferrari, and Li Fei-Fei. What's the point: Semantic segmentation with point supervision. In Bastian Leibe, Jiri Matas, Nicu Sebe, and Max Welling, editors, *Computer Vision – ECCV 2016*, pages 549–565, Cham, 2016. Springer International Publishing. 1, 2, 3, 4, 6

[5] Rodrigo Benenson, Stefan Popov, and Vittorio Ferrari. Large-Scale Interactive Object Segmentation With Human Annotators. *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 11692–11701, 2019. 3

[6] Sebastian Bodenstedt, Max Allan, Anthony Agustinos, Xiaofei Du, Luis C. García-Peraza-Herrera, Hannes Kenngott, Thomas Kurmann, Beat P. Müller-Stich, Sébastien Ourselin, Daniil Pakhomov, Raphael Sznitman, Marvin Teichmann, Martin Thoma, Tom Vercauteren, Sandrine Voros, Martin Wagner, Pamela Wochner, Lena Maier-Hein, Danail Stoyanov, and Stefanie Speidel. Comparative evaluation of instrument segmentation and tracking methods in minimally invasive surgery. *ArXiv*, abs/1805.02475, 2018. 1

[7] Lile Cai, Xun Xu, Jun Hao Liew, and Chuan Sheng Foo. Revisiting superpixels for active learning in semantic segmentation with realistic annotation costs. In *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 10983–10992, 2021. 1

[8] Arantxa Casanova, Pedro O. Pinheiro, Negar Rostamzadeh, and Christopher J. Pal. Reinforced active learning for image segmentation. In *International Conference on Learning Representations*, 2020. 1

[9] Chen Chen, Chen Qin, Huaqi Qiu, Giacomo Tarroni, Jinming Duan, Wenjia Bai, and Daniel Rueckert. Deep learning for cardiac image segmentation: a review. *Frontiers in Cardiovascular Medicine*, page 25, 2020. 1

[10] Liang-Chieh Chen, George Papandreou, Florian Schroff, and Hartwig Adam. Rethinking atrous convolution for semantic image segmentation. *arXiv preprint arXiv:1706.05587*, 2017. 6

[11] 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. 1, 2, 6

[12] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255. Ieee, 2009. 3

[13] Yiming Ding, Jae Ho Sohn, Michael G Kawczynski, Hari Trivedi, Roy Harnish, Nathaniel W Jenkins, Dmytro Lituiev, Timothy P Copeland, Mariam S Aboian, Carina Mari Aparici, Spencer C Behr, Robert R Flavell, Shih-Ying Huang, Kelly A Zalocusky, Lorenzo Nardo, Youngho Seo, Randall A Hawkins, Miguel Hernandez Pampaloni, Dexter Hadley, and Benjamin L Franc. A Deep Learning Model to Predict a Diagnosis of Alzheimer Disease by Using 18F-FDG PET of the Brain. *Radiology*, 290(2):456–464, 2019. 1, 3

[14] Andre Esteva, Brett Kuprel, Roberto A Novoa, Justin Ko, Susan M Swetter, Helen M Blau, and Sebastian Thrun. Dermatologist-level classification of skin cancer with deep neural networks. *Nature*, 542(7639):115–118, 2017. 3

[15] M Everingham, L Van˜Gool, C K I Williams, J Winn, and A Zisserman. The PASCAL Visual Object Classes Challenge 2012 VOC2012 Results. http://www.pascal-network.org/challenges/VOC/voc2012/workshop/index.html. 1, 6

[16] Junsong Fan, Zhaoxiang Zhang, Chunfeng Song, and Tieniu Tan. Learning integral objects with intra-class discriminator for weakly-supervised semantic segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4283–4292, 2020. 3

[17] Michal Heker and Hayit Greenspan. Joint liver lesion segmentation and classification via transfer learning. *arXiv preprint arXiv:2004.12352*, 2020. 1, 3

[18] Mohammad Hesam Hesamian, Wenjing Jia, Xiangjian He, and Paul Kennedy. Deep learning techniques for medical image segmentation: achievements and challenges. *Journal of digital imaging*, 32(4):582–596, 2019. 1

[19] Md Jahidul Islam, Chelsey Edge, Yuyang Xiao, Peigen Luo, Muntaqim Mehtaz, Christopher Morse, Sadman Sakib Enan, and Junaed Sattar. Semantic segmentation of underwater imagery: Dataset and benchmark. In *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 1769–1776. IEEE, 2020. 1, 2, 6

[20] Peng-Tao Jiang, Qibin Hou, Yang Cao, Ming-Ming Cheng, Yunchao Wei, and Hong-Kai Xiong. Integral object mining via online attention accumulation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 2070–2079, 2019. 3

[21] Donald R Jones, Matthias Schonlau, and William J Welch. Efficient global optimization of expensive black-box functions. *Journal of Global optimization*, 13(4):455–492, 1998. 4

[22] 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. 3

[23] Minsong Ki, Youngjung Uh, Wonyoung Lee, and Hyeran Byun. Contrastive and consistent feature learning for weakly supervised object localization and semantic segmentation. *Neurocomputing*, 445:244–254, jul 2021. 3

[24] Alexander Kolesnikov, Lucas Beyer, Xiaohua Zhai, Joan Puigcerver, Jessica Yung, Sylvain Gelly, and Neil Houlsby. Big transfer (bit): General visual representation learning. In *European conference on computer vision*, pages 491–507. Springer, 2020. 1, 3

[25] Alexander Kolesnikov, Alexey Dosovitskiy, Dirk Weissenborn, Georg Heigold, Jakob Uszkoreit, Lucas Beyer, Matthias Minderer, Mostafa Dehghani, Neil Houlsby, Sylvain Gelly, Thomas Unterthiner, and Xiaohua Zhai. An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale. 2021. 1

[26] Ksenia Konyushkova, Raphael Sznitman, and Pascal Fua. Introducing geometry in active learning for image segmentation. In *2015 IEEE International Conference on Computer Vision (ICCV)*, pages 2974–2982, 2015. 1

[27] Ksenia Konyushkova, Jasper Uijlings, Christoph H Lampert, and Vittorio Ferrari. Learning Intelligent Dialogs for Bounding Box Annotation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, jun 2018. 3

[28] 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. 3

[29] Lin Li, Eric Rigall, Junyu Dong, and Geng Chen. MAS3K: An Open Dataset for Marine Animal Segmentation. In *International Symposium on Benchmarking, Measuring and Optimization*, pages 194–212. Springer, 2020. 1

[30] Gaobo Liang and Lixin Zheng. A transfer learning method with deep residual network for pediatric pneumonia diagnosis. *Computer Methods and Programs in Biomedicine*, 187:104964, apr 2020. 1, 3

[31] 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. 3

[32] Shenlan Liu, Xiang Liu, Gao Huang, Lin Feng, Lianyu Hu, Dong Jiang, Aibin Zhang, Yang Liu, and Hong Qiao. FSD-10: a dataset for competitive sports content analysis. *arXiv preprint arXiv:2002.03312*, 2020. 1

[33] Rafid Mahmood, James Lucas, David Acuna, Daiqing Li, Jonah Philion, Jose M Alvarez, Zhiding Yu, Sanja Fidler, and Marc T Law. How much more data do i need? estimating requirements for downstream tasks. *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 275–284, 2022. 2, 3, 4

[34] Rafid Mahmood, James Lucas, Jose M. Alvarez, Sanja Fidler, and Marc T. Law. Optimizing data collection for machine learning. *Advances in Neural Information Processing Systems (NeurIPS)*, 10 2022. 2, 3

[35] Michael McCloskey and Neal J Cohen. Catastrophic interference in connectionist networks: The sequential learning problem. In *Psychology of learning and motivation*, volume 24, pages 109–165. Elsevier, 1989. 6

[36] Afonso Menegola, Michel Fornaciali, Ramon Pires, Flávia Vasques Bittencourt, Sandra Avila, and Eduardo Valle. Knowledge transfer for melanoma screening with deep learning. In *2017 IEEE 14th International Symposium on Biomedical Imaging (ISBI 2017)*, pages 297–300. IEEE, 2017. 1, 3

[37] Thomas Mensink, Jasper Uijlings, Alina Kuznetsova, Michael Gygli, and Vittorio Ferrari. Factors of Influence for Transfer Learning across Diverse Appearance Domains and Task Types. *arXiv preprint arXiv:2103.13318*, 2021. 1, 3, 4

[38] Dim P Papadopoulos, Alasdair D F Clarke, Frank Keller, and Vittorio Ferrari. Training Object Class Detectors from Eye Tracking Data. In David Fleet, Tomas Pajdla, Bernt Schiele, and Tinne Tuytelaars, editors, *Computer Vision – ECCV 2014*, pages 361–376, Cham, 2014. Springer International Publishing. 3

[39] Dim P Papadopoulos, Jasper R R Uijlings, Frank Keller, and Vittorio Ferrari. Extreme clicking for efficient object annotation. In *Proceedings of the IEEE international conference on computer vision*, pages 4930–4939, 2017. 3

[40] 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 *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, December 2015. 1, 2, 6

[41] Hiranmayi Ranganathan, Hemanth Venkateswara, Shayok Chakraborty, and Sethuraman Panchanathan. Deep active learning for image classification. In *2017 IEEE International Conference on Image Processing (ICIP)*, pages 3934–3938, 2017. 3

[42] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical image computing and computer-assisted intervention*, pages 234–241. Springer, 2015. 6

[43] Mennatullah Siam, Sara Elkerdawy, Martin Jagersand, and Senthil Yogamani. Deep semantic segmentation for automated driving: Taxonomy, roadmap and challenges. In *2017 IEEE 20th international conference on intelligent transportation systems (ITSC)*, pages 1–8. IEEE, 2017. 1

[44] 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. 3

[45] Liat Sless, Bat El Shlomo, Gilad Cohen, and Shaul Oron. Road scene understanding by occupancy grid learning from sparse radar clusters using semantic segmentation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision Workshops*, page 0, 2019. 1

[46] Chunfeng Song, Yan Huang, Wanli Ouyang, and Liang Wang. Box-driven class-wise region masking and filling rate guided loss for weakly supervised semantic segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3136–3145, 2019. 3

[47] Chen Sun, Abhinav Shrivastava, Saurabh Singh, and Abhinav Gupta. Revisiting unreasonable effectiveness of data in deep learning era. In *Proceedings of the IEEE international conference on computer vision*, pages 843–852, 2017. 4

[48] Nima Tajbakhsh, Jae Y. Shin, Suryakanth R. Gurudu, R. Todd Hurst, Christopher B. Kendall, Michael B. Gotway, and Jianming Liang. Convolutional Neural Networks for Medical Image Analysis: Full Training or Fine Tuning? *IEEE Transactions on Medical Imaging*, 35(5):1299–1312, may 2016. 3

[49] Meng Tang, Abdelaziz Djelouah, Federico Perazzi, Yuri Boykov, and Christopher Schroers. Normalized cut loss for weakly-supervised cnn segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1818–1827, 2018. 3

[50] Chien-Hao Tseng, Chia-Chien Hsieh, Dah-Jing Jwo, Jyh-Horng Wu, Ruey-Kai Sheu, and Lun-Chi Chen. Person Retrieval in Video Surveillance Using Deep Learning–Based Instance Segmentation. *Journal of Sensors*, 2021, 2021. 1

[51] Xiaosong Wang, Yifan Peng, Le Lu, Zhiyong Lu, Mohammadhadi Bagheri, and Ronald Summers. ChestX-ray8: Hospital-scale Chest X-ray Database and Benchmarks on Weakly-Supervised Classification and Localization of Common Thorax Diseases. In *2017 IEEE Conference on Computer Vision and Pattern Recognition(CVPR)*, pages 3462–3471, 2017. 3

[52] P Welinder, S Branson, T Mita, C Wah, F Schroff, S Belongie, and P Perona. Caltech-UCSD Birds 200. Technical Report CNS-TR-2010-001, California Institute of Technology, 2010. 1

[53] Amir R Zamir, Alexander Sax, William Shen, Leonidas Guibas, Jitendra Malik, and Silvio Savarese. Taskonomy: Disentangling Task Transfer Learning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3712–3722, jun 2018. 3

[54] Bingfeng Zhang, Jimin Xiao, Jianbo Jiao, Yunchao Wei, and Yao Zhao. Affinity Attention Graph Neural Network for Weakly Supervised Semantic Segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pages 1–1, 2021. 3

[55] Bolei Zhou, Aditya Khosla, Agata Lapedriza, Aude Oliva, and Antonio Torralba. Learning deep features for discriminative localization. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2921–2929, 2016. 3