

Clickstream analysis for crowd-based object segmentation with confidence

Eric Heim, Alexander Seitel, Jonas Andrulis, Fabian Isensee, Christian Stock, Tobias Ross
and Lena Maier-Hein

Abstract—With the rapidly increasing interest in machine learning based solutions for automatic image annotation, the availability of reference annotations for algorithm training is one of the major bottlenecks in the field. Crowdsourcing has evolved as a valuable option for low-cost and large-scale data annotation; however, quality control remains a major issue which needs to be addressed. To our knowledge, we are the first to analyze the *annotation process* to improve crowd-sourced image segmentation. Our method involves training a regressor to estimate the quality of a segmentation from the annotator’s clickstream data. The quality estimation can be used to identify spam and weight individual annotations by their (estimated) quality when merging multiple segmentations of one image. Using a total of 29,000 crowd annotations performed on publicly available data of different object classes, we show that (1) our method is highly accurate in estimating the segmentation quality based on clickstream data, (2) outperforms state-of-the-art methods for merging multiple annotations. As the regressor does not need to be trained on the object class that it is applied to it can be regarded as a low-cost option for quality control and confidence analysis in the context of crowd-based image annotation.

Index Terms—Crowdsourcing, Quality Control, Object Segmentation, Confidence Estimation, Clickstream Analysis

1 INTRODUCTION

THE interest in machine learning techniques for data processing has been rapidly growing in various fields including the automotive industry [1], computer vision [2], and biomedical image processing [3]. The major bottleneck of most of those techniques - especially with the rise of deep learning algorithms - is the annotation of the often large amount of required training data [4]. Crowdsourcing has become popular in this context as it is based on outsourcing cognitive tasks to many anonymous, untrained individual *workers* from an online community [5], [6]. It thus provides a valuable tool for low-cost and large-scale data annotation. One of the major challenges in crowd-based image annotation is quality control. Although many workers are highly skilled and motivated, the presence of *spammers* is a severe problem as they are mainly interested in receiving the reward for a given task by investing the minimum amount of time [7]. As detailed in section 2, methods proposed to address this issue have led to better overall results. However, they either require expert workers to perform time-wasting tasks on images for which the reference is already known, use inter worker agreement on segmentations of multiple workers on the same image or

restrict the pool of potential workers to those that have a history of exceptionally good rating by the task providers (and thus reduce annotation speed). These quality control approaches all rely on the annotation result.

The hypothesis of this paper is that the quality of a segmentation is reflected in the way a worker annotated the image (Fig. 1). Specifically, we assume that spammers behave differently (e.g. have a faster mouse movement) compared to workers that produce high-quality annotations. Our work therefore focuses on the annotation process itself to estimate the quality of crowd-sourced image segmentations. For this purpose, all user traces during the annotation process are recorded as *clickstreams*. A clickstream consists of consecutive time stamped events triggered by a user while interacting with a web application.

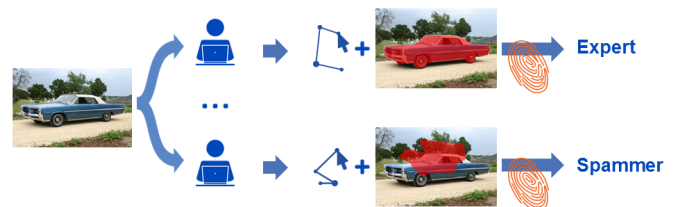


Fig. 1: Annotation process analysis based segmentation quality estimation in crowd-sourced image segmentation. The quality of the segmentation is derived from the worker’s mouse actions recorded in the clickstream.

To our knowledge, we are the first to use *annotation process-based features* for quality control in crowd-sourced image segmentation. Our contribution is illustrated in Fig. 2-3 and comprises (1) an approach to estimate the segmentation quality based on clickstream analysis and (2) a method for accurate crowd-based object segmentation

- E. Heim, A. Seitel, T. Ross and L. Maier-Hein work at the Division of Computer Assisted Medical Interventions, German Cancer Research Center (DKFZ), Heidelberg, Germany.
E-mail: {e.heim, a.seitel, t.ross, l.maier-heim}@dkfz.de
- F. Isensee works at the Division of Medical Image Computing, German Cancer Research Center (DKFZ), Heidelberg, Germany.
E-mail: f.isensee@dkfz.de
- J. Andrulis works at Pallas Ludens GmbH, Heidelberg, Germany.
E-mail: jonas.andrulis@pallas-ludens.awsapps.com
- C. Stock works at the Division of Clinical Epidemiology and Aging Research at the German Cancer Research Center (DKFZ), Heidelberg, Germany, and the Institute of Medical Biometry and Informatics, University Hospital Heidelberg, Heidelberg, Germany.
E-mail: stock@imbi.uni-heidelberg.de

based on confidence-based weighting of individual crowd segmentations. The outline of the paper is as follows: Section 2 provides an overview of related work that has previously been published on this topic. The quality estimation concept for crowd-sourced image segmentation is detailed in section 3, including the description of the performed validation. Finally, experimental results are presented and discussed in sections 4 and 5 respectively.

2 RELATED WORK

We review related work in view of the two main research fields related to our approach. Section 2.1 discusses the methods proposed for quality control in crowdsourcing, section 2.2 summarizes the research on user behavior analysis outside the field of crowdsourcing.

2.1 Quality control in crowdsourcing tasks

Approaches to quality control in crowd-sourced data annotation can roughly be divided into the following categories:

2.1.1 Integration of reference data into annotation task

Von Ahn’s reCaptcha [8], one of the first approaches to ensure the quality of crowd-sourced annotations, proposes pairing an unknown label with a label for which a reference is available (captcha) and rejecting the new label when the performed captcha has failed. For the final result, the consensus of labels from different workers is used. Even though this is an online approach, where false labels are directly rejected, it requires pre-labeled reference data and is not cost-effective, since the crowd has to additionally annotate already known labels, which is time consuming, especially in the context of object segmentation. In an initial quality control step, Lin et al. [9] rejected workers based on their performance in segmenting images of a training task and verified trusted worker’s annotations via a manual verification step. This approach still relies on reference data and does not incorporate the expertise of specific workers. Kazai et al. [7] conducted worker behavioral analyses and grouped them into spammers, sloppy, incompetent, competent and diligent workers by a worker survey and crowd annotations of already labelled digitized books. They did not explore the application of the worker’s expertise on other task types such as image annotation. An approach to categorize the workers into spammers, workers who did not understand the task, and reliable workers from their annotation on reference images was presented by Oleson et al. [10]. Cabezas et al. [11] proposed a quality control approach using interactive object segmentation based on background and foreground clicks in the image. Incorrect clicks are detected by their spatial neighborhood inside superpixels. Finally the clicks were weighted based on the worker’s expertise. The workers had to perform segmentations on gold standard images to determine their level of expertise. However the approach only uses an initial expertise estimation on a reference task and can thus not assure the quality of each individual label during the annotation process.

2.1.2 Majority voting

Another approach to detect low-quality labels is to acquire multiple labels from each sample and rank them against the majority of the acquired labels [2], [12], [13], [14], [15]. A pixel of the image to be annotated is classified as belonging to the object if the majority of workers have classified it as object (e.g. [14]). The assumption behind the majority voting approach is that the majority of labels are of good quality, which is likely to result in a larger amount of labelling data being acquired from the crowd. When merging a majority of accurate object segmentations, majority voting can provide solutions close to the optima [16].

2.1.3 Manual grading of annotation quality

A commonly used means for ensuring quality of crowd-sourced annotations is to let peer crowd-workers rate the quality of performed annotations. This is among others integrated in the annotation pipelines for generating large image databases such as LabelMe [17] and COCO [9]. This manual verification task imposes additional annotation costs that could be avoided when quality estimation is performed automatically.

2.1.4 Automatic annotation quality estimation

Vittayakorn et al. [18] investigated several scoring functions to assess the quality of crowd-sourced object annotations. Quality measures include features derived from the image itself (e.g. image edges) and parameters obtained from the resulting annotation contour or segmentation (e.g. number of control points, annotation size). Features related to the actual *process* of performing the annotation have not been applied for quality control. Hence, the method does not rely on recording the users clickstream. Welinder et al. [19] proposed a general method to determine a reference value of some properties from multiple noisy annotations; this is a method which inherently evaluates the annotator’s expertise and reliability. An expectation maximization (EM) approach is used to infer the target annotation as well as a confidence value for the worker’s reliability. This method has currently only been tested on simple binary decisions and bounding boxes. Similarly, Long et al. [20] introduced a Joint Gaussian Process Model for active visual recognition and expertise estimation. This approach has also not been tested on more complex annotations such as object segmentations. A Bayesian probabilistic model for rating the competence of workers creating binary labels of images out of multiple label aggregations per image from different workers was presented by Welinder et al. [21]. Several methods have been proposed that estimate the annotation quality by training a regressor on features derived from the image and/or final annotation contour [22], [23], [24], [25], [26]. Sameki et al. [27] recently presented a method that detects the worker’s behavior from the number of mouse clicks, the time per task and the time per user click. They were able to show that the number of contour points in the final annotation and the annotation time are good predictors for segmentation quality. The authors, however, state that the improvements to be gained from this limited number of features might be minimal for certain applications such as the segmentation of bio-medical images.

Other quality control approaches in crowd-based image annotation have been designed for specific applications, such as correspondence search in endoscopic images [28] and cannot be generalized to other tasks, such as segmentation. Mao et al. [29] investigated how different payment methods are related to the task outcome and came to the conclusion that financial incentives can be used to boost annotation speed at the cost of quality.

2.2 User behavior analysis

Several groups have investigated using clickstream data or mouse dynamics for user behavior analysis outside the field of crowdsourcing. Successful approaches have been presented to identify specific users by making use of biometric features based on mouse dynamics [30], [31] for user authentication and intrusion detection in computer systems. In Ahmed et al. [30] a neural network is trained to recognize a specific user via a feature set of mouse dynamics, e.g. velocity, acceleration, clicks, moving direction. The method of Feher et al. [31] extended this approach by using a random forest classifier trained on a similar features and combining the results of individual mouse actions in contrast to the histogram-based approach by Ahmed et al. [30]. While these approaches have been widely used for user authentication and intrusion detection, their application to segmentation and annotation tasks has not yet been presented. A method to detect fake identities and sybil accounts in online communities, based on features extracted out of clickstreams, was presented by Wang et al. [32]. In contrast to the mouse dynamics based approaches, user actions such as uploading photos, chatting with friends, pressing a specific button, etc. are used here to train the classifier. Wang et al. [33] showed that clickstream analysis can be used to cluster social network users based on their behaviour via an unsupervised learning approach. Lee et al. [34] presented a machine learning based approach to detect content in social networks that was manipulated through crowdsourcing. They trained a classifier on social network features, e.g. number of friends, posts on websites, overall activity time in the social network and achieved a high level of accuracy in detecting crowd manipulated content. Rzeszotarski et al. [35] presented a user-behavior-based method for estimating the quality of crowdsourcing for classification tasks, comprehensive reading tasks and text generation tasks.

In conclusion, while there has been a considerable effort in controlling the quality of crowd-sourced annotations and analysing user behaviour, we are not aware of any prior work on using annotation process-based features for quality control in crowdsourcing. Furthermore, confidence-based annotation merging using quality estimation has not been introduced to date.

3 METHODS

This section presents the concept of our confidence-based method for annotation merging (Sec. 3.1), a prototype implementation of the concept (Sec. 3.2) and the experimental design of our validation study (Sec. 3.3).

3.1 Segmentation concept

The purpose of this contribution was to develop a quality control method for crowd-based object segmentation that does not rely on (1) additional tasks (with known outcome) to be performed or (2) prior knowledge of a specific worker’s annotation history. Inspired by previous work on clickstream analysis for user identification [32], [33], the hypothesis of our work is that the clickstream data itself is sufficient to estimate a worker’s performance. Our concept involves a training process (Fig. 2), in which a regressor is trained to estimate the quality of a given segmentation by using features extracted from clickstream data. To segment an unseen image, the image is repeatedly distributed to the crowd until a certain number of segmentations with a high estimated quality is reached (Fig. 3). The DICE similarity coefficient (DSC) [36] is used as a measure for segmentation quality, which is defined by comparing a segmentation U to its corresponding reference segmentation V (Eq. 1).

$$DSC = \frac{2|V \cap U|}{|V| + |U|} \quad (1)$$

The obtained segmentations are then merged in a weighted manner, according to their estimated quality. An implementation of this concept is presented in the following section 3.2.

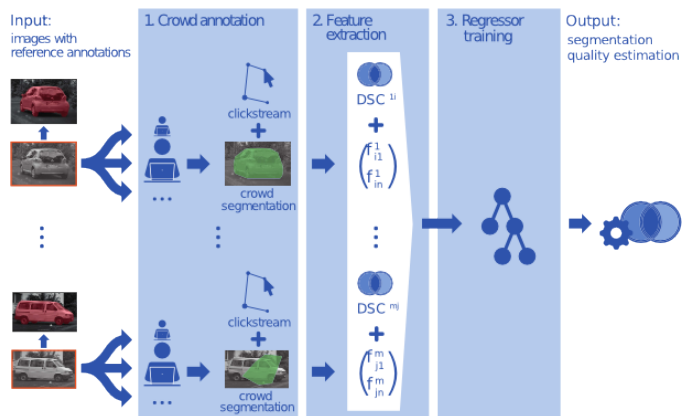


Fig. 2: Training of segmentation quality estimator. Initially, images with known reference segmentations are distributed to multiple crowd workers. While the workers are segmenting the images, the system records their annotation behavior (clickstreams). For each annotated image, the clickstream is converted into a feature vector characterizing the worker’s interaction behavior. The DSC is computed using the reference annotation. The set of all collected feature vectors with corresponding DSC values is then used to train a regressor to estimate the DSC solely based on a worker’s clickstream.

3.2 Prototype implementation

With our prototype implementation, we focus on single-object segmentation. The prototype comprises a user interface for crowd-based image segmentation (Sec. 3.2.1) as well as capabilities for clickstream data collection (Sec. 3.2.2), feature extraction (Sec. 3.2.3), segmentation quality estimation (Sec. 3.2.4) and confidence-based annotation merging (Sec. 3.2.5).

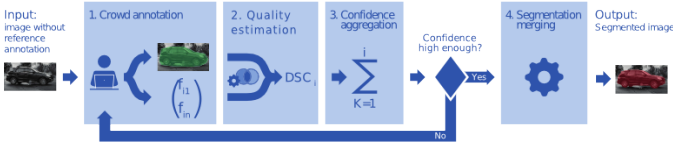


Fig. 3: Concept for crowd-based image segmentation based on a trained segmentation quality estimator (Fig. 2). The image to be annotated is repeatedly distributed to the crowd until a certain confidence level is reached. The obtained segmentations are merged in a weighted manner, where the weight of the worker’s annotation increases with the estimated DSC on that specific image.

3.2.1 User Interface

The prototype implementation incorporates a web-based user interface implemented in HTML and JavaScript to perform the image segmentations and collect the corresponding clickstream data. It provides basic functionalities to create, delete and correct contours. The worker can draw a contour by either pressing and holding the left mouse button while dragging the cursor or define points via clicks on the canvas. These points are then successively connected by lines, resulting in the segmentation contour. Typically, workers use a combination of both modes, e.g. if the object contains curvy regions combined with sharp corners or long lines, the worker might only set points to draw lines or corners and continues drawing the curves by dragging the mouse and vice versa. To help create accurate contours it is possible to zoom into the image by using the mouse wheel. If the worker is not satisfied with the created segmentation, the contour can be corrected by selecting and dragging single points to the desired position or by deleting redundant points by a double click. It is also possible to delete the complete contour and restart the segmentation from scratch.

3.2.2 Clickstream data collection

During the segmentation task, every action triggered by the worker’s mouse is saved to the clickstream. In addition to worker-triggered events, the current mouse position is continuously recorded. We represent the clickstream as a sequence of successively occurring mouse events $E = \{e_1, \dots, e_i\}$. The clickstream events are sorted by their time stamp t_i in their recorded chronological order. Each recorded event provides x and y coordinates of a point $\vec{p} \in \mathbb{R}^2$ in the canvas coordinate system, their corresponding coordinates transformed into the image coordinate system, the event type and the object upon the event was triggered. We differentiate between the following event types and actions: *mouse-down*, *mouse-up*, *mouse wheel*, *double-click*, *mouse-move*. The following objects are present in the user interface: *canvas*, *delete contour button*, *zoom button*, *save button*.

3.2.3 Feature extraction

Based on the assumption that reliable workers will interact differently with the program than malicious workers, we derive our feature set from the recorded clickstream. We assume that reliable workers will put more time and effort into creating accurate segmentations than malicious workers will do. Furthermore, we assume that the worker’s behavior will

change based on their level of expertise. Trained experts, for example, might create high-quality segmentations in less time with less effort than inexperienced, untrained workers. Inexperienced workers might instead create a high amount of user input that will not necessarily result in accurate segmentations. In order to estimate the segmentation quality, we need a feature set that is able to classify the quality of segmentations of different object types without incorporating knowledge of the underlying system, object types or worker identity. We define the following feature set that includes features calculated from both the clickstream and the image itself.

Clickstream-based features

Velocity: $\forall e_i \in E$ we compute a velocity vector \vec{v}_i based on the elapsed time Δt_i (in milliseconds) and travelled distance $\Delta \vec{p}_i$ on the canvas with the positions \vec{p}_i and \vec{p}_{i-1} of two successive events e_i and $e_{i-1} \in E$:

$$\vec{v}_i = \frac{\Delta \vec{p}_i}{\Delta t_i} = \frac{\vec{p}_{i-1} - \vec{p}_i}{t_{i-1} - t_i} \quad (2)$$

The velocity is only computed for mouse-move operations. On mouse or button clicks the velocity is set to zero. The mean, median, standard deviation and 95% quantile of the velocity $\forall e_i \in E$ are used as features.

Acceleration: The acceleration \vec{a}_i of every event in the clickstream is derived from the velocity change $\Delta \vec{v}_i$ and elapsed time Δt_i between two successive events $e_i \in E$ and $e_{i-1} \in E$:

$$\vec{a}_i = \frac{\Delta \vec{v}_i}{\Delta t_i} \quad (3)$$

The mean, median, standard deviation and 95% quantile of the acceleration $\forall e_i \in E$ are used as features.

Zoom: We extract the total number of zoom events out of the clickstream by using the event and object types detailed in Sec. 3.2.2.

Canvas clicks: The total number of mouse-clicks that are executed on the canvas are used as a feature.

Double clicks: The total number of double-clicks that are executed on the canvas are used as a feature.

Elapsed time: The duration of the whole task is calculated as the difference between the time stamps of the first and last event in the clickstream: $\Delta T = t_n - t_0$. The time is normalized by the mouse clicks and actions as described in [27], i.e.: $\frac{\Delta T}{\text{canvas clicks}}$.

Ratio of traveled mouse distance and length of the segmented contour: In contrast to the absolute size of a contour used by Vittayakorn et al. [18], we assume that the length of a created contour will be in relation to the total traveled mouse distance in the image space. We assume that a spammer will typically try to create a random contour on the canvas that will be similar in length to the total traveled distance, whereas reliable users will more likely perform different mouse movements such as zooming or moving the mouse to adjust the view or correcting created contours.

Mouse strokes: The total number of mouse strokes is used as a feature. We define a mouse stroke as a sequence of mouse-move events $S \subseteq E$ that occur between a mouse down and up event. As explained in section 3.2.1, mouse strokes are used to draw a contour or to select and drag points to correct existing contours.

Draw operations and contour correction: To distinguish whether a mouse stroke is used to draw a new contour or correct an existing one, the clickstream is processed using Algorithm 1 based on the event types and objects previously described.

```

Data: mouse strokes  $S$ , clickstream  $E$ 
Result: set of draw operations  $D$ , set of corrections  $K$ 
begin
   $i \leftarrow 0$ ;
  for  $S_i \subseteq E$  do
    if  $S_i.down.\vec{p} \equiv S_{i-1}.up.\vec{p}$ , with  $S_{i-1} \in D$  then
       $D.add(S_i)$ ;
    else
      if  $\exists e \in E : S_i.down.\vec{p} \equiv e.\vec{p} \wedge S_i.t \leq e.t$ 
        then
           $K.add(S_i)$ ;
        else
           $D.add(S_i)$ ;
        end
      end
    end
  end

```

Algorithm 1: Clickstream processing to identify if a mouse stroke S is a draw operation D or a correction K . We distinguish between three different cases: (1) If the mouse down event of the current mouse stroke occurs at the same position as the mouse up event that terminated the last draw event, the current mouse stroke S_i is considered as a drawing event. (2) If this did not occur and the down event of the current stroke occurred at the same position as a previous event in the clickstream, the current mouse stroke S_i is considered as a correction event. (3) Otherwise a new contour is started and the current mouse stroke S_i is a draw event. A kd-tree [37] is used to speed up the search of events in the clickstream based on their canvas position.

We use the absolute number of draw events and executed corrections as features. In addition we use the acceleration and velocity for the executed draw and corrections and use the mean, median, standard deviation and the 95% quantile as features.

Image-based features

Similar to Vittayakorn et al. [18], we assume that for an accurate segmentation, the contour will mainly be located on or next to image edges. While creating a contour, the worker will try to follow edges in the image and the mouse will move perpendicular to the gradient direction. The contour of a spammer, who is not segmenting an object, will, in contrast, not be created perpendicular to the gradient direction. In addition to the relation between the mouse-move direction and the gradient, the quality of the resulting contour is ranked based on the gradients and the interpolated vertex normals of the created polygon. In this case, we assume that vertex normals are collinear to the gradient direction.

We thereby define features that are based on the image gradient and compute the angle γ_i between the gradient direction $\vec{g}_{x,y}$ at the image coordinates x, y and the

mouse-move direction \vec{d}_i for each event e_i . The gradient is computed using recursive Gaussian filtering [38] as implemented in the Insight Toolkit (ITK) [39] while the mouse-move direction \vec{d}_i is derived by normalizing the velocity v_i for the event e_i :

$$\vec{d}_i = \frac{\vec{v}_i}{\|\vec{v}_i\|} \quad (4)$$

In the first step we compute the angle between $\vec{g}_{x,y}$ and \vec{d}_i according to Eq. 5:

$$\omega(\vec{d}_i, \vec{g}_{x,y}) = \text{acos}\left(\frac{\vec{d}_i \bullet \vec{g}_{x,y}}{\|\vec{d}_i\| \cdot \|\vec{g}_{x,y}\|}\right) \cdot \frac{180}{\pi} \quad (5)$$

use the smallest angle γ_i :

$$\gamma_i = \begin{cases} \omega(\vec{d}_i, \vec{g}_{x,y}), & \text{if } 0^\circ \leq \omega(\vec{d}_i, \vec{g}_{x,y}) \leq 180^\circ \\ 360^\circ - \omega(\vec{d}_i, \vec{g}_{x,y}), & \text{otherwise} \end{cases} \quad (6)$$

and normalize to an angle between 0° and 90° :

$$\bar{\gamma}_i = \epsilon_\gamma - |\epsilon_\gamma - \gamma_i| \quad (7)$$

with $\epsilon_\gamma = 90^\circ$. We then define the following image-based features:

Features extracted from contour drawing and correction events: The mean, standard deviation, median and 95% quantile of the angles $\bar{\gamma}_i$ are calculated as features for all contour drawing events D , all contour corrections K and all consecutive mouse click events. For the mouse click events, the direction vector \vec{d}_i is calculated as the line segment connecting the current mouse click with a previous one.

Features extracted from the final contour: The final contour is defined as a set of consecutive connected 2D vertices represented by points in the image coordinate system $X = \{\vec{p}_1, \dots, \vec{p}_n\}$, with $\vec{p}_i \in \mathbb{R}^2$. The normal for each line segment \vec{n}_i is calculated with two consecutive vertices $\vec{p}_i, \vec{p}_{i+1} \in X$:

$$\vec{n}_i = \begin{bmatrix} -1 \cdot (\vec{p}_{i+1,y} - \vec{p}_{i,y}) \\ (\vec{p}_{i+1,x} - \vec{p}_{i,x}) \end{bmatrix} \quad (8)$$

The interpolated vertex normal \vec{n}_i is calculated by a linear interpolation of the line segment normals of two adjacent line segments \vec{n}_{i-1} and \vec{n}_i .

$$\vec{n}_i = \frac{1}{2}\vec{n}_{i-1} + \frac{1}{2}\vec{n}_i \quad (9)$$

With the vertex normals and the gradient directions, we compute $\bar{\gamma}_i \forall p \in X$ according to Eq. 7 with the image gradient $\vec{g}_{x,y}$ and the interpolated vertex normal \vec{n}_i instead of the drawing direction \vec{d}_i . We use the mean, standard, deviation, median and 95% quantile of all computed angles as features.

3.2.4 Estimation of segmentation quality

With the set of features introduced in the previous Sec. 3.2.3 and the DSC of segmentations for which a reference segmentation exists, we train a random forest regressor [40] to estimate the quality of unseen crowd segmentations. We use the random forest regressor implementation from scikit-learn [41] to estimate the DSC \hat{s}_j of an unseen crowd segmentation U_j . To determine the parameters for our random

forest regressor we used a data set consisting of 20 images of cars that were not part of the validation data set. For each image, we obtained reference segmentations with the method described in [42]. In addition, we acquired a total of 500 crowd segmentations for each image with the platform provided by the Pallas Ludens GmbH¹, resulting in 10,000 segmentations for cross-validation. We determined the tree depth and the minimum number of samples per leaf of our random forest regressor by running a 10 fold labeled cross-validation optimizing the R^2 score with the estimated DSC \hat{s}_i , the corresponding real DSC s_j for every segmentation and the mean DSC \bar{s} of the data set (Eq. 10):

$$R^2 = 1 - \frac{\sum_{j=1}^n (s_j - \hat{s}_j)^2}{\sum_{j=1}^n (s_j - \bar{s})^2} \quad (10)$$

Before running the cross-validation we ensured that neither the same workers or the same images were included in the test and training data set simultaneously. For a random forest regressor with 500 trees, a minimum of three samples per leaf and extending the tree depth until all leaf nodes are pure, we achieved a mean R^2 score of 0.71 ± 0.04 .

3.2.5 Confidence-based segmentation merging

To generate a new segmentation, the workers annotations were merged according to their estimated quality. We investigated two different methods for merging the segmentations: (1) A confidence-weighted majority voting approach based on the estimated segmentation quality and (2) a simultaneous truth and performance level estimation (STAPLE) algorithm based approach [43].

Confidence-weighted majority voting

Based on the estimated quality, segmentations are discarded if the estimated DSC \hat{s}_j for a segmentation U_j falls under a predefined DSC threshold $\epsilon_t \in [0, 1]$. With the estimated DSC values \hat{s}_j we compute a normalized confidence value $\kappa(\hat{s}_j) \in [0, 1]$ for the remaining segmentations:

$$\kappa(\hat{s}_j) = \frac{\hat{s}_j - \epsilon_t}{1 - \epsilon_t} \quad (11)$$

We denote the two dimensional segmentations images $U_j(x, y)$ with the width m and height n , where (x, y) is the coordinate of the pixel in the image with $U_j(x, y) \in \{0, 1\}$. Each segmentation $U_j(x, y)$ is weighted with the estimated confidence (Eq. 11) and the confidence-weighted pixel values are accumulated in the image $\Delta U(x, y)$:

$$\Delta U(x, y) = \sum_{j=1}^{\lambda} U_j(x, y) \cdot \kappa(\hat{s}_j) \quad (12)$$

The smallest integer value μ representing the majority of λ segmentations is used to calculate the fraction of the maximum accumulated confidence value ψ in $\Delta U(x, y)$, that is required to classify a pixel as belonging to the segmented object:

$$\psi = \frac{\max_{x=1, y=1}^{m, n} \Delta U(x, y)}{\lambda} \cdot \mu \quad (13)$$

The final confidence-weighted segmentation $H(x, y)$ is calculated by applying a binary decision to each pixel of the image $\Delta U(x, y)$:

$$H(x, y) = \begin{cases} 1, & \text{if } \Delta U(x, y) \geq \psi \\ 0, & \text{otherwise} \end{cases} \quad (14)$$

Simultaneous truth and performance level estimation (STAPLE) with segmentation quality estimation

Instead of weighting each segmentation, low quality segmentations with a DSC under a predefined DSC threshold $\epsilon_t \in [0, 1]$ are discarded based on their estimated quality and the remaining segmentations with a high estimated DSC are fused with the native STAPLE algorithm implementation [43].

3.3 Validation

In our experiments we investigated the following aspects: (1) What is the quality of the proposed segmentation quality estimation (Sec. 3.3.1)? (2) What is the quality of the proposed confidence-based approach to segmentation fusion compared to state-of-the-art methods for annotation merging (Sec. 3.3.2)? (3) How well does the segmentation quality estimation generalize to new object types (Sec. 3.3.3)? and (4) what are the costs of the proposed method compared to state-of-the-art methods (Sec. 3.3.4)?

3.3.1 Segmentation quality estimation

We validated our proposed segmentation concept on a subset of the publicly available PASCAL Visual Object Classes (VOC) [44]. The regressor was validated on two object classes within the VOC challenge data: $\{cat, car\}$. For each class, we used 100 out of the first 150 images, making sure that a broad range of degree of difficulty (e.g. fully visible objects and partially occluded objects) was covered. Using the prototype implementation of our concept (Sec. 3.1) we acquired 10,000 segmentations per class (100 segmentations per image) in a gaming crowd provided by the Pallas Ludens GmbH with their proprietary user interface. Example segmentations with the segmentation outline were provided to the workers. No filters for quality management, e.g. captchas, tutorials or blocking of known spammers, were applied so that clickstreams from a variety of worker types with a high fluctuation in the segmentation quality would be collected. Furthermore we assured that each worker segmented each image at most once per class. This resulted in a total 20,000 segmented images with their corresponding clickstreams for validation.

For each object class we performed a leave-one-out cross validation, thereby ensuring that only annotations of workers that were not involved in the annotations of the training images were considered. To quantify estimation quality, we determined the absolute difference between the true DSC and the estimated DSC of all test data.

To investigate the number of annotations required for regressor training, we further determined the performance in terms of the R^2 score (Eq. 10) as a function of the number of training annotations. Again, we ensured that no worker was included in the test and training data at the same time. For training, we used 100 random permutations of

1. <https://pallas-ludens.com>

n annotations (for $n = 10,000$ only one permutation was possible).

Due to the success of wrapper and filter methods for optimal feature selection, we applied these methods for quantifying the relevance of our various features used by the segmentation quality estimation. To avoid bias towards a specific feature selection method, we performed the analysis by applying the most commonly used methods. In particular we employed the wrapper methods *Sequential Forward Selection (SFS)* [45] and *Best First Search (BFS)* [46] with a mean squared error criterion and a feature set size penalty to sequentially build an optimal feature set. Furthermore a wide array of filter methods for feature selection was applied to derive feature sets using the same criterion employed for the wrapper methods. The filter methods include *Conditional Mutual Information Maximization (CMIM)* [47], *Interaction Capping (ICAP)* [48], *Joint Mutual Information (JMI)* [49], *Conditional Infomax Feature Extraction (CIFE)* [50], and *Mutual Information for Feature Selection (MIFS)* with a nearest neighbor mutual information estimator [51], [52]. These filter methods construct feature sets sequentially, but do not return which number of features must be selected for optimal results. We addressed this shortcoming by running a cross-validation on the training set using the selected feature sets of increasing size and applying the same selection criterion as in the wrapper method for determining the ideal number of features. Feature selection was solely carried out on the training set.

3.3.2 Confidence-weighted annotation merging

We compared our method to create confidence-weighted crowd-sourced segmentations (as presented in Sec. 3.2.5) with the widely used majority voting method [14] (Sec. 2.1.2). Furthermore we compared the STAPLE algorithm using raw crowd generated segmentations with the quality estimation enhanced STAPLE algorithm based approach (Sec. 3.2.5), where low quality segmentations are filtered out by applying the segmentation quality estimation presented in this paper. To ensure that we only used high quality annotations we executed our method for the DSC at a threshold of $\epsilon_t = 0.9$. According to experiments on a separate data set, this threshold provides a good trade off between quality and excluding high quality annotations. To investigate the performance of our approach as a function of the number of images, we used $\lambda \in B = \{1, \dots, 10\}$ annotations per image that had an equal or higher estimated DSC than ϵ_t . For $\lambda = 1$ the approach basically reduces to the segmentation quality estimation without any further merging of annotations. Considering the fact that poor annotations may be rejected on common crowdsourcing platforms, we compared our method with majority voting using λ annotations. In addition, we computed the average number of annotations φ required to obtain λ annotations with a estimated DSC above ϵ_t by computing $\varphi = \lambda + r$, where r is the mean number of rejected annotations. For each $\lambda \in B$ our method was compared to majority voting with λ and φ annotations. Analogously, our STAPLE approach with quality estimation was compared to the native STAPLE algorithm using λ and φ annotations respectively. The experiments for annotation merging were conducted

separately for each class, i.e. training and testing on only cars or cats.

3.3.3 Generalization capabilities

To investigate the generalization capabilities of our approach, we applied it to a different crowd and additional validation data using an open re-implementation of the annotation software. Specifically, we re-implemented the annotation concept with a user interface based on the Openlayers² [53] library for the widely used crowd-sourcing platform Amazon Mechanical Turk (MTurk)³ [54]. The set of object classes obtained from the VOC validation set, namely $\{car, cat\}$, was extended by seven further object classes from the COCO data set [9] yielding the following three object categories: Vehicles $\{car, airplane, motorcycle\}$, animals $\{cat, bird, dog\}$, rectangular-shaped objects $\{laptop, refrigerator, tv\}$. For each class, we acquired 1,000 annotations (10 per image; 100 images per class). To investigate how our method degrades for target classes that are further away from the training classes, we determined the performance of segmentation quality estimation when training on the animal and vehicle classes respectively and testing the segmentation quality estimation on (1) the same class, (2) the same category (here: different animals/vehicles), (3) a similar category (here: vehicles for animal classes; animals for vehicle classes) and (4) a different category (here: rectangular-shaped object)).

3.3.4 Comparison of annotation costs

Depending on the number of requested segmentations a , the total cost for $c(a)$ image segmentations can roughly be approximated for the proposed method, a baseline method and the manual grading method (Sec. 2.1.3) applied in [9].

Proposed method

The total costs to acquire a segmentations with our proposed method can be approximated as follows, taking into account the average number of annotations used to perform the confidence-weighted majority voting step a_{mv} , the number of annotations used to train the regressor a_t and the percentage of spam s to be expected:

$$c(a) = a_t + \left(a + \frac{s}{1-s} a \right) \cdot a_{mv} \quad (15)$$

Baseline method

A widely used method for simple quality control in crowd based annotation merging using majority voting is to let each crowd worker annotate an image with known reference annotation every a_w annotations. For a_{mv} annotations needed for majority voting typically 10-30% of quality control tasks are mixed in between the tasks [55]. The total cost of annotating a images can then be calculated as follows, also taking into account the number of reference annotations a_r :

$$c(a) = a_{mv} a + \frac{a_{mv} a}{a_w - 1} + a_r \quad (16)$$

2. <https://openlayers.org>

3. <https://www.mturk.com>

Manual grading of annotation quality

A general estimation of the annotation costs entailed by the method applied in [9] can be determined by considering the average number of annotations performed by one worker A_{aw} , the number of categories n_c annotations are requested for, the total number of recruited workers n_w , the total number of approved workers n_{aw} to create $c(a)$ annotations, the number of banned approved workers r and the costs associated with the verification stage v for a percentage of spam s :

$$c(a) = \frac{a}{1-s} + n_c n_w + A_{aw} n_{aw} \cdot r + v \quad (17)$$

4 RESULTS

4.1 Segmentation quality estimation

By filtering the segmentations with our segmentation quality estimation method (using $\epsilon_t = 0.9$, $\lambda = 3$) we improve the mean, median (inter quartile range) (IQR) quality of the pool of segmentations by 16%, 4% (IQR:18%,3%) from 0.80, 0.91 (IQR: 0.79,0.94) to 0.93, 0.95 (IQR: 0.93,0.97). The mean and median (IQR) absolute difference between the true DSC and the estimated DSC were 0.18, 0.12 (IQR: 0.06, 0.21) for training and testing on cars and 0.09, 0.05 (IQR: 0.02, 0.11) for training and testing on cats (Fig. 5a).

In order to create high quality annotations, it is particularly crucial to omit low quality segmentations. To assess the performance of our method in view of this crucial aspect, we identified all annotations with a poor true DSC below 0.5 and a good estimated DSC over 0.8. These false positive (FP) estimations made up 2% of all annotations used for validation. They are subdivided into the following four categories, as illustrated in Fig. 4: (a) Wrong object: The

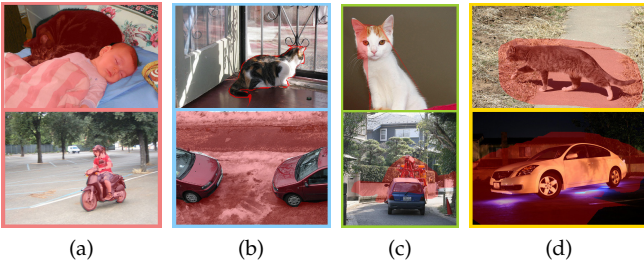


Fig. 4: Examples of poor estimations: (a) Accurate segmentation of wrong objects. (b) Wrong usage of the segmentation tool: Crowd users draw outlines with polygons (top). Inverted segmentation of the object (bottom). (c) Spam. (d) Bounding box.

worker creates an accurate segmentation of a wrong object (possibly on purpose). (b) Wrong tool usage: Wrong usage of the annotation tool, e.g. inverted segmentation or workers try to draw outlines with polygons rather than covering the objects with a polygon. (c) Spam: Obvious spam. (d) Bounding box: Workers draw a bounding box around the object of interest rather than an accurate contour. The distribution of false positive quality estimations according to these error classes is shown in Fig. 5b. The segmentation quality as a function of the number of images used to train the regressor

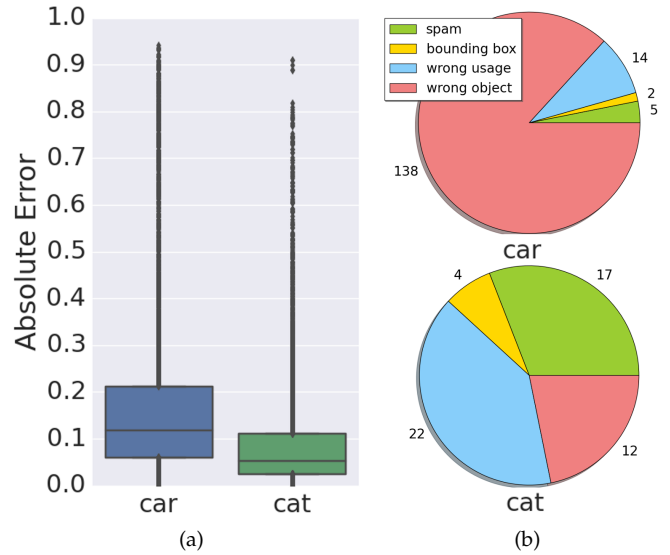


Fig. 5: (a) Absolute error of the estimated segmentation quality for training and testing on the same class. (b) Distribution of crowd segmentations that were estimated to have a high DSC but had a low true DSC (false positives) divided into the error classes introduced in Sec. 4.1 with the absolute amount for each error class. The total amounts of false positives relative to all estimations for each class were: 3% (car) and 1% (cat).

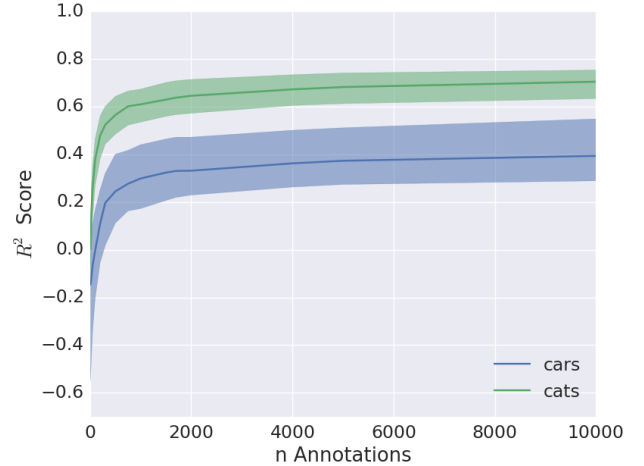


Fig. 6: Median R^2 score and IQR as a function of the number of images used to train the segmentation quality estimation.

is displayed in Fig. 6. For both data sets (cat and car), the mean angle between image gradient direction and interpolated vertex normal and the ratio of traveled mouse distance to length of the segmented contour were found to be important features by all feature selection methods. A potentially high impact on the estimation accuracy could also be found for the median mouse velocity, the median mouse velocity for draw events, the number of events in the clickstream, and the median angle between image gradient direction and mouse move direction for draw events features. The best feature selection methods

data set	method	no. features	mean error
cat	CMIM	7	0.07
cat	ICAP	33	0.06
cat	JMI	11	0.06
cat	CIFE	6	0.06
cat	MIFS	7	0.06
cat	SFS	8	0.07
cat	BFS	7	0.07
cat	BASE	53	0.06
car	CMIM	11	0.10
car	ICAP	6	0.10
car	JMI	7	0.11
car	CIFE	6	0.10
car	MIFS	21	0.11
car	SFS	7	0.11
car	BFS	8	0.11
car	BASE	53	0.10

TABLE 1: Mean estimation error for each feature selection method. The minimal chosen feature set achieved a similar classification performance compared to all features (BASE).

found feature sets as small as 6 features that achieved the same performance in the test set as we obtained when using the full feature set (TABLE 1). Of note, when using only image-based features that are calculated on the result and do not rely on any additional clickstream information, the mean error is approximately twice as high for both data sets compared to the errors reported in TABLE 1.

4.2 Confidence-weighted annotation merging

The resulting DSC values for different numbers of annotations λ for the two confidence-based segmentation merging approaches and the number of rejected crowd segmentations are shown in Fig. 7 and Fig. 8. Both approaches using the segmentation quality estimation outperformed their baseline methods in terms of the segmentation quality and were robust to outliers.

Our presented approach for confidence-weighted majority voting (Sec. 3.2.5) produced statistically significant better results compared to conventional majority voting for the same amount of annotations (Fig. 7). To obtain a median DSC of 0.95 with our method, the number of required annotations ranged from 1 to 3 for the two experiments depicted in Fig. 5a. Compared to conventional majority voting (Majority Voting λ in Fig. 7), the number of annotations was reduced by 75% on average.

When we used the STAPLE algorithm in conjunction with our segmentation quality estimation (Sec. 3.2.5) statistically significant better results were produced compared to the native STAPLE algorithm for the same amount of annotations (Fig. 8). To obtain a median DSC of 0.95 with our method, the number of required annotations ranged from 1 to 2 for the two experiments depicted in Fig. 5a. Compared to the native STAPLE algorithm (STAPLE λ in Fig. 8), the number of annotations was reduced by 73% on average.

Mean differences between conventional majority voting λ and confidence-weighted majority voting and mean differences between the STAPLE algorithm λ and our STAPLE approach with segmentation quality estimation both ranged

from 0.02 (bootstrapped 95%-confidence interval: 0.01, 0.02; 10 annotations, cat) to 0.13 (0.1, 0.16; 4 annotations, car). Non parametric Mann-Whitney U tests for all comparisons with $\lambda \in \{1, \dots, 10\}$ yielded p values that were statistically significant at the significance level of 0.0001, even after a conservative adjustment for multiple testing by the Bonferroni method.

4.3 Generalization capabilities

When applied to another crowd with additional validation data, intra-class performance of our segmentation quality estimation remains high, even when less than 1,000 annotations are used for training (Fig. 9). As shown in Fig. 10, the estimation performance degrades the further the target class is away from the training class. Yet, even when training on an animal class and testing on a vehicle (or vice versa), the mean/median estimation error was still below 0.1 and the number of annotations compared to conventional majority voting can be reduced by 50 %.

4.4 Comparison of segmentation costs

Annotation costs of our method compared to majority voting and the manual grading method are shown in Fig. 11 and Fig. 12, respectively.

It can be seen that our method outperforms the baseline method (majority voting) in terms of costs when a typical number of $> 1,000$ annotations is acquired. Our approximations further indicate that the manual grading method for contour drawing is more expensive than our method instantiated with $\lambda = 1$ and less expensive for $\lambda \geq 2$ when assuming the percentage of spam to be $\sim 25\%$. Note that the costs v for verification required by manual grading were not considered in our analysis (as the numbers were not available). As our method may potentially be combined with a verification step as well, instantiation with $\lambda = 1$ would be feasible.

5 DISCUSSION

To our knowledge, we have presented the first approach to crowd-based object segmentation that uses annotation process-based features derived from clickstreams for quality control. In contrast to previous approaches, this enables us to estimate segmentation quality without using any prior knowledge of specific workers and without having to perform any additional tasks depending on known reference data once the segmentation quality estimation is trained. Our work was inspired by user behavior analysis with clickstreams [33], [56], [57], [58] which has already successfully been explored outside the field of crowdsourcing for e-commerce applications [59], social networks [32] and web browsing behavior analysis [60], [61].

Our experiments support the following three hypotheses:

- 1) Clickstream features are very good predictors for segmentation quality
- 2) These features generalize over (similar) object classes

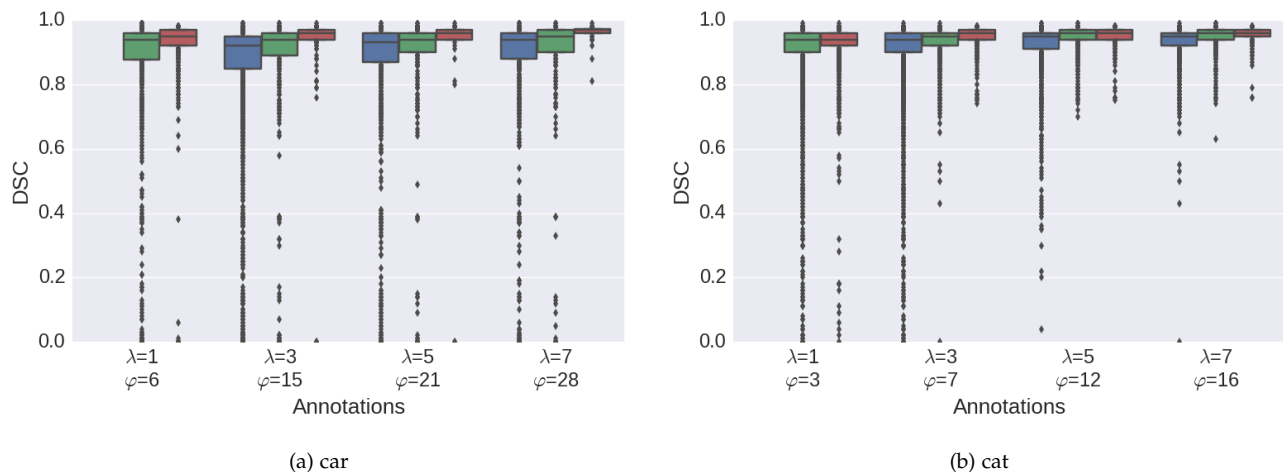


Fig. 7: Confidence-weighted majority voting (red) compared to conventional majority voting with λ annotations (blue) and φ annotations (green) for different training and testing classes (car and/or cat), where φ represents the average number of annotations to obtain λ annotations with an estimated DSC above ϵ_t . Performance is assessed for an estimated DSC threshold of $\epsilon_t = 0.9$ and a varying number of annotations λ . For clarity only subsets of the experiments ($\lambda \in \{1, 3, 5, 7\}$) are visualized.

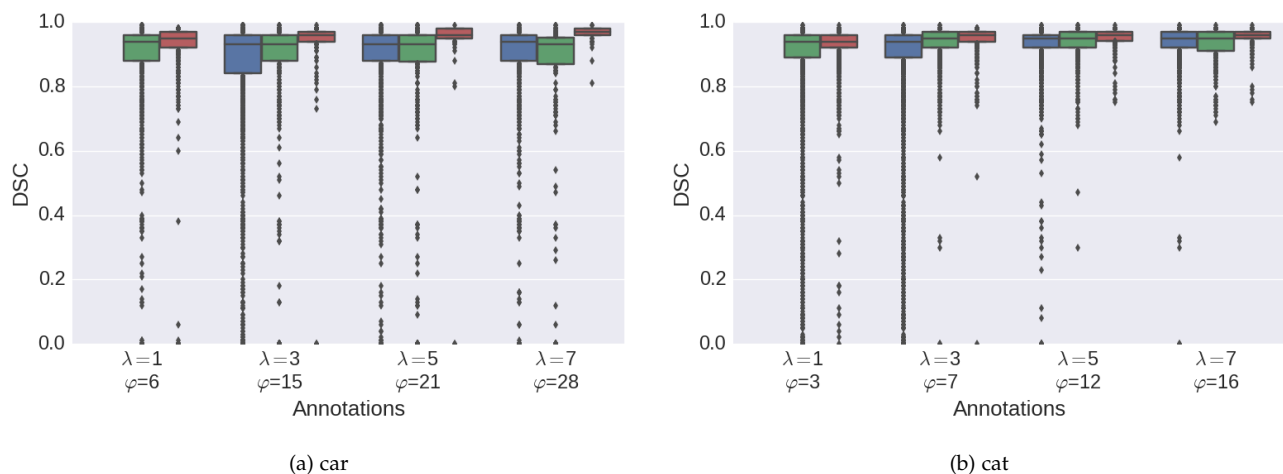


Fig. 8: STAPLE with segmentation quality estimation (red) compared to conventional STAPLE with λ annotations (blue) and φ annotations (green) for different training and testing classes (car and/or cat), where φ represents the average number of annotations to obtain λ annotations with an estimated DSC above ϵ_t . Performance is assessed for an estimated DSC threshold of $\epsilon_t = 0.9$ and a varying number of annotations λ . For clarity only subsets of the experiments ($\lambda \in \{1, 3, 5, 7\}$) are visualized.

- 3) Clickstream-based annotation quality estimation can be applied for confidence-based annotation merging

In a scenario where bad quality segmentations can be directly rejected without rewarding the workers, we were able to achieve the same segmentation accuracies while reducing the crowd annotation costs by up to 75% compared to conventional methods. Even when using only a single annotation through our segmentation quality estimation we were already able to outperform the baseline methods in terms of segmentation quality. Importantly, our experiments related to feature selection suggest that the annotation

process-based features are crucial for the success of our method. In addition, our presented method is resilient to outliers by rejecting bad segmentations through the quality estimation step and weighting them based on their estimated quality, unlike to conventional majority voting and the STAPLE algorithm. We can also show that our regressor does not need to be trained on the object class it is applied to, but generalizes well across classes, rendering the costs required for training the regressor negligible. According to initial experiments presented in Sec. 4.3 the training class should be chosen as closely as possible to the testing class.

To create highly accurate object segmentations it is crucial that inaccurate annotations are detected with a high

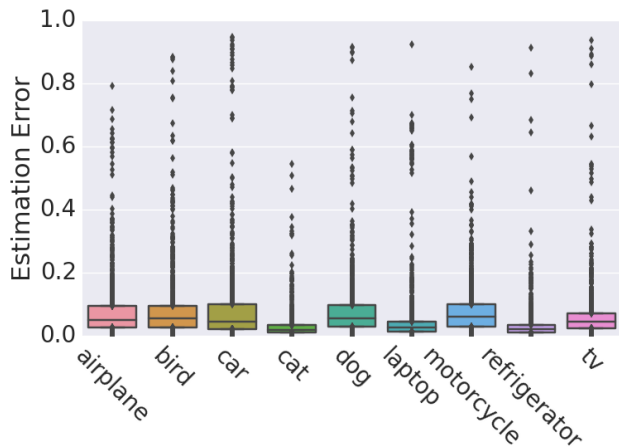


Fig. 9: Intra-class segmentation quality estimation performance for all classes (see Sec. 4.3).



Fig. 10: Descriptive statistics of the quality estimation error when training on animals or vehicles and testing on (1) the same class, (2) the same category (here: different animals or vehicles), (3) a similar category (here: vehicles or animals) and (4) a different category (here: rectangular-shaped objects).

reliability. The majority of estimations with incorrect high quality estimations could be traced back to accurate segmentations of the wrong object on the car data set (Fig. 4a). This occurred particularly frequently on rather difficult tasks, e.g. where the car is hidden in the background (Fig. 4a) or partially occluded and the workers tend to segment foreground objects like pedestrians, animals or motorcycles in addition to or instead of the object of interest. In contrast to the cat data set, where the object of interest is mostly visible in the foreground, some images of cars were taken in an environment with traffic showing a higher amount of different objects. For example trucks and busses are considered as a different class, but pick-up trucks and mini vans belong to the class car. This can be misleading for some workers and in some images cars and trucks are hard to distinguish from each other. In this case the workers tend to

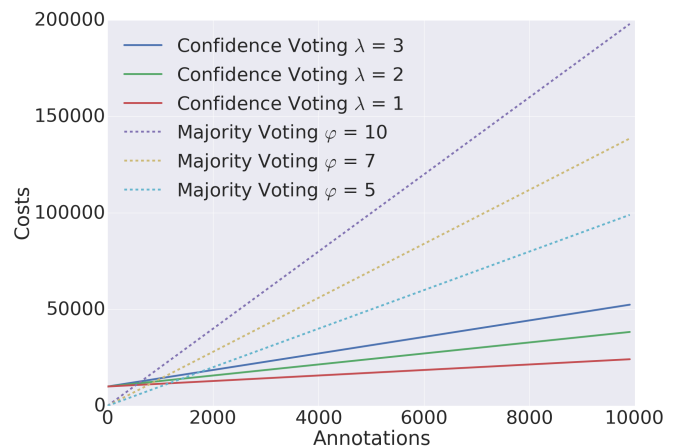


Fig. 11: Comparison of the annotation costs of the proposed method for different λ with a baseline method based on majority voting for different φ with $s = 30\%$ of spam, $a_t = 10,000$ training annotations for the proposed method, and $a_w = 10$ quality control tasks for the baseline method. The annotation costs are plotted for the number of requested annotations.

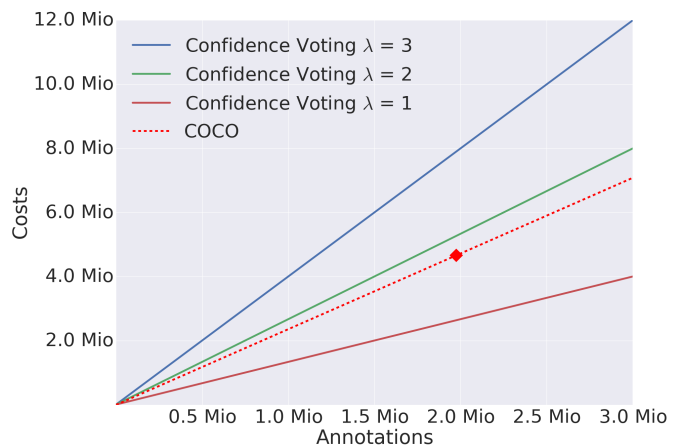


Fig. 12: Comparison of the annotation costs of the proposed method for different λ with the estimated costs of the approach applied by the method of Lin et al. [9] (COCO) with $s = 24.9\%$ of spam. The annotation costs are plotted for the number of segmentations requested. The diamond represents the estimated total cost for annotating the full COCO data set.

create accurate segmentations of both vehicles, while only the segmentation of the car is included in the reference data. In contrast to the car data set, the cat data set does not suffer from particularly difficult tasks, but has a slightly higher amount of misclassified bounding box segmentations and spam. With the low overall estimation error (Fig. 5a), the amount of misclassified spam can be considered as negligible. The largest proportion of erroneous classifications on the images of cats was produced by workers that used the segmentation tool in a wrong way, e.g. inverted segmentations (Fig. 4b), drawing outlines with polygons. It should be

pointed out that all of these problems are related to issues with the annotator instructions rather than to the quality estimation method itself. Of note, the method was designed for estimating the quality of a single object segmentation. It can easily be extended with instance spotting step to point the worker to the object of interest and assure the segmentation the correct object, as presented in [9]. Clear task instructions and training workers with tutorials might also help to minimize further problems [62].

We are currently also not considering the history of a specific worker. It is likely that a worker who has provided low-quality annotations several times will not be a reliable worker for future tasks. A pre-selection of workers could be performed in this case to achieve the desired segmentation result more quickly. We opted not to perform additional steps for quality assurance to better validate our method. Furthermore, spammers trying to cheat the system can create new accounts as soon they are blocked and will continue spamming the system until they are identified. Using a similar mouse dynamic based features set classification approach as described in Feher et al. [31] for user identity verification and taking into account the work for clickstream based sybil account detection in online communities [32] and user behaviour clustering [33] our concept could be extended to detect known malicious workers on different accounts.

The presented segmentation concept is rather general and open in the way it can be implemented. For this paper we chose standard and widely used methods for the individual components. We believe that more sophisticated approaches for annotation merging, like the maximum a-posteriori STAPLE algorithm approach [63] for merging single object annotations could further improve segmentation quality.

Given that the chosen features only rely on the clickstream data and gradient information extracted from the image and that no absolute pixel values are considered, we believe that our method can be applied to a wider range of different domains such as bio-medical imaging without needing to retrain the classifier. In view of the lack of publicly available reference data in the bio-medical imaging domain [64], this could result in a huge benefit for data annotation in this field. Additionally, future work will investigate adapting our method to other classes of annotation, such as localization using bounding boxes. Finally, methods for crowd-algorithm collaboration could be investigated (e.g. [65], [66]) to further reduce annotation costs. In conclusion, we believe that our method has a great potential for use in large-scale low-cost data annotation.

ACKNOWLEDGEMENTS

The authors would like to thank Caroline Feldmann for her help with the figures and Pallas Ludens GmbH for supporting us with their crowdsourcing platform. This work has been financed by the Klaus Tschira Foundation (project: "Endoskopie meets Informatik – Präzise Navigation für die minimalinvasive Chirurgie") and SFB/TRR 125 - Cognition-Guided Surgery. Finally, the authors would like to thank Esther Stenau, Sebastian Wirkert, Caroline Feldmann, Janek Gröhl, Annika Reinke, Hellena Hempe, Sarina Thomas,

Clemens Hentschke and Matthias Baumhauer for helping with the data acquisition as well as Tsung-Yi Lin from Cornell Tech, Cornell University for providing further statistics on the COCO data set.

REFERENCES

- [1] A. Luckow, K. Kennedy, F. Manhardt, E. Djerekarov, B. Vorster, and A. Apon, "Automotive big data: Applications, workloads and infrastructures," in *Proc. IEEE International Conference on Big Data (Big Data)*, pp. 1201–1210, 2015.
- [2] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "ImageNet: A Large-Scale Hierarchical Image Database," in *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, 2009.
- [3] L. Zhou, L. Wang, Q. Wang, and Y. Shi, eds., *Machine Learning in Medical Imaging*, vol. LNCS 9352, 2015.
- [4] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Advances in Neural Information Processing Systems*, pp. 1097–1105, 2012.
- [5] S. Albarqouni, C. Baur, F. Achilles, V. Belagiannis, S. Demirci, and N. Navab, "Aggnet: Deep learning from crowds for mitosis detection in breast cancer histology images," *IEEE Transactions on Medical Imaging*, vol. 35, no. 5, pp. 1313–1321, 2016.
- [6] V. C. Raykar, S. Yu, L. H. Zhao, G. H. Valadez, C. Florin, L. Bogoni, and L. Moy, "Learning from crowds," *Journal of Machine Learning Research*, vol. 11, pp. 1297–1322, 2010.
- [7] G. Kazai, J. Kamps, and N. Milic-Frayling, "Worker types and personality traits in crowdsourcing relevance labels," in *Proc. ACM International Conference on Information and Knowledge Management*, pp. 1941–1944, 2011.
- [8] L. Von Ahn, B. Maurer, C. McMillen, D. Abraham, and M. Blum, "recaptcha: Human-based character recognition via web security measures," *Science*, vol. 321, no. 5895, pp. 1465–1468, 2008.
- [9] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick, "Microsoft COCO: Common objects in context," in *Computer Vision – ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6–12, 2014, Proceedings, Part V*, pp. 740–755, 2014.
- [10] D. Oleson, A. Sorokin, G. P. Laughlin, V. Hester, J. Le, and L. Biewald, "Programmatic gold: Targeted and scalable quality assurance in crowdsourcing," *Human computation*, vol. 11, no. 11, pp. 43–48, 2011.
- [11] F. Cabezas, A. Carlier, V. Charvillat, A. Salvador, and X. G. i Nieto, "Quality control in crowdsourced object segmentation," in *2015 IEEE International Conference on Image Processing (ICIP)*, pp. 4243–4247, Sept 2015.
- [12] S. Vijayanarasimhan and K. Grauman, "Large-scale live active learning: Training object detectors with crawled data and crowds," *International Journal of Computer Vision*, vol. 108, no. 1–2, pp. 97–114, 2014.
- [13] L. Von Ahn and L. Dabbish, "Labeling images with a computer game," in *Proc. SIGCHI Conference on Human Factors in Computing Systems*, pp. 319–326, 2004.
- [14] L. Maier-Hein, S. Mersmann, D. Kondermann, S. Bodenstedt, A. Sanchez, C. Stock, H. G. Kennigott, M. Eisenmann, and S. Speidel, "Can masses of non-experts train highly accurate image classifiers?," in *Proc. International Conference on Medical Image Computing and Computer-Assisted Intervention*, pp. 438–445, 2014.
- [15] D. Gurari, D. Theriault, M. Sameki, B. Isenberg, T. A. Pham, A. Purwada, P. Solski, M. Walker, C. Zhang, J. Y. Wong, et al., "How to collect segmentations for biomedical images? a benchmark evaluating the performance of experts, crowdsourced non-experts, and algorithms," in *Proc. IEEE Winter Conference on Applications of Computer Vision*, pp. 1169–1176, 2015.
- [16] W. Wang, Y. Zhu, X. Huang, D. Lopresti, Z. Xue, R. Long, S. Antani, and G. Thoma, "A classifier ensemble based on performance level estimation," in *Proc. IEEE International Symposium on Biomedical Imaging: From Nano to Macro*, pp. 342–345, 2009.
- [17] B. C. Russell, A. Torralba, K. P. Murphy, and W. T. Freeman, "Labelme: A database and web-based tool for image annotation," *International Journal of Computer Vision*, vol. 77, pp. 157–173, May 2008.
- [18] S. Vittayakorn and J. Hays, "Quality assessment for crowdsourced object annotations," in *Proc. of British Machine Vision Conference*, pp. 109.1–109.11, 2011.

- [19] P. Welinder and P. Perona, "Online crowdsourcing: rating annotators and obtaining cost-effective labels," in *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, 2010.
- [20] C. Long, G. Hua, and A. Kapoor, "A joint gaussian process model for active visual recognition with expertise estimation in crowdsourcing," *International Journal of Computer Vision*, vol. 116, no. 2, pp. 136–160, 2016.
- [21] P. Welinder, S. Branson, P. Perona, and S. J. Belongie, "The multi-dimensional wisdom of crowds," in *Advances in neural information processing systems*, pp. 2424–2432, 2010.
- [22] D. Gurari, S. D. Jain, M. Betke, and K. Grauman, "Pull the plug? predicting if computers or humans should segment images," in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 382–391, June 2016.
- [23] J. Carreira and C. Sminchisescu, "Constrained parametric min-cuts for automatic object segmentation," in *2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pp. 3241–3248, June 2010.
- [24] P. Arbelez, J. Pont-Tuset, J. Barron, F. Marques, and J. Malik, "Multiscale combinatorial grouping," in *2014 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 328–335, June 2014.
- [25] S. Dutt Jain and K. Grauman, "Predicting sufficient annotation strength for interactive foreground segmentation," in *The IEEE International Conference on Computer Vision (ICCV)*, December 2013.
- [26] S. D. Jain and K. Grauman, "Active image segmentation propagation," in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 2864–2873, June 2016.
- [27] M. Sameki, D. Gurari, and M. Betke, "Predicting quality of crowd-sourced image segmentations from crowd behavior," in *Proc. AAAI Conference on Human Computation and Crowdsourcing*, 2015.
- [28] L. Maier-Hein, D. Kondermann, T. Roß, S. Mersmann, E. Heim, S. Bodenstedt, H. G. Kemngott, A. Sanchez, M. Wagner, A. Preukschas, et al., "Crowdtruth validation: a new paradigm for validating algorithms that rely on image correspondences," *International Journal of Computer Assisted Radiology and Surgery*, vol. 10, no. 8, pp. 1201–1212, 2015.
- [29] A. Mao, E. Kamar, Y. Chen, E. Horvitz, M. E. Schwamb, C. J. Lintott, and A. M. Smith, "Volunteering versus work for pay: Incentives and tradeoffs in crowdsourcing," in *Proc. AAAI Conference on Human Computation and Crowdsourcing*, 2013.
- [30] A. A. E. Ahmed and I. Traore, "A new biometric technology based on mouse dynamics," *IEEE Transactions on Dependable and Secure Computing*, vol. 4, no. 3, p. 165, 2007.
- [31] C. Feher, Y. Elovici, R. Moskovitch, L. Rokach, and A. Schlar, "User identity verification via mouse dynamics," *Information Sciences*, vol. 201, pp. 19–36, 2012.
- [32] G. Wang, T. Konolige, C. Wilson, X. Wang, H. Zheng, and B. Y. Zhao, "You are how you click: Clickstream analysis for sybil detection," in *Proc. USENIX Security*, pp. 1–15, 2013.
- [33] G. Wang, X. Zhang, S. Tang, H. Zheng, and B. Y. Zhao, "Unsupervised clickstream clustering for user behavior analysis," in *Proc. SIGCHI Conference on Human Factors in Computing Systems*, 2016.
- [34] K. Lee, S. Webb, and H. Ge, "Characterizing and automatically detecting crowd-turfing in fiverr and twitter," *Social Network Analysis and Mining*, vol. 5, no. 1, pp. 1–16, 2015.
- [35] J. M. Rzeszotarski and A. Kittur, "Instrumenting the crowd: using implicit behavioral measures to predict task performance," in *Proc. of the 24th annual ACM symposium on User interface software and technology*, pp. 13–22, ACM, 2011.
- [36] L. R. Dice, "Measures of the amount of ecologic association between species," *Ecology*, vol. 26, no. 3, pp. 297–302, 1945.
- [37] J. H. Friedman, J. L. Bentley, and R. A. Finkel, "An algorithm for finding best matches in logarithmic expected time," *ACM Transactions on Mathematical Software*, vol. 3, no. 3, pp. 209–226, 1977.
- [38] R. Deriche, "Recursively implementing the Gaussian and its derivatives," Research Report RR-1893, INRIA, 1993.
- [39] L. Ibanez, W. Schroeder, L. Ng, and J. Cates, *The ITK Software Guide*. Kitware, Inc., second ed., 2005. ISBN 1-930934-15-7.
- [40] T. K. Ho, "Random decision forests," in *Proc. IEEE International Conference on Document Analysis and Recognition*, vol. 1, pp. 278–282, 1995.
- [41] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, et al., "Scikit-learn: Machine learning in python," *The Journal of Machine Learning Research*, vol. 12, pp. 2825–2830, 2011.
- [42] D. Kondermann, R. Nair, S. Meister, W. Mischler, B. Gusefeld, K. Honauer, S. Hofmann, C. Brenner, and B. Jähne, "Stereo ground truth with error bars," in *Proc. Asian Conference on Computer Vision*, pp. 595–610, 2014.
- [43] O. Commowick and S. K. Warfield, "Incorporating priors on expert performance parameters for segmentation validation and label fusion: a maximum a posteriori STAPLE," in *Proc. Medical Image Computing and Computer-Assisted Intervention*, pp. 25–32, Springer, 2010.
- [44] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman, "The Pascal Visual Object Classes (VOC) Challenge," *International Journal of Computer Vision*, vol. 88, no. 2, pp. 303–338, 2010.
- [45] A. W. Whitney, "A direct method of nonparametric measurement selection," *IEEE Transactions on Computers*, vol. 100, no. 9, pp. 1100–1103, 1971.
- [46] R. Kohavi and G. H. John, "Wrappers for feature subset selection," *Artificial intelligence*, vol. 97, no. 1–2, pp. 273–324, 1997.
- [47] F. Fleuret, "Fast binary feature selection with conditional mutual information," *Journal of Machine Learning Research*, vol. 5, no. Nov, pp. 1531–1555, 2004.
- [48] A. Jakulin and I. Bratko, *Machine learning based on attribute interactions: phd dissertation*. PhD thesis, Univerza v Ljubljani, 2005.
- [49] H. H. Yang and J. Moody, "Feature selection based on joint mutual information," in *In Proceedings of International ICSC Symposium on Advances in Intelligent Data Analysis*, pp. 22–25, 1999.
- [50] D. Lin and X. Tang, "Conditional infomax learning: An integrated framework for feature extraction and fusion," in *Proceedings of the 9th European Conference on Computer Vision - Volume Part I, ECCV'06*, pp. 68–82, 2006.
- [51] R. Battiti, "Using mutual information for selecting features in supervised neural net learning," *IEEE Transactions on neural networks*, vol. 5, no. 4, pp. 537–550, 1994.
- [52] A. Kraskov, H. Stögbauer, and P. Grassberger, "Estimating mutual information," *Physical review E*, vol. 69, no. 6, p. 066138, 2004.
- [53] E. Hazzard, *Openlayers 2.10 beginner's guide*. Packt Publishing Ltd, 2011.
- [54] A. Kittur, E. H. Chi, and B. Suh, "Crowdsourcing user studies with mechanical turk," in *Proceedings of the SIGCHI conference on human factors in computing systems*, pp. 453–456, ACM, 2008.
- [55] J. Bragg, D. S. Weld, et al., "Optimal testing for crowd workers," in *Proceedings of the 2016 International Conference on Autonomous Agents and Multiagent Systems*, pp. 966–974, International Foundation for Autonomous Agents and Multiagent Systems, 2016.
- [56] I.-H. Ting, C. Kimble, and D. Kudenko, "Ubb mining: finding unexpected browsing behaviour in clickstream data to improve a web site's design," in *Web Intelligence, 2005. Proceedings. The 2005 IEEE/WIC/ACM International Conference on*, pp. 179–185, IEEE, 2005.
- [57] J. Srivastava, R. Cooley, M. Deshpande, and P.-N. Tan, "Web usage mining: Discovery and applications of usage patterns from web data," *Acm Sigkdd Explorations Newsletter*, vol. 1, no. 2, pp. 12–23, 2000.
- [58] Ş. Gündüz and M. T. Özsu, "A web page prediction model based on click-stream tree representation of user behavior," in *Proceedings of the ninth ACM SIGKDD international conference on Knowledge discovery and data mining*, pp. 535–540, ACM, 2003.
- [59] Q. Su and L. Chen, "A method for discovering clusters of e-commerce interest patterns using click-stream data," *Electronic Commerce Research and Applications*, vol. 14, no. 1, pp. 1–13, 2015.
- [60] J. Heer and E. H. Chi, "Separating the swarm: categorization methods for user sessions on the web," in *Proceedings of the SIGCHI Conference on Human factors in Computing Systems*, pp. 243–250, ACM, 2002.
- [61] H. Obendorf, H. Weinreich, E. Herder, and M. Mayer, "Web page revisitation revisited: implications of a long-term click-stream study of browser usage," in *Proceedings of the SIGCHI conference on Human factors in computing systems*, pp. 597–606, ACM, 2007.
- [62] L. Gottlieb, J. Choi, P. Kelm, T. Sikora, and G. Friedland, "Pushing the limits of mechanical turk: qualifying the crowd for video geo-location," in *Proc. ACM Multimedia 2012 Workshop on Crowdsourcing for Multimedia*, pp. 23–28, ACM, 2012.
- [63] O. Commowick, A. Akhondi-Asl, and S. K. Warfield, "Estimating a reference standard segmentation with spatially varying performance parameters: local MAP STAPLE," *IEEE Transactions on Medical Imaging*, vol. 31, no. 8, pp. 1593–1606, 2012.

- [64] H. Greenspan, B. van Ginneken, and R. M. Summers, "Guest editorial deep learning in medical imaging: Overview and future promise of an exciting new technique," *IEEE Transactions on Medical Imaging*, vol. 35, no. 5, pp. 1153–1159, 2016.
- [65] L. Maier-Hein, T. Ross, J. Gröhl, B. Glocker, S. Bodenstedt, C. Stock, E. Heim, M. Götz, S. Wirkert, H. Kennigott, *et al.*, "Crowd-algorithm collaboration for large-scale endoscopic image annotation with confidence," in *Proc. International Conference on Medical Image Computing and Computer-Assisted Intervention*, pp. 616–623, 2016.
- [66] A. Carlier, V. Charvillat, A. Salvador, X. Giro-i Nieto, and O. Marques, "Click'n'cut: crowdsourced interactive segmentation with object candidates," in *Proc. International ACM Workshop on Crowdsourcing for Multimedia*, pp. 53–56, 2014.



Eric Heim received his MSc in Applied Computer Science from the University of Heidelberg in 2013. He is currently working in the Division of Computer Assisted Medical Interventions at the German Cancer Research Center (DKFZ) in Heidelberg towards his PhD degree at the Interdisciplinary Center for Scientific Computing (IWR) University of Heidelberg. His research focuses on crowdsourcing enhanced algorithms in the context of medical procedures.



Alexander Seitel received his Doctorate in Medical Informatics from DKFZ / University of Heidelberg in 2012 and holds a Diploma degree in Computer Science from the Karlsruhe Institute of Technology (2007). He is currently a post-doctoral researcher at the German Cancer Research Center (DKFZ) focussing on computer-assisted interventions in particular navigation solutions for percutaneous needle insertions.



Jonas Andrulis received a MSc in Economics Engineering from Karlsruhe Institute of Technology KIT in 2008 specialising in statistical modeling. In his masters thesis he evaluated counterparty reputation risk in a multi-layer Bayesian network. Since then he increasingly investigated user-incorporating models to support ground truth generation with uncertainty.



Fabian Isensee holds a MSc in Molecular Biotechnology from the University of Heidelberg. In his masters thesis he investigated feature selection and feature importance in the context of image segmentation. He is currently working as a PhD student in the Division of Medical Image Computing at the German Cancer Research Center (DKFZ) where his research focuses on the development of deep learning algorithms for medical image classification and segmentation.



applied statistics for program and method evaluation studies in the biomedical field.

Christian Stock holds an MSc in Biostatistics from Heidelberg University, a MSc in Health Services Research from York University (UK), and a Doctorate from Heidelberg University. He is a senior statistician and epidemiologist at the Division of Clinical Epidemiology and Aging Research at the German Cancer Research Center (DKFZ) and at the Institute of Medical Biometry and Informatics at Heidelberg University Hospital. His work focuses on the development of statistical models for complex observational data and on



Tobias Ross received his MSc in Medical Informatics from the University of Heidelberg/Heilbronn. His masters thesis was about crowd-algorithm collaboration for large-scale image-annotation. He is currently working in the Division of Computer Assisted Medical Interventions at the German Cancer Research Center (DKFZ) in Heidelberg as a PhD student. His research focuses on robotics and surgical workflow analysis.



Lena Maier-Hein received her PhD from Karlsruhe Institute of Technology KIT with distinction in 2009 and conducted her postdoctoral research in the Division of Medical and Biological Informatics at the German Cancer Research Center (DKFZ) and at the Hamlyn Centre for Robotics Surgery at Imperial College London. As a full professor at the DKFZ, she is now working in the field of computer assisted medical interventions, focusing on multi-modal image processing, surgical data science and computational biophotonics. She has and continues to fulfill the role (co-) principal investigator on a number of national and international grants including a European Research Council (ERC) starting grant 2014.