CLICKSTREAM ANALYSIS FOR CROWD-BASED OBJECT SEGMENTATION WITH CONFIDENCE

Seminar Report

Submitted in partial fulfillment of the requirements for the award of degree of

BACHELOR OF TECHNOLOGY

In

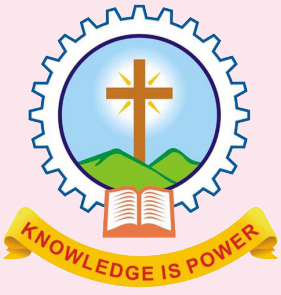
COMPUTER SCIENCE AND ENGINEERING

of

APJ ABDUL KALAM TECHNOLOGICAL UNIVERSITY

Submitted By

DEEPUL NAIR



Department of Computer Science & Engineering
Mar Athanasius College Of Engineering Kothamangalam

CLICKSTREAM ANALYSIS FOR CROWD-BASED OBJECT SEGMENTATION WITH CONFIDENCE

Seminar Report

Submitted in partial fulfillment of the requirements for the award of degree of

BACHELOR OF TECHNOLOGY

In

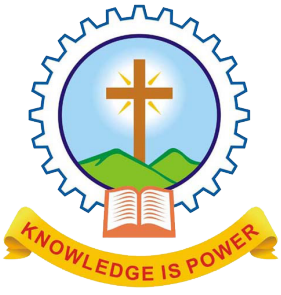
COMPUTER SCIENCE AND ENGINEERING

of

APJ ABDUL KALAM TECHNOLOGICAL UNIVERSITY

Submitted By

DEEPUL NAIR



Department of Computer Science & Engineering
Mar Athanasius College Of Engineering Kothamangalam

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING
MAR ATHANASIUS COLLEGE OF ENGINEERING
KOTHAMANGALAM

CERTIFICATE

This is to certify that the report entitled **Clickstream Analysis For Crowd-Based Object Segmentation with Confidence** *submitted by* **Mr. DEEPUL NAIR**, *Reg.No.***MAC15CS024** *towards partial fulfillment of the requirement for the award of Degree of Bachelor of Technology in Computer science and Engineering from APJ Abdul Kalam Technological University for December 2018 is a bonafide record of the seminar carried out by him under our supervision and guidance.*

.....	.....	.....
Prof. Joby George	**Prof. Neethu Subash**	**Dr. Surekha Mariam Varghese**
Faculty Guide	***Faculty Guide***	***Head Of Department***

Date:

Dept. Seal

ACKNOWLEDGEMENT

First and foremost, I sincerely thank the 'God Almighty' for his grace for the successful and timely completion of the seminar.

I express my sincere gratitude and thanks to Dr. Solly George, Principal and Dr. Surekha Mariam Varghese, Head Of the Department for providing the necessary facilities and their encouragement and support.

I owe special thanks to the staff-in-charge Prof. Joby george, Prof. Neethu Subash and Prof. Joby Anu Mathew for their corrections, suggestions and sincere efforts to co-ordinate the seminar under a tight schedule.

I express my sincere thanks to staff members in the Department of Computer Science and Engineering who have taken sincere efforts in helping me to conduct this seminar.

Finally, I would like to acknowledge the heartfelt efforts, comments, criticisms, co-operation and tremendous support given to me by my dear friends during the preparation of the seminar and also during the presentation without whose support this work would have been all the more difficult to accomplish.

Abstract

The availability of reference annotations for algorithm training is one of the major bottlenecks in the field of machine learning based solutions for automatic image annotation. Crowdsourcing has evolved as a valuable option for low-cost and large-scale data annotation but quality control remains a major issue which needs to be addressed. The paper analyze the annotation process to improve crowd-sourced image segmentation. The 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 quality when merging multiple segmentations of one image. The method is highly accurate in estimating the segmentation quality based on clickstream data. The method outperforms state-of-the-art methods for merging multiple annotations.

Contents

List of Figures

List Of Abbreviation

BFS	Best First Search
DSC	DICE similarity coefficient
STAPLE	Simultaneous truth and performance level estimation
CMIM	Conditional Mutual Information Maximization
ICAP	Interaction Capping

Introduction

The interest in machine learning techniques for data processing has been rapidly growing in various fields including the automotive industry,computer vision and bio-medical image processing. 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. 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.It provides a valuable tool for low-cost and large-scale data annotation.

One of the major challenges in crowdbased 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.The 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 segmentation 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. 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.

Image annotation is a difficult task for two main reasons: First is the well-known pixel-to-predicate or semantic gap problem, which points to the fact that it is hard to extract semantically meaningful entities using just low level image features, e.g. color and texture. Doing explicit recognition of thousands of objects or classes reliably is currently an unsolved problem. The second difficulty arises due to the lack of correspondence between the keywords and

image regions in the training data. For each image, one has access to keywords assigned to the entire image and it is not known which regions of the image correspond to these keywords. This makes difficult the direct learning of classifiers by assuming each keyword to be a separate class shown in figure.

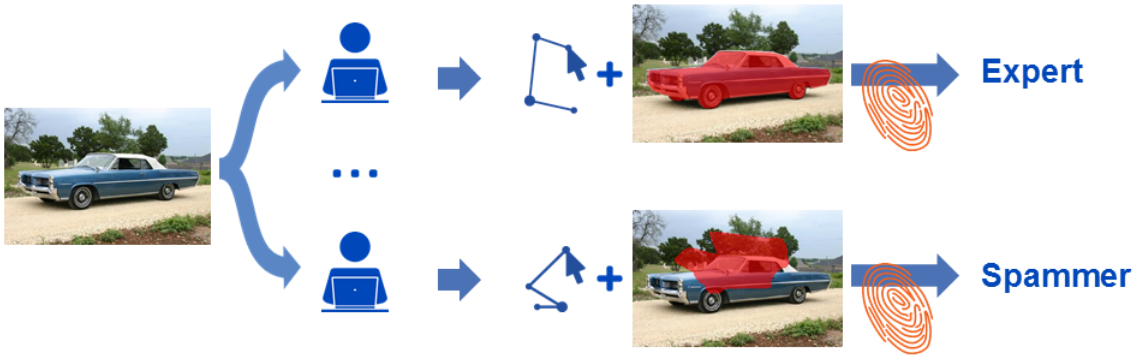


Figure 1.1: The difference between spam and expert.

The advantages of automatic image annotation versus content-based image retrieval are that queries can be more naturally specified by the user.CBIR generally requires users to search by image concepts such as color and texture, or finding example queries. Certain image features in example images may override the concept that the user is really focusing on. The traditional methods of image retrieval such as those used by libraries have relied on manually annotated images, which is expensive and time-consuming, especially given the large and constantly growing image databases in existence.

The alternative of doing explicit identification of thousands of image objects or classes reliably is currently an unsolved problem. The second difficulty is due to the lack of association

between the image regions and keywords in the training data. For each image, one has access to the keywords assigned to the entire image and it is not known which regions of the image correspond to these keywords. Types Of Automatic Image Annotation Techniques:

1 Single labelling annotation using binary classification-Low Level features are extracted and fed to Binary Classifier. 1.a Image annotation using Support Vector Machines (SVM) 1.b Image annotation using Artificial Neural Network 1.c Image annotation using Decision Tree

2 Multi labelling annotation using Bayesian Methods-Multi–instance multi label(MIML) concept is followed. 2.a Non Parametric approach 2.b Parametric approach

3.Image annotation incorporating metadata:WWW is a rich source of imagery and text information

The alternative of doing explicit identification of thousands of image objects or classes reliably is currently an unsolved problem. The second difficulty is due to the lack of association between the image regions and keywords in the training data. For each image, one has access to the keywords assigned to the entire image and it is not known which regions of the image correspond to these keywords.

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 and comprises an approach to estimate the segmentation quality based on clickstream analysis and a method for accurate crowd-based object segmentation 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.The final section draws a conclusion.

Related Works

Von Ahn's reCaptcha [1], 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. Weilender[2] proposed a general method to determine a reference value of objects.

In an initial quality control step, Long[3] 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 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.

Cabezas [4] 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.

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 [5]. 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 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 .

A commonly used means for ensuring quality of crowdsourced 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 and COCO . This manual verification task imposes additional annotation costs that could be avoided when quality estimation is performed automatically.

Vittayakorn [1]. investigated several scoring functions to assess the quality of crowdsourced 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 [2] 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 [3]. 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 .

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. Sameki [5] re-

cently 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.

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 for user authentication and intrusion detection in computer systems. In Ahmed [6] 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[7]. 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

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. 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 showed that clickstream analysis can be used to cluster social network users based on their behaviour via an unsupervised learning approach. Branson [7] a 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. Carreira [8]. presented a user-behaviorbased method for estimating the quality of crowdsourcing for classification tasks, comprehensive reading tasks and text generation tasks.

Earlier research used clickstream data for Web Usage Mining . Researchers applied simple methods such as Markov Chains to capture users' navigation paths within a website . However, these models focus on the simple aspects of user behavior and are incapable of

modeling more sophisticated user behavior. Other approaches use clustering techniques to identify user groups that share similar clickstream activities . The resulting clusters can be used to infer user interests or predict future user behaviors . However, existing clustering based models are largely supervised (or semi-supervised), requiring large samples of ground-truth data to train or fine-tune the model parameters . Also, many behavioral models are built as "black boxes" for classification tasks, offering little explanations on how users behave and why . Our work seeks to build unsupervised clickstream behavioral models and produce intuitive explanations on the models.

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.

Propsed System

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 additional tasks (with known outcome) to be performed or prior knowledge of a specific worker's annotation history. Inspired by previous work on clickstream analysis for user identification , 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 , 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. The DICE similarity coefficient (DSC) is used as a measure for segmentation quality, which is defined by comparing a segmentation U to its corresponding reference segmentation V

$$DSC = \frac{2|V \cap U|}{|V| + |U|} \tag{3.1}$$

The obtained segmentations are then merged in a weighted manner, according to their estimated quality. An implementation of this concept is presented in the figure 3.1.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.

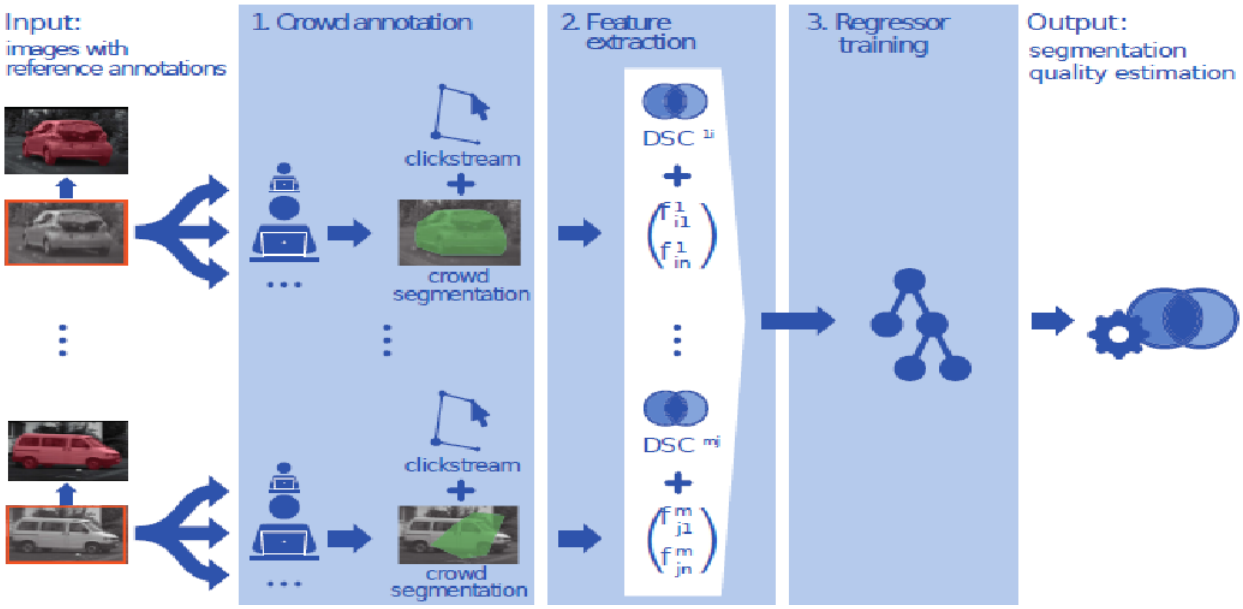


Figure 3.1: Training of segmentation quality estimator.

With our prototype implementation, we focus on singleobject segmentation. The prototype comprises a user interface for crowd-based image segmentation as well as capabilities for clickstream data collection , feature extraction , segmentation quality estimation and confidence-based annotation merging. The concept is shoen in figure 3.2.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.

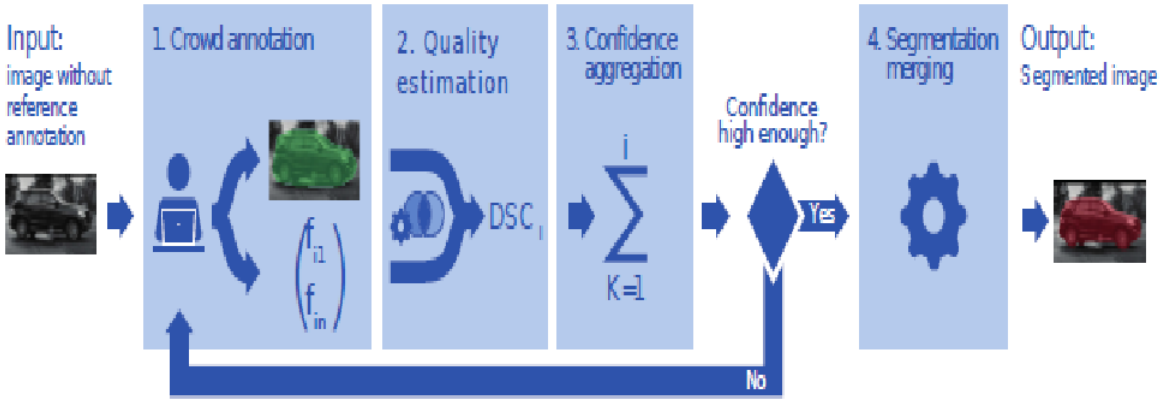


Figure 3.2: Concept for crowd-based image segmentation based on a trained segmentation quality estimator.

3.2 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.

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. The clickstream events are sorted by their time stamp ti in their recorded chronolog-

ical order. Each recorded event provides x and y coordinates of a point p 2 R2 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, mousemove. The following objects are present in the user interface: canvas, delete contour button, zoom button, save button.

3.3 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 willchange 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.
sectionClickstream-based features

Clickstream-based features are as follows. Velocity:We compute a velocity vector based on the elapsed time and travelled distance on the canvas with the positions of two successive events

$$\vec{vi} = \frac{\Delta \vec{pi}}{\Delta \vec{ti}} \tag{3.2}$$

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 are used as features.

Acceleration: The acceleration of every event in the clickstream is derived from the velocity change and elapsed time between two successive events

$$\vec{ai} = \frac{\Delta\vec{vi}}{\Delta\vec{ti}} \tag{3.3}$$

The mean, median, standard deviation and 95% quantile of the acceleration are used as features.

Zoom: We extract the total number of zoom events out of the clickstream by using the event and object types

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. The time is normalized by the mouse clicks and actions. 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 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 E that occur between a mouse down and up event. 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 (figure 3.3)based on the event types and objects

Data: mouse strokes S, clickstream E
Result: set of draw operations D, set of corrections K
begin
  $i \longleftarrow 0$;
  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

Figure 3.3: The algorithm

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 Si 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 Si is considered as a correction event. (3) Otherwise a new contour is started and the current mouse stroke Si is a draw event. A kd-tree is used to speed up the search of events in the clickstream based on their canvas position.

Similar to Vittayakorn 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 i between the gradient direction at the image coordinates and the mouse-move direction for each event. The gradient is computed using recursive Gaussian filtering as implemented in the Insight Toolkit while the mouse move direction is derived by normalizing the velocity vi for the event

$$\vec{d_i} = \frac{\vec{vi}}{||\vec{vi}||} \tag{3.4}$$

In the first step we compute the angle between gradient and velocity

$$w(\vec{d_i}, \vec{g_x y}) = \cos \frac{\vec{d_i} \cdot \vec{g_x y}}{||\vec{d_i}|| \cdot ||\vec{g_x y}||} \cdot \frac{180}{\Pi} \tag{3.5}$$

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 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 di 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 . The normal for each line segment is calculated with two consecutive vertices.The interpolated vertex normal is calculated by a linear interpolation of the line segment normals of two adjacent line segments

$$\vec{n_i} = \frac{\vec{n_i - 1}}{2|} + \frac{\vec{n_i}}{2} \tag{3.6}$$

With the vertex normal and the gradient directions, we compute angle required with the image gradient and the interpolated vertex normal instead of the drawing direction . We use the mean, standard, deviation, median and 95% quantile of all computed angles as features.

3.4 Estimation of Segmentation Quality

With the set of features introduced and the DSC of segmentations for which a reference segmentation exists, we train a random forest regressor to estimate the quality of unseen crowd segmentations. We use the random forest regressor implementation from scikitlearn to estimate the DSC of an unseen crowd segmentation . 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 . 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 crossvalidation optimizing the R2 score with the estimated DSC , the corresponding real DSC for every segmentation and the mean DSC s of the data set

$$R^2 = \frac{\sum_{j=1}(s_j - \vec{s_j})^2}{\sum_{j=1}(s_j - \vec{s})^2} \tag{3.7}$$

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 R2 score

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 .

Based on the estimated quality, segmentations are discarded if the estimated DSC for a segmentation falls under a predefined DSC threshold . With the estimated DSC values we compute a normalized confidence value for the remaining segmentations:

$$\kappa(\vec{s_j}) = \frac{\vec{s_j} - \varepsilon t}{1 - \varepsilon t} \tag{3.8}$$

We denote the two dimensional segmentations images Uj(x; y) with the width m and height n, where (x,y) is the coordinate of the pixel in the image with Uj(x, y) Each segmentation Uj(x, y) is weighted with the estimated confidence and the confidence-weighted pixel values are accumulated in the image

$$\Delta U(x, y) = \sum_{j=1}^{\lambda} U_j(x, y) \cdot \kappa(\vec{s_j}) \tag{3.9}$$

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

The final confidence-weighted segmentation H(x, y) is calculated by applying a binary decision to each pixel of the image and checking with

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 are discarded based on their estimated quality and the remaining segmentations with a high estimated DSC are fused with the native STAPLE algorithm implementation.

We validated our proposed segmentation concept on a subset of the publicly available PASCAL Visual Object Classes (VOC) . 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 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 R2 score 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 n annotations

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) and Best First Search (BFS) 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) , Interaction Capping (ICAP) , Joint Mutual Information (JMI) , Conditional Infomax Feature Extraction (CIFE) , and Mutual Information for Feature Selection (MIFS) with a nearest neighbor mutual information estimator . 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 selec-

tion criterion as in the wrapper method for determining the ideal number of features. Feature selection was solely carried out on the training set.

We compared our method to create confidence-weighted crowd-sourced segmentations with the widely used majority voting method . Furthermore we compared the STAPLE algorithm using raw crowd generated segmentations with the quality estimation enhanced STAPLE algorithm based approach , 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. 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 annotations per image that had an equal or higher estimated DSC than 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 annotations. In addition, we computed the average number of annotations required to obtain , annotations with a estimated DSC above by computing where r is the mean number of rejected annotations. For each our method was compared to majority voting with annotations. Analogously, our STAPLE approach with quality estimation was compared to the native STAPLE algorithm using annotations respectively. The experiments for annotation merging were conducted separately for each class, i.e. training and testing on only cars or cats.

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 library for the widely used crowdsourcing platform Amazon Mechanical Turk (MTurk)3 . The set of object classes obtained from the VOC validation set, namely fcar; catg, was extended by seven further object classes from the COCO data set yielding the following three object categories: Vehicles fcar; airplane; motorcycleg, animals fcat; bird; dogg, rectangular-shapted objects flaptop; refrigerator; tvg. 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)).

3.5 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

3.5.1 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 am, the number of annotations used to train the regressor at and the percentage of spam s to be expected:

$$c(a) = a_t + (a + \frac{s}{1-s}a) \cdot a_m \tag{3.10}$$

3.5.2 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 aw annotations. For am annotations needed for majority voting typically 10-30% of quality control tasks are mixed in between the tasks . The total cost of annotating a images can then be calculated as follows, also taking into account the number of reference annotations

$$c(a) = a_m a + \frac{a_m a}{a_w - 1}a + a_r \tag{3.11}$$

3.5.3 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 , the number of categories annotations are requested for, the total number of recruited workers , the total number of approved workers 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:

$$c(a) = \frac{a}{1-s} + n_c n_w + A_a n_a \cdot r + v \tag{3.12}$$

3.6 Segmentation Quality Estimation

By filtering the segmentations with our segmentation quality estimation method , 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 .

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. 3.5: (a) Wrong object: The

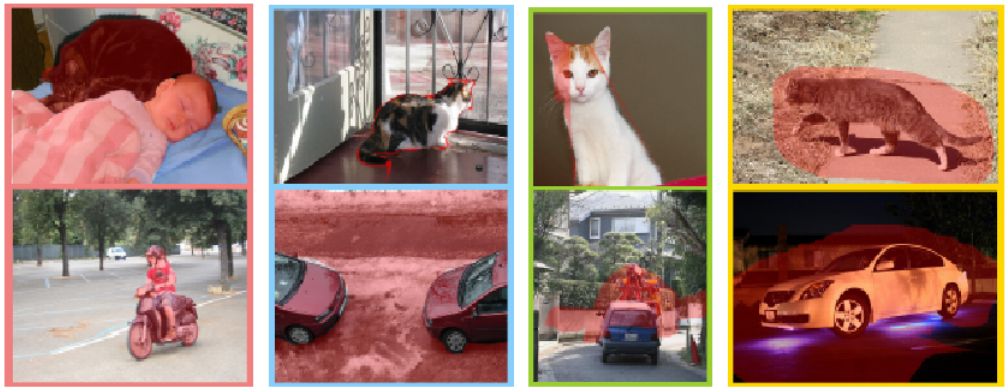


Figure 3.4: Examples of poor estimations.

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 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.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 figure 3.7. The absolute error of quality estimation compariosn is shown in fig 3.5.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.For acccuracy , the median score with image function is taken as shown in fig3.6.The mean estimation error for each feature selection method. The minimal chosen feature set achieved a similar classification performance compared to all features

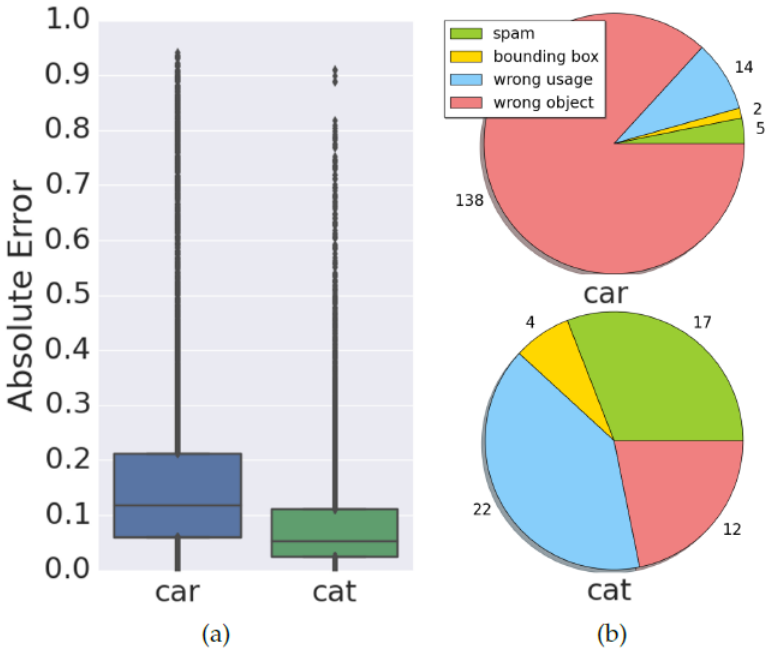


Figure 3.5: Absolute error of the estimated segmentation quality for training and testing on the same class.

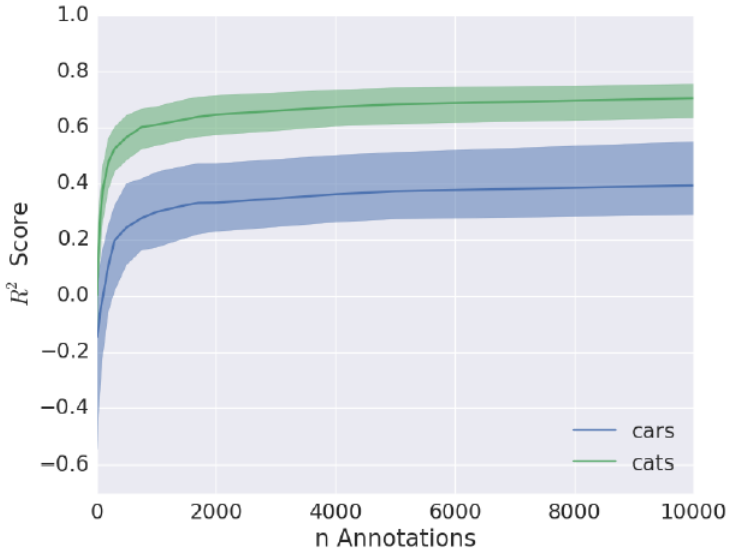


Figure 3.6: Median R2 score and IQR as a function of the number of images

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

Figure 3.7: Mean estimation error for each feature selection method.

The resulting DSC values for different numbers of annotations,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 outlines.

Our presented approach for confidence-weighted majority voting produced statistically significant better results compared to conventional majority voting for the same amount of annotations . 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 , the number of annotations was reduced by 75% on average.

When we used the STAPLE algorithm in conjunction with our segmentation quality estimation statistically significant better results were produced compared to the native STAPLE algorithm for the same amount of annotations . 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 , the number of annotations was reduced by 73% on average.

Mean differences between conventional majority voting and confidence-weighted majority voting and mean differences between the STAPLE algorithm 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 2 f1; ; 10g 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.

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 . As shown in Fig.3.10, 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

Annotation costs of our method compared to majority voting and the manual grading method are shown 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 and less expensive for when assuming the

percentage of spam to be 025%. 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 would be feasible.

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 which has already successfully been explored outside the field of crowdsouring for e-commerce applications , social networks and web browsing behavior analysis . The figures3.8 shown comparison of methods.For clarity, taking car as class,shown fig 3.9

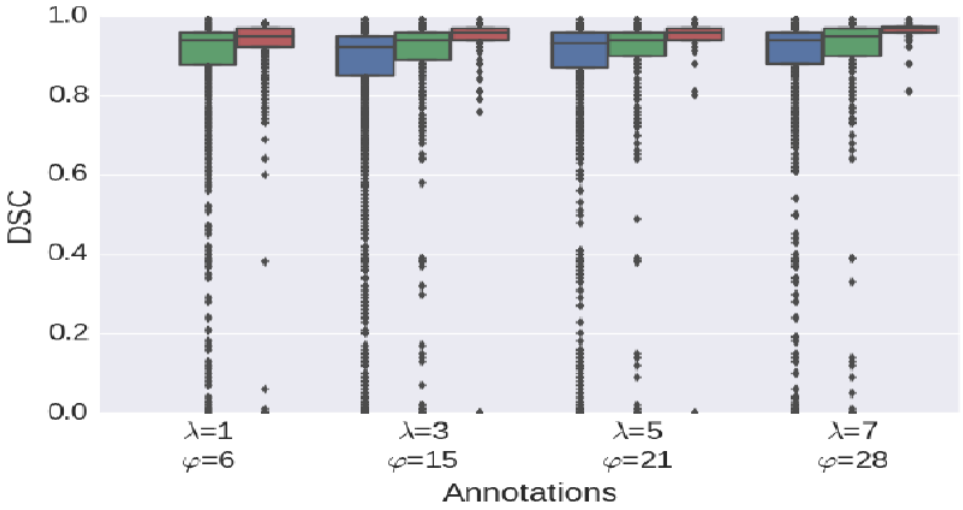


Figure 3.8: Confidence-weighted majority voting.

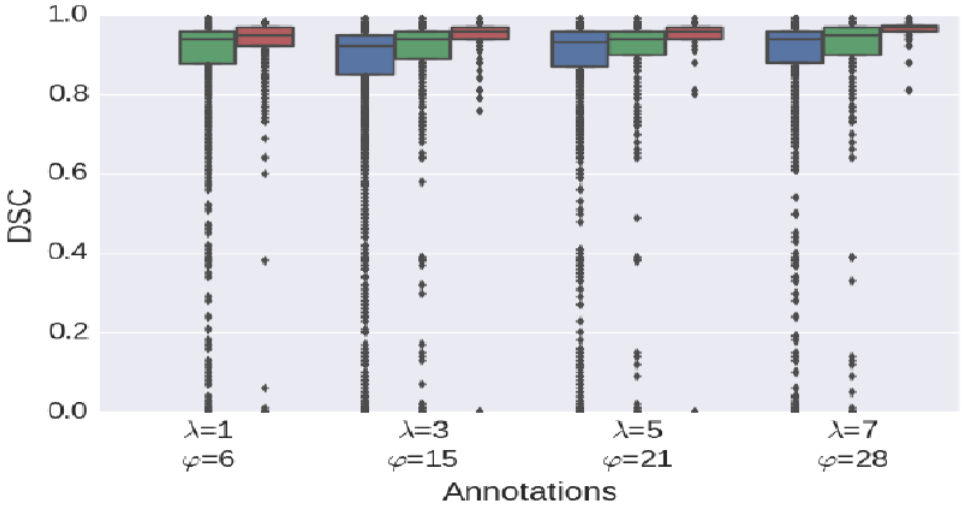


Figure 3.9: Confidence-weighted majority voting annotations of car

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 estimatedquality, 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.

To create highly accurate object segmentations it is crucial that inaccurate annotations are detected with a high 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 . This occurred particularly frequently on rather difficult tasks, e.g. where the car is hidden in the background 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 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 higheramount of misclassified bounding box segmentations and spam. With the low overall estimation error , 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, 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 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 .For user identity verification and taking into account the work for clickstream based Sybil account detection in online communities and user behaviour clustering 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 aposteriori STAPLE algorithm approach for merging single object annotations could further improve segmentation quality.Intra-class segmentation quality estimation performance for all classes is shown in fig 3.10 showing the efficiency of the proposed algorithm. It is also seen that cost is less in comparison as shown fig 3.11. The statistics and comparison is shown fig 3.12 and fig 3.13 respectively ,proving the cost effectiveness against conventional methods.

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 , 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 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.

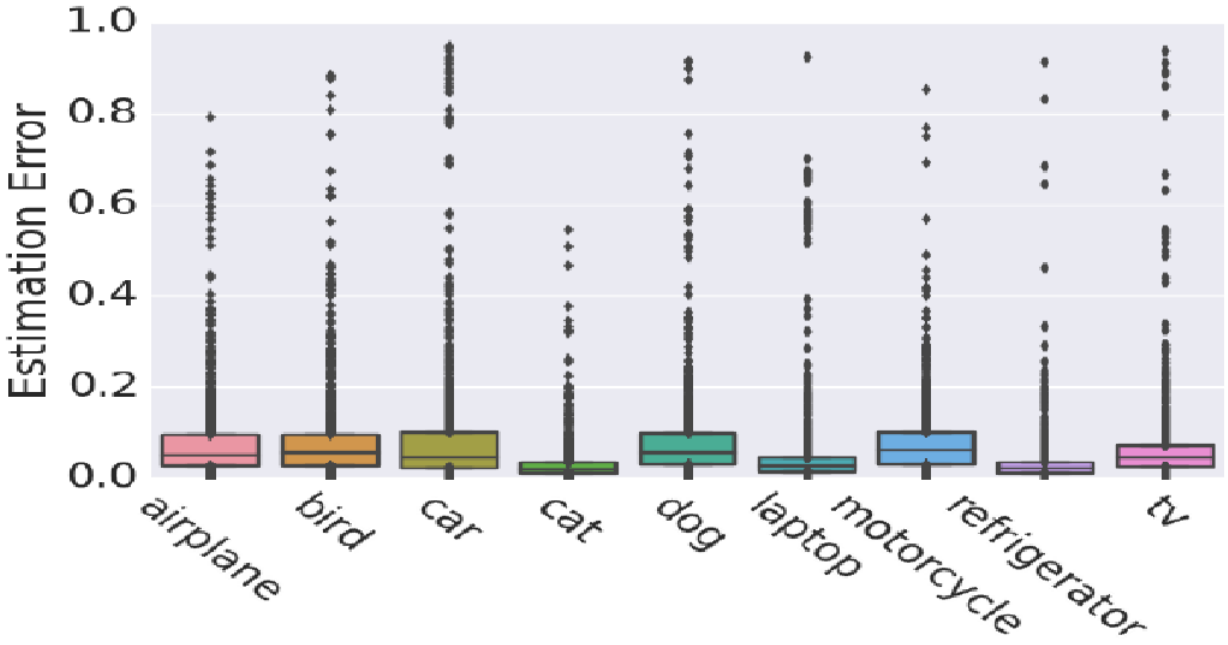


Figure 3.10: Intra-class segmentation quality estimation performance for all classes

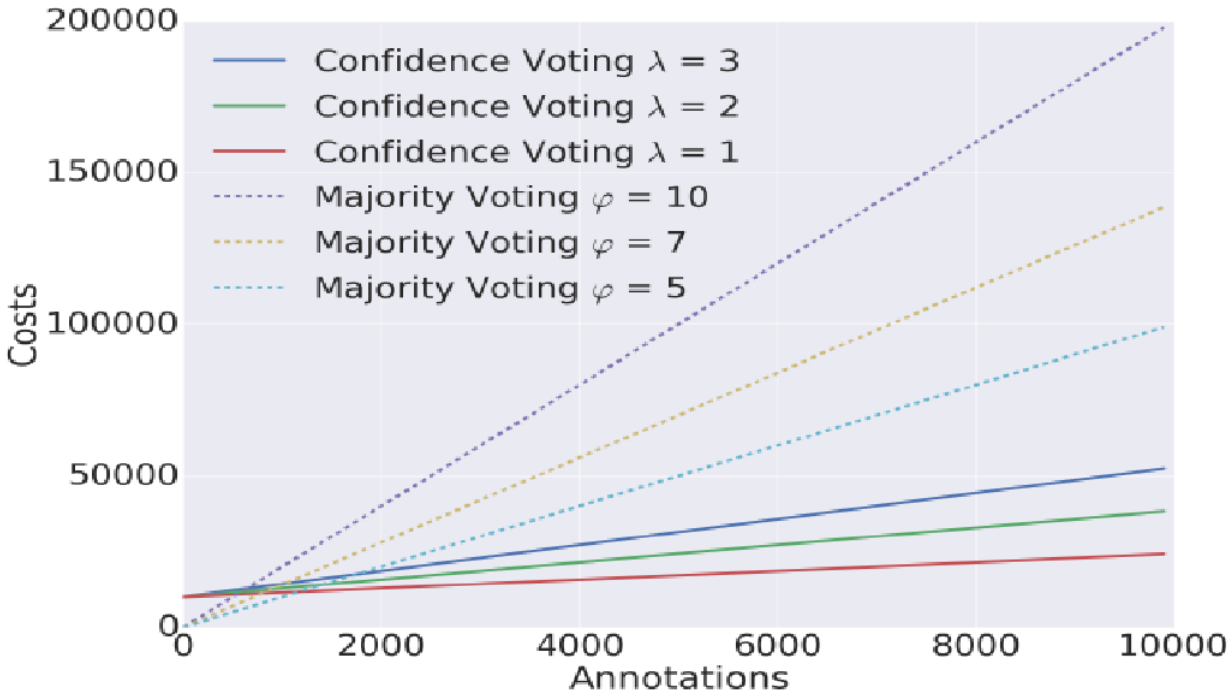


Figure 3.11: Comparison of the annotation costs

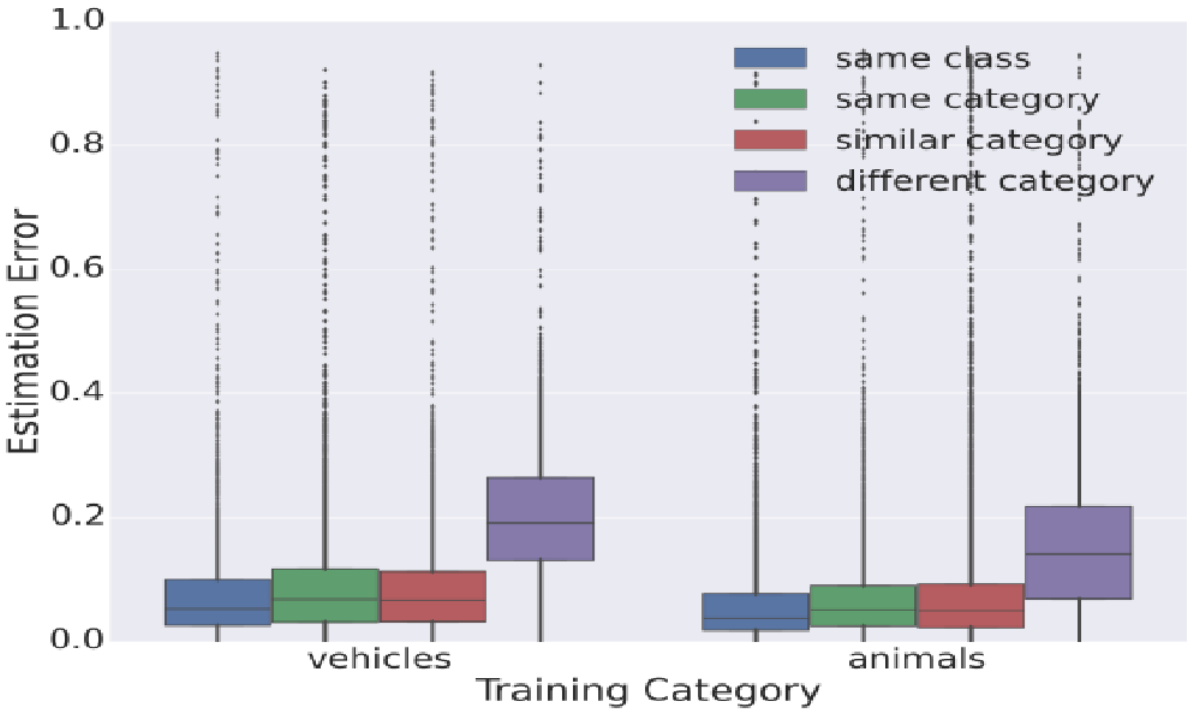


Figure 3.12: Descriptive statistics

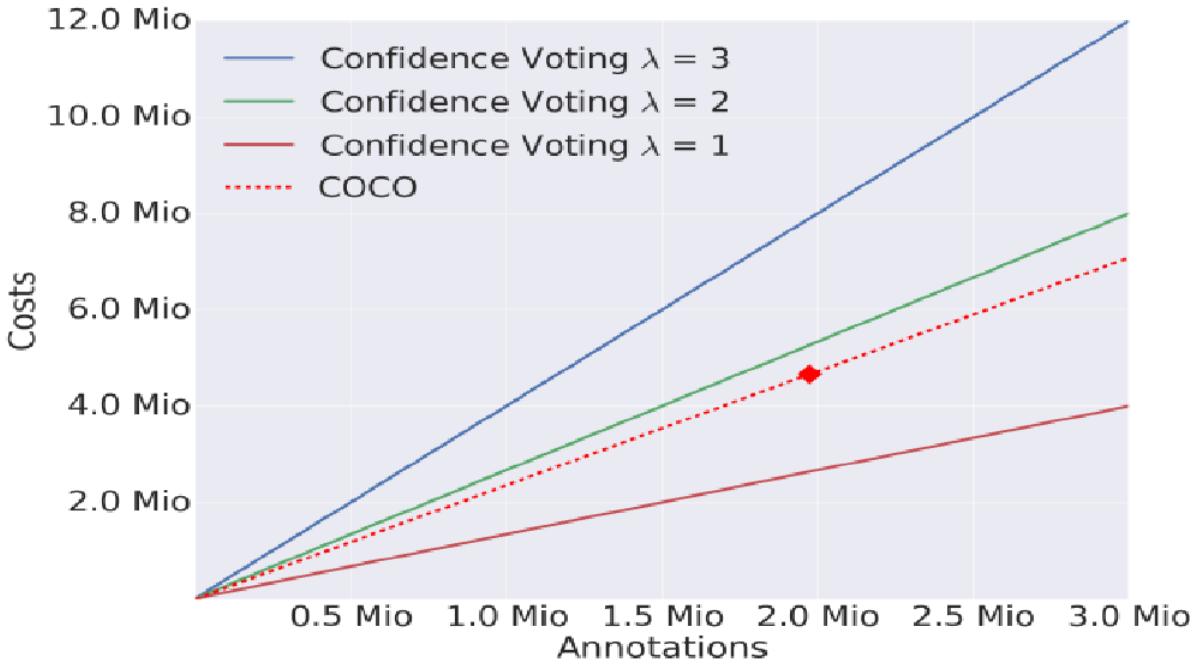


Figure 3.13: Comparison of the annotation costs of the proposed method with respect to COCO

Conclusion

This paper presents 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 enable 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. The hypothesis of our work is that the clickstream data itself is sufficient to estimate a worker's performance. The paper support the following three hypotheses:

- Clickstream features are very good predictors for segmentation quality
- These features generalize over (similar) object classes
- Clickstream-based annotation quality estimation can be applied for confidence-based annotation merging

REFERENCES

[1] A. Luckow, K. Kennedy, F. Manhardt, E. Djerekarov, B. Vorster, and A. Apon big data: Applications, workloads and infrastructures ", *IEEE International Conference on Big Data*, 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,", *IEEE Conference on Computer Vision and Pattern Recognition, 2009* , 1201–1210, 2009.

[3] A. Krizhevsky, I. Sutskever, and G. E. Hinton "Imagenet classification with deep convolutional neural networks ", *Advances in Neural Information Processing Systems* pp. 1097–1105, 2012.

[4] S. Albarqouni, C. Baur, F. Achilles, V. Belagiannis, S. Demirci, and N. Navab : Deep learning from crowds for mitosis detection in breast cancer histology images ", *IEEE Transactions on Medical Imaging*, vol. 35, 2016.

[5] S. Vittayakorn and J. Hays"Quality assessment for crowdsourced object annotations", *Proc. of British Machine Vision Conference* , . 109.1–109.11, 2011.

[6] P. Welinder and P. Perona"Online crowdsourcing: rating annotators and obtaining cost-effective labels ", *n Proc. IEEE Conference on Computer Vision and Pattern Recognition* , 2010.

[7] P. Welinder, S. Branson, P. Perona, and S. J. Belongie multidimensional wisdom of crowds *Advances in neural information processing systems, 2010* 2010.

[8] J. Carreira and C. Sminchisescu , "Constrained parametric min-cuts for automatic object segmentation," *IEEE Computer Society Conference on Computer Vision and Pattern Recognition* June 2010.