# Annotation quality vs. quantity for deep-learned medical image segmentation

Wesemeyer, Tim, Jauer, Malte-Levin, Deserno, Thomas

Tim Wesemeyer, Malte-Levin Jauer, Thomas M. Deserno, "Annotation quality vs. quantity for deep-learned medical image segmentation," Proc. SPIE 11601, Medical Imaging 2021: Imaging Informatics for Healthcare, Research, and Applications, 116010C (15 February 2021); doi: 10.1117/12.2582226

SPIE.

Event: SPIE Medical Imaging, 2021, Online Only

# Annotation quality vs. quantity for deep-learned medical image segmentation

Tim Wesemeyer[b], Malte-Levin Jauer[a], and Thomas M. Deserno[a]

[a]Peter L. Reichertz Institute for Medical Informatics of TU Braunschweig and Hannover Medical School, Mühlenpfordtstr. 23, Braunschweig, Germany
[b]TU Braunschweig, Mühlenpfordtstr. 23, Braunschweig, Germany

## ABSTRACT

For medical image segmentation, deep learning approaches using convolutional neural networks (CNNs) are currently superseding classical methods. For good accuracy, large annotated training data sets are required. As expert annotations are costly to acquire, crowdsourcing–obtaining several annotations from a large group of non-experts–has been proposed. Medical applications, however, require a high accuracy of the segmented regions. It is agreed that a larger training set yields increased CNN performance. However, it is unclear, to which quality standards the annotations need to comply to for sufficient accuracy. In case of crowdsourcing, this translates to the question on how many annotations per image need to be obtained.

In this work, we investigate the effect of the annotation quality used for model training on the predicted results of a CNN. Several annotation sets with different quality levels were generated using the Simultaneous Truth and Performance Level Estimation (STAPLE) algorithm on crowdsourced segmentations. CNN models were trained using these annotations and the results were compared to a ground-truth. It was found that increasing annotation quality results in a better performance of the CNN in a logarithmic way.

Furthermore, we evaluated whether a higher number of annotations can compensate lower annotation quality by comparing CNN predictions from models trained on differently sized training data sets. We found that when a minimum quality of at least 3 annotations per image can be acquired, it is more efficient to then distribute crowdsourced annotations over as many images as possible.

The results can serve as a guideline for the image assignment mechanism of future crowdsourcing applications. The usage of gamification, i.e., getting users to segment as many images of a data set as possible for fun, is motivated.

**Keywords:** Crowdsourcing, STAPLE, Ground-truth, Silver-standards, CNN

## 1. INTRODUCTION

Image segmentation is the task of extracting one or multiple coherent regions from an image. Segmentation is relevant for several current challenges in signal and image processing. Typical examples are scene understanding and relationship modeling for autonomous driving or handwriting recognition for advanced user interaction.[1]

Further author information: (Address correspondence to T.D.)
T.W.: E-mail: t.wesemeyer@tu-braunschweig.de
M.J.: E-mail: malte-levin.jauer@plri.de
T.D.: E-mail: thomas.deserno@plri.de

Medical Imaging 2021: Imaging Informatics for Healthcare, Research, and Applications
edited by Thomas M. Deserno, Brian J. Park, Proc. of SPIE Vol. 11601, 116010C
 · doi: 10.1117/12.2582226

### 1.1 Background

In medicine, image segmentation is a crucial part of magnetic resonance imaging (MRI) and computed tomography (CT) image analysis. While in other domains, localization of a squared region of interest (ROI) is sufficient, medical imaging usually requires the precise delineation of boundaries. Common use cases include the determination of size and shape of cancer tissue or the distinction between healthy and pathological tissue areas, e.g., to determine where a tumor resection or ray treatment should be applied. In other cases, segmentation is a preprocessing step for subsequent analysis, e.g., color analysis within a spatial area. For example, in the use case of automatic grading of hyperemia, a precise segmentation is necessary to reduce the noise for the following stage of redness calculation.[2]

To obtain a reliable segmentation, highly skilled experts are needed who manually annotate images using specialized software tools. However, this approach suffers from two drawbacks: First, a subjective bias might be introduced depending on the education and interpretation of the particular expert. To reduce this bias, more experts could be hired to obtain multiple annotations per image which then can be averaged. But, evidently increasing the number of experts increases the costs. Second, many scientific studies require the evaluation of very large data sets, which increases the annotation cost again. To reduce manual annotation effort, automation approaches have been investigated. Most successfully, deep learning methods have been applied. However, these methods rely on annotated reference data sets as well. It is agreed that the best results can be obtained when the reference data set is large. Unfortunately, in the context of medical use cases, large-scale human-annotated data sets are rarely publicly available and often costly.[3–6]

To overcome the mentioned problems, the method of crowdsourcing was proposed. Crowdsourcing describes the idea of using the "wisdom of the crowds", namely using a larger group of non-expert raters, to substitute for costly expert annotations.[7–9] The quality of these crowdsourced annotations is necessarily dependent on the difficulty of the performed task and the skill of the human raters forming the crowd.[10] Therefore, when these annotations are used for the training of neural networks (NN), other authors recommend applying some form of quality control.[6,11] The general idea is to let multiple raters annotate the same image and then calculate a consensus of all available annotations. Using this approach, a less biased set of annotations with nearly expert-like quality is expected.[12–14]

### 1.2 Problem

To sum up, it is consensus that the segmentation performance of a neural network depends on the size of the training data set.[3,15,16] To get a sufficient amount of training data, crowdsourcing can be used.[11,17,18]

In a setting with limited resources however, only a limited number of annotations can be obtained (regardless of using expert or crowdsourced annotations). Little to no evidence however is available on guidelines how to distribute crowdsourced annotations over a data set, when the segmentation using deep neural networks is desired. The parameter of interest in comparative overviews is prominently the amount of training data.[16] A quantitative analysis on the impact of annotation quality on training performance in the context of crowdsourcing is not known to the authors.

### 1.3 Research questions

In our setting, we assume that an extensive image data set is at hand which is too large to spend time and money for a manual segmentation of all data by experts. The problem shall be solved using a neural network trained on crowdsourced annotations from non-experts. We need to decide, how to distribute the available resources (workforce/money). It is the question of how much of the data set has to be segmented, how often, and by whom.[11] Therefore, we pose two research questions for the following work:

- **RQ-A**: How does the quality of the annotations influence the segmentation results of the NN?
- **RQ-B**: Can we compensate the lack of high-quality annotations for few images with low-quality annotations on more images?

To tackle these questions, we first summarize existing automatic segmentation methods in the following, before later on setting up our experiment and analyze the results.

## 2. STATE OF THE ART

Formally, segmentation reflects a classification at pixel level: Each pixel needs to be classified as belonging to a certain class. However, as an isolated pixel does not contain enough information, surrounding pixels must be considered as well. The most simple form of image segmentation is binary classification into foreground (or ROI) and background. Multi-class segmentation detects several classes of regions in an image, e.g., A, B, C and D.

### 2.1 Classical segmentation techniques

Depending on the layer of abstraction, classical segmentation techniques can be divided into pixel-, edge-, texture- or region-based methods.[19] Each class of methods has their own advantages and drawbacks. Pixel-based algorithms only use information from one pixel and thresholds to determine the class that this pixel belongs to. Algorithms operating on a higher abstraction level are more widely used. For example, a popular region-based algorithm is "region growing", where adjacent pixels are assigned to the same class if their pixel values are close according to some predefined closeness criterion.[20] Furthermore, edge-based algorithms like the "watershed algorithm" apply operators to the image for detecting edges. By doing so, they detect separate classes of pixels. These gradient operators depend on a well-defined step edge to reduce false positive and false negative classifications. In many cases, a combination of multiple of these heuristic techniques are necessary to achieve the segmentation goal.[20]

In summary, a wide variety of classical segmentation methods is available. However, selecting the "right" technology and the best parameters for a particular application is difficult as they often lack generalizability, e.g., over different data sets and use cases.[20]

### 2.2 Machine learning techniques

To overcome manually tuned parameters, machine learning is applied. These methods "learn" the best parameters for a use case by applying optimization techniques to a given set of training images. Either the pixel classes are known (ground truth) for the training data or a reward function is used. However, this requires the image features (e.g., certain pixel values, detected edges) on which the method operates, to be defined a priori.

An example for a machine learning-based segmentation method is the trainable Weka segmentation.[21] It uses machine learning algorithms on selected image features to create a texture-based segmentation. Fig. 1 shows two Weka segmentations. In this case, the classifier was trained on two different image data sets, with the first being images from Sirazitdinova et al.[22] and the second being a non-public data set which contains similar images. In the image from the first data set, the classifier segments the majority of the sclera correctly. In contrast, the image from the second data set contains heterogeneous lighting conditions and is not segmented with a satisfying accuracy. Large artifacts can be seen in the brighter areas.

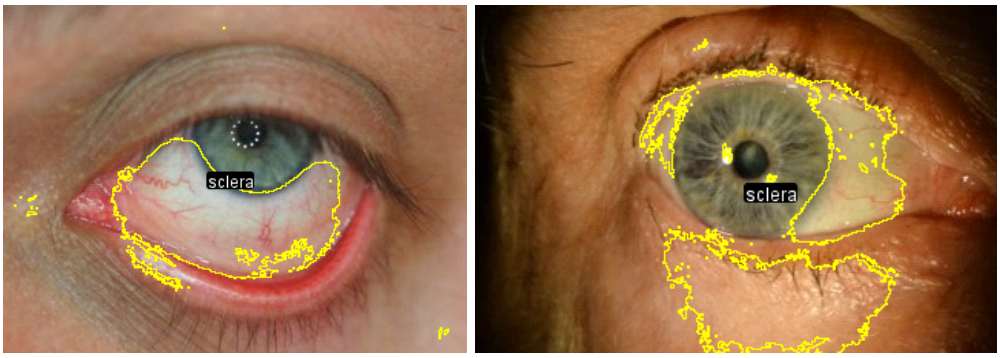


Figure 1. **Machine learning results** from trainable Weka segmentation

As demonstrated by this example, the results of segmentation largely depends on the selection of training images, features and their similarity to the images which are segmented. If available, using more images that are similar to the desired use case can improve the accuracy of segmentation. However, this quickly leads to the risk of "overfitting", i.e., specializing the classifier too much to a given data set that it performs poorly when applied to a slightly different case.

### 2.3 Deep learning approaches

To overcome the need for manual feature selection and tuning, the concept of deep neural networks has emerged from the family of machine learning algorithms. In contrast to the hand-crafted features from classical image processing techniques, the features are learned by the network during training. In this way, the network learns to recognize edges and corners, then outlines and shapes, if this suits the desired output. Typical use cases are classification of image content, for example, people, cars, or street signs.

#### 2.3.1 General concept

The key concept of neural networks is the usage of artificial neurons that store information about their state of activation. These neurons are usually organized in layers (groups of neurons) connected to some or all other neurons of adjacent layers using a set of weights. If several layers are used, the methods are referred to as "deep learning" (DL) methods.

A lot of DL networks have been proposed. They differ in architecture, data basis, training strategy and implementation, and achieve different performance in solving a specific problem.[15] Successful approaches exist, among others, in handwriting recognition and object classification.[1] There is also an active interest in directly adopting DL architectures for pixel-wise labeling.[23–25]

#### 2.3.2 Commonly used architectures for image segmentation

For image segmentation, convolutional neural networks (CNNs) are used frequently. Their strength relies on the inclusion of locality, i.e., the relationship between neighboring pixel. This is done by recurring and shared parameters to mimic convolution operators. In popular segmentation architectures such as U-Net or SegNet,[1,26] the information represented in the annotation is simplified in the encoder (downsampling) to learn the features. The decoder then attempts to reconstruct the original information (upsampling). This is done by means of learned features or architectural techniques that make previously lost information available again for later layers.

CNNs are known to perform more efficiently on grid-like data (like images) than other networks[15] with supervised CNNs currently known as the most successful method for segmentation.[3] Furthermore, the feasibility of using crowdsourced annotations for training CNNs has been demonstrated by related works.[6,11]

## 3. MATERIALS AND METHODS

We first define the variables under considerations, namely what we mean by data sets, ground truth (GT), quality of annotations, and segmentation performance. Subsequently, define testing scenarios on the available data sets to compute the variables of interest.

### 3.1 Definitions

#### 3.1.1 Data sets

An image can be seen as a matrix of pixels $M$. Let's define a set $I$ consisting of $n$ images as:

$$I = \{ M_i \mid i \in [1 \dots n] \} \tag{1}$$

and a set of annotations to these images as $A$ with $m$ available annotations per image as:

$$A = \{ A_{i,j} \mid i \in [1 \dots n], j \in [1 \dots m] \} \tag{2}$$

where the annotation $A_{i,j}$ is the $j$-th annotation to image $i$. The annotations take again the form of a matrix of pixels with the same dimensions as the image. For the binary segmentation, pixel values of 1 indicate that a pixel belongs to the ROI; a value 0 indicates the classification as background.

### 3.1.2 Ground truth

The term "ground truth" (GT) usually refers to the correct (true) segmentation, where all pixels are assigned to the semantically correct class. An automatic segmentation algorithm can then be compared to the GT, to evaluate its performance.

Especially in medical image segmentation, it is often hard to determine a clear GT. Even when the reference is obtained from domain experts, considered the "gold standard",[27] results will vary depending on the particular rater.

A consensus algorithm can be used to estimate the best guess for the truth using multiple expert ratings. These results are called the best "gold standard" available. Recently, it was also investigated how a higher number of less experienced raters can compensate for lacking gold standards.[6,7,13,28] Most authors agree that this "silver standard" can yield satisfying results if enough user ratings are taken into account so that the consensus algorithm converges.

Due to lack of gold standard GT for the data sets considered in this work, we follow this approach. The consensus algorithm used is the Simultaneous Truth and Performance Level Estimation (STAPLE) algorithm.

### 3.1.3 STAPLE

The STAPLE algorithm is used among others for segmentation involving multiple inputs and generation of GT data.[29] The algorithm was originally presented by Warfield et al.[30] Using an expectation-maximization method, the STAPLE algorithm yields a consensus segmentation as well as performance measures of each rater. This allows an a posteriori assessment of all user ratings with respect to the estimated GT. It was found that the training of CNNs on STAPLE masks results in a better generalization.[6,31,32]

Let $S_{i,k}(A)$ be the result of the STAPLE algorithm running on image $i$ with the annotations $A_{i,1} \dots A_{i,k}$. Then, we define the set of STAPLE results obtained on the given image data set $I$, using $k$ annotations per image as

$$S_k(A) = \{\ S_{1,k}(A), S_{2,k}(A), \dots, S_{n,k}(A)\ \} \mid k \leq m \tag{3}$$

where $m$ was defined above as number of annotations per image.

### 3.1.4 Annotation quality levels

To investigate research question RQ-A, the term of "annotation quality" needs to be defined. We use this term to describe how well a particular annotation complies to the GT. The highest annotation quality would therefore be reached, if the annotation is equivalent to the GT. As describe above, we use the STAPLE algorithm to generate the GT from all available annotations per image. To evaluate how a CNN would perform when trained with annotations of lower quality, we need to develop a way of generating annotations of different quality levels.

In previous work,[13] we investigated how the relative error with respect to the GT is reduced when taking different subsets of annotations image. After running STAPLE for all available annotations, we also obtain an a posteriori estimate for each rater's performance. When performing STAPLE on subsets of annotations, and including always the worst raters in these subsets, we can see a steep decrease in relative error when increasing the number of annotations per image. In our case, a subset of the worst 5 annotations per image resulted in a relative error of around 10%. For comparison, a subset of the 22 worst annotations yielded relative error of 2%, when the GT was generated using 30 annotations per image.[13]

In this work, we re-use this knowledge to generate artificial annotations with different levels of quality using the definition of STAPLE results above. In this sense, e.g., $S_5(A)$ (taking the worst 5 annotations per image) would be a set of low-quality annotations while $S_{15}(A)$ (15 worst annotations per image) would be a set of higher quality. In the same way, the GT is defined using all available annotations per image as $S_m(A)$.

#### 3.1.5 Segmentation performance

In contrast to our previous work, the aim is not to compare different STAPLE results with each other, but instead training a CNN with annotations of different quality levels. Afterwards, the CNN is used to predict segmentations, which are compared to the GT to analyze the effect of different annotation quality levels. This analysis is performed by using the F-score (also F1-score, F-measure). The F-score is based on precision $p$ and recall $r$ and combines both measures into one metric:

$$F(A_{\text{in}}, A_{\text{ref}}) = 2 \cdot \frac{p(A_{\text{in}}, A_{\text{ref}}) \cdot r(A_{\text{in}}, A_{\text{ref}})}{p(A_{\text{in}}, A_{\text{ref}}) + r(A_{\text{in}}, A_{\text{ref}})} \tag{4}$$

### 3.2 Available data sets

Two data sets with images of human eyes and their corresponding annotations of the sclera are used. Both data sets were collected within the evaluation study of the online segmentation tool WeLineation.[13] All photos are 24-bit RGB images in the JPEG format and were originally taken in the resolution of 2992 x 2000 pixels. In both studies, amateurs without medical background and expertise were asked to segment the sclera manually using a point-based annotation tool.

The first data record contains 75 images with at least 24 user annotations per image. Due to the high number of user annotations compared to the number of images, the data set is suitable for investigating how the annotation quality affects the segmentation performance (RQ-A). For simplicity, we call the set of annotations for this data set $A$ in the following:

$$A = \{\ A_{i,j}\ \mid\ i \in [1 \dots 75], j \in [1 \dots 24]\ \} \tag{5}$$

The second available data set has a size of 936 images. A minimum of three annotations are available for each image. Let $B$ be the set of annotations to these images. Due to the high number of images compared to number of annotations, the data set is suitable for investing the effect of high number of annotated images with lower annotation quality level. We denote the second data set as:

$$B = \{\ B_{i,j}\ \mid\ i \in [1 \dots 936], j \in [1 \dots 3]\} \tag{6}$$

### 3.3 Experimental setup

We used the U-Net as network architecture,[26] which is specially designed for segmentation tasks of medical images with small training data sets.[4] Specifically, we use the implementation of Yakubovskiy,[33] which provides a VGG-16 pre-trained model for the pyTorch framework (Table 1). We use an initial learning rate of $10^{-4}$ for the decoder and $10^{-6}$ for the encoder. Both learning rates are adapted during training using the Adam optimizer.[34] As preprocessing, input images and annotations are scaled to a uniform resolution of 640 x 427 pixels. To be suitable for the CNN, we applied random padding such that the image dimensions are divisible by 64. Data augmentations is applied using the albumentations framework.[35] We use dynamic learning rate reduction and early stopping as follows: if within three consecutive epochs, there is no improvement of the validation error by at least 0.0008, the LR of the decoder is set to $10^{-5}$. If this happens again within the next three iterations, we expect no further improvements, the training of the current model is stopped.

The following data is collected during the training of the networks:

- number of last evaluated epoch
- F-score for training and validation set
- time stamp
- current learning rates and result of the early stopping mechanism

Table 1. Hard- and software configuration for network training and testing.

| **Component** | **Version/model** |
|---|---|
| Operating system | Ubuntu 18.04. LTS |
| Interpreter | Python 3.6 |
| Processor | Intel i7 7700K |
| Graphics unit | NVIDIA GTX1070 |
| CUDA runtime | 10.1 |
| pyTorch | torch 1.4<br>torchvision 0.5<br>segmentation-models-pytorch 0.1<br>albumentations 0.4.5 |

### 3.4 Testing scenarios

For the comparison of the CNNs performance, several models need to be trained. First, the splitting of data sets into training, validation and test set is described. Then, the different subsets used for generating the models are explained.

#### 3.4.1 Data splits

The available data sets need to be partitioned into training, validation and test set. An existing data set $I$ with known annotation set $A$ can therefore be dived in these three disjoint sets.

$$I = I^{\text{train}} \cup I^{\text{valid}} \cup I^{\text{test}} \tag{7}$$

$$A = A^{\text{train}} \cup A^{\text{valid}} \cup A^{\text{test}} \tag{8}$$

Despite the widely used training to test data ratio of 75:25[36] and training to validation set ratio of 80:20,[15] we use a ratio of 80:20 in each case because of the small data set $A$.

#### 3.4.2 RQ-A: investigating annotation quality

To answer RQ-A, annotations of different quality levels are simulated with the STAPLE algorithm. Thereby, different subsets of $A$ are used. The corresponding image set $I$ consists of $n = 75$ images.

In our specific scenario, we evaluated the subsets of the worst 1, 2, 3, 5, 10, 15, 20, and all existing annotations per image to obtain the STAPLE results sets:

$$S_k(A^{\text{train}}) \mid k \in \{\, 1, 2, 3, 5, 10, 15, 20, m \,\} \tag{9}$$

For each of these sets, two models are trained using the U-Net architecture. Thereby for RQ-A the training ($I^{\text{train}}$), validation ($I^{\text{valid}}$) and test sets ($I^{\text{test}}$) consist of 48, 12 and 15 images with corresponding annotation.

As the set of annotations $A$ has a comparably high number of annotations per image ($j \geq 24$), we use the STAPLE results generated from all annotations per image respectively as a high-quality and reliable GT. Therefore, the test set $S_m(A^{\text{test}})$ is used in both testing scenarios for the comparison with the neural network's predictions.

### 3.4.3 RQ-B: investigating annotation quantity

For the second research question, we investigate different sizes of training data sets. Therefore, we generated several models using subsets of the available training data and compare the network's predictions using the test set defined for RQ-A.

In this case, we define subsets of the image set $J$ and annotation set $B$. The total number of images contained in $J$ is 936 with at least 3 annotations per image available. Therefore, we perform the investigation on two quality levels: using either one or three annotations per image. This allows a comparison between quality levels also for this experiment.

Let $J_l$ be a subset of images of data set $J$ with a size of $l$ images. Let $B_l$ be the set of annotations for these images. Necessarily, we have to divide the data sets again into a training and validation set for the model generation: $B_l = B_l^{\text{train}} \cup B_l^{\text{valid}}$. The splitting is again done using an $80:20$ ratio. As explained in the previous section, we do not need a test split in this case as we re-use $A^{\text{test}}$ for comparability. Therefore, the ratio of the splits is not constant for these testing scenarios, as the number of test images is fixed at 15 while number of training images is changed (Fig. 2).

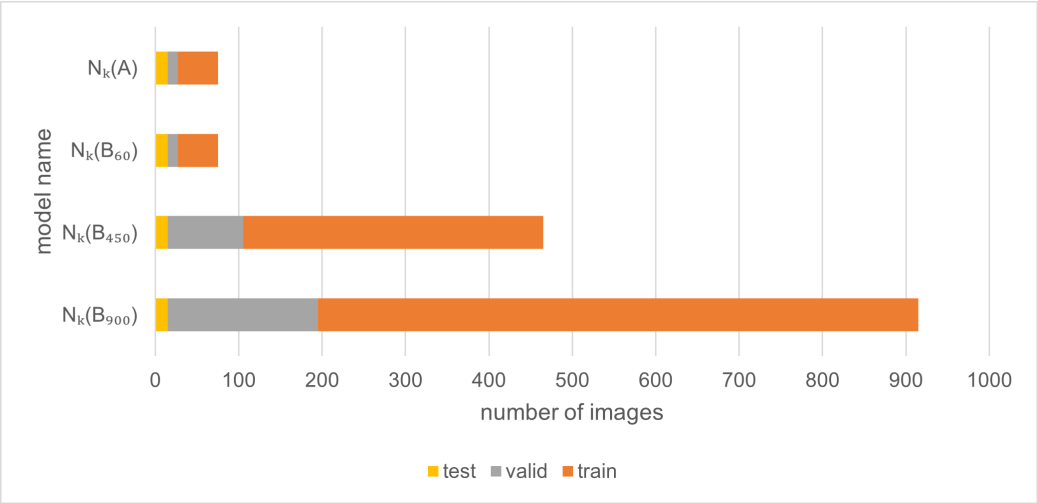


Figure 2. Distribution of the training, validation and test splits for exemplary scenarios

As a result, we consider the following sets of STAPLE results for each of these subsets to train the neural network models:

$$S_k(B_l^{\text{train}}) \mid k \in \{\, 1, 3 \,\},\ l \in \{\, 60, 120, 180, 300, 450, 600, 900 \,\} \tag{10}$$

These chosen subset quantities are chosen to obtain several directly comparable results. With the exception of 450, each size is a multiple of 60, the number of images in $I^{\text{train}}$. Furthermore, some models can be compared with respect to quality or quantity, as they share the same amount of annotations necessary but used in different ways: e.g., $S_3(B_{60})$ and $S_1(B_{180})$ are using 180 user annotations, but with different number of annotations per image. The same holds for $S_3(B_{300})$ and $S_1(B_{900})$. In this way, these models can be compared to evaluate to what extent higher numbers of training images compensates for the lower accuracy of the training annotations.

### 3.4.4 Model overview and comparison

Several models are generated using the different sets of STAPLE results. For easier distinction, we define the following abbreviations. Let $N_k(A)$ be the model generated by the training process of the U-Net on the training data set $A^{\text{train}}$. For RQ-A, this results in models $N_1(A), N_2(A), \ldots N_m(A)$. For RQ-B, we get models $N_1(B_{60}), N_1(B_{120}), \ldots N_1(B_{900}), N_3(B_{60}), N_3(B_{120}), \ldots, N_3(B_{900})$ (Table 2).

Table 2. Overview of all trained models for RQ-A and RQ-B.

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| **RQ-A** | $N_1(A)$ | $N_2(A)$ | $N_3(A)$ | $N_5(A)$ | $N_{10}(A)$ | $N_{15}(A)$ | $N_{20}(A)$ | $N_m(A)$ |
| **RQ-B** | $N_1(B_{60})$ | $N_1(B_{120})$ | $N_1(B_{180})$ | $N_1(B_{300})$ | $N_1(B_{450})$ | $N_1(B_{600})$ | $N_1(B_{900})$ | |
| | $N_3(B_{60})$ | $N_3(B_{120})$ | $N_3(B_{180})$ | $N_3(B_{300})$ | $N_3(B_{450})$ | $N_3(B_{600})$ | $N_3(B_{900})$ | |

Due to the network's implementation to semi-randomly find weights and then determine the performance of these, the training is not deterministic. This causes a deviation of the model parameters despite the same settings for each trained model. This means that the result between neural networks varies even with the same configuration. Due to this fact, two networks are trained per configuration and the better result will be evaluated owing to the very limited optimization. On averaged results the finding are still accurate.

The predicted result of the neural network $P(N, M_i)$ is defined using weights from the model $N$, evaluating the given input image $M_i$. For each test scenario, we calculate the F-score between the GT $S_m$ and the predictions of the neural network with model $N$ evaluating the test set. This yields our average performance measure $\bar{F}$ for the segmentation results:

$$\bar{F}(N, S_m) = \frac{1}{60} \sum\nolimits_{M_i \in I^{\text{test}}} F\Big( P(N, M_i)), S_{i,m}(A) \Big) \tag{11}$$

## 4. RESULTS

### 4.1 Results regarding the research questions

For RQ-A, 16 models have been trained. An increasing F-score can be observed with increasing annotation quality level. Depending on the $k$ annotations taken per image–fused to a single STAPLE result–the increase in F-score takes a logarithmic form (Fig. 3).

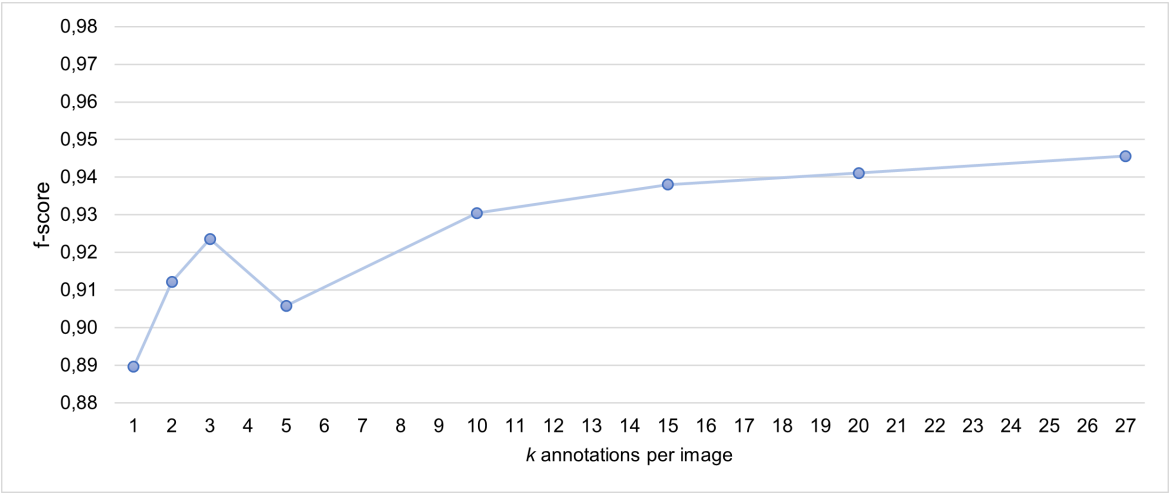


Figure 3. Comparison of segmentation performance using different annotation quality levels. The evaluation on the test set shows a minimum F-score of 0.89 with one annotation per image and a maximum of 0.946 using 27 annotations per image.

In the second setting, we used varying sizes of the training data set. Increasing the size of the training data set also logarithmically increases the F-score, as expected (Fig. 4, left). Using three annotations per image further improves the results. The progression again takes a logarithmic form (Fig. 4, right). Also in this setting, the usage of a higher quality GT leads to an increased F-score, with the highest value of 0.974 for the training scenario with 900 images and three annotations per image.


Figure 4. Comparison of segmentation performance using different numbers of training annotations. Left: one segmentation per image (lowest annotation quality) is used for training. Right: using three annotations per image.

As shown so far, the pure number of different images in DL segmentations using STAPLE masks does not reflect a clear measure of the success of the DL segmentation. To investigate the relationship between the associated number of annotations and the resulting F-score, we compared all trained models (Fig. 5). In principle, the more segmentations are divided among different images, the better the segmentation result of the neural network becomes.

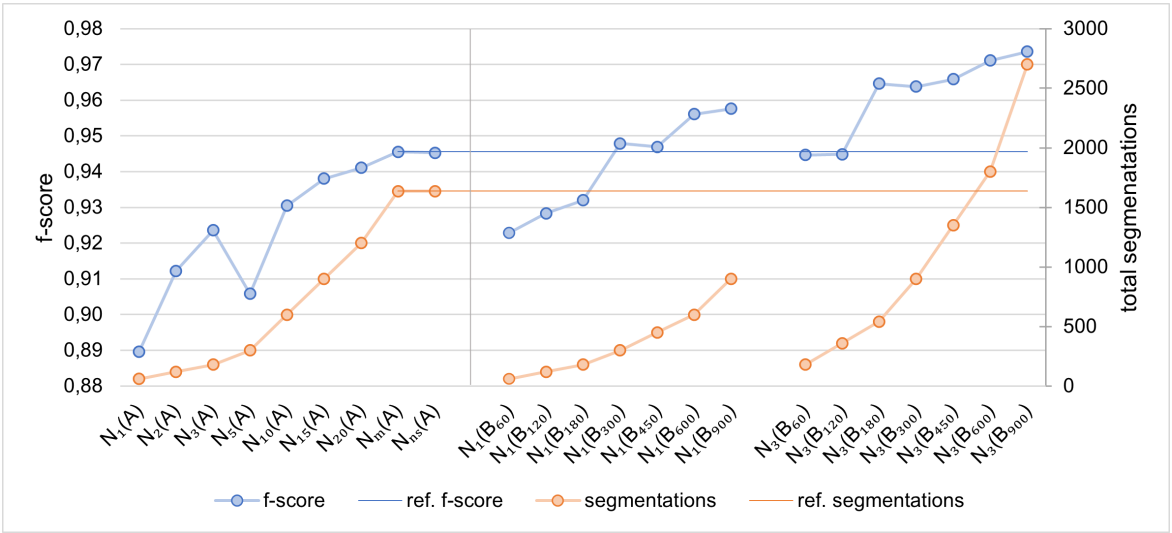


Figure 5. Overall comparison of segmentation performance of all trained models with respect to F-score (blue curve, left y-axis) and number of annotations needed for the models (orange curve, right y-axis). The colored horizontal lines serve as a reference to the values of the best STAPLE trained model of RQ-A. $N_{ns}(A)$ is another model trained without the use of STAPLE we will discuss.

### 4.2 Side results regarding STAPLE

Besides the models for RQ-A we trained another two on the same data set without using the STAPLE algorithm. We used all annotations and therefore the images of set $I$ multiple times. The training, validation and test set

were divided accordingly to the other trained networks and guaranteeing that the same images were used in the same sets. As a promising result of similar performance compared to the best STAPLE training based network (Fig. 5, STAPLE based $N_m(A)$ versus no STAPLE network $N_{ns}(A)$) we conducted another experiment. We slightly improved training and optimization, used the data data set $B$ for training and validation of networks and tested the networks performance on our defined ground truth level data from whole data set $A$ including all 75 images. We trained six models with and without using STAPLE. We perceive similar performances resulting in just small differences in the averaged F-score (Table. 3).

Table 3. Comparison between STAPLE and purely annotation based network performances

| **Method** | **F-score** |
|---|---|
| STAPLE based network training | 0,9652 |
| Annotation based network training | 0,9642 |

## 5. DISCUSSION

Regarding the poor performance of $N_5(A)$ (Fig. 3, 5) we trained more models on the same configuration. While gaining performance only when modifying training parameters we concluded that training of some models seem to be overly influenced by the used implementation. Additionally the variance of model results is much higher for RQ-A models when $k \leq 3$. While averaging results of models based on the same data set would be beneficial for the interpretation of results, we saw that averaging the results of only two models seems insufficient to reliably identify outliers. Considering these points a thoroughly optimized training process is recommended for future work.

### 5.1 Evaluation of research questions

To answer RQ-A, we observed increased F-score with higher annotation quality levels, which means increasing the number of available annotations. However, with more than 10 annotations per image, the F-score increases only marginally due to the logarithmic nature of the F-score progression. This implies that for the training of a DL network, a mediocre level annotation quality is sufficient.

Regarding RQ-B, we showed that distributing the segmentation work over more images rather than acquiring numerous annotations for each image, quickly makes up for the lower quality of the annotations. In our case, a better F-score is obtained when more than 300 images are segmented compared to the reference model of 60 segmented images with a high annotation quality level.

In this case, a model that was created using 300 low-quality annotations outperformed the high-quality model based on 1600 annotations. This is a large difference in data efficiency.

However, the results of using three segmentations per image show that a distribution of user segmentations over as many images as possible is not the one-and-only way. Comparing the models $N_3(B_{300})$ and $N_1(B_{900})$– both using 900 user annotations–we can observe that the former performs better ($F(N_3(B_{300})) = 0.964$ vs. $F(N_1(B_{900}) = 0.958$). The same phenomenon can be observed when comparing the 180 annotations model $N_3(B_{60})$ with the $N_1(B_{180})$ model ($F(N_3(B_{60})) = 0,945 > F(N_1(B_{180})) = 0,932$).

Although the distribution of annotations over many images is desired, best results are achieved when the annotation quality level does not fall under a certain minimum. We observe a steep increase in F-Score between one and three annotations per image (Fig. 3). As this observation is confirmed by the results of RQ-B, three annotations per image can be seen as a recommended lower bound to reduce ambiguities and reach sufficient annotation quality. This is especially interesting when gathering data using crowdsourcing.

Naturally, the estimation of an upper bound is difficult as it depends on the accuracy desired in a use case and on the real expertise of the contributing raters. In our case, we find that around 15 annotations per image are a reasonable upper bound.

### 5.2 Limitations & relation to other's findings

The result that the best outcome can be reached when taking all available crowdsourced annotations into account is inline with Albarquoni et al.[17] They examined the approach of aggregating user annotations for mitosis detection in breast cancer histology images and produced the best results when training a CNN on aggregated training annotations. However, we can not confirm the necessity of a separate aggregation layer (STAPLE in our case) to improve the training performance significantly.

Albarquoni et al. conclude the concept of gamification is promising to motivate users segment a high number of images,[17] which then in turn leads to more annotations per image when more users are performing the task. Based on our results, we agree to this proposal, as gamification measures are known to increase rater quality as well as long-term motivation.[37] In this sense, this work lays some of the grounds for future gamification applications by providing general guidelines on how to distribute the user annotations over the set of images.

However, our evaluation has been done solely for the use case of sclera segmentation. It is unclear, how the results can be generalized to different applications using different images and segmentation characteristics. Despite this the comparability of our data sets $A$ and $B$ is limited as well. Both data sets were collected in separate studies with different users annotating the images and are small due to limited available data, especially $A$. Therefore we look forward if the results are validated in applications like histology, with significantly larger images and multi-class segmentation tasks.

## 6. CONCLUSION AND OUTLOOK

We conclude that

- training a CNN with higher quality annotations increases the segmentation performance, but
- distributing annotations over as many images as possible is more efficient.
- a minimum of at least three annotations per image is nevertheless advisable.
- additionally, we saw no need for preprocessing the crowdsourced annotations but propose to review this step and alternatively feed the user masks directly into the CNN.

To achieve the aforementioned points, we recommend future crowdsourcing applications to motivate users to segment as many different images as possible, e.g., by the long-term motivation effects of gamification approaches. This will help increasing the number of annotations and segmented images. Second, new users shall at first segment images that have already an annotation from another user. This way, more annotations per image can be obtained as recommended. Finally, when multiple annotations for a certain image are already available, new users could be "rewarded" for good accordance to others increasing the overall rater performance in the long run.

## REFERENCES


[1] Badrinarayanan V, Kendall A, Cipolla R. SegNet: A deep convolutional encoder-decoder architecture for image segmentation. IEEE Trans Pattern Anal Mach Intell. 2017 Dec;39(12):2481–95.

[2] Sánchez Brea L, Barreira Rodríguez N, Mosquera González A, Pena-Verdeal H, Yebra-Pimentel Vilar E. Precise segmentation of the bulbar conjunctiva for hyperaemia images. Pattern Anal Applic. 2018 May;21(2):563–77.

[3] Litjens G, Kooi T, Bejnordi BE, Setio AAA, Ciompi F, Ghafoorian M, et al. A survey on deep learning in medical image analysis. Med Image Anal. 2017 Dec;42:60–88.

[4] Rot P, Vitek M, Grm K, Emeršič Ž, Peer P, Štruc V. Deep sclera segmentation and recognition. In: Uhl A, Busch C, Marcel S, Veldhuis R, editors. Handbook of Vascular Biometrics. Springer International Publishing; 2020. p. 395–432.

[5] Chartrand G, Cheng PM, Vorontsov E, Drozdzal M, Turcotte S, Pal CJ, et al. Deep learning: A primer for radiologists. RadioGraphics. 2017 Nov;37(7):2113–31.

[6] Lucena O, Souza R, Rittner L, Frayne R, Lotufo R. Silver standard masks for data augmentation applied to deep-learning-based skull-stripping. In: 2018 IEEE 15th International Symposium on Biomedical Imaging (ISBI 2018). Washington, DC: IEEE; 2018. p. 1114–7.
[7] Cheng J, Manoharan M, Zhang Y, Lease M. Is there a doctor in the crowd? Diagnosis needed! (for less than $5). IConference 2015 Proc. 2015 Mar;.
[8] Heim E. Large-scale medical image annotation with quality-controlled crowdsourcing [Dissertation]. University of Heidelberg; 2018.
[9] Wazny K. Applications of crowdsourcing in health: An overview. J. Glob. Health. 2018 Jun;8(1):010502.
[10] Meakin JR, Ames RM, Jeynes JCG, Welsman J, Gundry M, Knapp K, et al. The feasibility of using citizens to segment anatomy from medical images: Accuracy and motivation. PLoS ONE. 2019 Oct;14(10):e0222523.
[11] Amgad M, Elfandy H, Hussein H, Atteya LA, Elsebaie MAT, Abo Elnasr LS, et al. Structured crowdsourcing enables convolutional segmentation of histology images. Bioinformatics. 2019 Sep;35(18):3461–7.
[12] Schlesinger D, Jug F, Myers G, Rother C, Kainmüller D. Crowd sourcing image segmentation with iaSTAPLE. arxiv:1702.06461 [cs]. 2017 Feb;.
[13] Jauer ML, Goel S, Sharma Y, Deserno TM, Gijs M, Berendshot TTJM, et al. STAPLE performance assessed on crowdsourced sclera segmentations. Proc SPIE. 2020 Mar;11318:113180K.
[14] Grote A, Schaadt NS, Forestier G, Wemmert C, Feuerhake F. Crowdsourcing of histological image labeling and object delineation by medical students. IEEE Trans Med Imaging. 2019 May;38(5):1284–94.
[15] Goodfellow I, Bengio Y, Courville A. Deep Learning. MIT Press; 2006.
[16] Hesamian MH, Jia W, He X, Kennedy P. Deep learning techniques for medical image segmentation: Achievements and challenges. J Digit Imaging. 2019 Aug;32(4):582–96.
[17] Albarqouni S, Baur C, Achilles F, Belagiannis V, Demirci S, Navab N. AggNet: Deep learning from crowds for mitosis detection in breast cancer histology images. IEEE Trans. Med. Imaging. 2016 May;35(5):1313–21.
[18] Keshavan A, Yeatman JD, Rokem A. Combining citizen science and deep learning to amplify expertise in neuroimaging. Front Neuroinform. 2019;13.
[19] Deserno TM, Meyer zu Bexten E, editors. Handbuch der Medizinischen Informatik. München: Hanser; 2002.
[20] Rogowska J. Overview and fundamentals of medical image segmentation. In: Handbook of Medical Imaging. Elsevier; 2000. p. 69–85.
[21] Arganda-Carreras I, Kaynig V, Rueden C, Eliceiri KW, Schindelin J, Cardona A, et al. Trainable Weka Segmentation: A machine learning tool for microscopy pixel classification. Bioinformatics. 2017 Aug;33(15):2424–26.
[22] Sirazitdinova E, Gijs M, Bertens CJF, Berendschot TTJM, Nuijts RMMA, Deserno TM. Validation of computerized quantification of ocular redness. Transl Vis Sci Technol. 2019 Nov;8(6):31.
[23] Chen LC, Papandreou G, Kokkinos I, Murphy K, Yuille AL. DeepLab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected CRFs. IEEE Trans Pattern Anal Mach Intell. 2018 Apr;40(4):834–48.
[24] Long J, Shelhamer E, Darrell T. Fully convolutional networks for semantic segmentation. In: 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR); 2015. p. 3431–40.
[25] Noh H, Hong S, Han B. Learning deconvolution network for semantic segmentation. In: 2015 IEEE International Conference on Computer Vision (ICCV). Santiago, Chile; 2015. p. 1520–8.
[26] Ronneberger O, Fischer P, Brox T. U-Net: Convolutional networks for biomedical image segmentation. arXiv:1505.04597 [cs]. 2015 May;.
[27] TM L. From plastic to gold: a unified classification scheme for reference standards in medical image processing. Proc SPIE. 2002;4684(3):1819–27.
[28] Turner AM, Kirchhoff K, Capurro D. Using crowdsourcing technology for testing multilingual public health promotion materials. J Med Internet Res. 2012 Jun;14(3):e79.
[29] Kashif M DT Jonas SM. Deterioration of r-wave detection in pathology and noise: a comprehensive analysis using simultaneous truth and performance level estimation. IEEE Trans Biomed Eng. 2017;64:2163–75.
[30] Warfield SK, Zou KH, Wells WM. Simultaneous Truth and Performance Level Estimation (STAPLE): An algorithm for the validation of image segmentation. IEEE Trans Med Imaging. 2004 Jul;23(7):903–21.

[31] Souza R, Lucena O, Bento M, Garrafa J, Rittner L, Appenzeller S, et al. Brain extraction network trained with "silver standard" data and fine-tuned with manual annotation for improved segmentation. In: 2019 32nd SIBGRAPI Conference on Graphics, Patterns and Images (SIBGRAPI). Rio de Janeiro, Brazil: IEEE; 2019. p. 234–40.

[32] Ganapathy N DT Swaminathan R. Adaptive learning and cross training improves r-wave detection in ecg. Comput Meth Prog Biomed. 2021;105931:online first.

[33] Yakubovskiy P. Segmentation Models Pytorch. GitHub repository, https://github.com/qubvel/segmentation_models.pytorch. 2020;.

[34] Kingma DP, Ba J. Adam: A method for stochastic optimization. arXiv:1412.6980 [cs]. 2017 Jan;.

[35] Buslaev A, Iglovikov VI, Khvedchenya E, Parinov A, Druzhinin M, Kalinin AA. Albumentations: Fast and flexible image augmentations. Information. 2020;11(2).

[36] Tokime RB, Elassady H, Akhloufi MA. Identifying the cells' nuclei using deep learning. In: 2018 IEEE Life Sciences Conference (LSC). Montreal, QC: IEEE; 2018. p. 61–4.

[37] Morschheuser B, Hamari J, Koivisto J. Gamification in crowdsourcing: A review. In: 2016 49th Hawaii International Conference on System Sciences (HICSS); 2016. p. 4375–84.