2023 45th Annual International Conference of the IEEE Engineering in Medicine & Biology Society (EMBC) | 979-8-3503-2447-1/23/$31.00 ©2023 IEEE | DOI: 10.1109/EMBC40787.2023.10341074

AI^{2}Seg: A Method and Tool for AI-based Annotation Inspection of Biomedical Instance Segmentation Datasets

Marcel P. Schilling[1], Lukas Klinger[1], Ulrike Schumacher[1], Svenja Schmelzer[2], Miguel Bordallo López[3], Britta Nestler[4], and Markus Reischl[1]

***Abstract*— In biomedical engineering, deep neural networks are commonly used for the diagnosis and assessment of diseases through the interpretation of medical images. The effectiveness of these networks relies heavily on the availability of annotated datasets for training. However, obtaining noise-free and consistent annotations from experts, such as pathologists, radiologists, and biologists, remains a significant challenge. One common task in clinical practice and biological imaging applications is instance segmentation. Though, there is currently a lack of methods and open-source tools for the automated inspection of biomedical instance segmentation datasets concerning noisy annotations. To address this issue, we propose a novel deep learning-based approach for inspecting noisy annotations and provide an accompanying software implementation, AI^{2}Seg, to facilitate its use by domain experts. The performance of the proposed algorithm is demonstrated on the medical MoNuSeg dataset and the biological LIVECell dataset.**

***Clinical relevance*— The contributed AI^{2}Seg method enables experts, such as practicing clinicians or biologists, to review noisy annotated datasets using a novel automated inspection algorithm embedded in a software package.**

I. INTRODUCTION

Recent advances in Deep Learning (DL) have led to impressive results in various computer vision applications [1], [2]. Deep Neural Networks (DNNs) can be used for accurate function approximation to solve complex tasks, i.e., assisted diagnosis in pathology [3], MRI image segmentation [4], analysis of high-throughput assays [5], or microscopy image processing [6].

However, training state-of-the-art supervised DL pipelines requires the annotation of datasets by specialists. According to [2], [7], [8], obtaining annotated datasets is a current challenge in DL because it is demanding, laborious, expensive, and necessitates domain expertise.

Since annotation is a workflow where different people with varying perspectives interact and experts' time is usually scarce, noisy annotations[1] pose an issue [9], especially when having small datasets like in biomedicine [10]. First, noisy annotations can impede the training of DNNs [9], [11], [12]. Second, in the presence of noisy annotations, quantitative metrics, i.e., in terms of DNN model performance [12], can produce misleading results, which may lead to not selecting the best model.

There are methods for automatically inspecting noisy annotations for classification or semantic segmentation tasks. However, the complexity of an inspection algorithm suited for noisy instance segmentation datasets is higher. Instances must be checked on a detection level and regarding their shape. To the best of our knowledge, there is no method for automatically inspecting noisy annotated instance segmentation datasets commonly used in biomedical applications. A tool with Graphical User Interface (GUI) is needed to make emerging methods usable for inspecting noisy annotated image datasets. However, there is a lack of non-commercial open-source software packages that integrate algorithms suitable for the automated inspection of instance segmentation annotations.

Therefore, we propose an approach utilizing uncertainty-aware DNNs to provide users with an ordered sequence of images that may contain noisy instances. Furthermore, our proposal yields a rating per instance to assist users by highlighting the most critical instances per image. To evaluate our approach, first, we simulate common types of annotation noise using two different biomedical image datasets (MoNuSeg [13] and LIVECell [14]). Second, the pipeline is evaluated using the noisy simulated datasets. Finally, the method is integrated into the open-source annotation tool KaIDA [8] as the plugin AI^{2}Seg. Hence, we enable the usage of our newly developed method by researchers.

Our contribution is organized as follows: Related work is presented in Section II. A description of our developed method is shown in Section III. The results are presented and discussed in Section IV. Finally, Section V summarizes the conclusions and provides an outlook for future work.

*This work was funded by the Future Fields initative "Screening Platform for Personalized Oncology" of the KIT strategy of excellence. The developed methods are also provided within the KNMFi program (no. 43.31.01) of the Helmholtz association. We also thank the Karlsruhe House of Young Scientists for funding the scientific exchange in terms of the "Networking Grant" that was useful in obtaining the results. This work was supported by the HoreKa Supercomputer through the Ministry of Science, Research, and the Arts Baden-Württemberg.

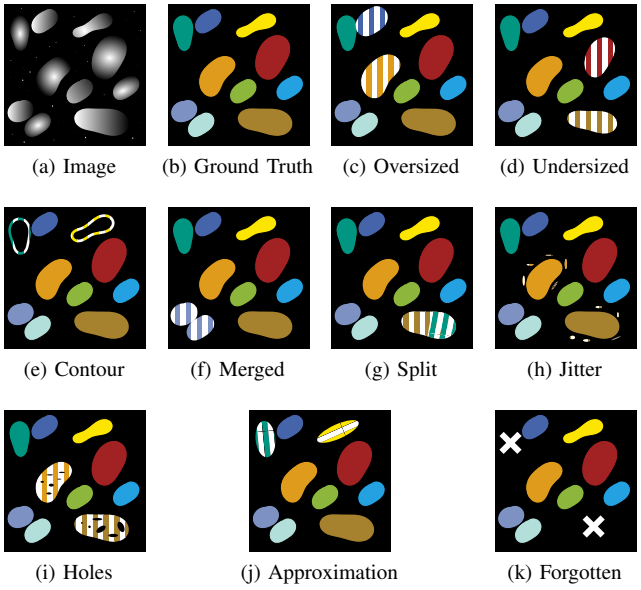
[1]Marcel P. Schilling, Lukas Klinger, Ulrike Schumacher, and Markus Reischl are with the Institute for Automation and Applied Informatics, Karlsruhe Institute of Technology, 76344 Eggenstein-Leopoldshafen, Germany marcel.schilling@kit.edu.

[2] Svenja Schmelzer is with the BioQuant, Heidelberg University, 69120 Heidelberg, Germany.

[3] Miguel Bordallo López is with the Center for Machine Vision and Signal Analysis, University of Oulu, 90570 Oulu, Finland.

[4] Britta Nestler is with the Institute for Applied Materials, Karlsruhe Institute of Technology, 76131 Karlsruhe, Germany.

[1]Deviating from the understanding of noise in measurement and control theory, namely stochastic deviations, the term "noise" is used as a collective term for annotation errors, biases, or similar effects.



(a) Image (b) Ground Truth (c) Oversized (d) Undersized

(e) Contour (f) Merged (g) Split (h) Jitter

(i) Holes (j) Approximation (k) Forgotten

Fig. 1: Types of Annotation Noise in Instance Segmentation. Considering annotation noise regarding instance segmentation, as depicted in [11], a synthetic image (a), the corresponding ground truth annotation (b), and various types of annotation noise (c-k) are given. Different colors encode instances. White line patterns indicate affected instances.

II. Related Work

According to Karimi et al. [9], annotation noise can be divided into i) variability regarding the annotation of a single expert (intra-annotator variability) and ii) variability between different experts (inter-annotator variability). When a division of labor is made during the annotation of a dataset, there is a risk of finding different annotation styles in the merged final dataset. For example, decision boundaries may differ between experts, such as when one pathologist annotates the edge of a cell and another pathologist does not mark this area. Fatigue, no defined annotation policy, pressure of time, varying qualifications of experts, or different hardware like varying monitor settings can be reasons for annotation variability. In addition, noisy annotations can occur when using the pre-annotation functionality of annotation tools. Issues may appear if a user only adheres to the initial annotations determined by a pre-annotation algorithm without controlling them. In the following, this work will focus on intra-annotator variability.

We base our work on the categorization of types of annotation noise given in [11] (cf. Figure 1). For instance, the shape of instances can be "oversized"/"undersized", only a "contour", "forgotten", or an "approximation" of the real one. In addition, "merging" two instances, "splitting" a single instance into two, small erroneous "jitter" instances, and "holes" within an instance are issues in practical projects.

Several works highlight that these noisy annotations can negatively affect the performance of DNNs, such as in classification [12], semantic segmentation [15], [16], tracking of cells [17], or instance segmentation [11]. When training a DNN, similar patterns in an image, e.g., a bright cell surrounded by a black background, but inconsistent annotations, would yield diverging optimization steps. Besides, it may cause the selection of suboptimal DL models, e.g., metrics are unreliable in the validation step dealing with noisy annotations [12]. Specifically, in [9], [10], the authors argue that noise has an enormous impact on small-scale datasets, as is often the case in biomedicine.

The reviews in [9], [18] discuss noisy annotations in DL. In general, two approaches exist for handling noisy annotated datasets: (i) directly detecting noisy annotations and correcting them or (ii) mitigating the effects of noisy annotations during the training of a DNN [19], [20]. As we focus on the detection of noisy annotations, related work of this strategy is presented in the following.

The CleanNet proposed by [21] involves computing a single feature vector per image and comparing it to a representative feature vector for a specific class. This method enables the detection of noisy annotations characterized by substantial deviations of feature vectors from the corresponding ones of the representative class. In the case of rank pruning [22], the DNN output is considered for detecting noisy annotations. The authors assume that probability distributions can be derived from network outputs (via softmax or sigmoid functions) and that a correct annotation is associated with a specific output.

Corbière et al. [23] argue that DNNs often lack calibration in their outputs and are too confident in failure cases. To overcome this problem, Köhler et al. [24] suggest using uncertainty-aware DNNs. Methods are, for instance, Monte Carlo dropout [25] to approximate Bayesian inference or Test Time Augmentation (TTA) [26] using augmented versions of an image to estimate the uncertainty. However, all methods mentioned above designed for classification problems are not transferable to instance segmentation, as the output in segmentation tasks is pixel-wise. In [15], we have already shown an automated approach suited for semantic segmentation. It is based on the idea of inspecting noisy datasets by searching for samples that show a large discrepancy between the noisy annotation and the DNN prediction, under the requirement that the prediction is confident.

Regarding tools, the open-source software CVAT [27] allows one to manually flag noisy samples by re-inspecting all images. However, only commercial software packages, such as hasty.ai [28], support an automated inspection. The tool hasty.ai integrates the idea of displaying images with wrong predictions to reveal noisy annotations. Nevertheless, this is only a beta function that is still under development. It lacks an open-source algorithm and a detailed method description.

In summary, two elements are missing in related work: i) an automated inspection pipeline for instance segmentation datasets and ii) an implementation of this method in an annotation tool to make it applicable to experts.

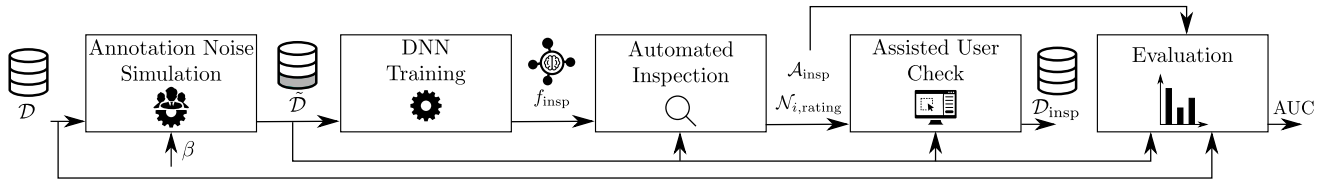


Fig. 2: Overview. A clean dataset $\mathcal{D}$ is corrupted, leading to a noisy dataset $\tilde{\mathcal{D}}$ controlled by β. Subsequently, a DNN is trained yielding f_{insp} on $\tilde{\mathcal{D}}$, which is used for the automated inspection. The inspection algorithm yields an ordered list $\mathcal{A}_{\text{insp}}$/an ordered rating list $\mathcal{N}_{i,\text{rating}}$. Using this information, users can specifically check the annotations yielding the inspected dataset $\mathcal{D}_{\text{insp}}$. The approach is evaluated using the AUC criterion.

Algorithm 1 Inspection Algorithm

Require: $\tilde{\mathcal{D}}, f_{\text{insp}}, \alpha, \gamma$
Ensure: $\mathcal{A}_{\text{insp}}, \{\ldots, \mathcal{N}_{i,\text{rating}}, \ldots\}$
  for $i = 1, \ldots, N$ **do**
    $\hat{\mathbf{y}}_i, \hat{\mathbf{u}}_i \leftarrow$ `predict_w_uncertainty`$(f_{\text{insp}}, \mathbf{x}_i)$
    $\mathcal{S}_i \leftarrow$ `state_assignment`$(\hat{\mathbf{y}}_i, \tilde{\mathbf{y}}_i)$
    $\mathcal{N}_{i,\text{rating}} \leftarrow$ `image_wise_inspection`$(\mathcal{S}_i, \hat{\mathbf{y}}_i, \tilde{\mathbf{y}}_i, \hat{\mathbf{u}}_i, \alpha, \gamma)$
  end for
  $\mathcal{A}_{\text{insp}} \leftarrow$ `overall_inspection`$(\{\ldots, \mathcal{N}_{i,\text{rating}}, \ldots\})$

III. MATERIAL AND METHODS

A. Overview

Figure 2 displays an overview of our developed pipeline for the automated inspection of noisy instance segmentation datasets. First, a noisy dataset $\tilde{\mathcal{D}}$ is generated from the dataset $\mathcal{D}$, where the annotations are assumed to be not noisy by simulating the annotation noise. The parameter $\beta \in [0, 1]$ controls the ratio of noisy instances per image. We refer to the types of annotation noise presented in [11] (cf. Figure 1, here $\beta = 0.2$ is illustrated). Second, a DNN f_{insp} is trained on the noisy dataset $\tilde{\mathcal{D}}$ to enable the subsequent automated inspection. It is assumed that most of the dataset is annotated correctly ($\beta < 0.5$). Details regarding the algorithm are discussed in Section III-B. The result of the automated inspection algorithm should yield an ordered list $\mathcal{A}_{\text{insp}}$ starting with images that contain many noisy annotations and ending with images that are mostly annotated correctly. In addition, the second output is a rating list $\mathcal{N}_{i,\text{rating}}$ containing score predictions $\hat{n}_{i,j} \in [0, 1]$ for each instance j given in the image i regarding their noise affection. Both information can be utilized for the assisted check by the user. After the assisted correction, the inspected cleaned dataset $\mathcal{D}_{\text{insp}}$ is obtained. Furthermore, we evaluate our pipeline considering the initial clean dataset $\mathcal{D}$, the noisy dataset $\tilde{\mathcal{D}}$, and the predictions $\mathcal{A}_{\text{insp}}$/ $\mathcal{N}_{i,\text{rating}}$ in terms of the Area Under Curve (AUC) score. Details in terms of our evaluation method are explained in Section III-C.

B. Inspection Algorithm

The pseudocode of our inspection algorithm is depicted in Algorithm 1. First, the trained DNN f_{insp} is used to generate an instance segmentation prediction $\hat{\mathbf{y}}_i$ given the image $\mathbf{x}_i \in \hat{\mathcal{D}}$ in order to compare it with the noisy annotation $\tilde{\mathbf{y}}_i$. Generally, two challenges occur in a naïve comparison approach: i) A prediction can contain wrong instances. ii) There can be the case of no matching between $\hat{\mathbf{y}}_i$ and $\tilde{\mathbf{y}}_i$, but considering the instance size may be worthwhile rather than treating the mismatch independently of the size.

To investigate the potential of using the network's uncertainty in our approach, we use an uncertainty-aware DNN predicting the uncertainty $\hat{\mathbf{u}}_i$ in parallel (cf. `predict_w_uncertainty`). TTA is used since the method is agnostic of the DNN. Inspired by the work in [29], the uncertainty of a single instance j is determined by cross-comparing the matching in each TTA prediction. Subsequently, the `state_assignment` method compares the noisy annotation $\tilde{\mathbf{y}}_i \in \hat{\mathcal{D}}$ and prediction $\hat{\mathbf{y}}_i$. We distinguish between the three states of instances j in $\tilde{\mathbf{y}}_i$/$\hat{\mathbf{y}}_i$: i) *odd* (intersection between prediction $\hat{\mathbf{y}}_i$ and annotation $\tilde{\mathbf{y}}_i$, but the same shape), ii) *additional* (instance only present in the annotation $\tilde{\mathbf{y}}_i$) and iii) *missing* (instance only present in the prediction $\hat{\mathbf{y}}_i$). The states are summarized in the list $\mathcal{S}_i$. Afterward, in the `image_wise_inspection`, the final rating $\mathcal{N}_{i,\text{rating}}$ is obtained. In general, using Intersection over Union (IoU) between annotated and predicted instances, we compute a mismatch score ($1 -$ IoU) per instance. To consider the size of an instance in the case of IoU $= 0$ (*additional*,*missing*), scaling the mismatch score is enabled. The size parameter γ can be used to scale the mismatch score w.r.t. the segment area by the maximum size of all given instances ($\gamma = 0$) or the maximum size of not matching instances ($\gamma = 1$). In addition, the final score per instance $\hat{n}_{i,j}$ is a weighted sum of the mismatch score and certainty of the predicted instance j controlled by an uncertainty weighting parameter $\alpha \in [0, 1]$ to focus on confident predictions. The parametrization $\alpha = 0$ means uncertainty is not taken into account. The inspection order regarding all samples, expressed by $\mathcal{A}_{\text{insp}}$, is obtained by averaging all inspection scores $\hat{n}_{i,j}$ per image, referred to as overall score (cf. `overall_inspection`). The parameters α, γ and their impact on the inspection performance are explored in Section IV-C.

C. Evaluation

To evaluate our presented method for automated instance segmentation inspection, we compute the Receiver Operating Characteristic (ROC). The ROC allows us to avoid defining a fixed threshold for $\hat{n}_{i,j}$ to mark an instance as noisy. Assuming that the initial dataset $\mathcal{D}$ is correct, we can determine whether an instance is noisy or not by

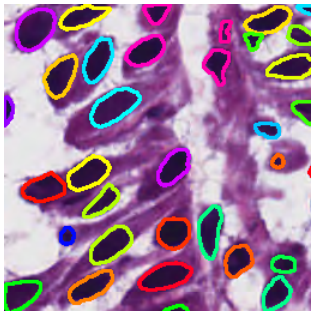
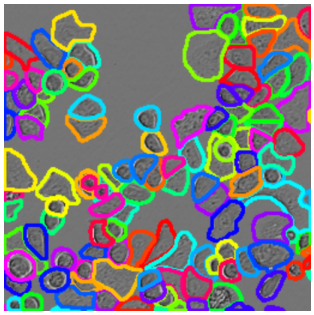



Fig. 3: Biomedical Datasets. Example images from MoNuSeg [13] (left) and LIVECell [14] (right) are shown. Color-coded contours indicate different instances.

comparing the annotations of the initial dataset $\mathcal{D}$ and the noisy dataset $\tilde{\mathcal{D}}$. It applies $n_{i,j} = 0$ for the case of no deviations between the annotations and conversely $n_{i,j} = 1$ for all deviations. Accordingly, the True Positive Rate (TPR) describes a successful detection of a noisy instance. In contrast, the False Positive Rate FPR characterizes the case of classifying a correct instance as a noisy one. The AUC $= \int_0^1$ TPR(FPR) dFPR combines both metrics and can be used to assess the quality of our annotation inspection. Note that an AUC > 0.5 characterizes better performance compared to random classifiers, while an AUC < 0.5 denotes classifiers that consistently indicate incorrect inspections. We empirically define a failed inspection for AUC ≤ 0.55 since the triggered inspection warnings seem arbitrary and are of little help from the user-experience perspective. In addition, we denote AUC* for the performance score using the optimal parameters $\alpha^*, \gamma^* = \arg\max_{\alpha,\beta}$ AUC(α, β) determined by a grid-search parameter study.

IV. RESULTS

A. Datasets

Two biomedical datasets are used to evaluate our proposal (cf. Figure 3). The MoNuSeg dataset [13] contains hematoxylin-eosin-stained histology images taken from seven different organs. This dataset aims to identify cell nuclei in the different tissue images. It contains 592 cropped images of size 256px $\times$ 256px. In addition, the LIVECell dataset [14] published by Sartorius, which includes eight different cell structures, is investigated. However, to reduce the computational cost, only a fraction (SkBr3 cells) of all samples is considered in our work. The extent of the reduced dataset is 2640 cropped samples of size 256px $\times$ 256px.

B. Implementation

1) DL: Each raw image is normalized to a range [0,1]. A U-Net [30] is utilized for predicting Euclidean distance maps, which are subsequently post-processed by a seed-based watershed algorithm to obtain instances [31]. We implement the DNN using the DL framework PyTorch Lightning. The Smooth L1 loss combined with Adam optimizer is considered for the training. Thereby, learning rate scheduling and early stopping are taken into account. The train-validation split is done randomly using a split ratio of 80/20. Additionally, we expand the training dataset using data augmentations (flipping, rotate/shift/scale, brightness/contrast adjustment) provided by Albumentations [32]. Finally, hyperparameters are determined using random search. The training is performed on SLURM cluster nodes equipped with four NVIDIA A100 Tensor Core GPUs and Intel Xeon Platinum 8368 CPU. All training runs are repeated five times using different random seeds.

2) AI^{2}Seg: The implementation of AI^{2}Seg is embedded in the `python`-based tool KaIDA [8] designed for assisted annotation by adding a new plugin. The GUI enables the direct application of our proposed method and is based on Qt. In addition, a user manual (`README` file) is provided to increase usability. The open-source software is publicly available at `https://git.scc.kit.edu/sc1357/kaida` in the form of a `pip` package. AI^{2}Seg is tested for various operating systems (including Windows 10 and Ubuntu 20.04) using `python` 3.8.5. In addition, a video to illustrate the workflow of our developed inspection pipeline is available at `https://osf.io/t8zx7/`.

Figure 4 displays the developed front-end for inspecting noisy instance segmentation datasets. The noisy user annotation, the DNN prediction, and an inspection visualization containing markings of potential noisy annotations are displayed. The user can decide whether to keep the annotation, accept the DL prediction, or re-annotate it. Based on the overall scores, the user can start with images that contain the noisiest predicted instances.

C. Inspection Performance and Discussion

Table I shows the results of the proposed evaluation (cf. Section III-C) for the two introduced datasets. The performance is evaluated for the mentioned types of annotation noise and various ratios $\beta \in \{0.10, 0.25, 0.50\}$ of affected instances. The mean values of five different initialized and trained DNNs are shown. We compare the performance of the naïve approach (AUC) and the elaborated approach considering the uncertainty of predictions/adapted size weighting (AUC*).

For 48 of 54 cases of different annotation noise types, the inspection performance is above the defined threshold AUC > 0.55. Therefore, our proposed AI^{2}Seg approach can be a helpful tool for users to detect/correct noisy annotations in most cases. Further, for most noise types, better results can be obtained with the advanced approach indicated by AUC* $>$ AUC. The results are consistent with the theoretical limits discussed when relying only on comparison and DNN prediction to obtain noisy instances. Taking uncertainty as well as size scaling into account is often meaningful. In addition, the optimal parameter setting depended on the noise type. Therefore, the parameters can be selected interactively in the GUI.

Failures can occur in the case of $\beta = 0.5$, which violates our assumption that most instances are correct.

In addition, there is no clear relationship between AUC/AUC* and β. On the one hand, this can be explained by

TABLE I: Inspection Performance. The inspection performance is evaluated for different types of annotation noise and noise ratios β regarding the datasets MoNuSeg and LIVECell. The metric AUC indicates the case of no uncertainty consideration/no size weighting. In contrast, AUC* shows the results of the optimal setting (α^*, γ^*) determined by a parameter study. A bold marking visualizes superior performance (AUC* $>$ AUC) using a more elaborated inspection method. Moreover, gray boxes ( ) indicate cases of failing inspection (AUC ≤ 0.55).

Noise Type	β	MoNuSeg		LIVECell	
		AUC	AUC*	AUC	AUC*
Oversized	0.10	0.728	**0.730**	0.773	**0.776**
	0.25	0.643	**0.644**	0.757	**0.761**
	0.50	0.392	0.392	0.608	**0.618**
Undersized	0.10	0.823	**0.837**	0.931	**0.945**
	0.25	0.812	**0.819**	0.935	**0.948**
	0.50	0.744	0.744	0.934	**0.942**
Contour	0.10	0.874	**0.879**	0.845	**0.860**
	0.25	0.791	**0.793**	0.812	**0.821**
	0.50	0.614	**0.628**	0.595	**0.635**
Merged	0.10	0.835	**0.836**	0.753	**0.761**
	0.25	0.829	**0.831**	0.754	**0.764**
	0.50	0.824	**0.825**	0.705	**0.710**
Split	0.10	0.854	**0.871**	0.910	**0.930**
	0.25	0.848	**0.857**	0.909	**0.922**
	0.50	0.811	**0.821**	0.891	**0.905**
Jitter	0.10	0.665	0.665	0.712	**0.717**
	0.25	0.776	0.776	0.806	0.806
	0.50	0.859	0.859	0.879	0.879
Holes	0.10	0.650	**0.656**	0.696	**0.697**
	0.25	0.650	**0.656**	0.704	**0.705**
	0.50	0.648	**0.653**	0.728	0.728
Approximation	0.10	0.505	0.505	0.623	**0.624**
	0.25	0.515	0.515	0.642	**0.643**
	0.50	0.509	**0.510**	0.679	0.679
Forgotten	0.10	0.696	**0.770**	0.736	**0.784**
	0.25	0.652	**0.712**	0.673	**0.725**
	0.50	0.436	**0.457**	0.479	**0.548**

the fact that the DNN used for inspection often degenerates as the number of noisy annotations increases. On the other hand, correct instances that are false positives have less impact on the metric due to a larger number of noisy instances.

Besides, considering the "approximation" noise type, which is only a minor deviation compared to the original clean annotation, the inspection is unsuccessful. Hence, another limitation of our developed method becomes apparent in the case of minor deviations. Note, however, that the initial dataset is assumed to be correct overall and annotation noise is generated only by our simulation. Consequently, small biases in the results may occur due to possibly noisy annotations included in the original dataset assumed to be correct.

V. CONCLUSIONS

Deep neural networks are potent methods for biomedical image processing. Nevertheless, the challenge of obtaining noise-free and consistent annotated datasets still remains. In particular, solutions for inspecting noisy biomedical instance segmentation datasets are lacking.

We contribute an elaborate approach to overcome this issue. Our algorithm utilizes an uncertainty-aware deep learning pipeline and a sophisticated comparison mechanism between noisy annotations and network predictions. To make it applicable, we provide the software implementation AI^{2}Seg integrated into the annotation tool KaIDA. Domain experts, such as biologists or practicing clinicians, can use the tool to check their annotated dataset, having assistance to increase the annotation quality in their projects. Our approach allows users to focus on the noisiest images or instances. Furthermore, we demonstrate the performance of the developed inspection pipeline using biological and medical datasets, respectively. Most of the given types of annotation noise can be managed by our proposal. We discussed the limitations of our approach in terms of too many noisy instances or noisy annotations in the assumed clean initial dataset.

Future work should investigate other computer vision tasks, such as object detection. Here, deep learning architectures and algorithms need to be redesigned for comparing noisy annotations with predictions. In addition, a coupling of our approach with the automated generation of annotations is of particular interest in the future. Further, testing our algorithm in other domains, i.e., autonomous driving, is another open work package. For further establishing the FAIR (Findable, Accessible, Interoperable, and Reusable) principles, the developed methods can be combined with research data infrastructures, e.g., Kadi4Mat [33]. Therefore, the integration of KaIDA in workflow management systems [34] can improve the usability to capture the whole research process.

REFERENCES

[1] C. Cao *et al.*, "Deep learning and its applications in biomedicine," *Genomics, Proteomics & Bioinformatics*, vol. 16, no. 1, pp. 17–32, 2018.

[2] N. O' Mahony *et al.*, "Deep learning vs. Traditional computer vision," in *Advances in Computer Vision*, 2019, pp. 128–144.

[3] K. Saednia, W. T. Tran, and A. Sadeghi-Naini, "A cascaded deep learning framework for segmentation of nuclei in digital histology images," in *Annual International Conference of the IEEE Engineering in Medicine & Biology Society*, 2022, pp. 4764–4767.

[4] A. Nejad, S. Masoudnia, and M.-R. Nazem-Zadeh, "A fast and memory-efficient brain mri segmentation framework for clinical applications," in *Annual International Conference of the IEEE Engineering in Medicine & Biology Society*, 2022, pp. 2140–2143.

[5] M. P. Schilling *et al.*, "Grid screener: A tool for automated high-throughput screening on biochemical and biological analysis platforms," *IEEE Access*, vol. 9, pp. 166 027–166 038, 2021.

[6] E. Moen, D. Bannon, T. Kudo *et al.*, "Deep learning for cellular image analysis," *Nature Methods*, vol. 16, no. 12, pp. 1233–1246, Dec. 2019.

[7] W. Chi *et al.*, "Deep learning-based medical image segmentation with limited labels," *Physics in Medicine & Biology*, vol. 65, no. 23, p. 235001, 2020.

[8] M. P. Schilling, S. Schmelzer, L. Klinger, and M. Reischl, "KaIDA: A modular tool for assisting image annotation in deep learning," *Journal of Integrative Bioinformatics*, p. 20220018, 2022.

[9] D. Karimi, H. Dou, S. K. Warfield, and A. Gholipour, "Deep learning with noisy labels: Exploring techniques and remedies in medical image analysis," *Medical Image Analysis*, vol. 65, p. 101759, 2020.

[10] D. Rolnick, A. Veit, S. J. Belongie, and N. Shavit, "Deep learning is robust to massive label noise," *CoRR*, vol. abs/1705.10694, 2017.

[11] M. P. Schilling, N. Ahuja, L. Rettenberger, T. Scherr, and M. Reischl, "Impact of Annotation Noise on Histopathology Nucleus Segmentation," *Current Directions in Biomedical Engineering*, vol. 8, no. 2, pp. 197–200, 2022.

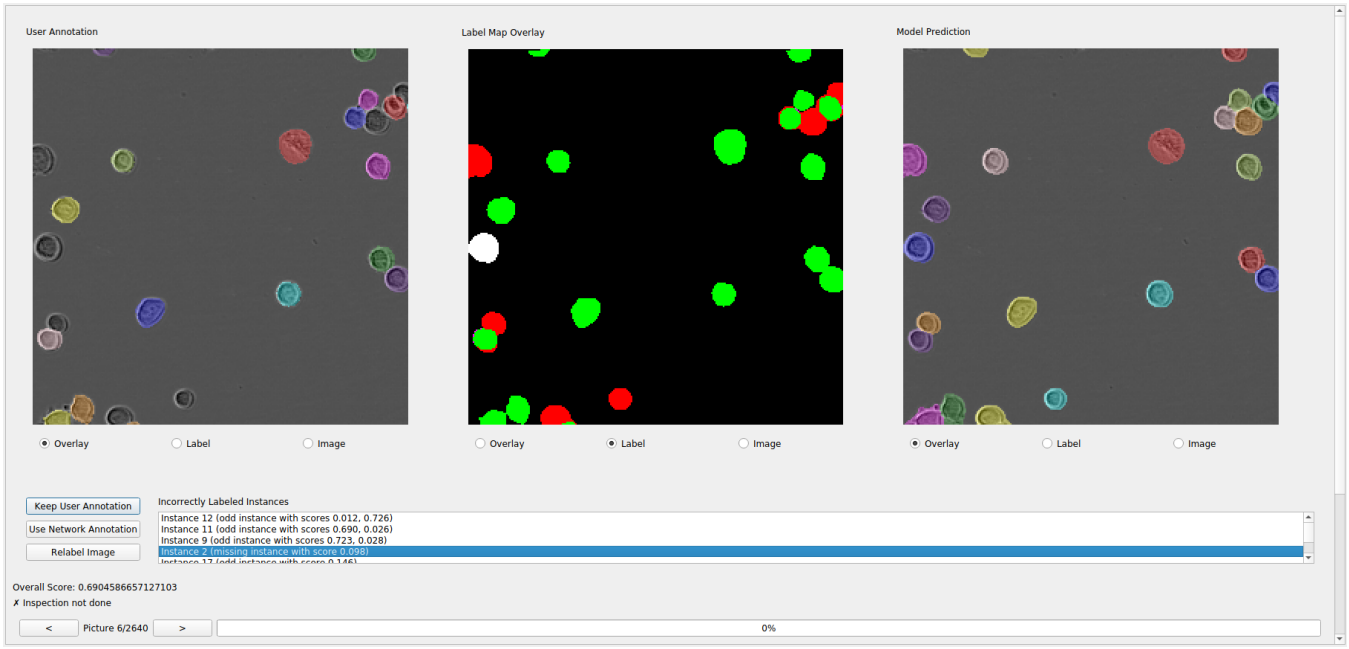


Fig. 4: AI^{2}Seg GUI. We present an example of the noise type "forgotten" ($\beta = 0.25$). Users are presented with three different views: i) the user annotation (left), ii) the prediction of the model (right), and iii) a combined image (center) showing correct instances in green while displaying noisy ones in red. Users can switch between an overlay image, only the annotation, or the raw image. Instance annotations are color-coded. In addition, all noisy instances are enumerated and the currently selected one is shown in white. The user can select to keep the annotation, use the DNN prediction, or re-annotate the image. The images are ordered based on the overall scores.

[12] C. G. Northcutt, A. Athalye, and J. Mueller, "Pervasive label errors in test sets destabilize machine learning benchmarks," in *Conference on Neural Information Processing Systems - Track on Datasets and Benchmarks*, 2021.

[13] N. Kumar *et al.*, "A multi-organ nucleus segmentation challenge," *IEEE Transactions on Medical Imaging*, vol. 39, no. 5, pp. 1380–1391, 2020.

[14] C. Edlund *et al.*, "LIVECell—A large-scale dataset for label-free live cell segmentation," *Nature Methods*, vol. 18, no. 9, pp. 1038–1045, 2021.

[15] M. P. Schilling *et al.*, "Automated annotator variability inspection for biomedical image segmentation," *IEEE Access*, vol. 10, pp. 2753–2765, 2022.

[16] S. Yu *et al.*, "Robustness study of noisy annotation in deep learning based medical image segmentation," *Physics in medicine and biology*, vol. 65, no. 17, 2020.

[17] K. Löffler, T. Scherr, and R. Mikut, "A graph-based cell tracking algorithm with few manually tunable parameters and automated segmentation error correction," *PLOS ONE*, vol. 16, no. 9, pp. 1–28, 2021.

[18] N. Tajbakhsh *et al.*, "Embracing imperfect datasets: A review of deep learning solutions for medical image segmentation," *Medical Image Analysis*, vol. 63, p. 101693, 2020.

[19] L. Yang, F. Meng, H. Li, Q. Wu, and Q. Cheng, "Learning with noisy class labels for instance segmentation," in *European Conference on Computer Vision*, 2020, pp. 38–53.

[20] L. Chen, Y. Fu, S. You, and H. Liu, "Hybrid supervised instance segmentation by learning label noise suppression," *Neurocomputing*, vol. 496, pp. 131–146, 2022.

[21] K.-H. Lee, X. He, L. Zhang, and L. Yang, "CleanNet: Transfer learning for scalable image classifier training with label noise," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2018.

[22] C. G. Northcutt, T. Wu, and I. L. Chuang, "Learning with confident examples: Rank pruning for robust classification with noisy labels," 2017.

[23] C. Corbière, N. Thome, A. Bar-Hen, M. Cord, and P. Pérez, "Addressing failure prediction by learning model confidence," in *Advances in Neural Information Processing Systems*, 2019, pp. 2902–2913.

[24] J. M. Köhler, M. Autenrieth, and W. H. Beluch, "Uncertainty based detection and relabeling of noisy image labels," in *IEEE Conference on Computer Vision and Pattern Recognition - Workshop*, 2019, pp. 33–37.

[25] Y. Gal and Z. Ghahramani, "Dropout as a Bayesian Approximation: Representing Model Uncertainty in Deep Learning," in *International Conference on Machine Learning*, vol. 48, 2016, pp. 1050–1059.

[26] N. Moshkov, B. Mathe, A. Kertesz-Farkas, R. Hollandi, and P. Horvath, "Test-time augmentation for deep learning-based cell segmentation on microscopy images," *Scientific Reports*, vol. 10, no. 1, p. 5068, 2020.

[27] B. Sekachev *et al.*, "Computer Vision Annotation Tool (CVAT)," 2020, [Online]. Available at: https://github.com/openvinotoolkit/cvat. Accessed: 2022-04-22.

[28] T. Rouillard, K. Proskudin, and A. Wennman, "Hasty.ai," [Online]. Available at: https://hasty.ai. Accessed: 2022-05-18, 2022.

[29] A. G. Roy, S. Conjeti, N. Navab, and C. Wachinger, "Bayesian QuickNAT: Model uncertainty in deep whole-brain segmentation for structure-wise quality control," *NeuroImage*, vol. 195, pp. 11–22, 2019.

[30] O. Ronneberger, P. Fischer, and T. Brox, "U-net: Convolutional networks for biomedical image segmentation," in *Medical Image Computing and Computer-Assisted Intervention*, 2015, pp. 234–241.

[31] T. Scherr, K. Löffler, M. Böhland, and R. Mikut, "Cell segmentation and tracking using CNN-based distance predictions and a graph-based matching strategy," *PLOS ONE*, vol. 15, no. 12, pp. 1–22, 2020.

[32] A. Buslaev *et al.*, "Albumentations: Fast and flexible image augmentations," *Information - an International Interdisciplinary Journal*, vol. 11, no. 2, 2020.

[33] N. Brandt *et al.*, "Kadi4Mat: A research data infrastructure for materials science," *Data Science Journal*, vol. 20, no. 8, pp. 1–14, 2021.

[34] L. Griem *et al.*, "KadiStudio: FAIR modelling of scientific research processes," *Data Science Journal*, vol. 21, no. 16, pp. 1–17, 2022.