Biological data annotation via a human-augmenting AI-based labeling interface

Andre Esteva (andre.esteva@gmail.com)
Salesforce AI Research https://orcid.org/0000-0003-1937-9682

Douwe van der Wal
Salesforce AI Research

Iny Jhun
Stanford University

Israa Laklouk
University of California, San Francisco https://orcid.org/0000-0001-9688-3167

Jeffrey Nirschl
Stanford Healthcare https://orcid.org/0000-0001-6857-341X

Lara Richer
https://orcid.org/0000-0002-1150-3610

Rebecca Rojansky
Stanford University

Talent Theparee
University of California, San Francisco https://orcid.org/0000-0002-7297-6040

Joshua Wheeler
Stanford University

Jorg Sander
Amsterdam University Medical Center

Felix Feng
University of California, San Francisco

Osama Mohamad
University of California, San Francisco

Richard Socher
Salesforce AI Research

Article

Keywords: artificial intelligence (AI), deep learning, HALI (Human-Augmenting Labeling Interface)

Posted Date: February 1st, 2021

DOI: https://doi.org/10.21203/rs.3.rs-146086/v1

Biological data annotation via a human-augmenting AI-based labeling interface

Douwe van der Wal[1], Iny Jhun[2*], Israa Laklouk[2*], Jeff Nirschl[2*], Lara Richer[3*], Rebecca Rojansky[2*], Talent Theparee[3*], Joshua Wheeler[2*], Jörg Sander[4], Felix Feng[3], Osama Mohamad[3], Richard Socher[1], Andre Esteva[1]

[*]Equal Contribution Authors

[1]Salesforce AI Research

[2]Stanford University

[3]University of California, San Francisco

[4]Amsterdam University Medical Center, University of Amsterdam

Abstract

Biology has become a prime area for the deployment of deep learning and artificial intelligence (AI), enabled largely by the massive datasets that the field can generate. Key to most AI tasks is the availability of a sufficiently large, labeled dataset with which to train AI models. In the context of microscopy, it is easy to generate image datasets containing millions of cells and structures. However, it is challenging to obtain large-scale high-quality annotations for AI models. Here we present HALI (Human-Augmenting Labeling Interface), a human-in-the-loop AI-based data labeling tool which begins un-initialized and learns annotations from a human, in real-time. Using a multi-part AI composed of three deep learning models, HALI learns from just a few examples and immediately increases both the efficiency of the annotator, and the quality of the annotations. Using a highly repetitive use-case --- annotating cell types --- and running experiments with seven pathologists --- experts at microscopic analysis of biological specimens --- we demonstrate an average human-efficiency improvement of 5.15x and an average data-quality boost of 4.34%, measured across four use-cases and two tissue stain types.

Introduction

The microscopic imaging of tissues, cells, and other relevant biological specimens is key to many areas of biological and medical research. Highly sophisticated tooling and workflows have developed around biological imaging. For instance, molecular staining protocols [1] -- chemical stains which selectively highlight different aspects of tissue (e.g. cell types, structures, organoids, etc.) -- are used from basic research to medical diagnostics. Furthermore, sample preparation has become highly standardized in a variety of domains (e.g. slide preparation in histopathological analysis [2]), enabling the large-scale digitization of data [3,4].

Digitization has fueled the advancement of computational methods to analyse data using a variety of techniques. The rise of deep learning (DL) methods [5] in the last decade have spurred progress across most fields that generate sufficiently abundant amounts of digital data. Visual

biology is a prime area for the deployment of deep-learning based computer vision (CV) techniques [6], as evidenced by a rapidly growing body of work[7]. At this intersection of fields, a number of remarkable capabilities have been developed. CV has demonstrated physician-level diagnostic performance on tissue slides [8], cellular segmentation performance that far surpasses classical techniques [9,10], the ability to virtually stain raw microscopic images as accurately as chemical stains [11], and many others.

Supervised learning -- in which computational models are trained using data points (e.g. histopathology image; raw microscopy image) and data annotations (e.g. 'cancerous' vs 'benign'; stained microscopy image) -- have been central to the success of CV in biology. Biologists have the distinct advantage of being able to generate massive amounts of data -- a single microscopy image can yield a gigabyte of visual data for algorithms to learn from. A disadvantage, however, is the difficulty and cost of obtaining complete annotations for datasets. Consider the ImageNet Large-scale Visual Recognition Challenge (ILSVRC) [6], a benchmark competition for object classification, localization, and detection in images of normal every-day objects (animals, furniture, etc.). It offered competitors a dataset of ~1 million images from 1000 object classes, made possible by the use of crowdsourced annotations from thousands of non-expert individuals. In contrast, computational biology competitions [4,12] typically offer only hundreds to thousands of labeled examples. The key bottlenecks are that annotators need to have certain levels of expertise and that annotation takes longer than conventional domains, making it difficult to obtain annotations at scale.

Practitioners typically rely on a number of computational advances in deep learning and related disciplines in order to work with smaller annotated datasets. Techniques like data augmentation[10] can synthetically expand the size of a dataset by creating modified copies of the original data that preserve its labels (e.g. distortions, rotations, changes in color balance, etc.), but their key effect is to smooth out data distributions -- they cannot synthesize new parts of the distribution. Generative Adversarial Networks (GANs)[13] have demonstrated remarkable abilities at creating synthetic data, but rely heavily on sufficiently large datasets from which to learn most of a dataset's distribution. Transfer learning, in which models are first trained on large datasets from a different domain (e.g. ImageNet) and then fine-tuned on small datasets for the task at hand (e.g. microscopy data), has become the standard technique across medical and biological CV use-cases [14]. Recently, self-supervised learning techniques -- in which synthetic labels are *extracted* from unlabeled data -- have started to mature, demonstrating promise in decreasing the need for abundant labeled data [15].

In spite of these advances, data annotations continue to be essential in training AIs, and supervised learning continues to be the standard technique. Significant efforts have been put into developing labeling interfaces that allow experts to efficiently label medical data[16–18]. However, annotating this data for the purposes of AI development continues to require substantial computational knowledge, both in terms of annotating the right data, and training AI models. For instance, variables such as staining inconsistency, scanned artifacts, and natural

changes in object appearance, combined with the large amount of data generated in microscopy, can adversely affect the quality of annotated data, and AIs trained from it.

Here we present a human-augmenting AI-based labeling interface (HALI), in which initially untrained deep learning models learn from human demonstration, train themselves, and begin to augment human annotation ability. The effect is to increase annotation speed and annotation quality, enabling the annotation of datasets which were previously cost-prohibitive.

Using challenging and mundane labeling tasks, we demonstrate that HALI can significantly improve the speed of annotation, and the quality of annotations. Specifically, we outfit a data annotation interface with three deep learning models - a segmentation model, a classifier, and an active learner - which work in synchrony to (1) learn the labels provided by an annotator (2) provide recommendations to that annotator designed to increase their speed, and (3) determine the next best data to label to increase the overall quality of annotations while minimizing total labeling burden. The models work passively in the background without the need for human intervention, essentially enabling a non-computationally savvy biologist to train their own personalized AI for workflow support, and downstream AI development.

To establish an approximate lower bound on human augmentation, we experiment with challenging tasks, working with highly trained expert annotators. Specifically, we select the task of cellular annotation on tissue images - a highly repetitive, time-intensive task, broadly useful across domains of biology - and we select pathologists as annotators. If this method can augment trained specialists on challenging tasks, it is likely to generalize well to less trained annotators on simpler tasks.

We run experiments using four different cellular labeling tasks on two visually distinct stains -- Hematoxylin and Eosin (H&E), and immunohistochemistry (IHC). Working with seven pathologists from Stanford and the University of California at San Francisco (UCSF), we demonstrate that our system can increase labeling speed by an average factor of 5.15x, and increase the effectiveness of the annotated data by 4.34%. The latter is determined by computing the AUC of accuracy vs number of training samples for an AI trained on data annotated with HALI, and comparing it to the AUC of an AI trained without human augmentation.

Results

System Architecture

The labeling workflow of our system is depicted in the illustration of Figure 1. Given a large microscopy image (e.g. histopathology whole slide images (WSI), in the provided example), an annotator will begin by labeling points within a small region of the WSI. Once they do, an untrained classifier will begin training itself on these annotations, learning to distinguish between

the various classes provided. Once the classifier sees sufficient data, it then starts performing two functions. First, it renders suggestions to the annotator, which the annotator may accept or change. In practice, we find that as the classifier's accuracy improves and the suggestions become indistinguishable from the annotator-provided labels, the speed of annotation significantly accelerates -- annotators can scan over a set of suggestions and approve/disapprove much faster than they can individually annotate each point. Second, the classifier converts square image patches that circumscribe the labeled data points into feature vectors which are fed into an active learning model. The active learner takes these features, along with features from the circumscribed squares of the remaining cells in the rest of the image, to determine the next best patch for annotation. The net effect of these two models is to essentially guide the annotator around the image, rapidly sampling from a diverse and representative set of points. Along with the rest of the system architecture, they form a human-in-the-loop AI interface that learns from demonstration and enhances human performance.

The specific technical steps required to annotate an image break down into two components: (1) a data pre-processing step to prepare the image for enhanced annotation, (2) human annotation through an AI-augmented labeling interface. The structure of this system is depicted in Figure 2. The first component (Fig 2a) uses a segmentation model (Hover-Net[9]) to segment each cell in the tissue, and determine the smallest containing bounding box for each. In our setup, we train a separate segmentation model for each of the two stains of interest, using the QuPath labeling interface[16] to generate the requisite cellular bounding boxes segmentation masks. Once a segmentation model is trained on a particular stain, it will work for new images that use that particular stain. The positions of these are then sent to the labeling interface for use in real-time. Adapting this step to a new stain type simply requires re-training the segmentation model with an example image.

HALI's system architecture (Fig 2b) is built through a microscopy labeling interface outfitted with two deep learning algorithms - a classification model, and an active learning model. We use the SlideRunner open-source labeling interface to build on[19], a PanNuke-dataset[20] pre-trained ResNet-18[21] as our classifier, and the Coreset[22] method for active learning. Both ResNets and Coreset are state-of-the-art models in image classification and active learning, and PanNuke is a dataset of 205,343 annotated cells, effective for pretraining this task. Each of these components is completely modular, and can be easily replaced with equivalent methods (e.g. a different labeling interface) for new use-cases. Once an annotator begins labeling data points (green and blue bounding boxes, Fig 2b), the system stores these data points alongside the unlabeled data pool, and finetunes the classifier on these labels. Once sufficiently many points are annotated (around 30, in our case), the classifier begins rendering predictions in the interface (muted blue and muted green bounding boxes, Fig 2b) which the annotator can accept or deny. Further, the classifier performs a feedforward pass over all the data (labeled and unlabeled) and feeds their resultant feature vectors - high-dimensional representations of the cells - into the active learner. This model then determines, from the unlabeled set U, an unlabeled subset S which is maximally diverse, and expected to most improve the performance

and generalizability of a model trained on *L* + *S,* where *L* is the labeled data. The subset is sent back to the labeling interface, which chooses, as the next regional patch to annotate, the patch that contains the most points of *S*.

Experiments

To test the impact of HALI on data annotation, we execute two experiments designed to test for improvements to the efficiency of annotation, and the effectiveness of the annotated data.

Annotation Efficiency: Here, an expert annotator is asked to find a patch in an image containing about 200 nuclei (around 30x magnification), in which the classes are roughly balanced. As a control, they are timed as they annotate all cells in the patch, without AI augmentation. Next, they find a new patch of equivalent composition, and are timed as they annotate all cells, with AI augmentation as described above. Their efficiency, as measured by the number of annotated cells per second, is compared in the two trials.

Annotation Effectiveness: In this experiment, an expert annotator begins by labeling 10 cells of each class, to initialize the classifier. They then begin annotating cells, following (and possibly correcting) the suggestions of the classifier, while being guided around the slide by the active learner. As a control, they repeat this experiment on the same interface but with all deep learning models deactivated.

For each of the experiments, we test 7 pathologists (from Stanford and UCSF) on four different binary use cases (see Fig 3):

1) **Tumor infiltrating lymphocytes (TIL) [H&E]:** The presence of sufficiently dense TILs can provide prognostic information and aid in measuring the response to treatments[23].
2) **Tumor cells [H&E]:** Quantifying the fraction of tumor cells in a tissue sample is a challenging task that suffers from pathologist variability, and is of value to therapeutic decision making as well as diagnostics[24].
3) **Eosinophils [H&E]**: Eosinophilic esophagitis is a chronic immune system disease. Quantitating eosinophils is necessary for diagnosis[25].
4) **Ki-67 [IHC]**: The Ki-67 stain is a marker of cellular proliferation. The ratio of positive to negative tumor cells can have prognostic significance[26].

In each use-case, the annotator labels two classes of cells: (1) the cell type of interest (2) all other cells in the tissue. All four use cases are real tasks with diagnostic value. The first three are stained with H&E, while the fourth - stained with IHC - is selectively chosen to demonstrate generalizability across stain types.

The results of these experiments are summarized in the table of Fig 4a. The efficiency boost across the pathologists, when using HALI, ranges from 3.83x to 8.03x their original efficiency, as measured by the number of cells per second annotated. The average efficiency boost is

5.15x. Intuitively, the efficiency boost is greater on tasks with greater visual differences between the two classes. Eosinophils (3.83x boost) are a type of white blood cell with multi-lobulated nuclei and granular eosinophilic cytoplasm. In contrast, the Ki-67 staining protein selectively attaches to proliferating cells. This intensity difference simplifies the Ki-67-based task (8.03x boost) of distinguishing proliferating cells from non-proliferating cells. TILs (4.46x boost) and Tumor Cells (6.16x boost) lie around the average value. Individual efficiency boosts across the use-cases are shown in Fig 4b. Variability can be observed within each use-case. Behaviorally, individual annotators interact differently with the interface, gaining or losing trust in model predictions as a function of model accuracy.

The effectiveness boost across pathologists, when using HALI, ranges from 1.38% to 6.43%, averaging 4.34%. The effectiveness of an annotated dataset is defined as the area under the curve (AUC) of validation accuracy versus N, the number of training samples, with $N < 200$, for a model trained with this dataset. The AUC of such a curve yields an intuitive measure of how *quickly* the dataset becomes of sufficient *quality* to learn the task at hand. The higher the AUC, the faster a model converges, the fewer data points are needed to learn the proper distribution. The exact impact of this value on the accuracy improvement of a model trained on the annotated dataset is a function of the individual shapes of the AUC curves. See Online Methods for full details on experimental parameters. The effectiveness improvement in one annotated dataset over another is then the AUC ratio between them. See Fig. 5 for an example comparison plot of a dataset annotated with HALI versus one annotated without. Here, the AUC ratio is 5.3%, and a model trained with 50, 75, and 100 training examples from HALI benefits from an 11%, 11%, and 5% boost in model validation accuracy, respectively. In benchmark machine learning competitions, top performing models typically win by fractions of a percent to single percentage points[6,27].

Note that the ResNet classifier used in these experiments is pre-trained on the PanNuke dataset -- a dataset of H&E-stained cells -- and observes performance on Ki-67 that is on par with the other use-cases, pointing to the generalizability of this method. Adapting HALI to new use-cases is an iterative process involving testing the platform as-is, then potentially replacing the classification and/or the segmentation models with ones trained on the stain at hand. The active learner is generic, operating on feature vectors as opposed to raw input data.

Discussion

Here we present HALI, a human-augmenting AI-based labeling interface designed to learn from human data annotators in real-time, augmenting their abilities and boosting both their annotation speed and annotation effectiveness. Using four highly-repetitive binary use-cases across two stain types, and working with expert pathologist annotators, we demonstrate a 5.15x average improvement in labeling efficiency and a 4.34% average improvement in labeling effectiveness. Decreasing the time and cost of data annotation has the potential to enable the development of previously inaccessible AI models across a range of valuable domains. HALI can serve

biologists in data analysis by allowing them to collect quality datasets on their specific use-cases, with minimal computational knowledge, for the training of AI models.

Future work in this direction will involve expanding the capabilities of the system across tasks and image types. This could be achieved by working with more complex biological targets, different stain types, or three-dimensional images (e.g. z-stacks of microscopy cross-sections). Efforts could involve training models that can effectively transfer-learn across similar tasks - for instance, using a single classifier for all cellular annotation tasks, possibly via meta-learning[28]. Eventually, computational layers could be added on top of HALI which build it into a distributed auto-ML[29] style platform, which can auto-detect the task at hand, select the best pre-trained model and model type, and concurrently learn from a number of annotators. As this type of technology matures in the research community, decreasing the time and cost of data annotation, more areas of biology will begin to benefit from the high value-add of AI data analysis.

Author Contributions

DW built the labeling interface, trained the requisite models, and tested the pathologists. IJ, IL, KN, LR, RR, TT, and JW served as expert pathologists for the annotation experiments. JS supervised the project. FF and OM provided clinical guidance. RS contributed ideas related to the inception of the platform and broad vision. AE supervised the project, designed the experiments, recruited the team, and tested pathologists.

References

1. Beveridge, T. J., Lawrence, J. R. & Murray, R. G. E. Sampling and Staining for Light Microscopy. *Methods for General and Molecular Microbiology* 19–33 (2007).
2. Slaoui, M. & Fiette, L. Histopathology procedures: from tissue sampling to histopathological evaluation. *Methods Mol. Biol.* **691**, 69–82 (2011).
3. Veta, M. *et al.* Predicting breast tumor proliferation from whole-slide images: The TUPAC16 challenge. *Med. Image Anal.* **54**, 111–121 (2019).
4. Litjens, G. *et al.* 1399 H&E-stained sentinel lymph node sections of breast cancer patients: the CAMELYON dataset. *Gigascience* **7**, (2018).
5. LeCun, Y., Bengio, Y. & Hinton, G. Deep learning. *Nature* **521**, 436–444 (2015).

6. Russakovsky, O. *et al.* ImageNet Large Scale Visual Recognition Challenge. *Int. J. Comput. Vis.* **115**, 211–252 (2015).
7. Litjens, G., Kooi, T., Bejnordi, B. E., Setio, A. A. A. & Ciompi, F. A survey on deep learning in medical image analysis. *Medical image* (2017).
8. Nagpal, K. *et al.* Development and validation of a deep learning algorithm for improving Gleason scoring of prostate cancer. *NPJ Digit Med* **2**, 48 (2019).
9. Graham, S. *et al.* Hover-Net: Simultaneous segmentation and classification of nuclei in multi-tissue histology images. *Med. Image Anal.* **58**, 101563 (2019).
10. Ronneberger, O., Fischer, P. & Brox, T. U-Net: Convolutional Networks for Biomedical Image Segmentation. in *Medical Image Computing and Computer-Assisted Intervention – MICCAI 2015* 234–241 (Springer International Publishing, 2015).
11. Christiansen, E. M. *et al.* In Silico Labeling: Predicting Fluorescent Labels in Unlabeled Images. *Cell* **173**, 792–803.e19 (2018).
12. Verma, R. *et al.* Multi-organ Nuclei Segmentation and Classification Challenge 2020. (2020) doi:10.13140/RG.2.2.12290.02244/1.
13. Goodfellow, I. *et al.* Generative Adversarial Nets. in *Advances in Neural Information Processing Systems* (eds. Ghahramani, Z., Welling, M., Cortes, C., Lawrence, N. & Weinberger, K. Q.) vol. 27 2672–2680 (Curran Associates, Inc., 2014).
14. Esteva, A. *et al.* Dermatologist-level classification of skin cancer with deep neural networks. *Nature* **542**, 115–118 (2017).
15. Jing, L. & Tian, Y. Self-supervised Visual Feature Learning with Deep Neural Networks: A Survey. *IEEE Trans. Pattern Anal. Mach. Intell.* **PP**, (2020).
16. Bankhead, P. *et al.* QuPath: Open source software for digital pathology image analysis. *Sci. Rep.* **7**, 16878 (2017).

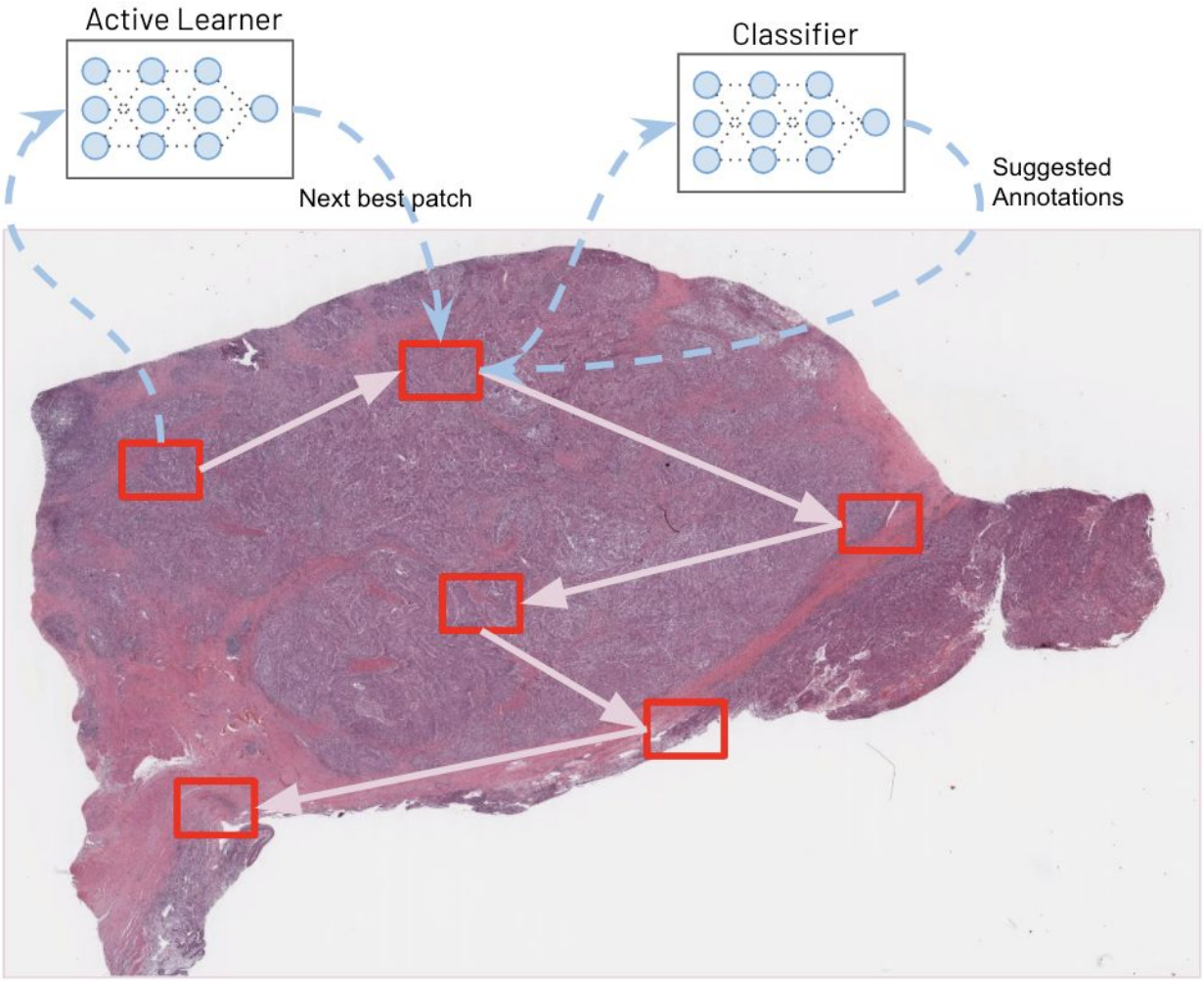
17. McQuin, C. *et al.* CellProfiler 3.0: Next-generation image processing for biology. *PLoS Biol.* **16**, e2005970 (2018).
18. Nalisnik, M. *et al.* Interactive phenotyping of large-scale histology imaging data with HistomicsML. *Sci. Rep.* **7**, 14588 (2017).
19. Aubreville, M., Bertram, C., Klopfleisch, R. & Maier, A. SlideRunner. in *Bildverarbeitung für die Medizin 2018* 309–314 (Springer Berlin Heidelberg, 2018).
20. Gamper, J., Koohbanani, N. A., Benet, K., Khuram, A. & Rajpoot, N. PanNuke: An Open Pan-Cancer Histology Dataset for Nuclei Instance Segmentation and Classification. *Digital Pathology* 11–19 (2019) doi:10.1007/978-3-030-23937-4_2.
21. He, K., Zhang, X., Ren, S. & Sun, J. Deep Residual Learning for Image Recognition. *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (2016) doi:10.1109/cvpr.2016.90.
22. Sener, O. & Savarese, S. Active Learning for Convolutional Neural Networks: A Core-Set Approach. in *International Conference on Learning Representations* (2018).
23. Hendry, S. *et al.* Assessing Tumor-infiltrating Lymphocytes in Solid Tumors: A Practical Review for Pathologists and Proposal for a Standardized Method From the International Immunooncology Biomarkers Working Group: Part 1: Assessing the Host Immune Response, TILs in Invasive Breast Carcinoma and Ductal Carcinoma In Situ, Metastatic Tumor Deposits and Areas for Further Research. *Adv. Anat. Pathol.* **24**, 235–251 (2017).
24. Smits, A. J. J. *et al.* The estimation of tumor cell percentage for molecular testing by pathologists is not accurate. *Mod. Pathol.* **27**, 168–174 (2014).
25. Dellon, E. S. Eosinophilic esophagitis: diagnostic tests and criteria. *Curr. Opin. Gastroenterol.* **28**, 382–388 (2012).
26. Ellis, M. J. *et al.* Ki67 Proliferation Index as a Tool for Chemotherapy Decisions During and

After Neoadjuvant Aromatase Inhibitor Treatment of Breast Cancer: Results From the American College of Surgeons Oncology Group Z1031 Trial (Alliance). *J. Clin. Oncol.* **35**, 1061–1069 (2017).

27. Aresta, G. *et al.* BACH: Grand challenge on breast cancer histology images. *Med. Image Anal.* **56**, 122–139 (2019).
28. Vanschoren, J. Meta-Learning: A Survey. *arXiv [cs.LG]* (2018).
29. He, X., Zhao, K. & Chu, X. AutoML: A Survey of the State-of-the-Art. *Knowledge-Based Systems* 106622 (2020).
30. Weinstein, J. N. *et al.* The Cancer Genome Atlas Pan-Cancer analysis project. *Nature Genetics* vol. 45 1113–1120 (2013).

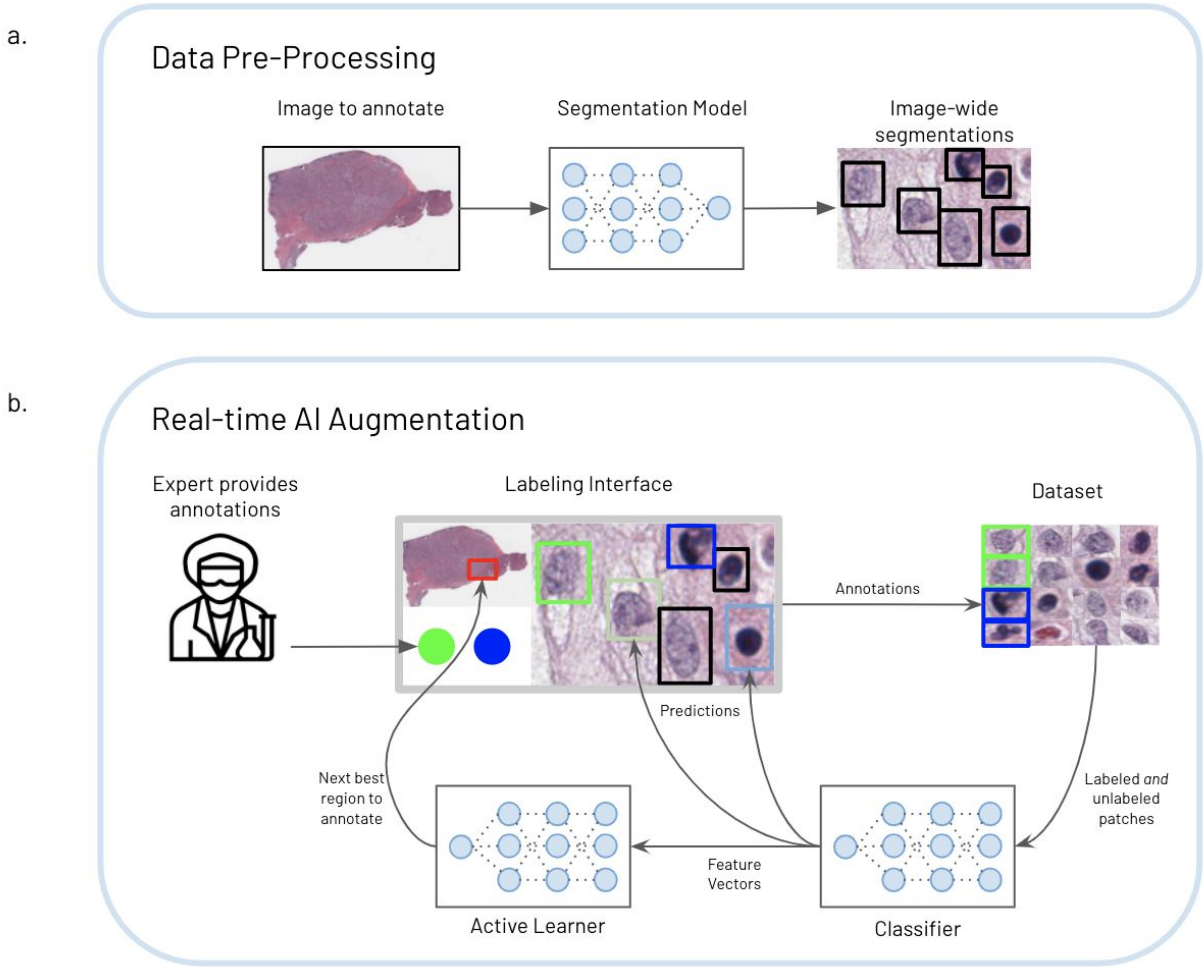
Figures

Figure 1. HALI: Human-Augmenting AI-based Labeling Interface.



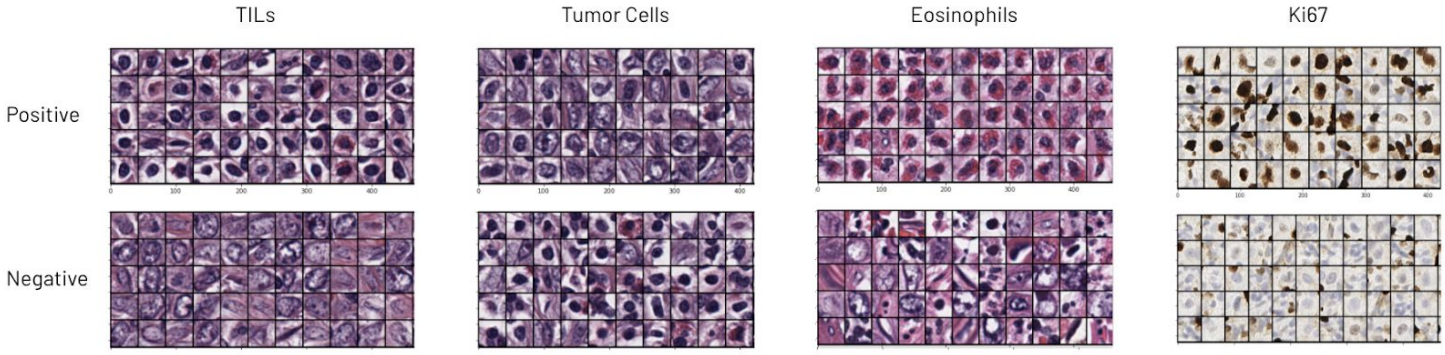
As a human annotator labels data, an active learning algorithm shuttles the annotator around the image by identifying the next best visual features to annotate. Simultaneously, other AIs make labeling suggestions designed to significantly accelerate annotation speed (not pictured - see Fig 3. System Architecture). Together, they give annotators the ability to train personalized AI models, enabling them to generate high-quality labeled datasets for otherwise intractably large images.

Figure 2. System Architecture.



(a) **Data Pre-Processing**. Digital images are first pre-processed by passing through a deep learning model (HoverNet[9]) which segments and generates bounding boxes for each cell. The image and bounding boxes are then used in real-time via a labeling interface outfitted with two AI models that serve to augment and accelerate expert labeling (b). **Real-time AI Augmentation**. As annotators provide cellular labels (bright green/blue boxes denote different classes), they are stored in a dataset and iteratively used to fine-tune a ResNet classification[21] model, pre-trained on the PanNuke cellular dataset, but initially *untrained* on the task at hand. As the classifier learns the annotations, it renders ever more accurate predictions (pastel green/blue boxes) to the annotator. Simultaneously, it feeds in high-dimensional feature vector representations of the labeled *and* unlabeled dataset to the active learner, which determines the next best patch that an annotator should label. Together they increase the speed of annotation, and the quality of the labeled dataset.

Figure 3. Experimental Use-cases.



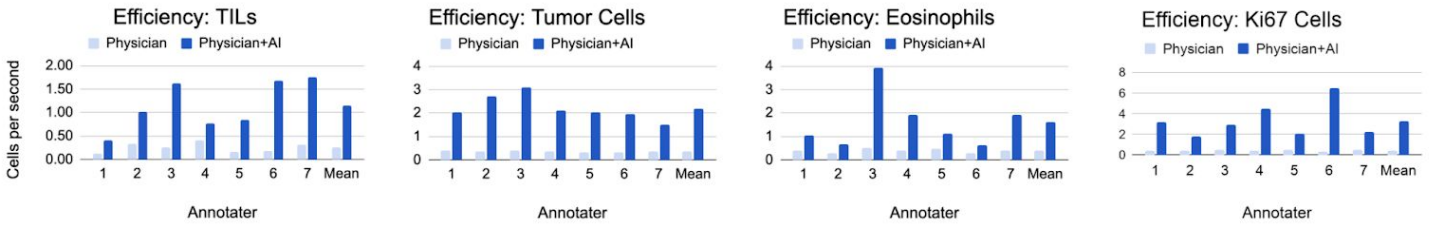
The task of cellular annotation is chosen, specifically due to its highly repetitive nature, and the difficulty of complete data annotation (e.g. a section of tissue may contain 900,000 cells). Four use-cases highlight the generalizability of this method across stain types and cell types. From left to right: tumor-infiltrating lymphocytes (TILs), Tumor Cells, Eosinophils, and Ki-67-stained cells of arbitrarily sufficient size, as determined by the annotators. Top row shows example cells of the specified type (positive), bottom row shows examples of all others (negative).

Figure 4. Experimental Results.

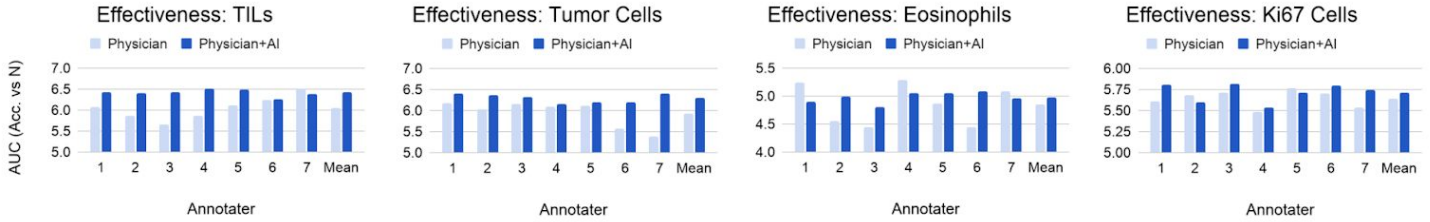
a.

Use-case	TILs	Tumor cells	Eosinophils	KI67 Cells	**Average**
Efficiency Increase	4.46x	6.16x	3.83x	8.03x	**5.15x**
Effectiveness Increase	6.33%	6.43%	3.21%	1.38%	**4.34%**

b.

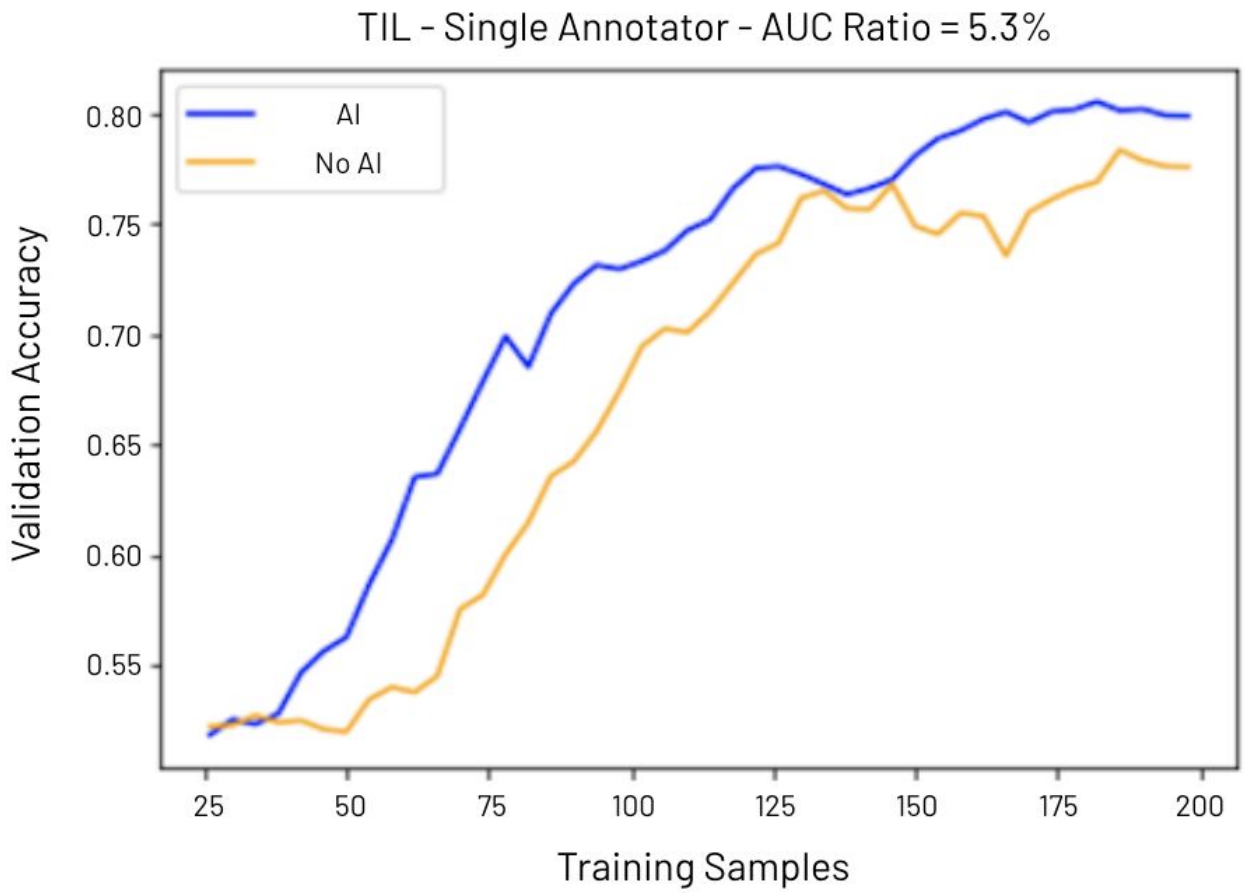


c.



a. The average efficiency and effectiveness increase for the four use-cases considered (TILs, Tumor Cells, Eosinophils, and Ki-67 cells), altogether averaging a 5.15x efficiency improvement and 4.34% effectiveness improvement. Efficiency is measured by the number of cells per second that an annotator can label. Effectiveness is measured as the area under the Accuracy vs Number of Samples (*N*) plot, bounded by $N < 200$, for an AI model trained on the resultant annotated dataset (see example in Fig. 5). b. Efficiency results on the seven tested annotators - top histopathology trainees from Stanford and the University of California, San Francisco. C. Effectiveness results on the same set of trainees.

Figure 5: Effectiveness Metric



We consider a dataset, D, to be more effective than another, E, if a model trained on D has a higher validation accuracy than a model trained on E, and both D and E cost the same to annotate (i.e. both have the same number of data points and annotations). To this end, we use the AUC of a model's validation accuracy vs number of training samples as a measure of effectiveness, with a higher AUC indicating higher effectiveness, and the ratio in AUC between two curves allowing a comparison between two models. In the example above (use-case: tumor infiltrating lymphocytes), the dataset generated with AI augmentation has an absolute validation accuracy improvement of 0.11, 0.11, and 0.05, over a dataset generated without AI augmentation, for 50, 75, and 100 training samples, respectively. The AUC ratio of the two curves is 5.3%.

Online methods

Labeling Platforms

SlideRunner[19] is an open-source tool to annotate objects in whole slide images written in Python. The tool supports annotating objects of any size, using point, bounding box or polygon annotations. It supports plugins that can interact with the data and potentially deploy AI techniques. A version of SlideRunner, augmented with the various AI modules described in Fig. 2, was used for all experiments.

QuPath is an extensive bioimage analysis program written in Java. It contains many built-in workflows for tasks such as cell segmentation and classification. In addition it is able to process slides in parallel and the program can be extended by writing scripts. QuPath was used to obtain segmentations (i.e. bounding boxes) for the Ki-67 experiments, using the watershed cell detection plugin.

Segmentation Model

Here we use a HoverNet model[9] for the H&E-stained use-cases, as it achieves state-of-the-art performance against other models (e.g. Unet[10]) at segmentation tasks in tissue. For the IHC-stained use-case (Ki-67), we fine-tune HoverNet using the segmentation data obtained from QuPath (see above).

Classification Model

Before we can use the ResNet-18 model with a small dataset, it needs to be pretrained.
For pretrained, the PanNuke[20] dataset is used, which is composed of 256 x 256 pixel images, containing just over 200,000 nuclei. The images originate from 19 different kinds of tissue and contain 5 different classes of nuclei. All nuclei in the patches have been labelled. We calculated the centroid of every nuclei, and extracted a 40 x 40 patch around the centroid, omitting nuclei at the edges of the images when the window did not fully fit within the image. These small patches are used to train the model, using PanNuke fold 1 and fold 2 for training and fold 3 for testing. Fold 1 and 2 have been combined to calculate the mean and standard deviation in the pixel values. All patches have been normalized by subtracting the mean and dividing by the standard deviation.

To finetune the model, we freeze all layers, and replace the final layer with two untrained fully-connected layers containing 32 nodes each, with a final output layer containing two nodes corresponding to the binary classification cases used in the experiments.

As the user starts labeling, data is obtained that is used to finetune the ResNet in real-time. The data labeled by the expert is split into a training and validation set. 75% of this data is used for

training, and the other 25% validation. When the classes are not balanced, the less common classes are oversampled, in order to present the model a balanced dataset.

The model is optimized using Stochastic Gradient Descent(SGD), with a learning rate of 0.00001 and momentum 0.9. The model is trained for 100 epochs, but is early stopped if the validation score does not improve for 10 epochs.

During system usage, the classification model finetunes itself for every 5 newly labelled data points (using all labelled data). The cells in the current image patch are predicted and rendered to the user as suggestions. The feature representation for all data (labeled and unlabeled) are then extracted from the second to last layer in the ResNet, and passed to the active learning model to select the next best points to annotate.

Active Learning Model

In order to create the best combination of labelled nuclei, we use the active learning method Coreset[22] to suggest which patch the expert should label next. Active learning methods operate by considering a labeled data pool and an unlabeled data pool, and determining the next best data points from the unlabeled pool to label next. They are optimized to find data points that are highly diverse, and will increase both the accuracy and generalizability of the resultant model. Coreset is considered state of the art for image data.

Efficiency Experiments

During the efficiency experiment, we first ask the annotator to find a patch in the slide that contains roughly 200 objects that are equally distributed over the classes that are to be labelled. When such a patch is found, a timer is started and the annotator begins labeling all cells. The system's classifier begins to train once 20 cells have been annotated, and continues to update itself regularly. The system renders predictions to the annotator, which they can confirm or deny, until all cells have annotated to their satisfaction. Once completed, the number of annotated cells per second is calculated. An equivalent experiment, without the AI predictions, is also run as a control. The ratio in cells per second between the two defines the efficiency boost of the system.

Effectiveness Experiments

At the start of an experiment, the annotator is asked to find and annotate 5 objects of each class, to begin the training of the classifier, and initialize the active learner's next-patch suggestion. The annotators then follow the guidance of the active learner until they have annotated 200 total cells. As a control, the experiment is repeated, but with the models disabled.

Due to the lack of ground truth labels for WSI's used in the experiments, labeled data across annotators is used as ground truth for evaluation for an experiment. That is, separate classification models (ResNet-18) are trained on a single annotator's dataset, and evaluated on

all other annotators' datasets. More precisely, if there are N annotators, each with two runs (with and without AI augmentation), then the evaluation of a single run from a single annotator is achieved by training a model on that run's data, and testing it on the 2*(N-1) runs completed by all other annotators.This ensures that the evaluation set is the same for both experiments performed by a single annotator, allowing us to fairly compare model improvements with and without AI-augmented data annotation.

The order in which samples are labelled is preserved during experiments and utilized in generating effectiveness curves (e.g. Fig 5). In such curves, if N training samples are used to plot a classifier's accuracy, those are the *first* N samples to be annotated by the user. We sweep N from 0 to 200 in all curves. 200 is chosen empirically as classifier performance plateaus around this number, in these use-cases. To account for performance fluctuations (particularly for low values of N), all models are trained 10 times and the average value is reported.

Experimental Image Data

For the tumor infiltrating lymphocytes and tumor cell use-cases we used slide TCGA-XF-AAN8-01Z-00-DX1 [Footnote [1]] from The Cancer Genome Atlas[30] (TCGA). For the Eosinophil use-case, we used slide TCGA-XP-A8T7-01Z-00-DX1 [Footnote [2]] from TCGA. For the KI67 use-case we use a slide provided by the University of California at Davis. Not all slides are scanned at the same resolution. Extracting patches around cells of a predefined size in pixels will result in cells appearing scaled. To mitigate this, the patch dimensions are changed such that the rescaled patch always covers the same area.

[1] http://quip1.bmi.stonybrook.edu/camicroscope/osdCamicroscope.php?tissueId=TCGA-XF-AAN8-01Z-00-DX1

[2] https://portal.gdc.cancer.gov/image-viewer?filters=%7B%22op%22%3A%22and%22%2C%22content%22%3A%5B%7B%22op%22%3A%22in%22%2C%22content%22%3A%7B%22field%22%3A%22cases.case_id%22%2C%22value%22%3A%5B%2259dd907e-c674-46c2-bce7-63517d5ae7a7%22%5D%7D%7D%5D%7D&selectedId=3dcaee06-a9ee-410e-b409-b3e1cbae90cd

Figures

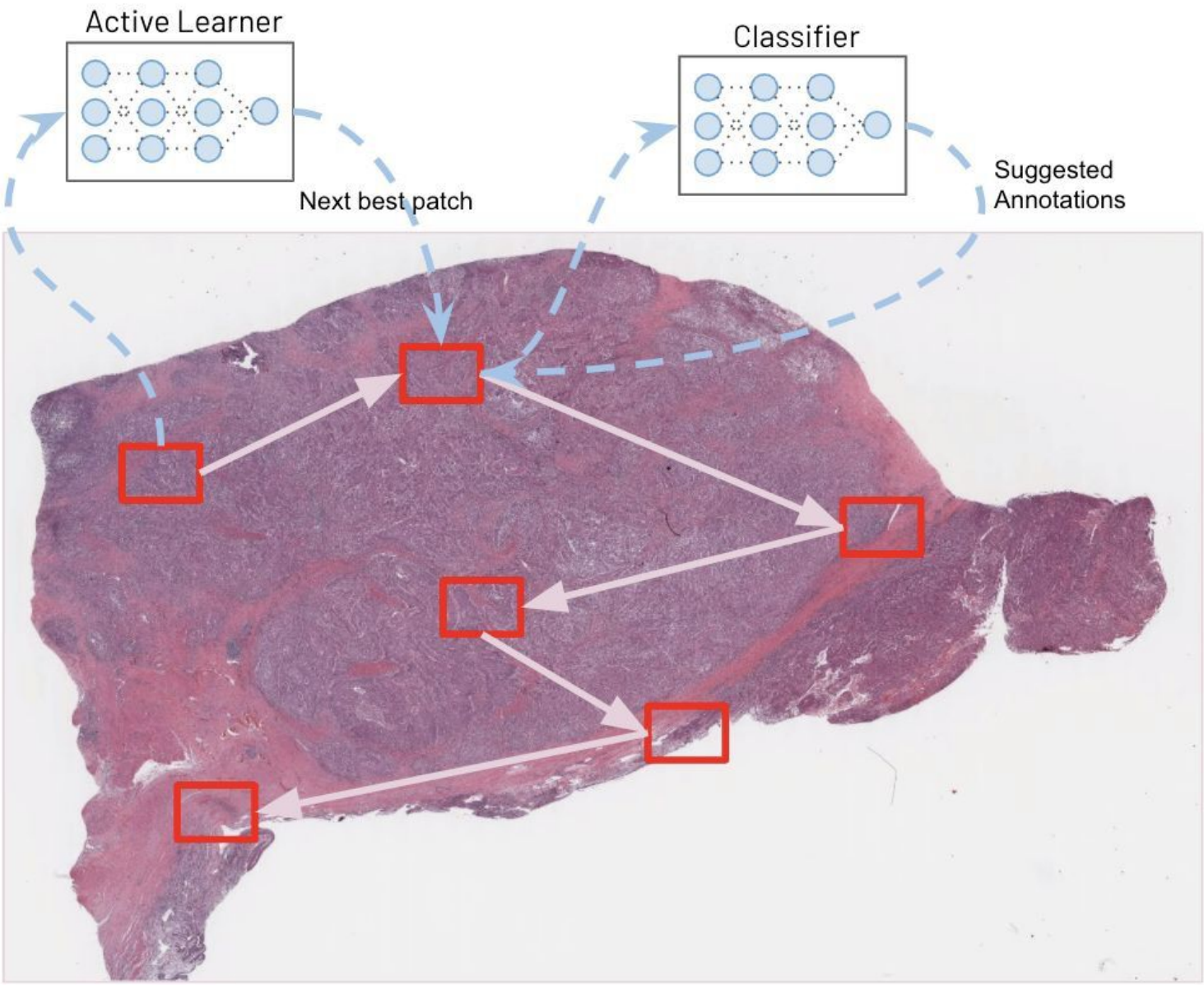


Figure 1

HALI: Human-Augmenting AI-based Labeling Interface. As a human annotator labels data, an active learning algorithm shuttles the annotator around the image by identifying the next best visual features to annotate. Simultaneously, other AIs make labeling suggestions designed to significantly accelerate annotation speed (not pictured -see Fig 3. System Architecture). Together, they give annotators the ability to train personalized AI models, enabling them to generate high-quality labeled datasets for otherwise intractably large images.

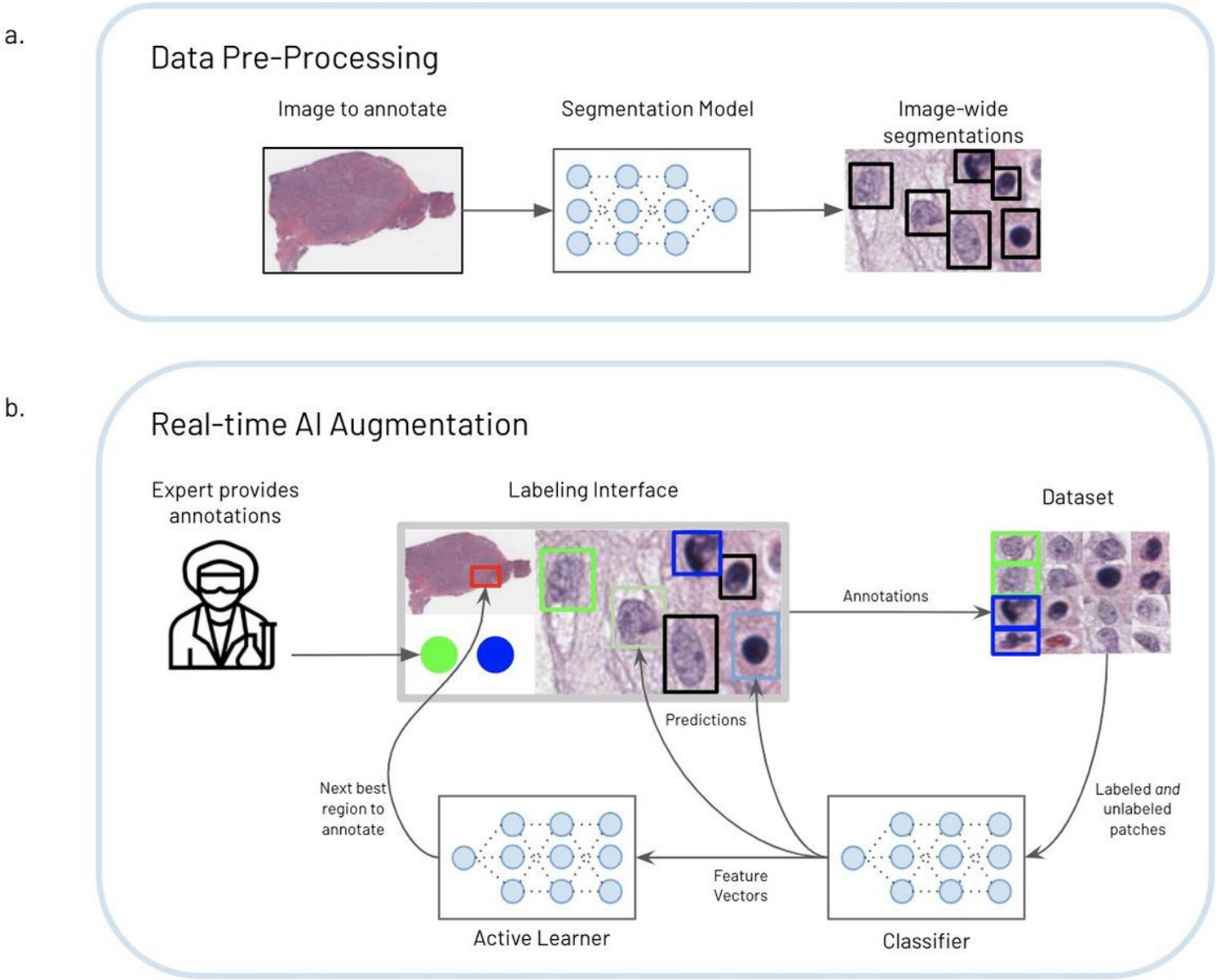


Figure 2

System Architecture. (a) Data Pre-Processing . Digital images are first pre-processed by passing through a deep learning model (HoverNet 9) which segments and generates bounding boxes for each cell. The image and bounding boxes are then used in real-time via a labeling interface outfitted with two AI models that serve to augment and accelerate expert labeling (b). Real-time AI Augmentation . As annotators provide cellular labels (bright green/blue boxes denote different classes), they are stored in a dataset and iteratively used to fine-tune a ResNet classification 21 model, pre-trained on the PanNuke cellular dataset, but initially untrained on the task at hand. As the classifier learns the annotations, it renders ever more accurate predictions (pastel green/blue boxes) to the annotator. Simultaneously, it feeds in high-dimensional feature vector representations of the labeled and unlabeled dataset to the active learner, which determines the next best patch that an annotator should label. Together they increase the speed of annotation, and the quality of the labeled dataset.

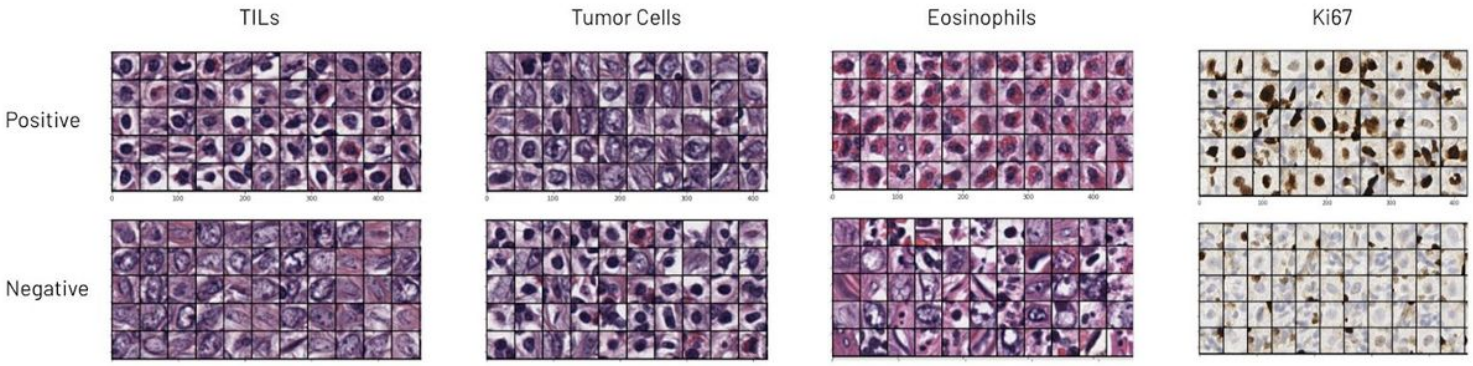


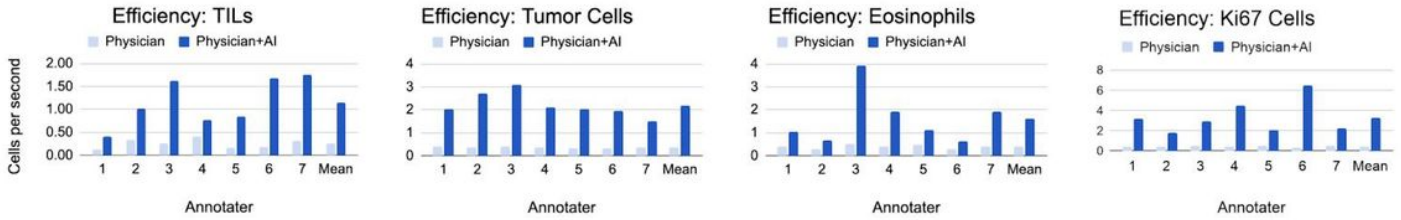
Figure 3

Experimental Use-cases. The task of cellular annotation is chosen, specifically due to its highly repetitive nature, and the difficulty of complete data annotation (e.g. a section of tissue may contain 900,000 cells). Four use-cases highlight the generalizability of this method across stain types and cell types. From left to right: tumor-infiltrating lymphocytes (TILs), Tumor Cells, Eosinophils, and Ki-67-stained cells of arbitrarily sufficient size, as determined by the annotators. Top row shows example cells of the specified type (positive), bottom row shows examples of all others (negative).

a.

Use-case	TILs	Tumor cells	Eosinophils	KI67 Cells	**Average**
Efficiency Increase	4.46x	6.16x	3.83x	8.03x	**5.15x**
Effectiveness Increase	6.33%	6.43%	3.21%	1.38%	**4.34%**

b.



c.

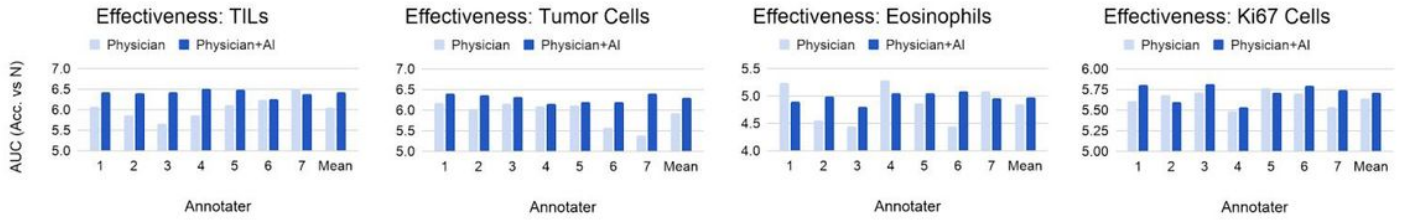


Figure 4

Experimental Results. a. The average efficiency and effectiveness increase for the four use-cases considered (TILs, Tumor Cells, Eosinophils, and Ki-67 cells), altogether averaging a 5.15x efficiency improvement and 4.34% effectiveness improvement. Efficiency is measured by the number of cells per second that an annotator can label. Effectiveness is measured as the area under the Accuracy vs Number of Samples (N) plot, bounded by N < 200, for an AI model trained on the resultant annotated dataset (see

example in Fig. 5). b. Efficiency results on the seven tested annotators - top histopathology trainees from Stanford and the University of California, San Francisco. C. Effectiveness results on the same set of trainees.

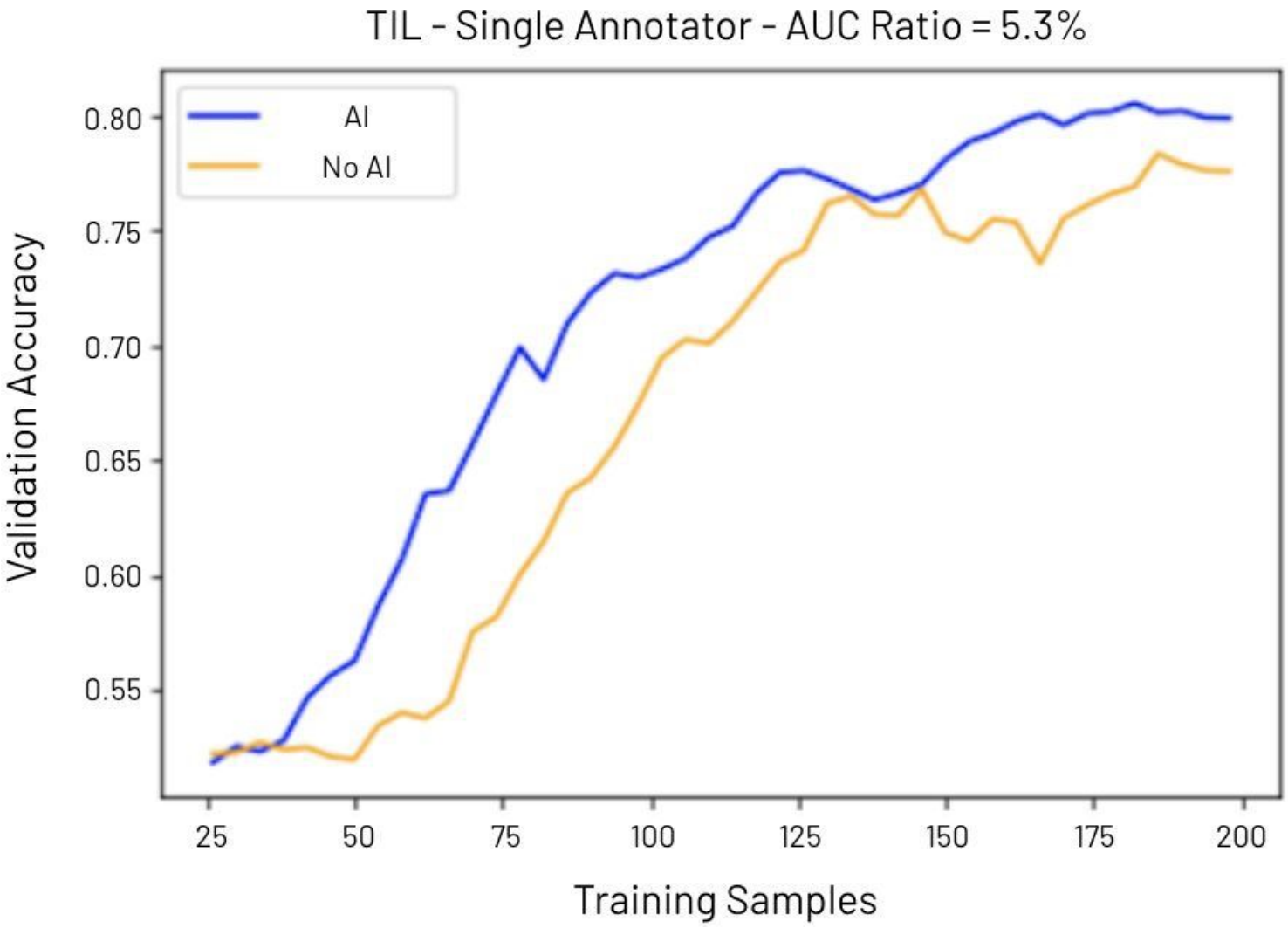


Figure 5

Effectiveness Metric. We consider a dataset, D, to be more effective than another, E, if a model trained on D has a higher validation accuracy than a model trained on E, and both D and E cost the same to annotate (i.e. both have the same number of data points and annotations). To this end, we use the AUC of a model's validation accuracy vs number of training samples as a measure of effectiveness, with a higher AUC indicating higher effectiveness, and the ratio in AUC between two curves allowing a comparison between two models. In the example above (use-case: tumor infiltrating lymphocytes), the dataset generated with AI augmentation has an absolute validation accuracy improvement of 0.11, 0.11, and 0.05, over a dataset generated without AI augmentation, for 50, 75, and 100 training samples, respectively. The AUC ratio of the two curves is 5.3%.

Supplementary Files

This is a list of supplementary files associated with this preprint. Click to download.

- NatureMethodsHALIOnboarding.pdf