Machine Learning Framework incorporating Expert Knowledge in Tissue Image Annotation

Florian Kromp*†, Inge Ambros*, Tamara Weiss*, Dominik Bogen*, Helena Dodig*, Maria Berneder*,
Teresa Gerber*, Sabine Taschner-Mandl*, Peter Ambros*‡ and Allan Hanbury†‡
*Children's Cancer Research Institute, Tumor Biology Lab
Zimmermannplatz 10, 1090 Vienna, Austria, Email: florian.kromp@ccri.at
†Vienna University of Technology, Institute of Software Technology and Interactive Systems
‡: Shared senior authors

***Abstract*—The annotation of cellular nuclei in images of tissue sections is a time consuming but crucial task in quantitative microscopy. We present a machine learning framework incorporating expert knowledge enabling biologists to annotate a large number of nuclear images in a reasonable time. The proposed system is designed to generate three successive levels of annotation, each presenting more details until single nuclei are annotated. Moreover, the output of each level is used to update the model of the next level, increasing the performance speed of the system. A crucial task is the separation of aggregated nuclei. This task is modeled as an Integer Linear Program (ILP), based on the output of an ensemble segmentation and solved by using a genetic algorithm. To incorporate user input at runtime, we use a modified version of an Online Random Forest (ORF). The proposed system was tested by biologists annotating images of ganglioneuroma tissue sections. Results show that the time biologists need to annotate an image is considerably reduced after the system has been trained.**

***Index Terms*—machine learning, online training, image annotation**

I. Introduction

In quantitative microscopy, a precise segmentation of nuclear images is crucial to generate reliable quantification results [1]. Automated or semi-automated segmentation of fluorescence images presenting cellular nuclei of grown or cytospinned cells is a well-studied topic. However, the segmentation of nuclear images of tissue sections remains challenging. This is due to the fact that nuclei present in tissue sections are of different cellular types, different morphology and overlap with other nuclei. Moreover, a section is a 2-dimensional representation of a 3-dimensional specimen resulting in nuclei presenting irregular shapes, varying morphology and differing size. In case of fluorescence imaging, the nuclei present blurred borders and partially weak signals due to damaged nuclei or out-of-focus light. All of these challenges enhance the complexity of automated or semi-automated image segmentation.

In a machine learning setting, features of manually segmented nuclei (i.e. texture, morphology) serve as training data used by automated systems designed to segment new images. The performance of these systems depends heavily on the accuracy of the manual segmentations, called annotations. Usually, a large number of training data is needed to achieve a precise automated segmentation of new images, especially in the context of tissue cytometry [2].

Due to the aforementiond challenges, the process of annotation is a time-consuming task in tissue imaging. To establish a reference time for a fully manual annotation, two experienced biologists, referred to as expert users, annotated a nuclear image of a ganglioneuroma tissue section by drawing polygons along the contour of each nucleus manually. The time needed to annotate the image was 105 minutes. This has to be reduced to at least 45 minutes due to personnel costs and lack of time to allow for the annotation of a large number of images used as input for machine learning.

Automated or semi-automated segmentation algorithms can support the annotation of nuclear images. A good overview of nuclear segmentation algorithms is presented by Dima et al. [3]. Many different modifications of the watershed algorithm [4] are used, based on a distance transform, the gradient of the input image and/or foreground/background markers. The watershed algorithm tends to oversegment images, thus, strategies correcting the output image were developed [5]. A nuclear segmentation algorithm based on finding edges to reconstruct nuclei if at least three nuclear borders were found was proposed by Arslan et al. [6]. Laurain et al. [7] proposed a method for detecting seed points in images of tissue sections, one for each nucleus, and apply a region growing algorithm subsequently.

Most of the segmentation algorithms are designed to work on a certain type of cell and/or of image captured in specific modalities. Depending on the setting, these algorithms perform well in segmenting nuclear images of certain specimens they were designed for. When applying these algorithms to nuclear images of different specimens, they may perform less optimally. In the case of nuclear images of tissue sections, nuclei of multiple types of cell are present appearing with all aforementioned kinds of morphology. Therefore, applying existing segmentation algorithms to images of tissue sections will perform well for certain types of nuclei present, but will fail for others. By applying an ensemble segmentation consisting of existing segmentation algorithms performing well for certain types of nuclei each, all different types of nuclei could be segmented.

The aim of this work was to create a machine learning framework supporting the expert user in annotation of multiple

nuclear images of tissue sections in a reasonable amount of time.

We propose a novel framework by splitting the task of nuclear image annotation into three levels, each guided by user interaction. Annotated objects are used to update classification models that predict classes of new objects, allowing to increase the performance speed of the overall framework. We use ensemble segmentations [8] composed of existing segmentation algorithms to segment the nuclear images within two steps of our workflow. To split aggregated nuclei, we formulate an Integer Linear Program (ILP) which is solved by using a genetic algorithm. To increase the performance speed of our system at run time, we use an Online Random Forest (ORF) [9] to update the models used online. Moreover, we modified the ORF to calculate confidence intervals reflecting the reliability of a prediction, used as an additional input to the genetic algorithm.

II. Machine learning framework

We propose a workflow for the annotation of nuclear images of tissue sections. The workflow is divided into three subsequently executed steps we call levels of annotation. Many nuclear segmentation algorithms are designed using some or all of these three levels while processing an image, thus, existing algorithms can be easily adapted to our framework. The complete workflow is presented in Figure 1.

The three levels of annotation are:

- Level one: Creation of a segmentation mask $S_1(x, y)$ of a raw nuclear image $I(x, y)$ separating foreground from background. $S_1(x, y)$ is a binary image where pixels holding a greyvalue of one represent foreground information in $I(x, y)$.
- Level two: Creation of labeled segmentation masks $S_{2S}(x, y)$, $S_{2C}(x, y)$ and $S_{2N}(x, y)$ based on $S_1(x, y)$. $S_{2S}(x, y)$ contains all objects classified as single nuclei. $S_{2C}(x, y)$ contains all objects classified as clumps (agglomerates of nuclei or single nuclei agglomerated with additional background pixels), and $S_{2C}(x, y)$ contains all objects classified as not-analyzable (i.e. objects made up of out-of-focus light, damaged nuclei).
- Level three: Creation of a segmentation mask $S_3(x, y)$ containing all distinct single nuclei present in $I(x, y)$. The objects in $S_{2C}(x, y)$ classified as clumps in level two have to be separated to obtain the single nuclei. The process will be referred to as 'clump split'. The final segmentation mask $S_3(x, y)$ contains the objects from all clump splits of level three and the objects of $S_{2S}(x, y)$, and thus, represent all distinct nuclei of image $I(x, y)$.

The system supports the expert user in creating multiple segmentation masks each representing different information of $I(x, y)$ until all distinct nuclei are annotated. The different levels of annotation are illustrated in Figure 2.

Each level of annotation is generated based on user interaction. All interactions are logged in a database for further processing and performance evaluation. Within level two and three of annotation, user interactions are processed and used to update

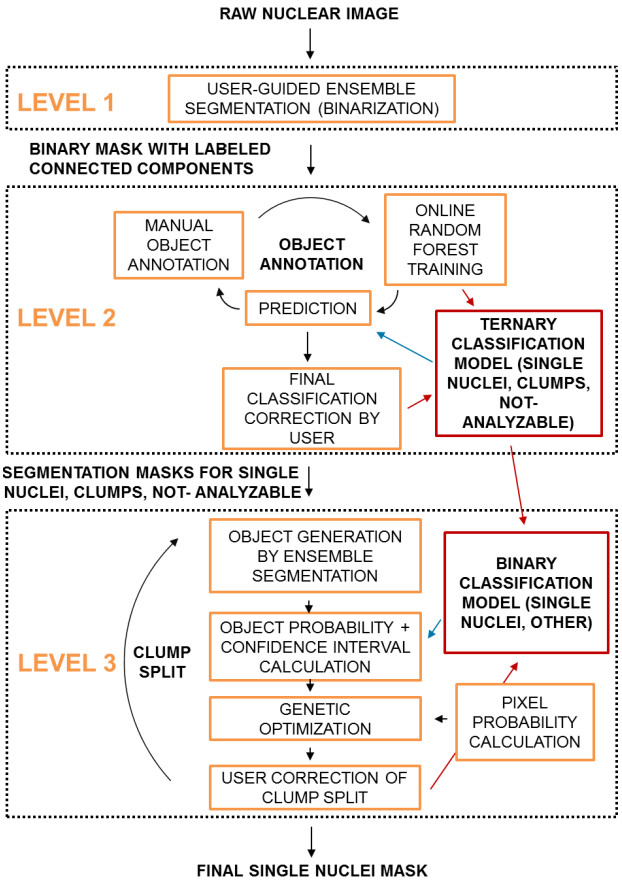

Fig. 1: Workflow of the proposed machine learning framework. Red arrows indicate user input is used to update a model, blue arrows indicate the model is used to predict object probabilities.

classification models at run time. After the annotation of level two and three have been finished, the trained models are stored and reloaded within annotation of those levels of a new image, increasing the performance speed of the framework.

A. Level one annotation

The first level addresses the separation of foreground and background regions in $I(x, y)$ to obtain $S_1(x, y)$. This is a challenging task since the nuclei in the raw image appear with blurred borders and light from different focus layers. If too much background is classified as foreground, spatially close objects tend to agglomerate to larger objects, complicating the clump splitting in level three. To overcome this challenge, we propose an ensemble segmentation to create a temporary image $S_{1T}(x, y)$. The ensemble consists of four binarization algorithms with two different parametrizations each to generate binary images:

- A kernel based graph cuts [10] with a smoothness term $\alpha = 0.001$ and $\alpha = 0.00001$ resulting in $S_{1T1}(x, y)$ and $S_{1T2}(x, y)$.
- Otsu global threshold [11] with an offset of 0 and -10 resulting in $S_{1T3}(x, y)$ and $S_{1T4}(x, y)$.

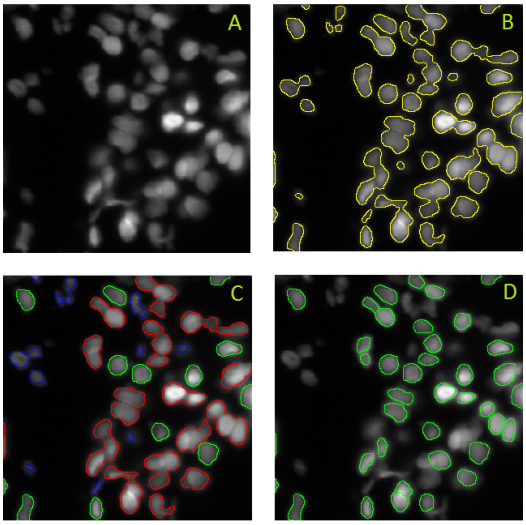

Fig. 2: Segmentation masks of the three levels of annotation presented as overlays of their contour to the raw image. (A) Cropped raw nuclear image. (B) Level one segmentation mask representing foreground region. (C) Level two segmentation mask: single nuclei (green), clumps (red) and not-analyzable objects (blue). (D) Level three segmentation mask: single nuclei (green).

- Hysteresis threshold with an offset of ± 0.1 and ± 0.2 resulting in $S_{1T5}(x, y)$ and $S_{1T6}(x, y)$.
- Adaptive threshold with a window size of 20 and 40 resulting in $S_{1T7}(x, y)$ and $S_{1T8}(x, y)$.

Based on the obtained binary images, we create a temporary image $S_{1T}(x, y) = S_{1T1}(x, y) + S_{1T2}(x, y) + S_{1T3}(x, y) + S_{1T4}(x, y) + S_{1T5}(x, y) + S_{1T6}(x, y) + S_{1T7}(x, y) + S_{1T8}(x, y)$ used for further processing. To obtain a precise representation of the foreground region in $I(x, y)$, the expert user is presented an overlay of the raw nuclear image and the contours of the region of $S_{1T}(x, y) > t$. t can be set by the expert user to values of 1 to 8, see Figure 3 for an example. A threshold of t denotes 'at least t algorithms have to agree on a pixel to be part of the foreground region' to classify pixels as being foreground. The expert user interactively increases or decreases the threshold t to find a threshold that fits the contour of $S_{1T}(x, y) > t$ to the borders of the nuclei, but includes almost all foreground pixels present in the raw image $I(x, y)$. After a threshold has been set, single pixels can be reclassified by using a pen-like tool, resulting in the binary segmentation mask $S_1(x, y)$.

B. Level two annotation

The output of level one annotation, $S_1(x, y)$, contains connected components we call objects. Connected components are foreground pixels connected by a local neighborhood. We use a ternary classification model to classify these objects as being either single nuclei, clumps or not-analyzable objects. Damaged or small nuclei and nuclei with very low mean intensity should be excluded from a quantitative analysis and thus be classified as not-analyzable.

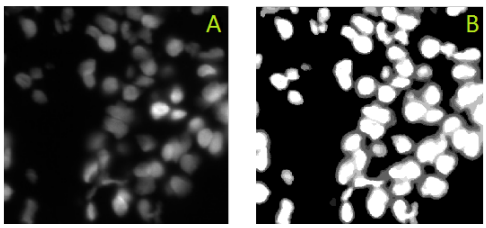

Fig. 3: Probability of pixels being foreground by combining the output of multiple binarization algorithms. (A) Patch of a raw nuclear image. (B) Probability of pixels from A being foreground or background. The grading of grey values corresponds to the number of algorithms agreeing on pixels to be foreground. Black pixels were classified as background by all binarization algorithms, white pixels were classified as foreground by all binarization algorithms.

To annotate all objects resulting from level one annotation, we use an iterative process. The expert user is presented an overlay of the contour of an arbitrary object in $S_1(x, y)$ to the raw nuclear image $I(x, y)$ and classifies the object according to one of the three classes. Morphological and texture features of the currently classified object are calculated and used to update an ORF at run time. After 10 objects have been annotated, we use the ORF to predict class labels of all non-annotated objects. Experiments showed that no meaningful object prediction is possible if less objects were annotated. Moreover, we calculate the difference in classifcation to the former prediction in case a former prediction already exists. The expert user is presented with a graph consisting of the number of objects that changed their class label after the current ORF prediction. The graph reflects the training state of the ORF. If it is not well trained, many class labels change when compared to the former prediction. The expert user continues classifying objects until only few objects change their class label over a certain number of classifications, based on the subjective impression of the expert user. If the expert user decides the overall classification performance to be sufficient, the iterative process is stopped and the expert user can correct the class labels of wrongly predicted objects in a second step. In Figure 4, an example of level two annotation is presented.

C. Level three annotation

To generate $S_3(x, y)$, all clumps in $S_{2C}(x, y)$ resulting from level two annotation have to be split. We call this process clump split. Objects resulting from clump splits are combined with the objects in $S_{2S}(x, y)$ to form $S_3(x, y)$.

We propose an approach to provide a suggestion on how to split clumps. We use $I(x, y)$ and $S_1(x, y)$ as input to an ensemble segmentation. The ensemble consists of four algorithms proven to perform well for the segmentation of nuclear images:

- A watershed algorithm based on a distance transform and the gradient of the raw image [12].

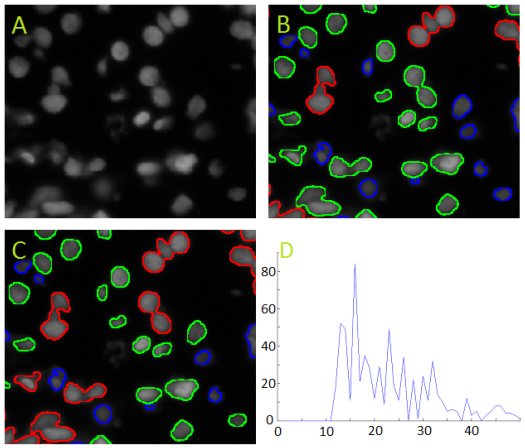

Fig. 4: Example of level two annotation. (A) Patch of a raw nuclear image. (B) Predicted objects after classification of 10 objects. Contours of single nuclei (green), clumps (red) and not-analyzable objects (blue). (C) Predicted objects after classification of 50 objects. (D) Graph showing how many class labels changed with the current input. On the x-axis, the number of annotated objects is shown. On the y-axis, the number of class labels that changed after annotation of an object and reprediction of non-annotated objects is shown. Prediction of class labels is performed for the first time after 10 objects have been annotated.

- A marker-controlled watershed algorithm [13].
- Attributed Relational Graphs [6].
- Attributed Relational Graphs based on a sigma filtered raw image [14].

The ensemble creates segmentation masks $S_{S1}(x,y)$, $S_{S2}(x,y)$, $S_{S3}(x,y)$ and $S_{S4}(x,y)$ that contain distinct, labeled objects. In Figure 5 B-E, segmentation masks of the clump present in image 5 A are shown. For each clump in $S_{2C}(x,y)$, we calculate morphological and texture features of all objects in $S_{S1}(x,y)$ to $S_{S4}(x,y)$ that are within the same region as the current clump in $S_{2C}(x,y)$. Now we want to find a subset of all extracted objects that possibly split the clump. We use a vector x with length equal to the number of objects extracted to represent a possible combination of these objects. If $x(i)$ corresponding to object i is present in a combination, $x(i)$ is set to 1, otherwise it is set to 0. The aim is to to find a combination of objects that is appropriate at splitting a clump. This problem is an Integer Linear Problem (ILP) which is np-hard in the worst case.

To solve this ILP in a reasonable amount of time, we use a genetic algorithm. To rate possible combinations of objects attempting to split a selected clump, we use an ORF to calculate the probability of all extracted objects to be single nuclei. If the expert user annotates level three of an image for the first time, we use the training set from the ternary model in level two of annotation to train the ORF initially. Since we only want to predict the probability of objects being single nuclei, we use a binary classification model in level three of annotation and combine clumps and not-analyzable objects from the training set of level two of annotation into one class. Note that single nuclei appear with a different morphology than nuclei involved in a clump since the shape and size of the object is influenced by a spatially close presence of other nuclei. Thus, the ORF trained on the training set of level two annotation will perform badly in predicting the class labels of single nuclei at the beginning but will increase performance after user interactions have been done and the model has been updated.

In addition to the probability of all objects that possibly split a clump, we use $S_{2S}(x,y)$ and $I(x,y)$ to calculate an offline Random Forest classifier designed to classify pixels of $I(x,y)$. For a subset of pixels of $I(x,y)$, we extract morphological features and train the Random Forest using the features and the corresponding class labels from $S_{2S}(x,y)$. Note that we use single nuclei objects as one class and the background pixels as well as the pixels of not-analyzable objects in $S_{2N}(x,y)$ as the other class. Subsequently, we use the classifier to predict the probabilities for all pixels in $I(x,y)$ to be pixels of single nuclei or not.

The fitness function used by the genetic algorithm to solve the ILP is $y = -1 * (a(x) + b(x) + (1 - c(x)) - d(x) - \alpha e(x) - \beta f(x) - \gamma(1 - g(x)))$. y is negative since we solve a minimization problem. $a(x)$ is the mean object probability of all objects involved in a proposed clump split to be single nuclei, $b(x)$ is the mean pixel probability of all pixels of the area covered by the objects available in the current combination, $c(x)$ is the mean Jackknife confidence interval of object predictions to be single nuclei of all objects available in the current combination, $d(x)$ is the mean probability of all pixels of the area not covered by the objects available in the current combination, $e(x)$ is the amount of overlap between involved objects, $f(x)$ is the amount of freespace not covered by objects, and $g(x)$ is the minimal object probability of all objects involved in the current combination to be single nuclei. α, β and γ are weights that were set empirically. In Figure 5 F, a clump split proposed by the system at an early stage of training is presented.

The clump split as proposed by the genetic optimization is screened and modified by the expert user, if necessary. This can be done by removing objects of the proposed clump split or by adding objects from the segmentation masks $S_{S1}(x,y)$ to $S_{S4}(x,y)$. If there are nuclei in a clump that are not represented by any of the objects in $S_{S1}(x,y)$ to $S_{S4}(x,y)$, the expert user has to draw a polygon along the contour of these nuclei manually. The area enclosed by the contour forms a new object.

After the expert user has finished the clump split modification, the selected clump consists of objects representing distinct nuclei. These objects are of class single nuclei. If objects proposed by the genetic optimization were removed but not replaced, they are not single nuclei or not useful for quantification, and thus, of class other. If the removed objects overlap with a chosen object in the completed clump split by more than 30 percent, we do not regard them as single

nuclei as they do not fit the contour of a nucleus properly. These objects are of class other as well. We calculate the features of all objects previously described to update the ORF. Subsequently, the single nuclei objects are added to the segmentation mask $S_3(x, y)$ and the next clump split is optimized, until all clumps of $S_{S2C}(x, y)$ have been processed. Finally, all objects of $S_{2S}(x, y)$ are added to $S_3(x, y)$.

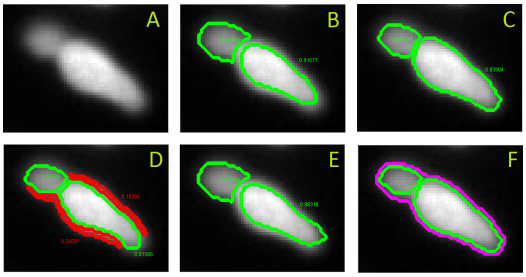

Fig. 5: Level three clump split. (A) Patch of a raw nuclear image. (B)-(E) Objects in segmentation masks: green objects have a probability of higher than 0.5 to be single nuclei, red objects have a probability of less than 0.5. (F) Clump split of the clump indicated by the purple contour as proposed by the genetic algorithm.

III. Incorporation of Expert Knowledge using a modified Online Random Forest

The proposed machine learning framework incorporates expert knowledge in two forms:

- We update an ORF in level two and three of annotation by processing user interactions at runtime.
- Within the ORF classifier in form of key features.

We modified the ORF proposed by Saffari et al. [15] in two aspects. First, we enabled the specification of certain features, we call key features, that are used in the root node of all decision ftrees of the forest. Key features are features that are essential for the classifcation of every nucleus. We selected key features size, mean intensity and compactness based on discussion with experts. The original implementation of the ORF uses a random subset of all features available in each node. The threshold of each feature used in a node deciding the next node is also chosen randomly, known as extremely randomized forest. This setting led to a slow learning rate for objects having similar texture, which was improved by the proposed modification.

Moreover, we modified the ORF to calculate a Jackknife confidence interval [16]. When using the ORF to predict objects that are not in the training set, we obtain a probability for both classes (nuclei or other) by the ratio of trees voting for the classes. Nevertheless, there is a difference in prediction reliabilty between training the ORF with only a small training set compared to training it with a large training set. We use the Jackknife confidence interval to reflect the certainty of the proposed classification for an object. The results are not consistent for small training data but become stable once the training set is large enough ($>$ 600 objects).

IV. Results

To test the performance of the proposed system, we selected five arbitrarily chosen images of tissue sections of a ganglioneuroma tumor. The nuclei were labeled with the fluorescent DAPI stain, allowing the visualization of nuclei present in a section. The number of single nuclei present in each image was between 250 and 350. Two expert users and two junior users annotated four out of five images. The sequence of images annotated by the users was not specified, thus, the users annotated four images in an self-selected order. The classification models used within the machine learning framework were trained for each of the users independently to prevent an increase in annotation time due to potentially contradictory annotations.

In Figure 6A, the total time each user required for a full annotation of a respective image is given.

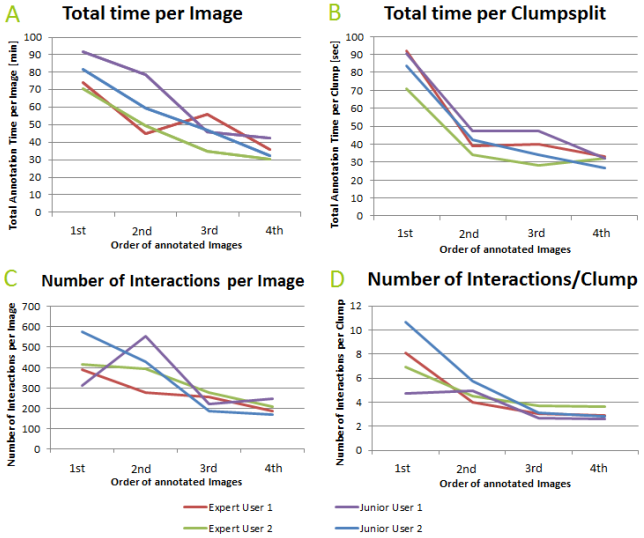

Fig. 6: Performance measures. (A) Total time a user required to use the proposed system. (B) Total time a user required to use the proposed system related to the number of clump splits. (C) The number of interactions taken by the user to annotate an image. (D) The number of interactions taken by the user to annotate an image related to the number of clump splits.

We were able to cut the time users had to sit in front of the computer in halve after three images were annotated. When compared to a former annotation that took 105 minutes for an image, we could remarkably reduce the time needed. The time required to annotate an image mainly depends on the number of clump splits necessary in level three annotation. It also depends on the number and appearance of nuclei aggregated to clumps. If the nuclei present sharp contours, it takes less time to split a clump than if there are overlaid nuclei or nuclei with blurred borders. Nevertheless, we decided to relate the total time to the number of clumps as the most appropriate measure available to rate the performance of the system. The results are presented in Figure 6B. We were able to reduce the time needed to split clumps from 70 to 90 seconds to approximately 30 seconds per clump for the 4th

image. The time is increased if manual polygons have to be drawn, meaning the segmentation algorithms did not propose objects reflecting single nuclei that could be chosen in level three annotation.

Another measure for the performance of the system is the number of interactions a user has to take to annotate an image. In Figure 6C, we show a marked reduction in the number of interactions taken by a user as the system becomes better trained. We could reduce the number of interactions from approx. 450 to 200 in the fourth image annotated. Again, the number of interactions also depends on the number of clumps resulting from level two annotation. The peak in the annotation time of the second image for junior user 1 was due to a high number of clumps from the level two annotation of this image, almost twice as high as the number of clumps resulting from level two annotation of image 1.

To obtain a more objective measure, we relate the number of interactions to the number of clumps resulting from level two annotation. The result is presented in Figure 6D. We were able to reduce the mean number of interactions per clump to three to four clicks after two images were annotated.

To test the increase in performance speed for statistical significance, we performed a repeated measures ANOVA followed by a Bonferroni multiple comparison test using the statistical software GraphPad Prism. Results as outlined in table I show that by using the proposed machine learning framework the time needed to annotate an image is significantly reduced after the first image has been annotated. Moreover, already after two images have been annotated the decrease of the number of interactions, related to the number of clumps, is statistically significant.

	1^{st} vs 2^{nd} Image	1^{st} vs 3^{rd} Image	1^{st} vs 4^{th} Image
Total annotation time / Image	*	**	***
Total annotation time / Clump	***	***	***
Number of interactions / Image	ns	ns	*
Number of interactions / Clump	ns	**	**

TABLE I

STATISTICAL ANALYSIS OF THE PERFORMANCE MEASURES ON A 0.05 SIGNIFICANCE LEVEL. NS: NOT SIGNIFICANT ($p > 0.05$), *: SIGNIFICANT ($p \leq 0.05$), **: SIGNIFICANT ($p \leq 0.01$), ***: SIGNIFICANT ($p \leq 0.001$).

V. CONCLUSION

In this work we showed that the proposed system accelerates the annotation of nuclear images of tissue sections significantly.

Nevertheless, the total time a user takes to annotate an image in an advanced system state of training mainly depends on two aspects: the number of clumps and the appearance of nuclei forming the clumps. If the ensemble segmentation in level three annotation does not result in objects reflecting single nuclei, the optimization on the clump split cannot be precise. Thus, the missing objects have to be drawn manually, which is a time consuming task. A further improvement of the proposed system could be an automated tuning of the segmentation parameters involved.

ACKNOWLEDGMENT

This work was facilitated by an EraSME grant (project TisQuant) of the Austrian Research Promotion Agency (FFG) under the grant no. 844198 and the St. Anna Kinderkrebsforschung. The authors would like to thank Andrea Ziegler and Diana Walder (Children's Cancer Research Institute, Tumor Biology Lab) for carefully preparing and staining the tissue sections.

REFERENCES

[1] J. C. Waters, "Accuracy and precision in quantitative fluorescence microscopy," *The Journal of Cell Biology*, vol. 185, no. 7, pp. 1135–1148, Jun. 2009.

[2] R. C. Ecker and G. E. Steiner, "Microscopy-based multicolor tissue cytometry at the single-cell level," *Cytometry*, vol. 59A, no. 2, pp. 182–190, Jun. 2004.

[3] A. A. Dima, J. T. Elliott, J. J. Filliben, M. Halter, A. Peskin, J. Bernal, M. Kociolek, M. C. Brady, H. C. Tang, and A. L. Plant, "Comparison of segmentation algorithms for fluorescence microscopy images of cells: Comparison of Segmentation Algorithms," *Cytometry Part A*, vol. 79A, no. 7, pp. 545–559, Jul. 2011.

[4] L. Vincent and P. Soille, "Watersheds in digital spaces: an efficient algorithm based on immersion simulations," *IEEE Transactions on Pattern Analysis & Machine Intelligence*, no. 6, pp. 583–598, 1991.

[5] P. U. Adiga and B. B. Chaudhuri, "An efficient method based on watershed and rule-based merging for segmentation of 3-D histo-pathological images," *Pattern Recognition*, vol. 34, no. 7, pp. 1449–1458, 2001.

[6] S. Arslan, T. Ersahin, R. Cetin-Atalay, and C. Gunduz-Demir, "Attributed Relational Graphs for Cell Nucleus Segmentation in Fluorescence Microscopy Images," *IEEE Transactions on Medical Imaging*, vol. 32, no. 6, pp. 1121–1131, Jun. 2013.

[7] V. Laurain, H. Ramoser, C. Nowak, G. E. Steiner, and R. Ecker, "Fast Automatic Segmentation of Nuclei in Microscopy Images of Tissue Sections," in *Engineering in Medicine and Biology Society, 2005. IEEE-EMBS 2005. 27th Annual International Conference of the*. IEEE, 2006, pp. 3367–3370.

[8] A. Alush and J. Goldberger, "Ensemble segmentation using efficient integer linear programming," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 34, no. 10, pp. 1966–1977, Oct 2012.

[9] A. Saffari, C. Leistner, J. Santner, M. Godec, and H. Bischof, "On-line random forests," in *Computer Vision Workshops, 2009 IEEE 12th International Conference on*, Sept 2009, pp. 1393–1400.

[10] M. B. Salah, A. Mitiche, and I. B. Ayed, "Multiregion Image Segmentation by Parametric Kernel Graph Cuts," *IEEE Transactions on Image Processing*, vol. 20, no. 2, pp. 545–557, Feb. 2011.

[11] R. W. Sittler, "An optimal data association problem in surveillance theory," *Military Electronics, IEEE Transactions on*, vol. 8, no. 2, pp. 125–139, 1964.

[12] A. Mouelhi, M. Sayadi, and F. Fnaiech, "Automatic segmentation of clustered breast cancer cells using watershed and concave vertex graph," in *Communications, Computing and Control Applications (CCCA), 2011 International Conference on*. IEEE, 2011, pp. 1–6.

[13] P. K. Geetha, R. Nidhya, A. D. Kumar, and K. T. Selvi, "Cell segmentation and nc ratio analysis for biopsy images using marker controlled watershed algorithm," in *Green Computing Communication and Electrical Engineering (ICGCCEE), 2014 International Conference on*, March 2014, pp. 1–5.

[14] J.-S. Lee, "Digital image smoothing and the sigma filter," *Computer Vision, Graphics, and Image Processing*, vol. 24, no. 2, pp. 255 – 269, 1983.

[15] A. Saffari, C. Leistner, J. Santner, M. Godec, and H. Bischof, "On-line random forests," in *Computer Vision Workshops (ICCV Workshops), 2009 IEEE 12th International Conference on*. IEEE, 2009, pp. 1393–1400.

[16] S. Wager, T. Hastie, and B. Efron, "Confidence intervals for random forests: The jackknife and the infinitesimal jackknife," *The Journal of Machine Learning Research*, vol. 15, no. 1, pp. 1625–1651, 2014.