Applications of deep learning in single-cell analysis

Summary of the PhD Thesis

Réka Hollandi

Supervisor: Péter Horváth, PhD

Department of Biochemistry, Biological Research Centre

Doctoral School of Interdisciplinary Medicine,

Faculty of Medicine, University of Szeged

Szeged

2021

Introduction

Cellular analysis generally relies on accurate detection of cells (typically based on nucleus segmentation). High-content experiments tend to rapidly produce thousands of images of either multi-well plates (cell cultures) or tissue sections (using slide scanners). Automated devices (e.g. appropriate microscopes) are used in such experiments and similarly high-throughput algorithms are also desired to process the large amounts of image data acquired. Deep learning (DL) is an efficient group of methods for tasks often attended to in single-cell analysis such as cell segmentation and classification (i.e. phenotyping); a specific type of DL, convolutional neural networks (CNNs) are ideal for such image information retrieval. Here two tools have been developed applying various DL methods to prepare training data properly fit for novel experiments: one to incorporate expert knowledge with minimal user interaction, the other to automatically adapt via synthetic data generation. A combination of them may also be used.

I. Annotation

DL methods typically require large training data of sufficiently reliable quality – ideally created by field experts which is expensive and time consuming in e.g. cellular analysis – to guarantee potentially high performance. AnnotatorJ is proposed as an ImageJ/Fiji plugin intended for the efficient and fast annotation of (not strictly) microscopy images to easily create training data for various DL applications (see **Fig. 1**). The often tedious and slow process of manually creating expert-curated annotations – which is required to e.g. train new DL models for specific tasks or properly benchmark new DL methods – may be accelerated with this plugin via its integrated pre-trained DL model (a U-Net) which suggests contours from a quickly initialized line over the object on the image (see **Fig. 2**). Alternatively, a classical image processing method (region growing) can also be used for contour suggestion.

Figure 1 shows the annotation and export data formats available in AnnotatorJ such as semantic annotation suitable to train semantic segmentation models (e.g. U-Net) where pixels of the target object class (e.g. nucleus) are marked without object separation, instance annotation for instance segmentation models (e.g. Mask R-CNN, CellPose etc.) where image regions corresponding to individual objects are marked and separated, or bounding boxes for object detection (e.g. YOLO) marking the smallest enclosing rectangle around the objects individually

– these can be exported as coordinates according to the COCO dataset format. Segmentations may be exported as binary, multi-labelled or multi-layered masks (optimal for overlapping objects such as cytoplasms in cell cultures). Class assignment and export are also possible by objects alongside further user-friendly functions for ease of usage.

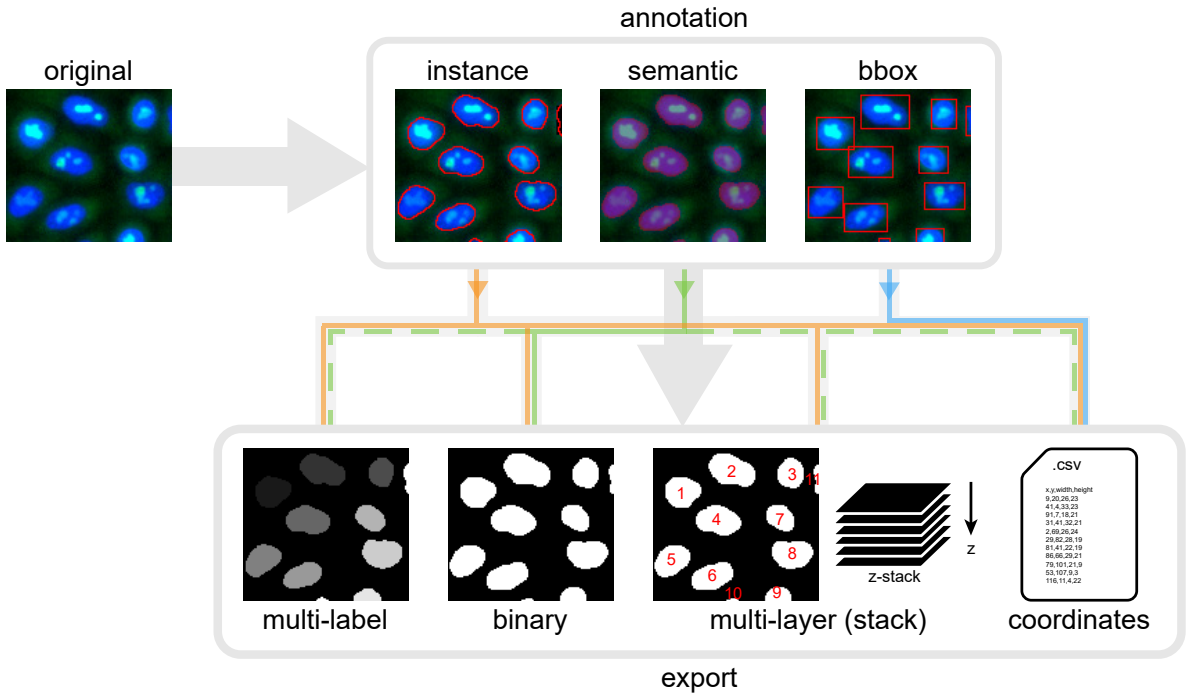

Figure 1. Annotation and export options supported in AnnotatorJ. According to the main data types expected as training data by DL models, users can annotate objects in semantic or instance (object-aware) mode, or mark object locations as bounding boxes (bbox on the figure), shown in red. Supported export formats corresponding to each annotation type are connected with coloured lines; dashed lines show possible export where the user must consider the limitations of the given data format (e.g. adjacent objects on semantic masks are not separated).

Contour assist mode (see on **Fig. 2**) allows users to concentrate on challenging image regions such as clumped or out-of-focus nuclei, potentially on complex background e.g. on histopathology images, by the introduction of contour suggestion using either U-Net or a classical method (region growing) based on a quickly initialized contour. The initial contour is extended to x-by-x pixels in 2D creating a bounding box around the object marked, then the selected suggestion method (explained here with the U-Net option) predicts pixel probabilities of the target object class in this bounding box and a fine approximation of the true object boundaries is returned to the user (as the suggested contour) for manual refinement (minor

correction) if needed to ensure the expert-curated manner of annotation; this is considered as semi-automatic annotation.

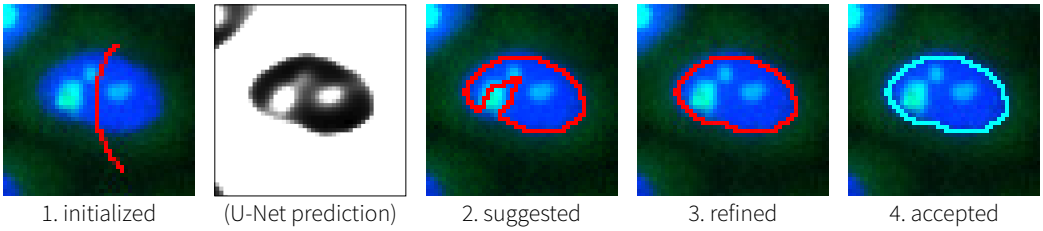

Figure 2. Contour assist mode of AnnotatorJ. Steps show how *Contour assist* mode may be used to create new annotations quickly and conveniently as follows. 1) Contour is roughly initialized by the user (red line drawn), 2) a new contour is suggested automatically using a U-Net prediction around the initial contour, 3) the user (optionally) refines the contour and 4) accepts it. See also the video supplementary in [1].

Efficiency in both time and accuracy was tested with the help of three expert annotators having relevant biological background. They were asked to annotate our four test sets as follows twice: with and without the *Contour assist* function (in the latter case using only the *automatic adding* function of AnnotatorJ). Test set *i* contained heterogeneous microscopy images of nuclei while *ii* of cytoplasms, both displaying fluorescent, brightfield, cell culture, tissue images acquired in different microscope types and quality etc., test set *iii* was adapted from ISBI 2012 containing electron microscopy (EM) recordings of neuronal structures, and finally set *iv* was an arbitrary selection from the Cityscapes dataset of traffic video frames (we focused on the car object class). Test set *iv* (cars) demonstrated applicability in general object annotation tasks (e.g. in industry). Based on the measured annotation times presented on **Figure 3** we concluded that AnnotatorJ indeed increased speed significantly in most cases. Notably, similar improvement could not be achieved (except for one annotator out of three) on test set *iii* (cytoplasm images) since these were the most difficult to annotate due to overlapping objects (in culture) and complex tissue background with barely visible and not obvious separating borders.

These tests were also used for annotation accuracy evaluation compared to ground truth annotations (GT, created by independent experts) where pixel-wise matching of masks was performed on object level, then averaged for all images in a test set. We used the IoU (intersection over union) metric – as for nucleAIzer too (see later) – determining the ratio of the overlap (intersection) area between predicted and ground truth objects and the area of their union. IoU is formulated as a function of the number of true positive (TP), false positive (FP)

and false negative (FN) objects identified: $IoU(t) = \frac{TP(t)}{TP(t)+FP(t)+FN(t)+\varepsilon}$ where t is such a threshold above which an object is considered TP. Our results demonstrated larger inter-expert than intra-expert differences allowing us to conclude that annotations created in AnnotatorJ are of sufficiently high quality to train DL models on.

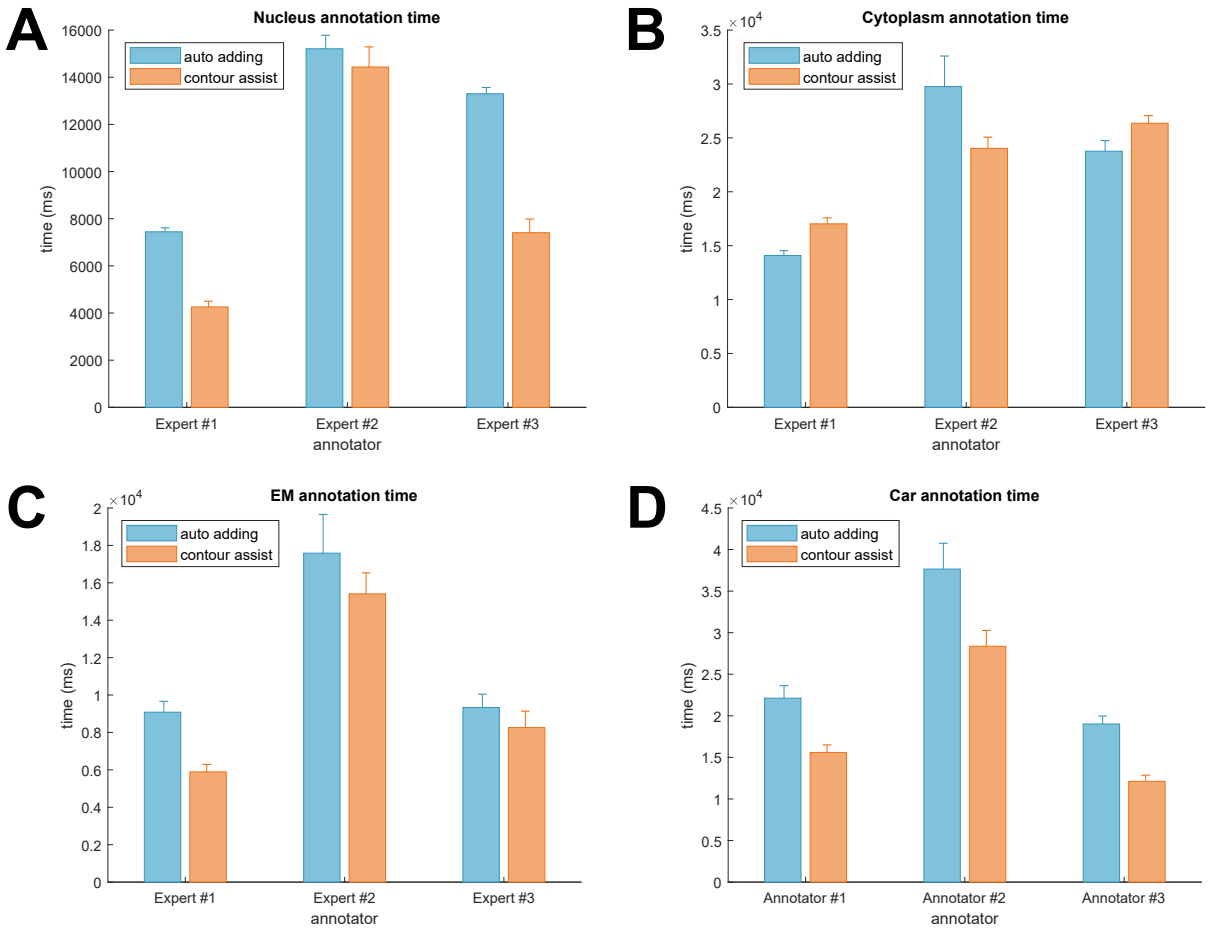

Figure 3. Annotation time efficiency. Experts compared *Contour assist* and *automatic adding* functions of AnnotatorJ on test sets *i-iv* (A-D), annotation times were measured in ms by objects, then averaged for all images in a set. Errors bars show SEM (standard error of the mean).

Furthermore, integration is available to OpSeF (Open Segmentation Framework): a user-friendly interactive platform to test different DL models in processing pipelines typically applied in single-cell analysis. Segmentation with optionally classification information included can be easily imported to AnnotatorJ, refined (both segmentation and classication) then exported in suitable format for OpSeF re-import. Images and masks are handled in simulated z-stacks (to enable 3D-like data processing).

The Author's main contributions to this work are the following:

- Conceived and implemented the AnnotatorJ plugin.
- Designed, executed and evaluated the experiments. Collected datasets and created figures.
- Implemented the OpSeF integration.

II. Image segmentation with deep learning

Even though several methods applying deep learning (DL) for image segmentation tasks have been published since the last decade (some motivated by worldwide challenges), single-cell analysis remains a topic frequently targeted (e.g. Cellpose released in 2020). As most cellular analysis pipelines depend on cell detection as a basis of downstream analysis such as cell phenotyping (classification), counting, molecular studies (e.g. sequencing, proteomic, lipidomic experiments etc.), high quality, reliable methods are required by bioimage analysts, ideally carrying robustness too. For this purpose, an international nucleus segmentation challenge Data Science Bowl (DSB) 2018 was launched inspiring numerous research groups to develop and push state-of-the-art solutions. We started developing nucleAIzer as such, aiming to prepare our method for novel image domains the training set lacked – despite being constructed of heterogeneous sets of images from public databases and various projects of our own laboratory showing distinct labelling, imaging and treatment conditions on different sample types (culture, tissue) with different quality, magnification etc. As an alternative to ground truth annotation (GT, as in e.g. AnnotatorJ) to incorporate the unknown experiment's domain in the training set we apply image style transfer learning to generate new artificial microscopy images nearly indistinguishable from real ones and train our segmentation model (a Mask R-CNN) on an augmented dataset of the synthetic and prior annotated images. Hence, considerable effort may be spared as no new expert-curated annotation is (necessarily) needed to train a model for new image modalities.

The nucleAIzer pipeline is depicted on **Figure 4A**. A various set of test images are firstly predicted with a pre-trained Mask R-CNN model (we refer as *preseg*) robust enough to generalize well on practically any new image domain – however, it is not expected to segment all objects present – thus object size and morphology estimation can be performed on these

segmentations and forwarded to the pipeline. Then, test images are clustered with *k*-means by their visual similarities and image style transfer learning models (pix2pix) are trained on each, separately, which learns how to create a realistic microscopy image from a labelled mask (obtained in the pre-segmentation step or prior annotation); see on **Fig. 4B**. Simultaneously, new mask images are created by fetching individual object (nucleus) masks from a previously collected set (mask database) that follow the morphological properties measured on the pre-segmented masks of test images, then placing them on a new blank mask image according to the spatial localization of objects on original images. Additionally, artificial objects are also simulated (contributing 50% of objects) on these masks with simcep (a cell image simulator software). Once trained, image style transfer learning models are applied on these masks and yield fake microscopy images resembling their clusters' style each (see examples on **Fig. 5**).

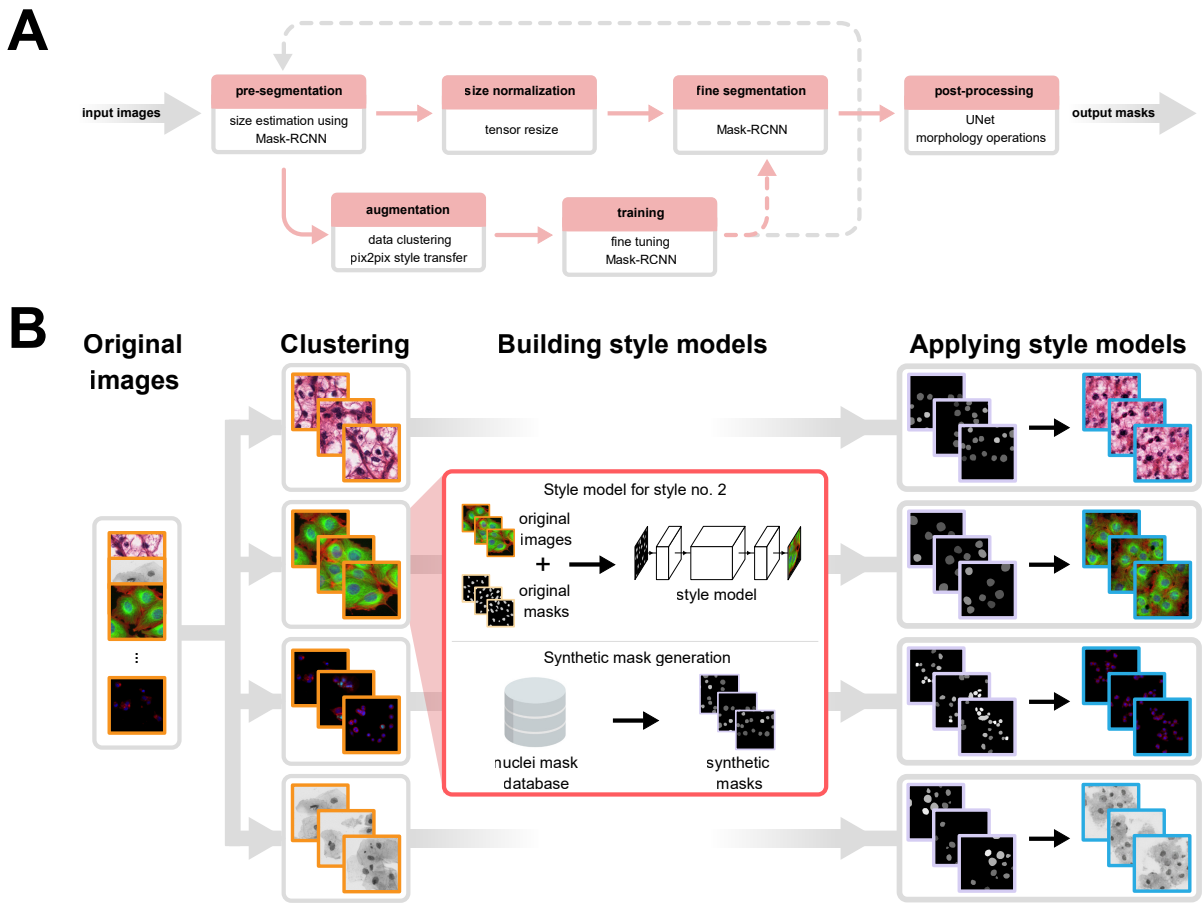

Figure 4. The nucleAIzer pipeline. A) Main processing steps: inference in top row, training in bottom row. Firstly, test images are pre-segmented with a Mask R-CNN model (*preseg*), object size and shape are estimated, then images are clustered by styles and forwarded to image style transfer (see on B) to generate more training data

which is used to train a segmentation model (Mask R-CNN). Images are rescaled according to size estimation and segmented with the refined model, then optionally post-processed (using U-Net and morphology steps) for higher precision. Training may be iterated as the dashed line suggests. B) Image style transfer. Test images are clustered by visual appearance, a separate image style transfer model is trained for each, new artificial masks are created based on the size and shape estimations, then models are applied on these masks to yield realistic artificial images.

The created artificial image-mask pairs are appended to the training set containing prior experiments' (annotated) image data which is augmented using conventional operations (e.g. rotations, flips, crops, resizing, blur, noise, intensity stretch, inversion etc.) and used to train (refine) a segmentation model (Mask R-CNN). Training images are resized to have uniform object sizes for we observed superior performance in this case. Since the pipeline produces masks for image style transfer, any desired number of synthetic image-mask pairs can be generated; this may be utilized to emphasize rare types of images originally underrepresented in the training set. Finally, segmentation masks are predicted by the trained Mask R-CNN model on test images rescaled according to the size estimation, then optionally post-processed using U-Net and classical morphology operations (e.g. dilation, object filtering, merges etc.).

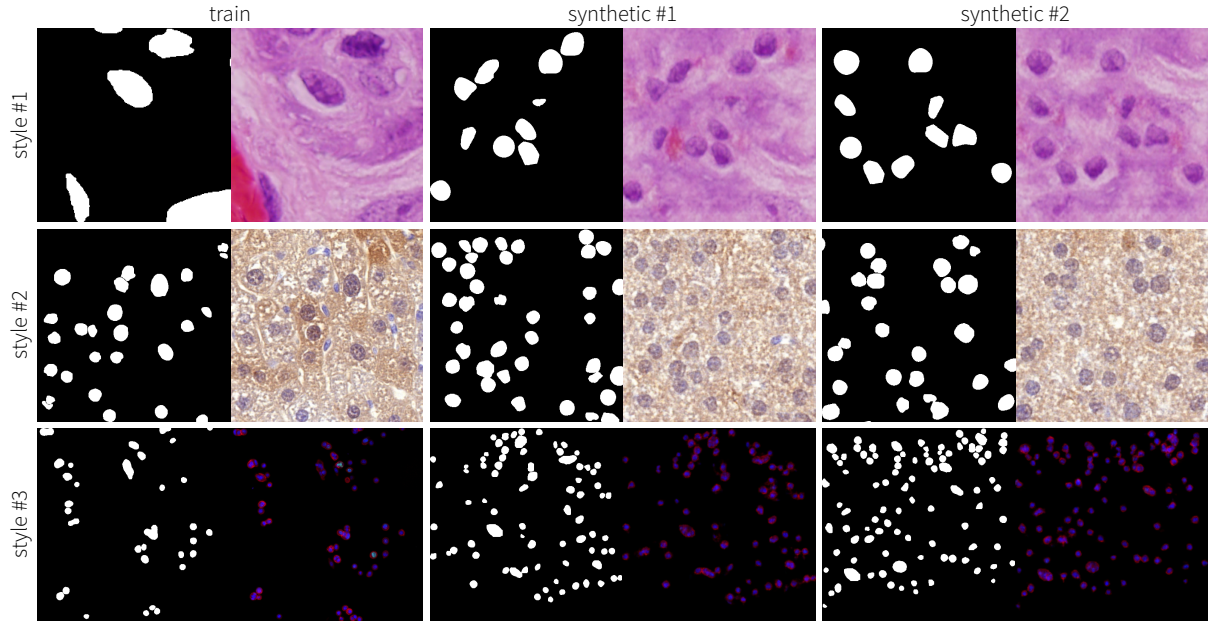

Figure 5. Example image style transfer-generated artificial image-mask pairs. The first column shows training mask-image pairs for each style, columns 2-3 displays two randomly selected synthetic mask-image pairs per style.

We performed proof-of-concept experiments comparing the segmentation accuracy of nucleAIzer to both classical and DL-based methods, and also validated the effectiveness of

images style transfer learning via internal comparison to training without synthetic images (referred as *NOstyle*); see results both quantitatively (A) and qualitatively (B) on **Figure 6**. IoU was our primarily utilized metric according to its definition in the DSB2018 (others were used in the study as well). Evaluation was performed on four test sets as follows. Official test sets of DSB in the first and second stage of the competition comprising of truly heterogeneous images (fluorescent, brightfield, culture, histology, imaging modalities etc.), and additionally our custom sets of florescent cell line images (referred as *fluo*, corresponds to BBBC039) and a set of various histopathology images (*hist*, from public datasets and our own laboratory).

The following methods were compared. Classical methods included 1) CellProfiler (CP): a widely used bioimage analysis software with classical image processing, 2) GVF (gradient vector flow): a sophisticated but still classical algorithm that finds gradients pointing to bright image regions on dark background and 3) ilastik: a pixel classification software using machine learning. DL-based solutions used were 4) unet4nuclei: a U-Net-based approach developed on test set *fluo* (BBBC039), 5) DSB1: first place solution of DSB using a large number of trained U-Nets, 6) DSB2: second place solution of DSB also based on U-Net and additionally 7) Cellpose: an also U-Net-based method using vector flow predictions (in the DVP project, see later). Additionally, we evaluated variations of nucleAIzer using different image style transfer-generated artificial images (*AUTOstyle* (8) with images generated on pre-segmented masks and *GTstyle* (9) on annotated GT masks) or without them (*NOstyle* (9)). Furthermore, our pre-trained models were also tested: 10) *preseg* (used in pre-segmentation) and 11) *postComp* (fine-tuned model on DSB data, corresponding to *AUTOstyle* on DSB stage 2 test). Our tests confirmed that our models yielded the highest scores on all sets when compared to external methods (CP, GVF, ilastik, unet4nuclei) while generally *postComp* performed the best in internal comparisons (*preseg*, *NOstyle*, *AUTOstyle*, *GTstyle*). As expected, GT masks (*GTstyle*) usually improved segmentation performance more than automatically retrieved ones (*AUTOstyle*), nonetheless, they both provided higher accuracy than omitting image style transfer entirely (*NOstyle*). When testing on DSB data, we could compare to 739 methods submitted to stage 2, among them DSB1 and DSB2, achieving the highest score with *postComp*. We note that in challenging image regions (such as clumped cells in culture, out-of-focus regions or occlusions in tissue surroundings) most methods produce segmentation errors

(misses, false extra detections, splits, merges); this is a well-known phenomenon. However, our method helps reduce such errors also prominently compared to other solutions.

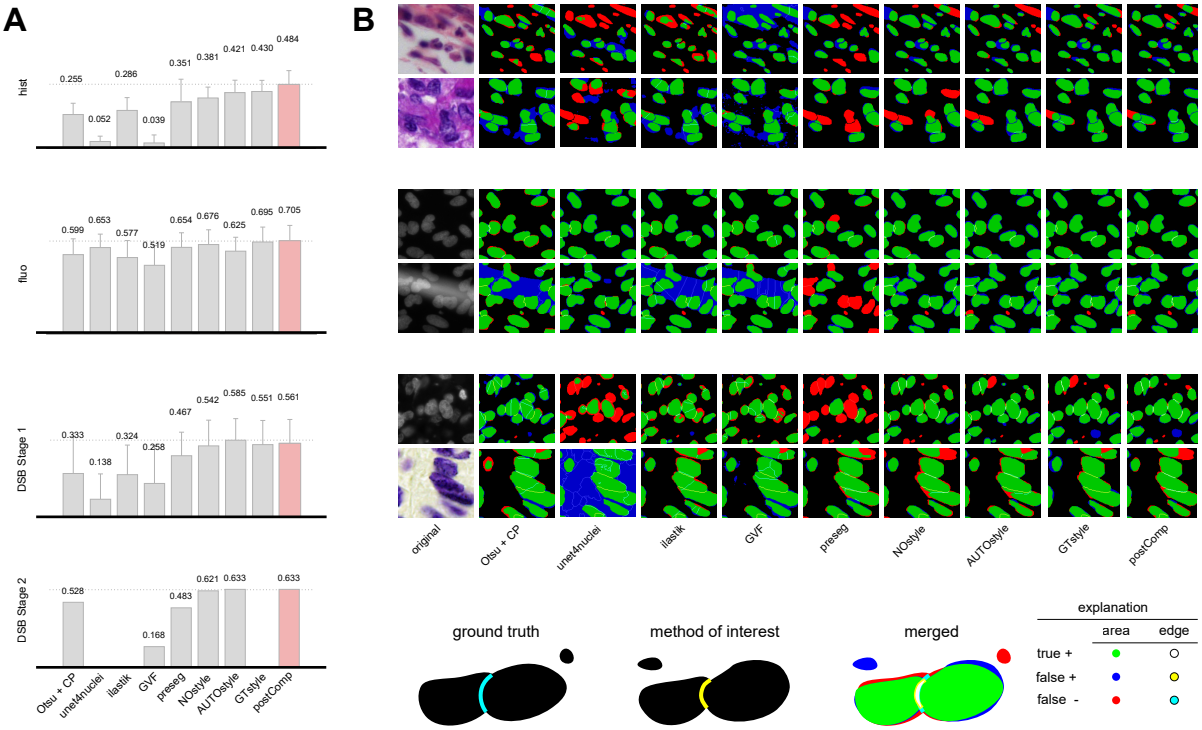

Figure 6. Accuracy evaluation of nucleAIzer and compared methods. The full nucleAIzer pipeline optimized to the DSB 2018 (denoted as *postComp*) was compared to its variants (*AUTOstyle*, *GTstyle*, *NOstyle* and *preseg*), as well as to classical methods (CP, GVF, ilastik) and a DL method unet4nuclei. A) Mean IoU scores with error bars (standard deviation) over the test sets, highest scores are shown in pink, B) visual representation of segmentation performance in a colour-coded way according to the legend (bottom), corresponding to test sets in A by two rows.

A quality control experiment was conducted as well to prove the high quality of synthetically generated microscopy images; field experts were asked to identify real image tiles against fake ones. We found an average performance of 57% close to random selection (50% in binary classification) suggesting applicability in real experiments.

An online tool is supplemented to nucleAIzer for quick and easy testing on custom experimental images; among others the general nucleus segmentation model (*postComp*) is also available.

After we had developed and validated nucleAIzer, we found another technique useful in its segmentation accuracy boosting on the DSB test set, namely test-time augmentation (TTA) i.e. to predict segmentation on augmented versions of test images then ensemble predictions.

Applications nucleAIzer has been applied in several real-life projects including our international collaborations. Our genome-wide screen (GWS, unpublished) collaboration project with the Kutay group at ETHZ (Switzerland) to study ribosome biogenesis via siRNAs motivated us to extend nucleAIzer for further cellular compartment segmentation tasks, in this case for the nucleoli (small spot-like structures inside the nucleus), thus we trained a separate model for nucleolus segmentation and applied it in the project. Alongside the general nucleus segmentation model (*postComp*), segmentations far more accurate and reliable were obtained and utilized in downstream analysis (e.g. phenotyping); see examples on **Figure 7**.

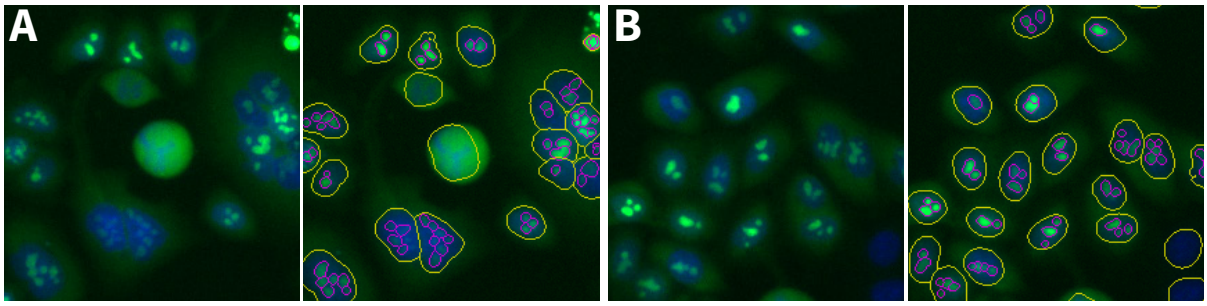

Figure 7. Nucleoli and nucleus segmentation in the GWS project using nucleAIzer. Nuclei are outlined in yellow, nucleoli in magenta. A-B) Different examples (e.g. mitotic on A).

Furthermore, we encountered the need to also segment the cytoplasm of cells more accurately than previously (using e.g. CellProfiler or Fiji) in multiple collaboration projects. As an example, in the Deep Visual Proteomics (DVP) project together with the Mann group (University of Copenhagen, Denmark and Max Planck Institute, Germany) and Lundberg group (Karolinska Institute, Sweden) nucleAIzer was used to train cytoplasm segmentation models (also for the nucleus). Precision achieved by these models was benchmarked against Cellpose (DL method) alongside CP and unet4nuclei (also DL) and was found superior on all sample types in the experiment: fluorescent U2OS cell culture, IHC-marked melanoma and salivary gland tissues; see **Fig. 8**. This increased segmentation accuracy allowed precise single-cell isolation and proteomic analysis (mass spectrometry) in the project. We also successfully applied nucleAIzer for nucleus segmentation in the Neuropilin-1 (NRP1) project which identified NRP1 as a target of SARS-COV-2 – in this project we used a U-Net-based approach for large, fused cell (*syncytia*) cytoplasm segmentation. Additionally, several further projects (whose results are unpublished as of writing this thesis) utilize nucleAIzer for cellular compartment segmentation and AnnotatorJ for annotation purposes.

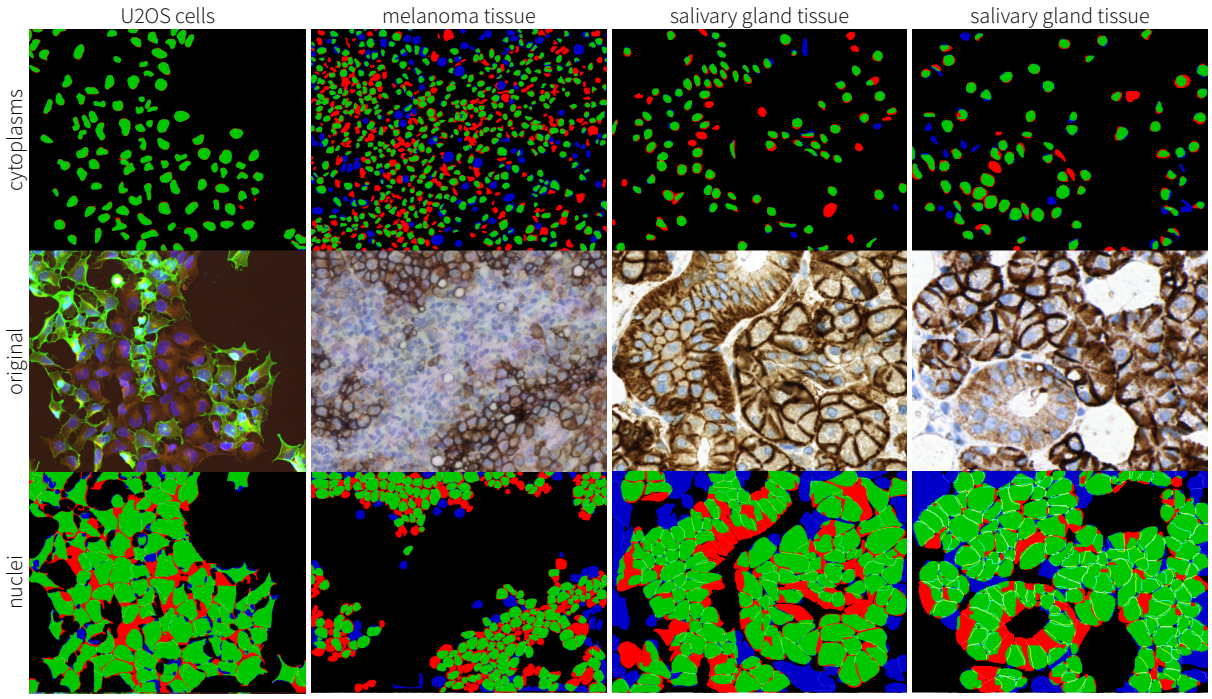

Figure 8. Cytoplasm and nucleus segmentation in the DVP project using nucleAIzer. Fluorescent cell culture (U2OS) and brightfield tissue (melanoma and salivary gland, IHC) images were accurately segmented, quality is shown colour coded as on Figure 6.

The author's main contributions to this work are the following:

- Actively participated in the conception and implementation of the nucleAIzer pipeline during the DSB2018 competition and optimized the pipeline. After the competition, refined and optimized the pipeline.
- Designed and performed experiments, evaluated results and created figures. Helped collect datasets.
- Supervised the TTA project and performed experiment on the DSB test set using nucleAIzer, created some figures.
- Trained and benchmarked models in the DVP project. Performed image analysis and training in the GWS and NRP1 projects.

Summary

In this thesis, a robust workflow of deep learning-based single-cell analysis has been proposed; **Figure 9** and **Table 1** demonstrate how the following main thesis points are related.

I. **Annotation** A deep learning-driven assisted annotation tool AnnotatorJ is proposed to help create large amounts of image annotations in a fast, efficient and convenient manner, in a proper format for direct forward to (various) appropriate deep learning (DL) models as training data. Object detection (via bounding box coordinates), semantic- (binary masks) or instance segmentation (multi-labelled or multi-layered masks) and classification (objects by classes) training data may be easily gathered with AnnotatorJ. Annotations may be assisted with either a U-Net (DL network) or classical image processing algorithms (regions growing, active contours) by contour suggestion which the user can either edit, accept or reject. Several further convenient functions are available in this ImageJ/Fiji plugin.
Additionally, connection to a user-friendly framework for multiple DL model-based segmentation OpSeF (Open Segmentation Framework) is also implemented.

II. **Image segmentation with deep learning** nucleAIzer is an instance segmentation pipeline using so-called image style transfer learning to synthesize such artificial microscopy images that faithfully represent the texture and visual appearance of real microscopy images. This allows adaptation to new domains corresponding to novel experimental setups erasing the instant need of creating new annotations each time, by training a style to each cluster (group or type) of image modalities – corresponding to different types of microscopy, sample origin, staining, quality etc. The pipeline uses a Mask R-CNN model fine-tuned (trained) on an augmented dataset of synthetic image-mask pairs and ground truth annotations from prior experiments to yield a single, multi-modal segmentation model with excellent generalization ability to unseen conditions not represented in the training set. Instance segmentations may be further refined with optional post-processing that includes U-Net probability predictions and morphology operators.
Pixel-wise precision achieved by nucleAIzer is on par with or even superior to recent DL methods (e.g. CellPose), let alone previous classical image processing-based solutions. Accuracy might be further increased via test-time augmentation (TTA) that is ensembling predictions on augmented versions of test images.

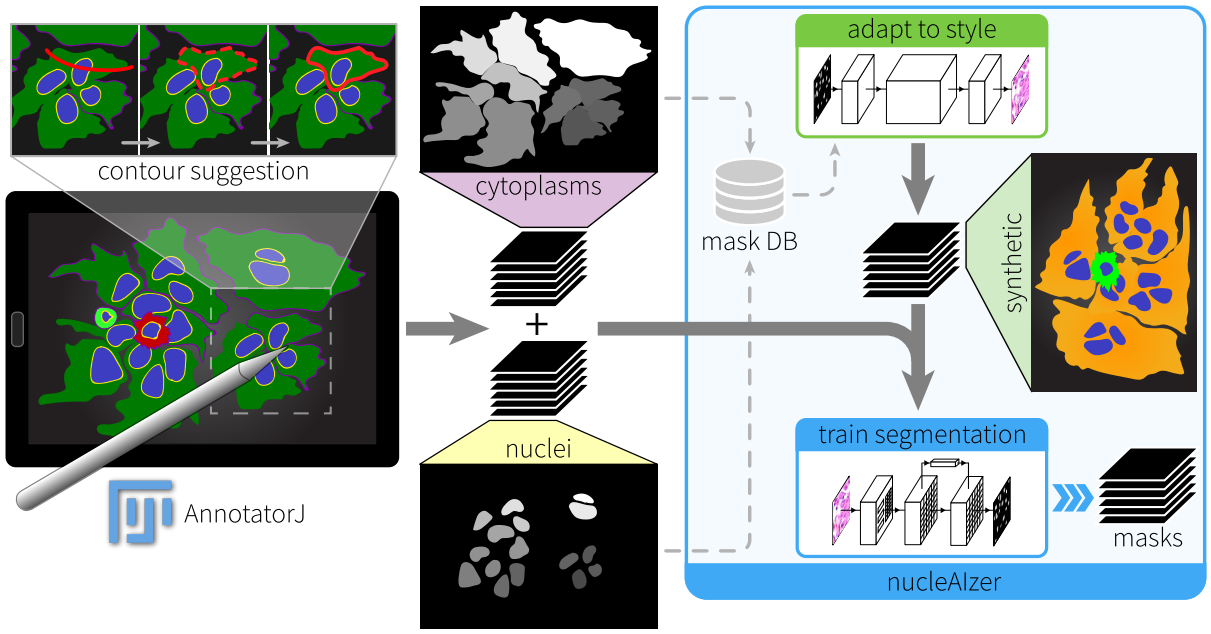

Figure 9. Connection of AnnotatorJ to nucleAIzer. Annotations created with ease in the ImageJ/Fiji plugin AnnotatorJ are exported to appropriate training data format suitable for multiple deep learning models. The annotation process can be significantly accelerated using contour suggestion (with integrated U-Net model prediction) via the *Contour assist* mode. Annotated masks exported from AnnotatorJ may be forwarded to a mask database (mask DB) and serve as input to both the image style transfer generation (green box) and segmentation model training (blue box) in the nucleAIzer pipeline. Novel synthetic microscopy images created in the unknown, new experiments' domain are forwarded to train segmentation as well, then finally masks are output.

Applications Both AnnotatorJ and nucleAIzer have been successfully applied in real experimental projects with our collaborators both domestically and internationally. We showed in the Deep Visual Proteomics (DVP) project [5] that nucleus and cytoplasm segmentations retrieved with nucleAIzer models have higher accuracy than compared methods, both classical and DL-based, suggesting its applicability in single-cell analysis – since single-cell isolation based on these segmentations yielded accurate proteomic data as expected according to scientific literature. Furthermore, in the Neuropilin-1 (NRP1) project identifying a novel target of SARS-COV-2 viral infection nucleAIzer was used for nucleus segmentation based on which phenotypic analysis (using machine learning) confirmed NRP1-related inhibition of infection. AnnotatorJ is used to create ground truth annotations for benchmarking purposes (such as shown in DVP). Several additional collaborative projects utilize these tools whose results are not yet published as of writing this thesis.

Table 1 summarizes the Author's contributions in publications and shows the highly relevant applicability of the thesis points in recent projects to which prominent publications are related.

	Publications					
Part	[1]	[2]	[3]	[4]	[5]	[6]
I.	•	•				
II.			•	•		
Applications					•	•

Table 1. Connection of the thesis contributions and their applications in publications.

References

[1] R. Hollandi, A. Szkalisity, T. Toth, E. Tasnadi, C. Molnar, B. Mathe, I. Grexa, J. Molnar, A. Balind, M. Gorbe, M. Kovacs, E. Migh, A. Goodman, T. Balassa, K. Koos, W. Wang, J. C. Caicedo, N. Bara, F. Kovacs, L. Paavolainen, T. Danka, A. Kriston, A. E. Carpenter, K. Smith and P. Horvath, "nucleAIzer: a parameter-free deep learning framework for nucleus segmentation using image style transfer," *Cell Systems,* vol. 10, no. 5, pp. 453-458.e6, 2020.

[2] 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, 2020.

[3] R. Hollandi, Á. Diósdi, G. Hollandi, N. Moshkov and P. Horváth, "AnnotatorJ: an ImageJ plugin to ease hand-annotation of cellular compartments," *Molecular Biology of the Cell,* vol. 31, no. 20, pp. 2179-2186, 2020.

[4] T. M. Rasse, R. Hollandi and P. Horvath, "OpSeF: Open Source Python Framework for Collaborative Instance Segmentation of Bioimages," *Frontiers in Bioengineering and Biotechnology,* vol. 8, p. 1171, 2020.

[5] A. Mund, F. Coscia, R. Hollandi, F. Kovacs, A. Kriston, A.-D. Brunner, M. Bzorek, S. Naimy, L. M. R. Gjerdrum, B. Dyring-Andersen, J. M. Bulkescher, C. Lukas, C. Gnann, E. Lundberg, P. Horvath and M. Mann, "AI-driven Deep Visual Proteomics defines cell identity and heterogeneity," *bioRxiv,* 2021.

[6] J. L. Daly, B. Simonetti, C. Antón-Plágaro, M. K. Williamson, D. K. Shoemark, L. Simón-Gracia, K. Klein, M. Bauer, R. Hollandi, U. F. Greber, P. Horvath, R. B. Sessions, A. Helenius, J. A. Hiscox, T. Teesalu, D. A. Matthews, A. D. Davidson, P. J. Cullen and Y. Yamauchi, "Neuropilin-1 is a host factor for SARS-CoV-2 infection," *Science,* vol. 370, no. 6518, pp. 861-865, 2020.