[This template contains all of the necessary sections for completing a document for the F1000R Hackathons channel. Do not remove any sections. Do not change the order of any sections. Figures should be submitted as separate files. Detailed author instructions are available at http://f1000research.com/for-authors/article-guidelines/software-tool-articles.]

Instance segmentation of nuclei in electron microscopy images.

Authors (Alphabetical)
Murat Bilgel: Conceptualization, Software
Author email address
Author affiliation, including full address with zip code

Patrick Fletcher: Metric Testing, Conceptualization

Carla Griffiths: Writing-Original Draft Preparation, Conceptualization
Author email address
Author affiliation, including full address with zip code

David Sarma: Conceptualization
George Zaki: Conceptualization

Abstract

To quantify the number of cells present within a Z-stack image, current methods in nuclei segmentation often rely on manual annotation rather than automated methods. To identify and segment nuclei in this image, a Mask-RCNN model was used, which had a user-defined input of four regions of interest. Next, post-processing after this convolutional neural network (CNN) was used to patch, or stitch each region of interest together into one unified image. This required constructing connected nodes through the compressed sparse graph routines package in Python, and then displaying each node as a final image.

The connectivity matrix stitching method was run both after one pass of the Mask-RCNN model using an electron microscopy image, and after a second pass using the same electron microscopy image rotated 180 degrees. The two-pass method achieved more accurate results (Fig. 3), and thus suggests a positive correlation between the number of passes of Mask-RCNN and the overall accuracy of nuclei segmentation.

Keywords

Nuclei, segmentation, Mask-RCNN, neural, network, image, processing

Introduction

Since the advent of image processing, a gap remains between the physical techniques used to capture cell morphology and automated segmentation techniques used to quantify the data. While electron microscopy is a commonly used technique to image cell development, often manual methods are used to count the number and positions of cell types.

Finetuned convolutional neural networks (CNNs) have been traditionally used in the defense and security industries to classify objects of interest. However, shallow CNNs such as AlexNet have already been used in pathology and radiology, namely to determine the malignancy of cancer tumors [1]. Thus, an opportunity remains for biomedical researchers to exploit ready-built networks designed for large-scale object recognition for microscopic biological systems.

However, image workflows can vary greatly between laboratories depending on the microscope model and proprietary software used to capture these images. This can cause variations in the pixel density and pixel grayscale intensity. Traditional methods such as Ostsu's thresholding can be over-deterministic and only work on a case-by-case basis. Overall, current methods often require weeks of preparation before implementation.

Methods

Implementation

Training images from the National Cancer Institute (NCI) were first run through a convolutional neural network to segment nuclei into bounding boxes. Rather than using a loop across the image array, using vectorized functions, the number of unique segmented label IDs was calculated by using the .shape function to calculate the size and then the numpy .unique function to find all unique label IDs across the Z-stack images. This array of matrices was then called the labels_to_combine matrix.

Using Jupyter Notebook (Python 3.0), the connectivity matrix was then generated from a sparse graph of each segmented label, where a matrix of the segmented ID label indexes and their column position was recorded into a separate matrix. The final resulting image was a plot of all identified nodes within the Z-stack image. Each node contains the segmented ID label and the relative column position of the segmented nucleus (relative to the far left corner). As a cross-checking measure, the distribution of the

area and shape of each segmented nucleus was used to determine nuclei that have the most uniform morphology relative to the whole sample.

Operation

Ideal operation should occur in Jupyter Notebook, which can be downloaded as part of the Anaconda library. The user should install the program by opening Jupyter Notebook and downloading the source file from the GitHub repository (https://github.com/NCBI-Hackathons/Biological-structure-segmentation-in-microscopy-images-using-deep-learning). In terms of language requirements, Python 3.0 is required to run this program. The compressed sparse graph routines, scikit-learn, and numpy packages should be installed to run the program.

Results

Include if the paper includes novel data or analyses; should be written as a traditional results section (otherwise, include a Use Cases section).

Use Cases

This Mask-RCNN post-processing script is targeted for researchers seeking an effective computational way to segment nuclei markers. By allowing users to quickly identify the positions and number of nuclei present, this script would aim to give an accurate measurement of the number of cells captured within a microscopy image. Ideally, this post-processing script could be incorporated into other image segmentation models and include use cases for cells without nuclei. Provided a more robust training set was used, future use cases could include segmenting individual cell types using this post-processing method.

In this Mask-RCNN model, a NCI training set was used. Bootstrapping algorithms were used to generate enough images for the training set. The model then identified nuclei in each user-defined region of interest. After running the model, post-processing required adjoining each region of interest and accounting for nuclei that appear in multiple bounding boxes. This stitching method was developed by manually computing a sparse graph and then determining each segmented nuclei connectivity (Figure 2). Compared to the original brute-force method, this connectivity method accounts for more segmented nuclei and achieves a more even spatial distribution of points. After this comparison, a second pass of the model was run using the same electron microscopy image rotated 180 degrees. The two-pass method achieved more accurate results (Fig. 3), and suggests a positive correlation between the number of model passes and the segmentation accuracy rate.

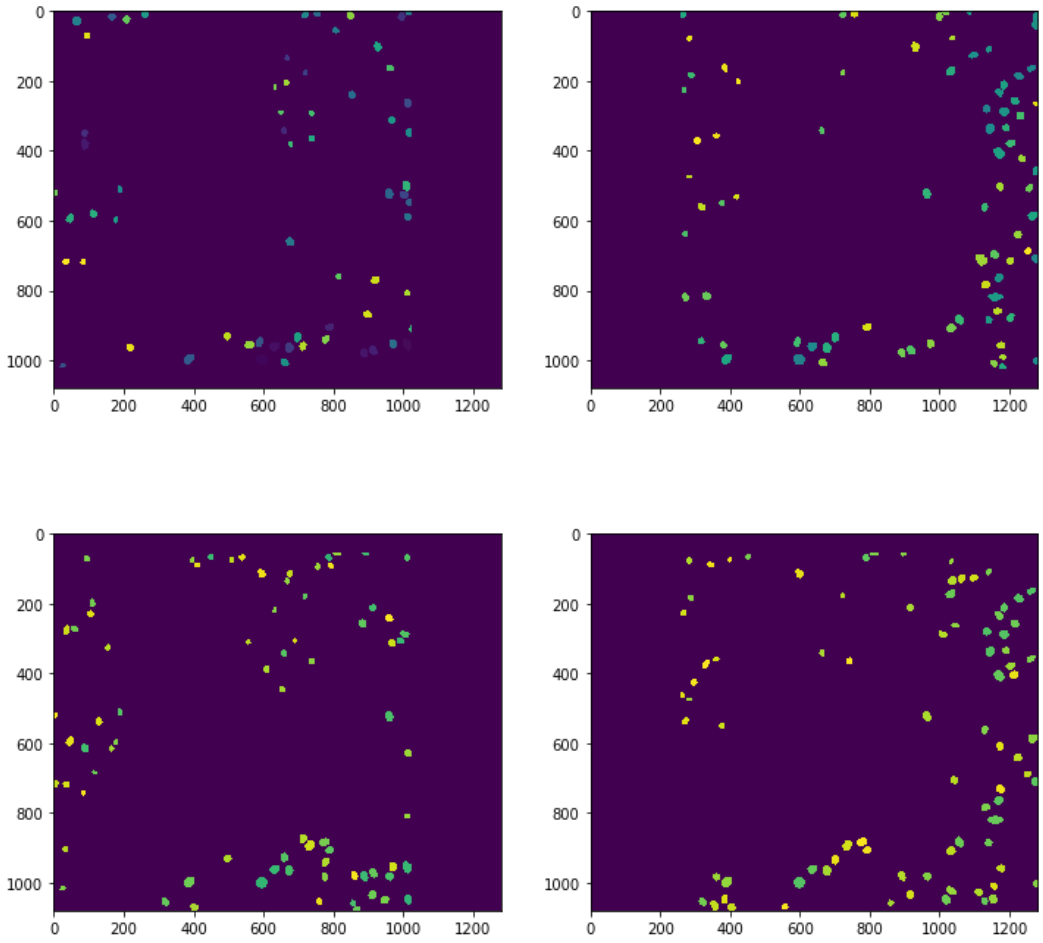

Figure 1. Original Output Given by the MASK-RCNN Model. Caption for Figure 1: Using the Mask-RCNN model, a user-defined output of 4 regions of interest was generated containing segmented nuclei.

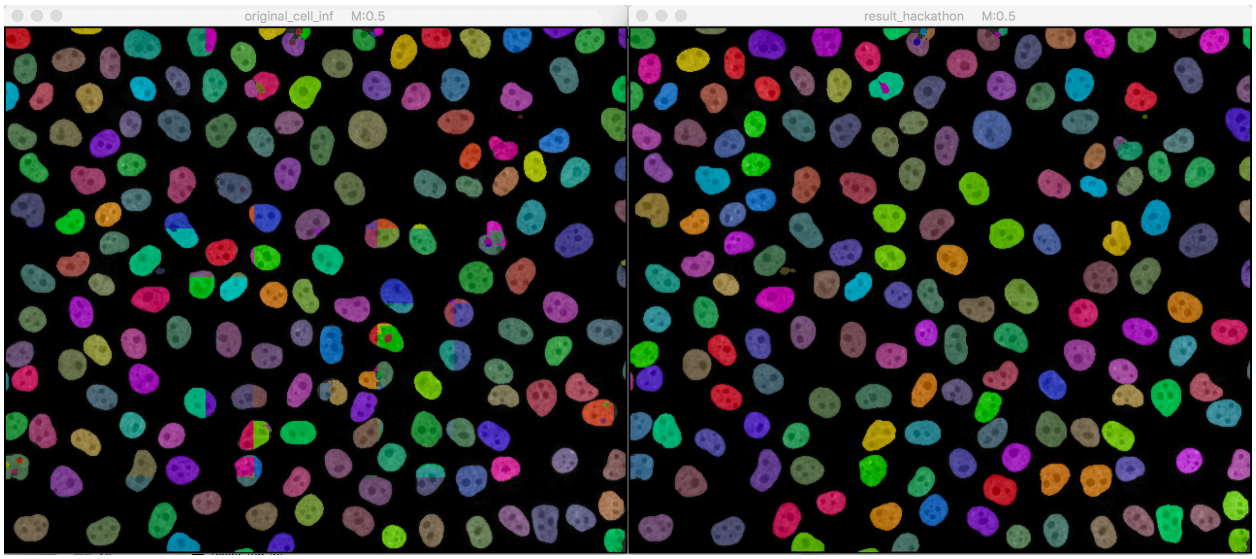

Figure 2. Previous Version of Stitched Nuclei Images (Left) and Connectivity Matrix Approach to Segmentation (Right).
Caption for Figure 2: The previous stitching method (credit: Justin Kim) relied on finding an overlap of identified segmented nuclei in each bounding box region. This still produced some over-division of nuclei, as indicated by multiple pseudo colors per nucleus. whereas the connectivity matrix captures more nuclei.

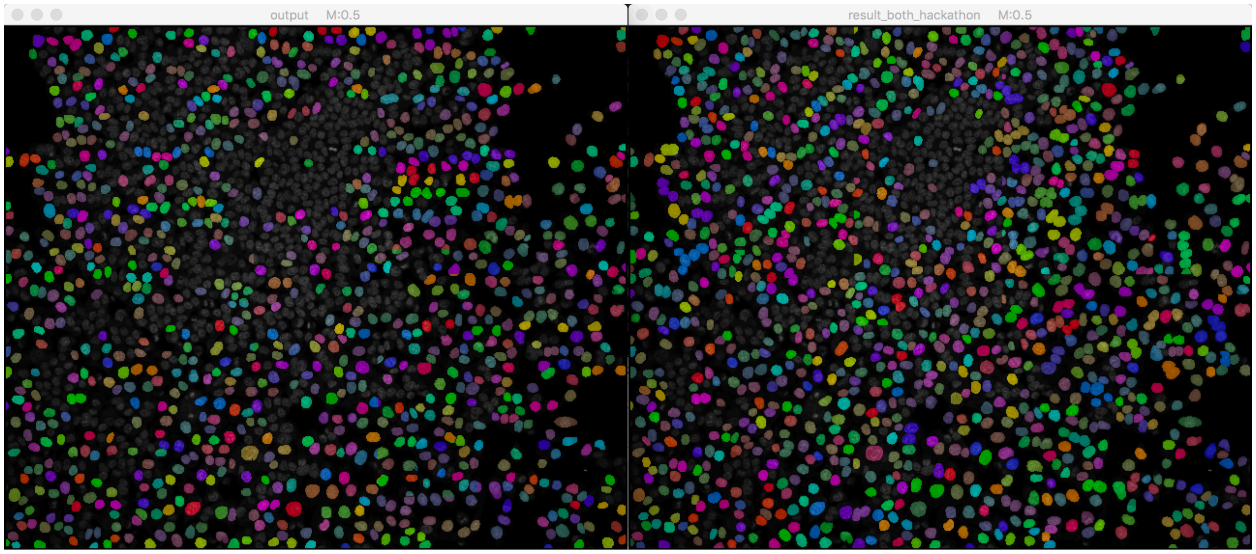

Figure 3. Comparison of the One-Pass Inference (Left) and Two-Pass Inference (Right).
Caption for Figure 3: The Mask-RCNN model was used used two approaches: (1) running a one-pass model on just the image, (2) running two passes of the model, first on the original image and then a second pass on the original image rotated 180 degrees.

Conclusion and Next Steps

One should consider the weight values used in the Mask-RCNN model, as these settings determine which shapes are individual nuclei. Moreover, the connectivity matrix stitching method relies on the assumption that there will always be some nuclei that appear in multiple regions of interest. This causes a form of overlap where some nuclei will stretch over multiple bounding boxes. In the rate case that there are no overlapping segmented nuclei, this stitching method will generate a compilation error. Further research could include the reliability of this method depending on the spatial placement of nuclei in the original microscopy image.

Data and software availability

A Github repository can be found at https://github.com/NCBI-Hackathons/Biological-structure-segmentation-in-microscopy-images-using-deep-learning. This release follows an MIT Open Source License.

Suggested Reviewers

Please pick ten suggested reviewers with whom you have not published in the last three years and who do not work in the same department. Papers can not be sent to F1000 research without such a list.

Author contributions

F1000R uses the CRediT Taxonomy for author contributions. For each author, list their contribution(s) from the list below. Anyone who contributed in another capacity or otherwise does not meet the criteria for authorship (e.g. they did not review the final manuscript) should be included in the acknowledgements.

Contributor Role	**Role Definition**
Conceptualization	Ideas; formulation or evolution of overarching research goals and aims.
Data Curation	Management activities to annotate (produce metadata), scrub data and maintain research data (including software code, where it is necessary for interpreting the data itself) for initial use and later reuse.
Formal Analysis	Application of statistical, mathematical, computational, or other formal techniques to analyze or synthesize study data.
Funding Acquisition	Acquisition of the financial support for the project leading to this publication.
Investigation	Conducting a research and investigation process, specifically performing the experiments, or data/evidence collection.
Methodology	Development or design of methodology; creation of models.
Project Administration	Management and coordination responsibility for the research activity planning and execution.
Resources	Provision of study materials, reagents, materials, patients, laboratory samples, animals, instrumentation, computing resources, or other analysis tools.

Software	Programming, software development; designing computer programs; implementation of the computer code and supporting algorithms; testing of existing code components.
Supervision	Oversight and leadership responsibility for the research activity planning and execution, including mentorship external to the core team.
Validation	Verification, whether as a part of the activity or separate, of the overall replication/reproducibility of results/experiments and other research outputs.
Visualization	Preparation, creation and/or presentation of the published work, specifically visualization/data presentation.
Writing – Original Draft Preparation	Creation and/or presentation of the published work, specifically writing the initial draft (including substantive translation).
Writing – Review & Editing	Preparation, creation and/or presentation of the published work by those from the original research group, specifically critical review, commentary or revision – including pre- or post-publication stages.

Competing interests

No competing interests were disclosed.

Grant information

The author(s) declared that no grants were involved in supporting this work

Acknowledgements

This section should acknowledge anyone who contributed to the research or the writing of the article but who does not qualify as an author; please clearly state how they contributed. Authors should obtain permission to include the name and affiliation, from all those mentioned in the Acknowledgments section.

References

Instructions on using the F1000R Google docs plug in for reference management:
http://f1000.com/work/faq/google-docs-add-on/1

Instructions on using the F1000R Word plug in for reference management:
http://f1000.com/work/faq/word-plugin

Figures and Tables

All figures and tables should be cited and discussed in the article text. Figure legends and tables should be added at the end of the manuscript. Tables should be formatted using the 'insert table' function in Word, or provided as an Excel file. Files for figures should be uploaded as separate files through the submission system. Each figure or table should have a concise title of no more than 15 words. A legend for each figure and table should also be provided that briefly describes the key points and explains any symbols and abbreviations used. The legend should be sufficiently detailed so that the figure or table can stand alone from the main text.