DeepLIIF: An Online Platform for Quantification of Clinical Pathology Slides

Parmida Ghahremani* Joseph Marino * Ricardo Dodds* Saad Nadeem

Memorial Sloan Kettering Cancer Center
{ghahremp,marinoj1,doddsr,nadeems}@mskcc.org

Abstract

In the clinic, resected tissue samples are stained with Hematoxylin-and-Eosin (H&E) and/or Immunhistochemistry (IHC) stains and presented to the pathologists on glass slides or as digital scans for diagnosis and assessment of disease progression. Cell-level quantification, e.g. in IHC protein expression scoring, can be extremely inefficient and subjective. We present DeepLIIF (https://deepliif.org), a first free online platform for efficient and reproducible IHC scoring. DeepLIIF outperforms current state-of-the-art approaches (relying on manual error-prone annotations) by virtually restaining clinical IHC slides with more informative multiplex immunofluorescence staining. Our DeepLIIF cloud-native platform supports (1) more than 150 proprietary/non-proprietary input formats via the Bio-Formats standard, (2) interactive adjustment, visualization, and downloading of the IHC quantification results and the accompanying restained images, (3) consumption of an exposed workflow API programmatically or through interactive plugins for open source whole slide image viewers such as QuPath/ImageJ, and (4) auto scaling to efficiently scale GPU resources based on user demand.

1. Introduction

Single cell protein expression quantification by pathologists and researchers in digital pathology serves a critical role in characterizing tissue microenvironment (TME) for clinical diagnostics to guide patient therapy as well as for developing research predictive and prognostic biomarkers. However, there are widely reported variations in manual singleplex/protein immunohistochemistry (IHC) scoring, even between experienced pathologists, that can negatively impact patient outcomes. Research multiplex imaging platforms can overcome these variations by allowing detection and co-visualization of multiple protein markers in a single tissue sample for more precise scoring of protein markers-of-interest in TME. An example of the visual difference between H&E, IHC, and multiplex stains is shown in Figure 1.

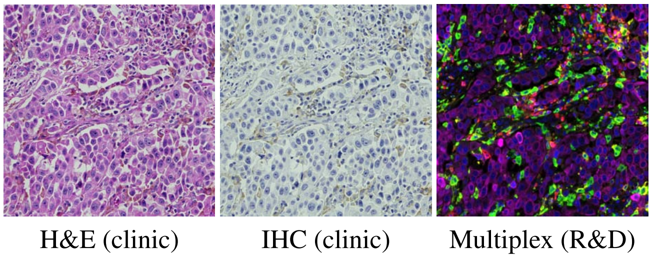


Figure 1. Brightfield Hematoxylin-Eosin (H&E) and singleplex/protein immunohistochemistry (IHC) stains are cheap, high throughput, and clinically deployed. Multiplex IHC staining, still in the research phase, allows detection and co-visualization of multiple protein markers, providing deeper characterization of tissue microenvironment. DeepLIIF bridges the gap between clinic and research to create more informative representations for diagnosis and assessment of disease progression.

There are several open source tools for segmenting cells in pathology slides stained with H&E (Hematoxylin & Eosin), IHC (Hematoxylin + brown DAB substrate), and multiplex [5]. These tools are available as (1) web apps (easiest to run with no prerequisite computational expertise), (2) plugins for image analysis toolboxes (mandates familiarity with these toolboxes), (3) coding notebooks (requires basic coding skills), and (4) code-based pipelines (requires significant coding expertise). In this demonstration paper, we focus on tools that are available as web apps and hence are accessible to a broad audience, including both image analysis experts and non-experts.

There are currently three tools that are available as web apps, namely DeepCell (https://deepcell.org/) [2,6], Cellpose (https://www.cellpose.org/) [7], and NucleAIzer (https://www.nucleaizer.org/) [3]. DeepCell and Cellpose focus on nuclei and cytoplasm segmentation in multiplex images whereas NucleAIzer can segment nuclei across H&E, IHC, and multi-

*Equal Contribution

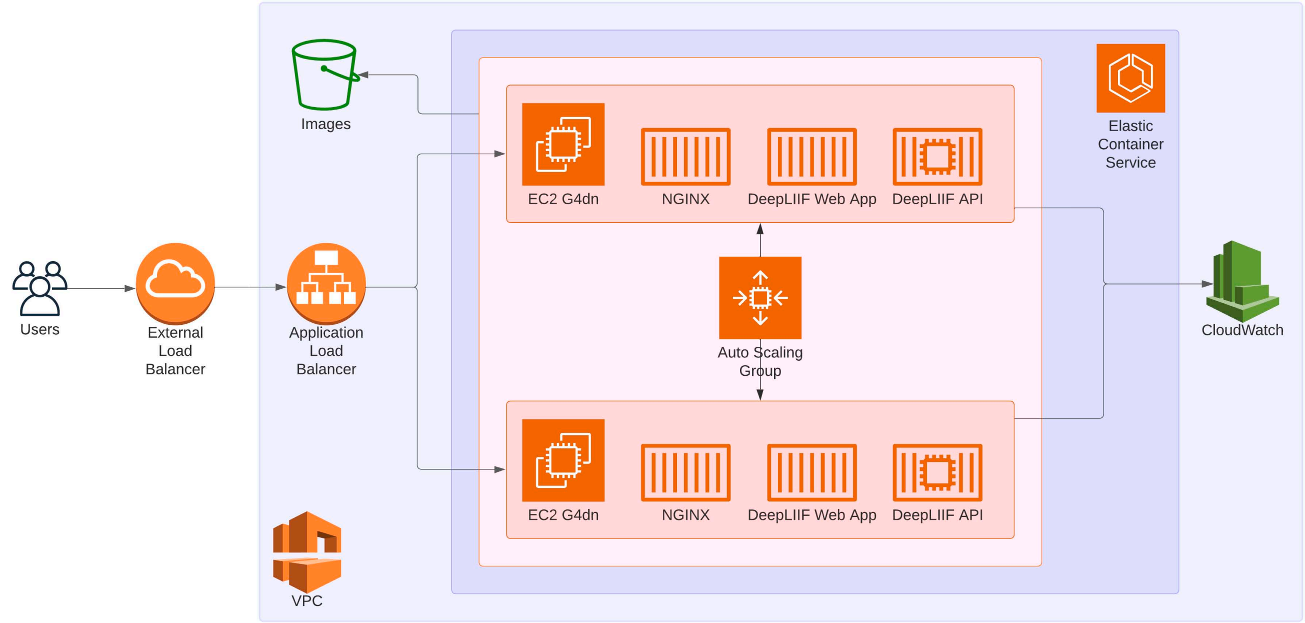


Figure 2. System design diagram for the DeepLIIF online platform deployed on AWS. The application is running on an Elastic Container Service (ECS) cluster with auto scaling. The Web App service handles requests through the https://deepliif.org website, and uses an S3 bucket to store images uploaded by the user, along with the resultant inference images and post-processed images. The API service runs the actual DeepLIIF model and performs post-processing on the images. The Web App communicates with the API service running on the same machine. Direct programmatic access to the API is possible, allowing for the creation of an ImageJ plugin.

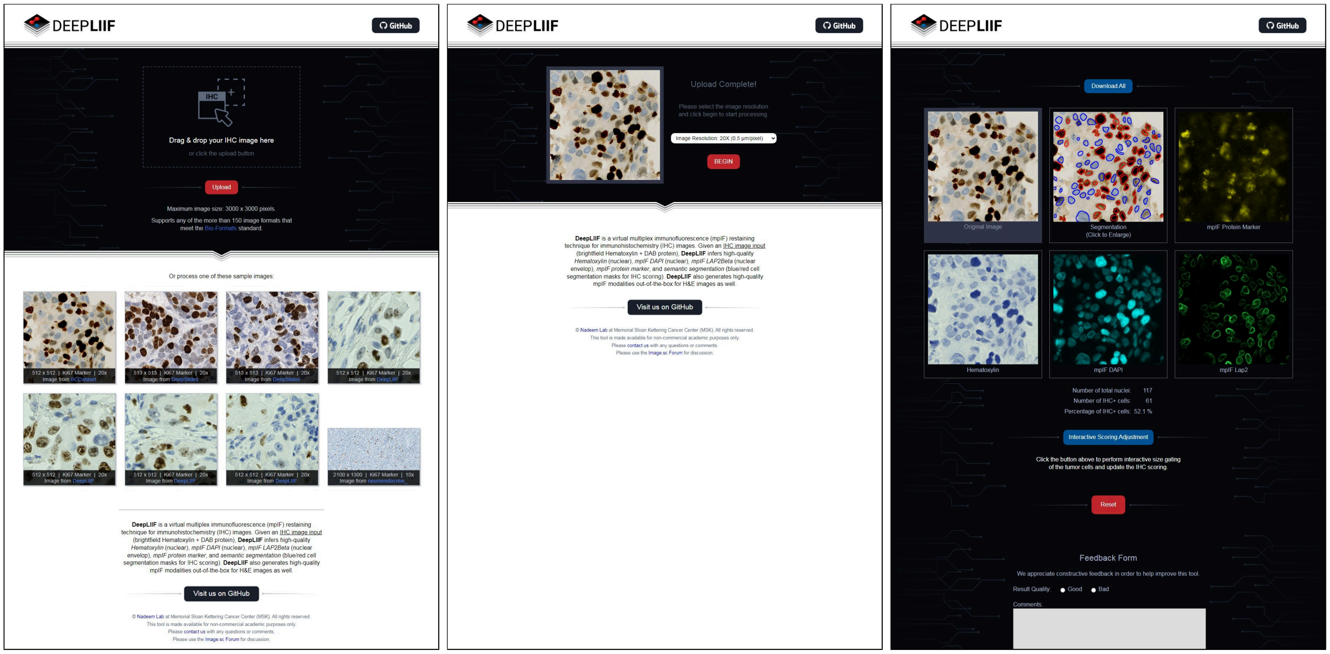
plex images (via style transfer). DeepCell and Cellpose can also perform H&E and IHC nuclei segmentation if these are provided as grayscale images. DeepCell is the only tool among the above three that allows interaction with the segmented results via the DeepCell-Label web module (https://label.deepcell.org/). The Cellpose web app allows 512 × 512 input patches whereas DeepCell restricts input to 2048 × 2048 pixels in order to output results in a reasonable amount of time (less than a minute).

None of the above web apps perform both cell segmentation and classification simultaneously. In brightfield H&E and IHC slides, the two channels (Hematoxylin and Eosin in H&E and Hematoxylin and DAB in IHC) cannot be visualized or analyzed separately without performing the necessary digital stain deconvolution pre-processing step. This is not the case with the research multiplex immunofluorescence platforms, for example, that output each marker as an independent channel that can be visualized and analyzed separately or as composites. Leveraging this insight, we developed DeepLIIF (published in *Nature Machine Intelligence* [1]) for virtual multiplex immunofluorescence restaining of standard IHC slides that performs stain deconvolution and cell segmentation/classification in a single step to output clinically relevant IHC scores (mostly quantified as relevant DAB brown cells divided by the total cells). We showed that DeepLIIF, trained on co-registered IHC and multiplex immunofluorescence images, not only achieves state-of-the-art results in IHC nuclear protein marker scoring (Ki67, ER, PR, P53) but also works out-of-the-box for H&E nuclei segmentation as well as cytoplasmic markers (that are expressed close to the nuclei, e.g. CD3/CD8).

This demonstration paper accompanies our *Nature Machine Intelligence* manuscript [1]. Specifically, here we present the DeepLIIF cloud-native platform which supports (1) more than 150 proprietary/non-proprietary input formats via the Bio-Formats standard, (2) interactive adjustment, visualization, and downloading of the IHC quantification results and the accompanying restained images, (3) consumption of an exposed workflow API programmatically or through interactive plugins for open source whole slide image viewers such as QuPath/ImageJ, and (4) auto scaling to automatically and efficiently scale GPU resources based on user demand.

2. System Design

Our DeepLIIF cloud native platform is hosted on Amazon Web Services (AWS). It is deployed using an Elastic Container Service (ECS) cluster powered by an Auto Scaling group to automatically adjust resources to demand. An Amazon S3 bucket is used for cloud storage for images uploaded through the website interface and the corresponding results. An overview of our system design architecture is shown in Figure 2.



Maximum image size: 3000 x 3000 pixels.

Supports any of the more than 150 image formats that meet the Bio-Formats standard.

Or process one of these sample images:

512 x 512 | Ki67 Marker | 20x Image from BCDataset

513 x 513 | Ki67 Marker | 20x Image from DeepSlides

513 x 513 | Ki67 Marker | 20x Image from DeepSlides

512 x 512 | Ki67 Marker | 20x Image from DeepLIIF

512 x 512 | Ki67 Marker | 20x Image from DeepLIIF

512 x 512 | Ki67 Marker | 20x Image from DeepLIIF

512 x 512 | Ki67 Marker | 20x Image from DeepLIIF

2100 x 1300 | Ki67 Marker | 10x Image from neuroendocrine_

DeepLIIF is a virtual multiplex immunofluorescence (mpIF) restaining technique for immunohistochemistry (IHC) images. Given an IHC image input (brightfield Hematoxylin + DAB protein), **DeepLIIF** infers high-quality *Hematoxylin* (nuclear), *mpIF DAPI* (nuclear), *mpIF LAP2Beta* (nuclear envelop), *mpIF protein marker*, and *semantic segmentation* (blue/red cell segmentation masks for IHC scoring). **DeepLIIF** also generates high-quality mpIF modalities out-of-the-box for H&E images as well.

© Nadeem Lab at Memorial Sloan Kettering Cancer Center (MSK). All rights reserved.
This tool is made available for non-commercial academic purposes only.
Please contact us with any questions or comments.
Please use the Image.sc Forum for discussion.

(a) Image upload.

Upload Complete!

Please select the image resolution and click begin to start processing.

Image Resolution: 20X (0.5 µm/pixel)

DeepLIIF is a virtual multiplex immunofluorescence (mpIF) restaining technique for immunohistochemistry (IHC) images. Given an IHC image input (brightfield Hematoxylin + DAB protein), **DeepLIIF** infers high-quality *Hematoxylin* (nuclear), *mpIF DAPI* (nuclear), *mpIF LAP2Beta* (nuclear envelop), *mpIF protein marker*, and *semantic segmentation* (blue/red cell segmentation masks for IHC scoring). **DeepLIIF** also generates high-quality mpIF modalities out-of-the-box for H&E images as well.

© Nadeem Lab at Memorial Sloan Kettering Cancer Center (MSK). All rights reserved.
This tool is made available for non-commercial academic purposes only.
Please contact us with any questions or comments.
Please use the Image.sc Forum for discussion.

(b) Image verification.

Number of total nuclei: 117
Number of IHC+ cells: 61
Percentage of IHC+ cells: 52.1 %

Click the button above to perform interactive size gating of the tumor cells and update the IHC scoring.

We appreciate constructive feedback in order to help improve this tool.

(c) Results view.

Figure 3. DeepLIIF website interface. (a) The DeepLIIF homepage, where the user has the option to upload their own image, or select one of several sample images to process. (b) Choice of resolution (10×, 20×, or 40×) along with preview of uploaded image. (c) Results page, showing the inferred modalities, segmentation, and IHC quantification score. The user can also download the full resolution inferred images and quantification scores, as well as leave feedback.

We have three services, each running in Docker Containers for ease of portability:

- Web App service for front-end website content.
- API service for back-end data processing.
- Nginx reverse proxy to serve requests.

Both the web application and the API are packaged using Flask. The API service requires use of a CUDA-enabled NVIDIA GPU with at least 4GB of video memory in order to run the DeepLIIF inference model. To meet this requirement, the Auto Scaling group spins up Elastic Compute Cloud (EC2) G4dn machines, each with a single GPU. Each machine that is started contains all three services, with the web service communicating with the API service on the same machine.

Web App Service. The Web App service serves all of the front end website content for each page of our web application. Additionally, user input is handled by this module, including verifying that uploaded files are valid images and within the specified limit of 3000 × 3000 pixels. All connections to our S3 cloud storage bucket occur from this service, including both the original uploaded image and the inference results returned by the API service.

API Service. The API service handles all of the back end computational tasks, including the DeepLIIF model, segmentation post-processing, and quantification. Splitting these computational tasks into a separate service allows the website interface to remain responsive to user requests even during long running processes. In addition to providing back-end service for the DeepLIIF website, the API can also be accessed programmatically, allowing users to write and integrate their own code. This capability has also facilitated our ability to develop a plugin that can be used to access DeepLIIF directly from within the ImageJ application.

Input Images. We utilize Bio-Formats [4] for reading the input image, which allows our application to support over 150 image formats, including all standard formats in the digital pathology domain. To maintain an acceptable processing time and ensure equitable usage of resources, input data is currently limited to images of size no greater than 3000 × 3000 pixels. Since the website handles a series of separate requests across a session, the image data is stored in an S3 bucket. Similarly, the result images and quantification are stored in the S3 bucket to allow for interaction and downloading of the complete result package. Images uploaded to the API are not stored in the cloud (and neither are the corresponding results), since the API consumer is responsible for handling all of the data.

License Agreement Upon the first visit to the DeepLIIF website in a session, the user is presented with the Terms of Use for the website and results. Memorial Sloan Kettering Cancer Center (MSK) makes no warranties as to the accuracy of the results. Submitted images must be the property of the user uploading said images, and MSK reserves the right to copy and use any submitted images as desired. Submitted images should not contain and personally identifiable information (PII) or personal health information (PHI). DeepLIIF content and results are for personal and academic research only, and may not be used in any commercial setting. The underlying DeepLIIF project (`https://github.com/nadeemlab/DeepLIIF`) is licensed under the Apache 2.0 with Commons Clause license, and is available for non-commercial academic purposes.

3. User Experience

We provide three methods with which users can access the capabilities of the online DeepLIIF platform. The first is through an ordinary website interface, where users can upload images, view/download results, and visualize and interact to adjust the quantification score. For users seeking a more automated approach, we have exposed an API endpoint which can be accessed programmatically. Additionally, we have utilized this endpoint to create an ImageJ plugin, allowing users to access DeepLIIF directly within the ImageJ application.

Website Interface

The DeepLIIF platform is made available through a conventional website user interface, which requires only a standards-compliant modern web browser and no other client software to be installed. The steps in using the website to upload and obtain the results from the DeepLIIF model are shown in Figure 3. These figures show the appearance of the site for desktop users, though the interface is fully responsive and can be used on any device. The general steps for a user on the website include:

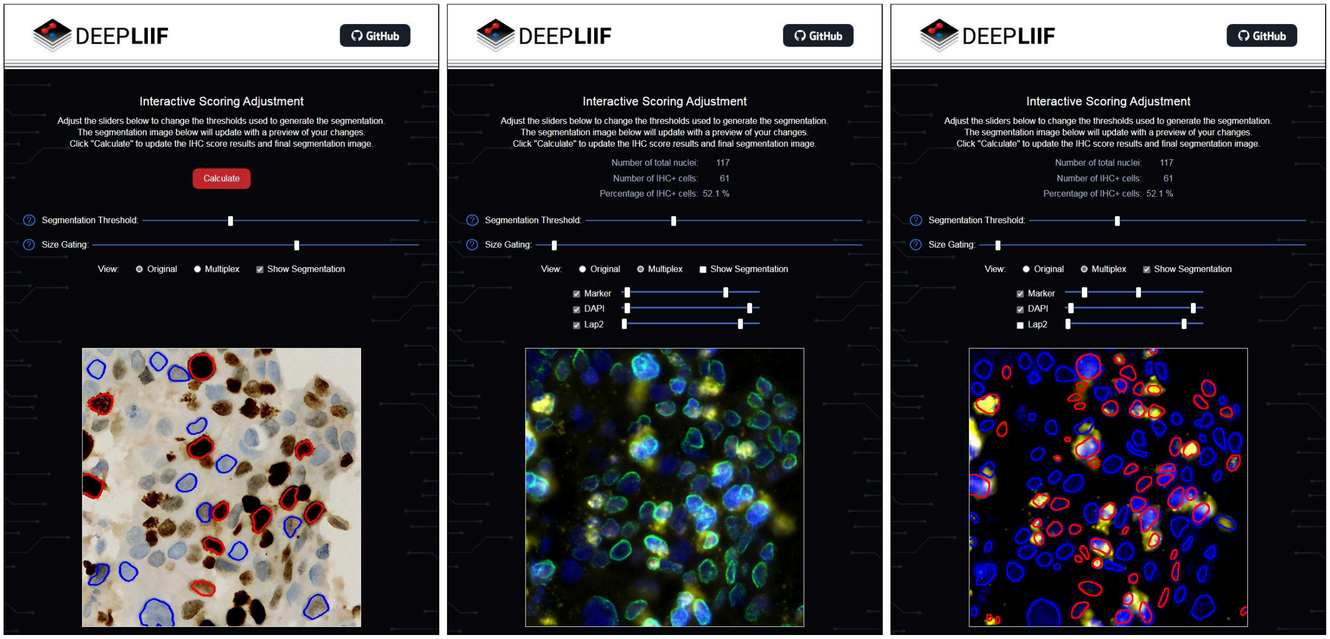
1. Upload a digital pathology image.
2. Verify thumbnail and select image resolution.
3. View/download image and quantification results.
4. Interactively adjust the segmentation and update the final results.

Image Upload. The homepage for DeepLIIF (Figure 3a) provides the user with the option to upload an image either through drag and drop on the target area or via a File Upload dialog box. Users can also choose from one of the sample images below the upload area if they want to try out the system but do not have their own pathology images. Precalculated results are not used for the sample images, so if a sample is selected, it is processed through the same pipeline as any image uploaded by a user would be. Once a file or sample is selected, the image is automatically sent to the server for verification.

Image Verification. Once the uploaded image has been verified by the server as an image and within the required size (currently limited to 3000×3000 pixels), the image data (without any metadata from the original upload) is stored in our S3 bucket. A thumbnail is generated and returned for display in the interface (Figure 3b). The user can verify that the image appears as expected, and is given the option to choose the resolution (magnification level) used when the image was captured ($10\times$, $20\times$, or $40\times$). Once the user is ready, processing through DeepLIIF begins.

Results View. Once the DeepLIIF model has run and post-processing is completed, the results view is presented to the user (Figure 3c). Thumbnails of the result inferred modalities (and original upload) are displayed, along with the classified segmentation outlines overlaid on the original image. The inferred modalities that are displayed include the protein marker, hematoxylin, DAPI, and Lap2 images. Beneath these thumbnails are the scoring results, including the total number of nuclei, the number of positive cells, and the percentage of positive cells. For further inspection and records, all of the full resolution images (along with the scoring results) can be downloaded in a single ZIP file. At the bottom of the page, a feedback form is provided so that the user can leave comments specifically about the image that was processed, which will allow us to retrospectively evaluate the performance of our system.

Interactive Adjustment. In addition to the automatic results provided by DeepLIIF, our online platform provides an interactive tool with which the user can adjust the segmentation results, and thus the quantification results. This view also allows the user to visualize the results in multiple ways, such as with or without segmentation outlines and various combinations of inferred multiplex data. The user can adjust the segmentation threshold and perform size gating on the image. A visual preview of the adjustments is shown in the page as the sliders are moved, implemented in JavaScript and using web workers for distributed processing to yield real-time performance even at the highest supported image resolution of 3000×3000. When the user is satisfied with the preview segmentation overlay, a button is clicked to perform the full update on the server for both the segmentation image and the quantification. The user can also download these updated results, if desired. The interface for this interactive adjustment is shown in Figure 4.



Interactive Scoring Adjustment

Adjust the sliders below to change the thresholds used to generate the segmentation. The segmentation image below will update with a preview of your changes. Click "Calculate" to update the IHC score results and final segmentation image.

Segmentation Threshold:

Size Gating:

View: ◉ Original ○ Multiplex ☑ Show Segmentation

(a) Original IHC with segmentation.

Interactive Scoring Adjustment

Adjust the sliders below to change the thresholds used to generate the segmentation. The segmentation image below will update with a preview of your changes. Click "Calculate" to update the IHC score results and final segmentation image.

Number of total nuclei: 117
Number of IHC+ cells: 61
Percentage of IHC+ cells: 52.1 %

Segmentation Threshold:

Size Gating:

View: ○ Original ◉ Multiplex ☐ Show Segmentation

☑ Marker
☑ DAPI
☑ Lap2

(b) Inferred multiplex DAPI + Ki67 + Lap2.

Interactive Scoring Adjustment

Adjust the sliders below to change the thresholds used to generate the segmentation. The segmentation image below will update with a preview of your changes. Click "Calculate" to update the IHC score results and final segmentation image.

Number of total nuclei: 117
Number of IHC+ cells: 61
Percentage of IHC+ cells: 52.1 %

Segmentation Threshold:

Size Gating:

View: ○ Original ◉ Multiplex ☑ Show Segmentation

☑ Marker
☑ DAPI
☐ Lap2

(c) Inferred DAPI + Ki67 with segmentation.

Figure 4. User interaction and multiplex visualization. The user can interactively adjust the segmentation threshold and perform size gating. The on-screen preview will update in real time to show the new cell segmentation based on the adjustments, and the IHC score can then be updated with these user-specified parameters. User controlled visualization of the inferred multiplex data is also provided. (a) Original IHC image with segmentation outlines. (b) Inferred multiplex image of DAPI + Ki67 + Lap2. (c) Inferred multiplex image of DAPI + Ki67 with segmentation outlines.

Multiplex Visualization In addition to the segmentation overlay on the original IHC image, we also provide an inferred multiplex view which is user controlled. The user can combine and adjust the following three inferred multiplex modalities, which are part of the DeepLIIF results:

- Protein marker, indicating IHC positive cells.
- DAPI stain, highlighting cell locations.
- Lap2 stain, highlighting cell boundaries.

These inferred multiplex images help the user to view the cellular data without background noise and evaluate the accuracy of the segmentation results. Each of these modalities can be toggled on or off, and the double slider can be used to adjust the histogram normalization window for each marker/stain channel. Additionally, the user can toggle the segmentation outlines on or off for both the inferred multiplex view and the original uploaded image view.

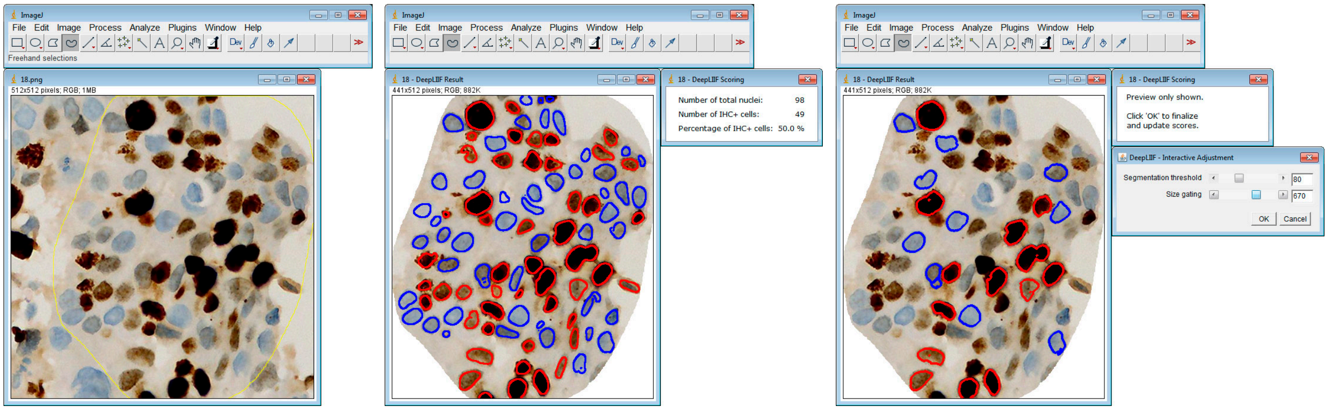
As an example, the combination of DAPI + Lap2 + Ki67 without the segmentation outlines (Figure 4b) allows the user to compare against the original IHC image to confirm that the DeepLIIF model is accurately identifying the individual cells along with protein markers. The combination of DAPI + Ki67 with the segmentation outlines (Figure 4c) allows the user to confirm the accuracy of the cell segmentation and classification.

API Endpoint

The image inference can also be accomplished through an API endpoint, giving users programmatic access to the inferred modalities, segmentation, and quantification results. This is accessed by posting a multipart-encoded request containing the original image file to the `/api/infer` endpoint. The response is JSON encoded data including an array of images with the results of the DeepLIIF model and the quantification scoring values. The result images are Base64 encoded Portable Network Graphics (PNG) image data, allowing them to be easily viewed or saved locally. The following query parameters can be passed:

- `resolution` (*string*): resolution used to scan the slide (10×, 20×, 40×); defaults to 20×.
- `pil` (*boolean*): if true, use the Python Imaging Library to load the image instead of Bio-Formats. For common image formats, this will reduce the time required to read the image on the server.
- `slim` (*boolean*): if true, return only the segmentation result image. This will reduce the amount of data that is returned.

A Python code sample showing how to access and obtain results from this API is given in Code 1.


ImageJ
File Edit Image Process Analyze Plugins Window Help
Freehand selections
18.png
18 - DeepLIIF Result
18 - DeepLIIF Scoring
DeepLIIF - Interactive Adjustment
OK
Cancel

512x512 pixels; RGB; 1MB

441x512 pixels; RGB; 882K

Number of total nuclei: 98
Number of IHC+ cells: 49
Percentage of IHC+ cells: 50.0 %

Preview only shown.

Click 'OK' to finalize and update scores.

Segmentation threshold 80
Size gating 670

(a) Select a region of interest (ROI).

(b) DeepLIIF results for ROI.

(c) Interactive adjustment.

Figure 5. ImageJ plugin. (a) A user has selected a region of interest (ROI), outlined in yellow, using the ImageJ freehand selection tool. (b) After running the DeepLIIF plugin to submit the image region and obtain the results, the segmentation and scoring for the ROI are displayed. (c) The user can invoke the interactive adjustment command of the plugin to adjust the results, as is possible on the DeepLIIF website.

ImageJ Plugin

The API endpoint also allows integration of our platform into open source whole slide viewers, such as QuPath/ImageJ, through custom plugins. We have developed and made available an ImageJ plugin which allows users to easily submit images and obtain results from the DeepLIIF cloud platform. This plugin provides a user experience that is similar to that of the website, but with the user able to take advantage of the additional tools provided by ImageJ. For example, Figure 5 shows DeepLIIF being run on a region of interest (ROI) that a user has selected within an image. All of the inferred image results, along with the scoring results, are saved as local files.

4. Limitations and Future Work

Currently, we enforce a limit on the size of uploaded images to no larger than 3000 × 3000 pixels. As we continue to improve the performance of our model, we will increase the size of images which are allowed. Moreover, at present we expect users to use our ImageJ/QuPath plugin to open whole slide images, crop regions of interest (normally tumor regions) for IHC scoring, and upload to the DeepLIIF platform. Users can obviously crop regions of interest in other whole slide image viewers and upload these manually as well. Our eventual goal is to incorporate an open source whole slide image viewer (e.g. HistomicsUI, https://digitalslidearchive.github.io/digital_slide_archive/) into the web interface, which will allow users to upload a whole slide image, annotate one or more regions of interest, and then process those selected regions to obtain the inferred modalities and quantification.

Our platform currently only supports nuclear markers (such as Ki67, ER, PR) and cytoplasmic markers (e.g. CD3/CD8) that express close to the boundary of the nuclei. In the future, we also plan to support cytoplasmic/membranous IHC protein markers, such as PD-L1, HER2, etc. We invite readers to visit our online platform at https://deepliif.org, which will be under continuous development as we refine our model and add additional features.

Acknowledgements

This project was supported by MSK Cancer Center Support Grant/Core Grant (P30 CA008748) and MSK DigITs Hybrid Research Initiative.

References

[1] Parmida Ghahremani, Yanyun Li, Arie Kaufman, Rami Vanguri, Noah Greenwald, Michael Angelo, Travis J. Hollmann, and Saad Nadeem. Deep learning-inferred multiplex immunofluorescence for ihc quantification. *Nature Machine Intelligence*, 2022. 2

[2] Noah F. Greenwald, Geneva Miller, Erick Moen, Alex Kong, Adam Kagel, Thomas Dougherty, Christine Camacho Fullaway, Brianna J. McIntosh, Ke Xuan Leowa, Morgan Sarah Schwartz, Cole Pavelchek, Sunny Cui, Isabella Camplisson, Omer Bar-Tal, Jaiveer Singh, Mara Fong, Gautam Chaudhry, Zion Abraham, Jackson Moseley, Shiri Warshawsky, Erin Soon, Shirley Greenbaum, Tyler Risom, Travis Hollmann, Sean C. Bendall, Leeat Keren, William Graf, Michael Angelo, and David Van Valen. Whole-cell segmentation of tissue images with human-level performance using

```
import os
import json
import base64
from io import BytesIO
import requests
from PIL import Image

# Use the sample images from the main DeepLIIF repo
images_dir = './Sample_Large_Tissues'
filename = 'ROI_1.png'

res = requests.post(
    url='https://deepliif.org/api/infer',
    files={
        'img': open(f'{images_dir}/{filename}', 'rb')
    },
    # optional param that can be 10x, 20x (default), or 40x
    params={
        'resolution': '20x'
    }
)

data = res.json()

def b64_to_pil(b):
    return Image.open(BytesIO(base64.b64decode(b.encode())))

for name, img in data['images'].items():
    output_filepath = f'{images_dir}/{os.path.splitext(filename)[0]}_{name}.png'
    with open(output_filepath, 'wb') as f:
        b64_to_pil(img).save(f, format='PNG')

print(json.dumps(data['scoring'], indent=2))
```

Code 1. Python code sample for using the DeepLIIF cloud API endpoint.

large-scale data annotation and deep learning. *Nature Biotechnology*, 2021. 1

[3] Reka Hollandi, Abel Szkalisity, Timea Toth, Ervin Tasnadi, Csaba Molnar, Botond Mathe, Istvan Grexa, Jozsef Molnar, Arpad Balind, Mate Gorbe, Maria Kovacs, Ede Migh, Allen Goodman, Tamas Balassa, Krisztian Koos, Wenyu Wang, Juan Carlos Caicedo, Norbert Bara, Ferenc Kovacs, Lassi Paavolainen, Tivadar Danka, Andras Kriston, Anne Elizabeth Carpenter, Kevin Smith, and Peter Horvath. NucleAIzer: A parameter-free deep learning framework for nucleus segmentation using image style transfer. *Cell Systems*, 10(5):453–458, 2020. 1

[4] Melissa Linkert, Curtis T. Rueden, Chris Allan, Jean-Marie Burel, Will Moore, Andrew Patterson, Brian Loranger, Josh Moore, Carlos Neves, Donald MacDonald, Aleksandra Tarkowska, Caitlin Sticco, Emma Hill, Mike Rossner, Kevin W. Eliceiri, and Jason R. Swedlow. Metadata matters: Access to image data in the real world. *The Journal of Cell Biology*, 189(5):777–782, 2010. 3

[5] Alice M. Lucas, Pearl V. Ryder, Bin Li, Beth A. Cimini, Kevin W. Eliceiri, and Anne E. Carpenter. Open-source deep-learning software for bioimage segmentation. *Molecular Biology of the Cell*, 32(9):823–829, 2021. 1

[6] Erick Moen, Dylan Bannon, Takamasa Kudo, William Graf, Markus Covert, and David Van Valen. Deep learning for cellular image analysis. *Nature Methods*, 16(12):1233–1246, 2019. 1

[7] Carsen Stringer, Tim Wang, Michalis Michaelos, and Marius Pachitariu. Cellpose: A generalist algorithm for cellular segmentation. *Nature Methods*, 18(1):100–106, 2021. 1