A Guide for Open-Source Deep Learning Tools in Bioimaging

Prateek Verma, Hao Van, Xintao Wu

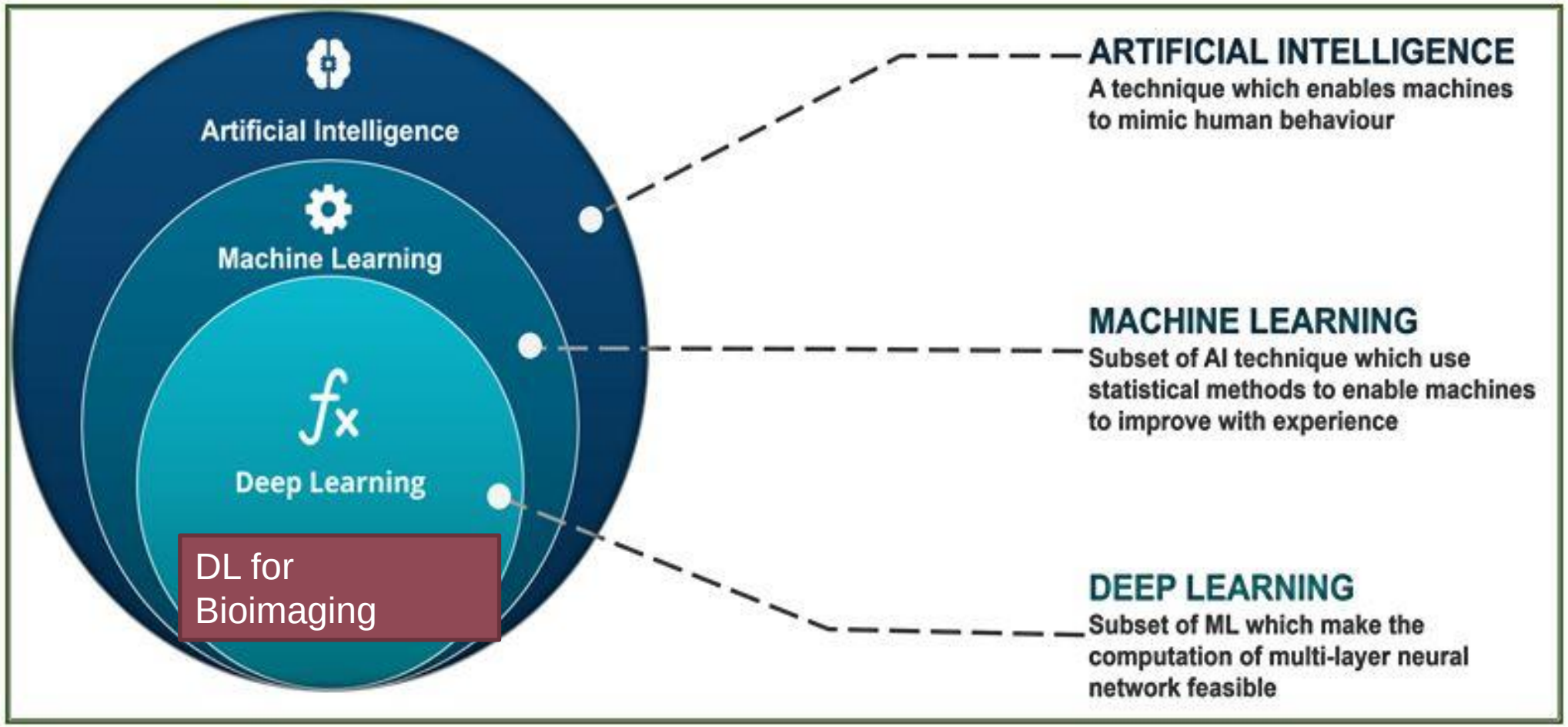
Aug 14, 2024

Arkansas Integrative Metabolic Research Center

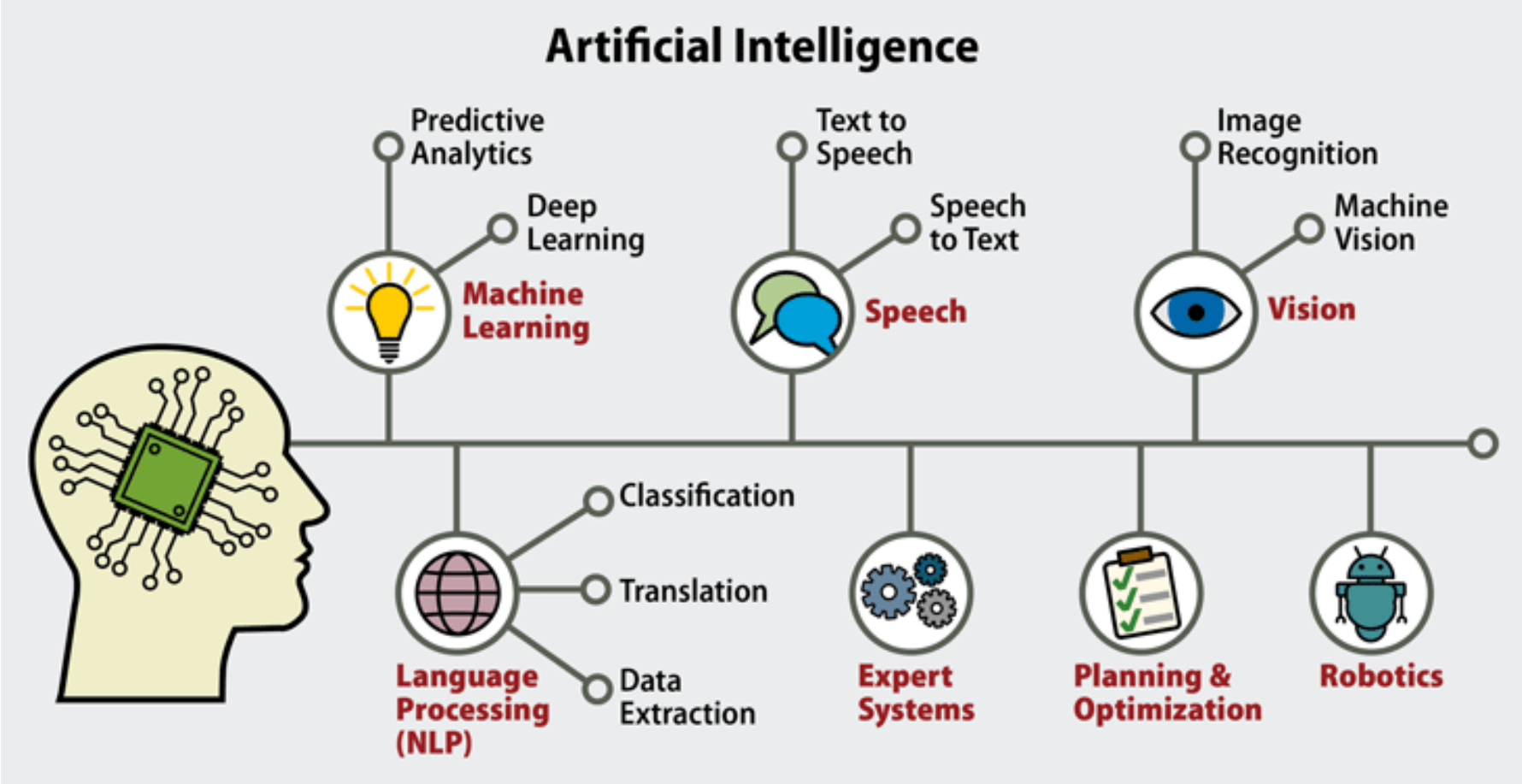
Outline

- Introduction to deep learning
  - Exercise 1
- Deep learning tools/platforms in bioimaging
  - Traditional ML tools: ImageJ, ImageJ2 and Fiji
  - DL tools: BioImage Model Zoo, DeepImageJ
  - Exercise 2
- Overview of DL algorithms in bioimaging
  - Segmentation (Exercise 3)
  - Denoising and image restoration
  - Super-resolution microscopy
  - Object detection
  - Image-to-image translation
- More tools
  - ZeroCostDL4Mic (Exercise 4)
  - CSBDeep, etc.

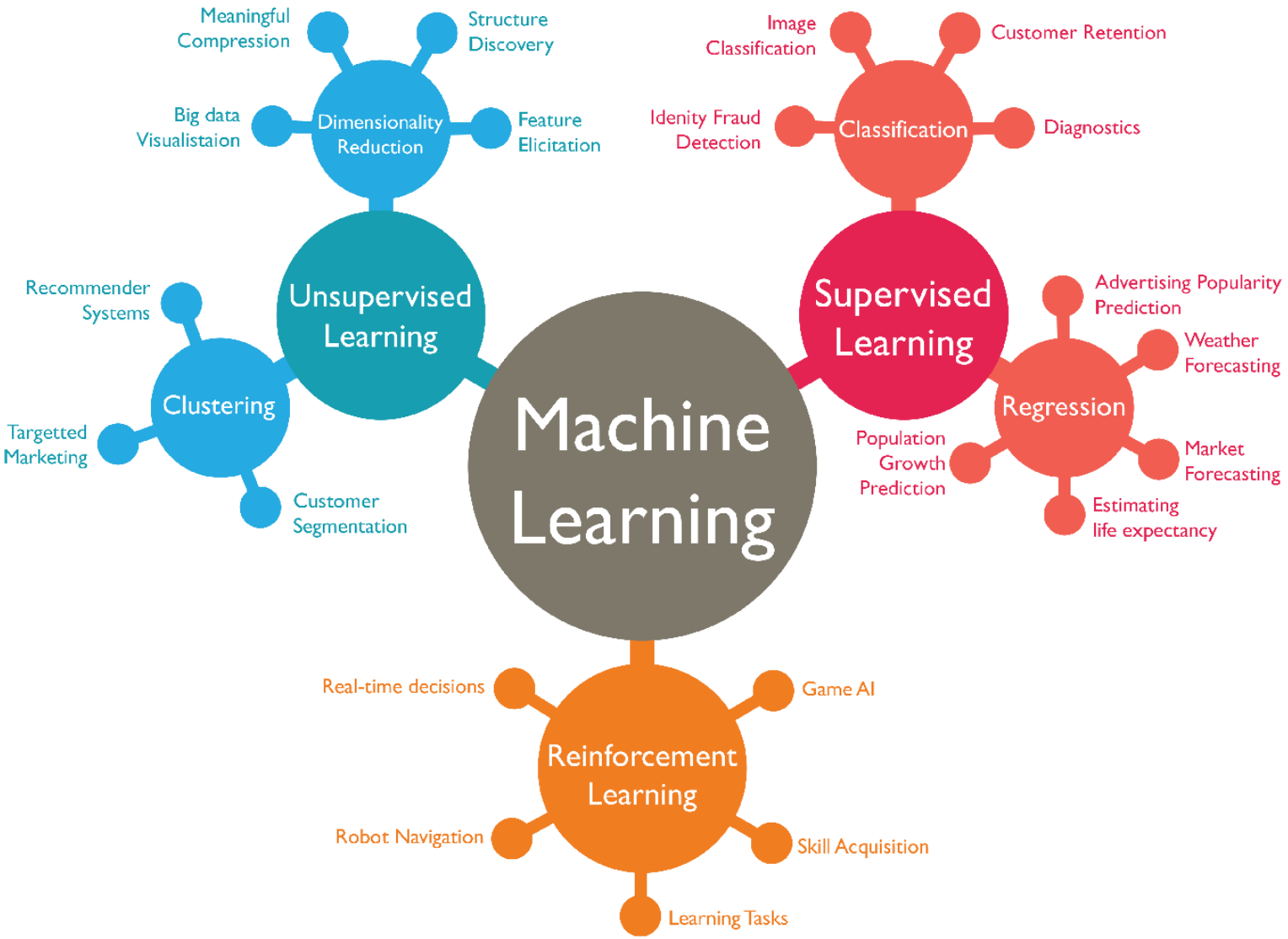
AI vs. ML vs. DL


Artificial Intelligence
Machine Learning
Deep Learning
DL for Bioimaging
ARTIFICIAL INTELLIGENCE
A technique which enables machines to mimic human behaviour
MACHINE LEARNING
Subset of AI technique which use statistical methods to enable machines to improve with experience
DEEP LEARNING
Subset of ML which make the computation of multi-layer neural network feasible

Artificial Intelligence

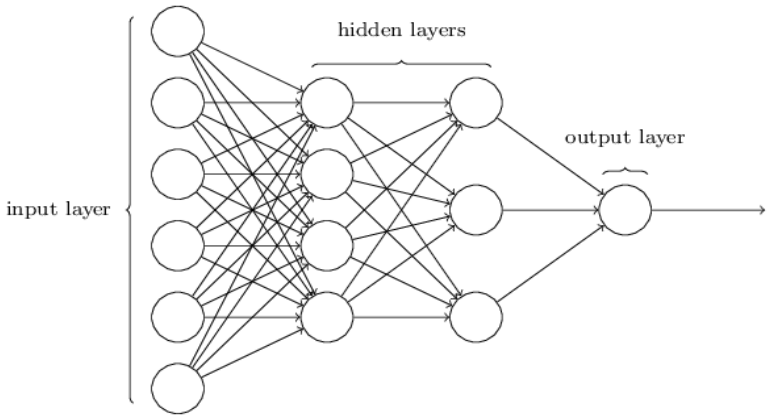


Machine Learning

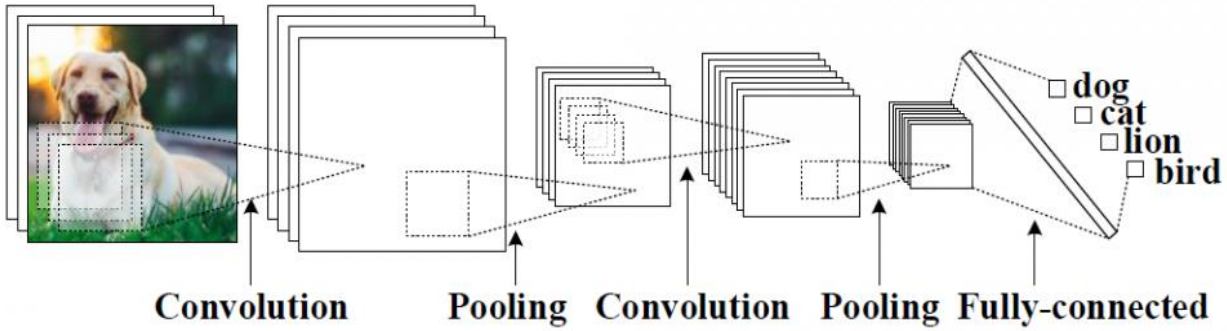


Basic Deep Learning Structures

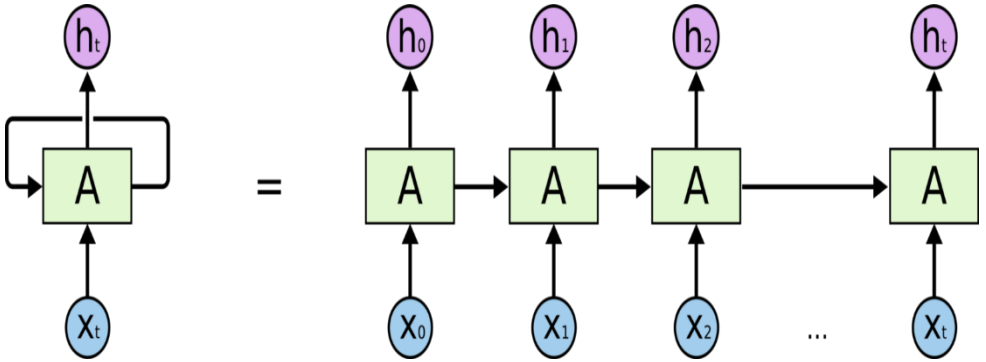
Feedforward Neural Network



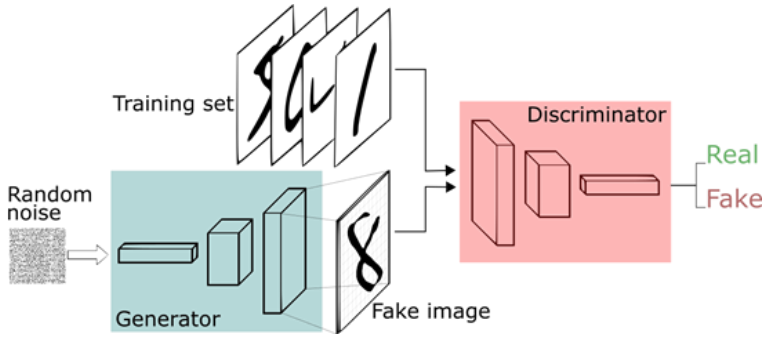
Convolutional Neural Network



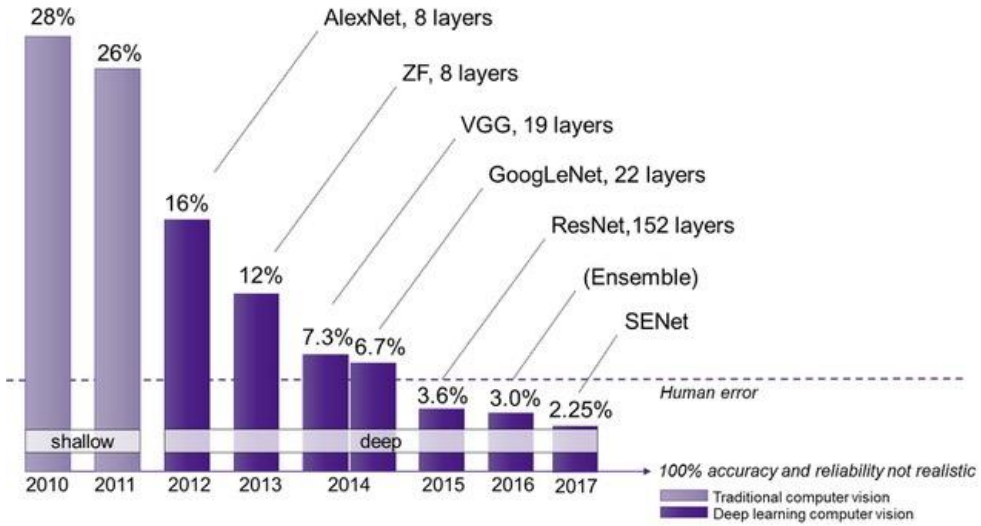
Recurrent Neural Network

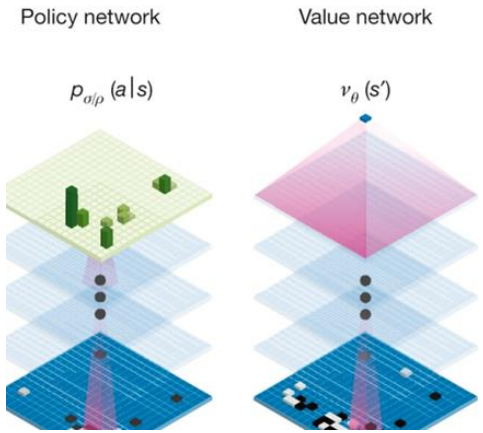


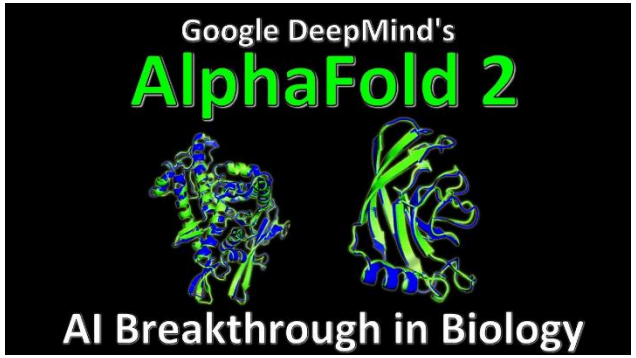
Generative Adversarial Network



Deep Learning

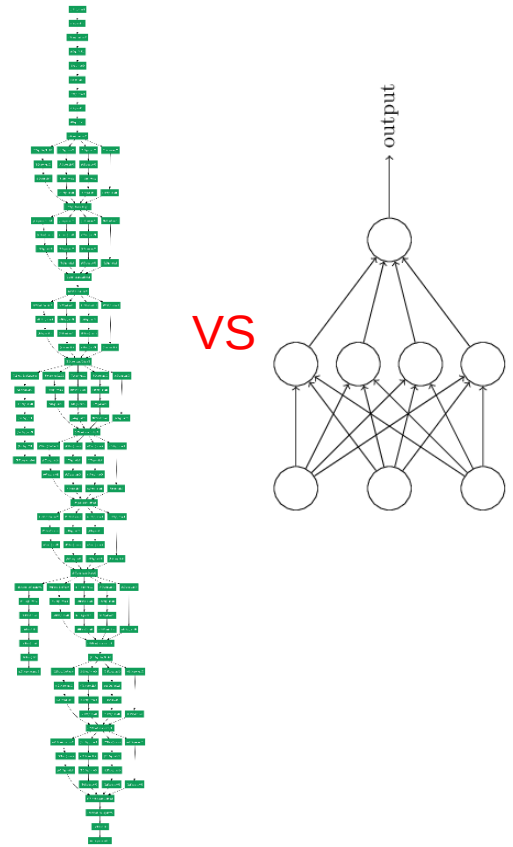




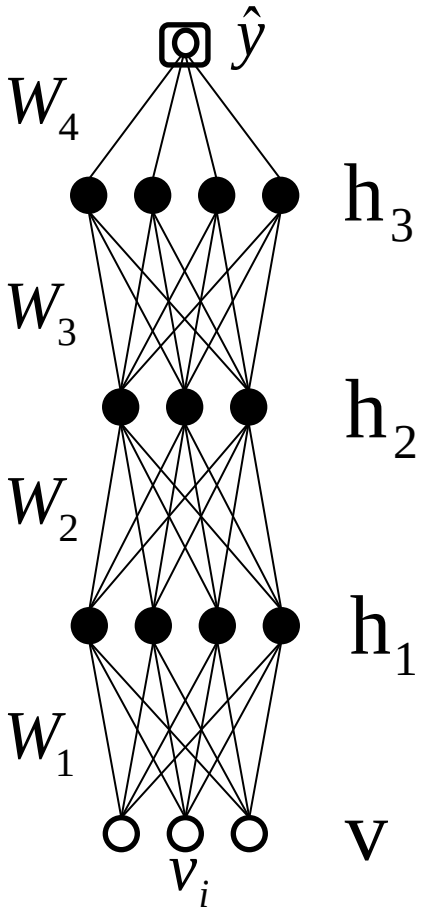


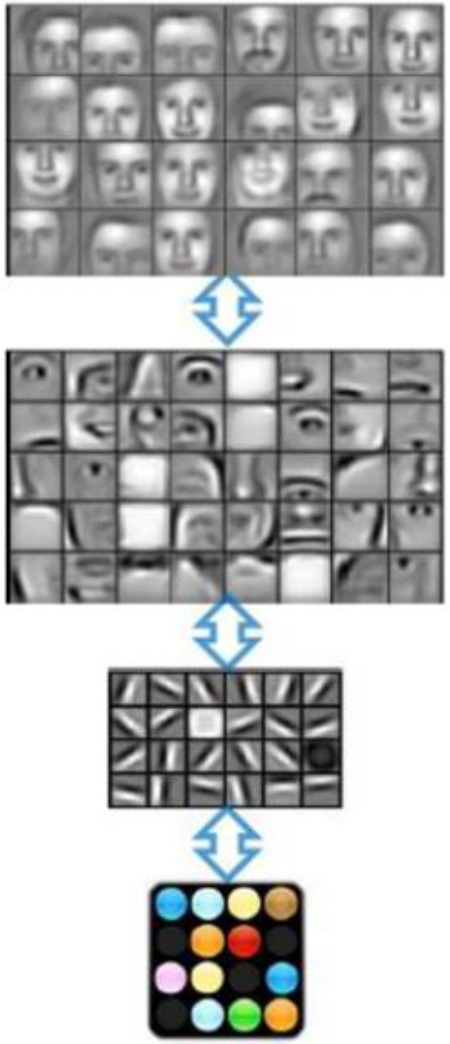
Why does DL work so well?

- lots of data (Big Data)
- Very flexible models
- GPGPU (powerful machines)
- Advanced algorithms for optimization, activation, regularization
- Huge research society (vision, speech, NLP, bioimaging, etc.)



Learning of Representations





3rd Layer
"Objects"

1st Layer
"Edges"

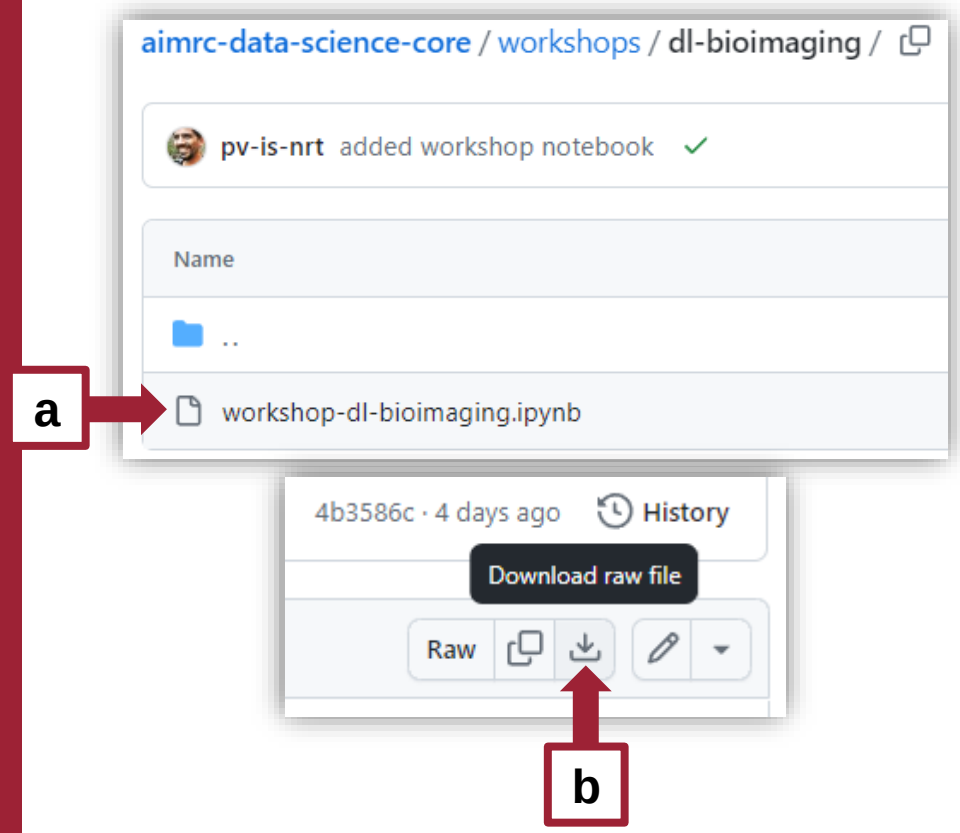
Pixels

[Andrew Ng]

Exercise 1 – **Classification**

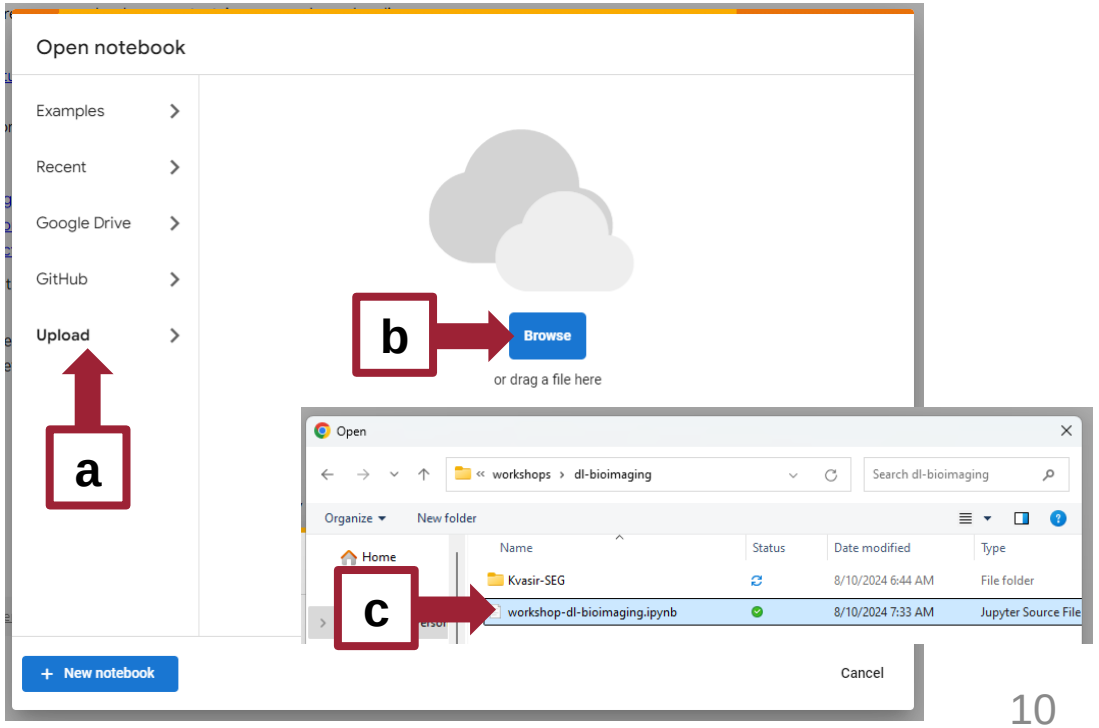
1. Download the Jupyter Notebook

a. AIMRC GitHub > workshops > dl-bioimaging
b. Click on “Download raw file” button in the upper right corner.

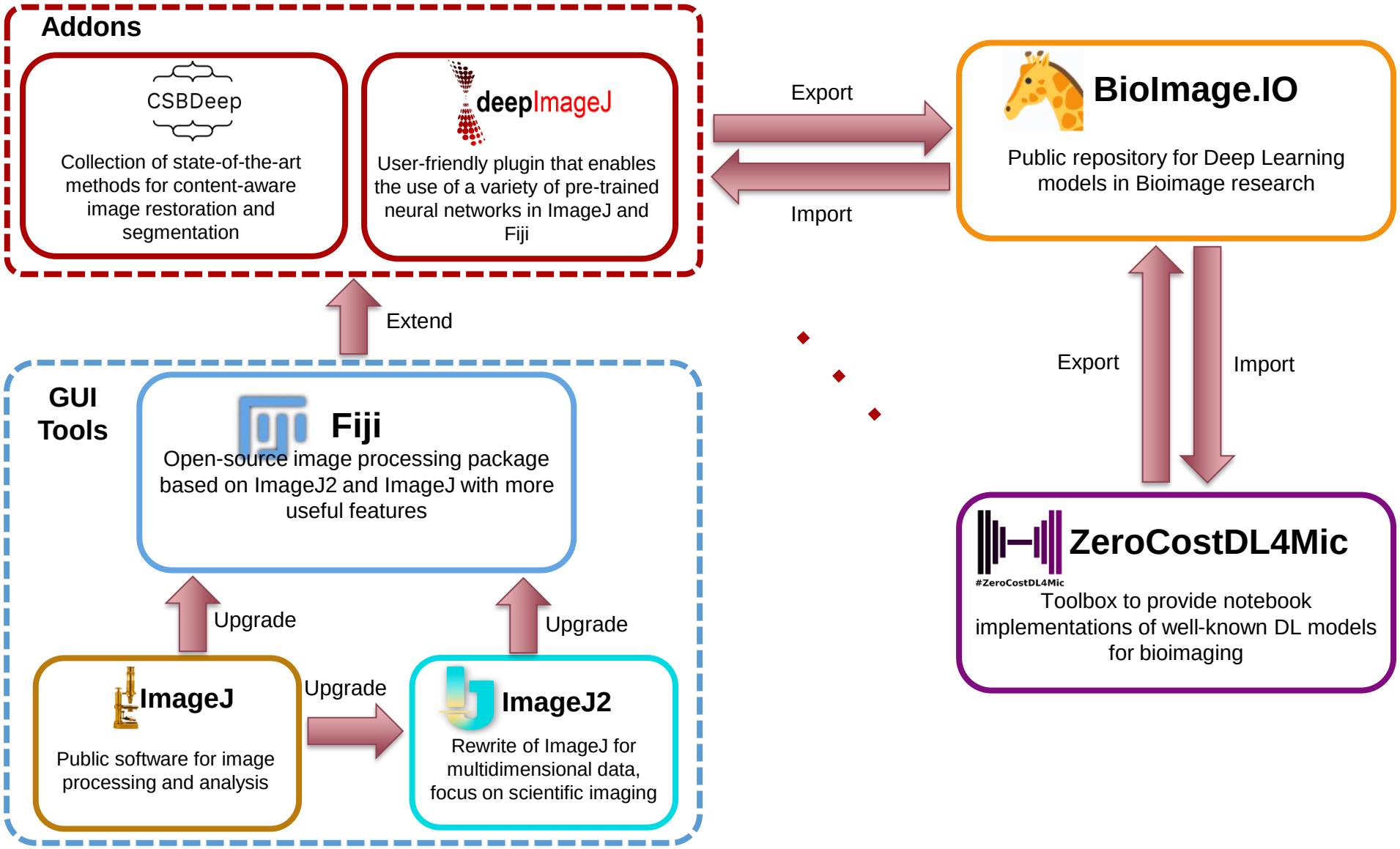


2. Upload the Notebook to Google Colab

a. Open Google Colab in your browser.
b. Upload > Browse. Upload the Notebook file (.ipynb) you downloaded in the previous step.
c. Follow steps written in the Notebook.



Software Tools



ImageJ, ImageJ2 and Fiji

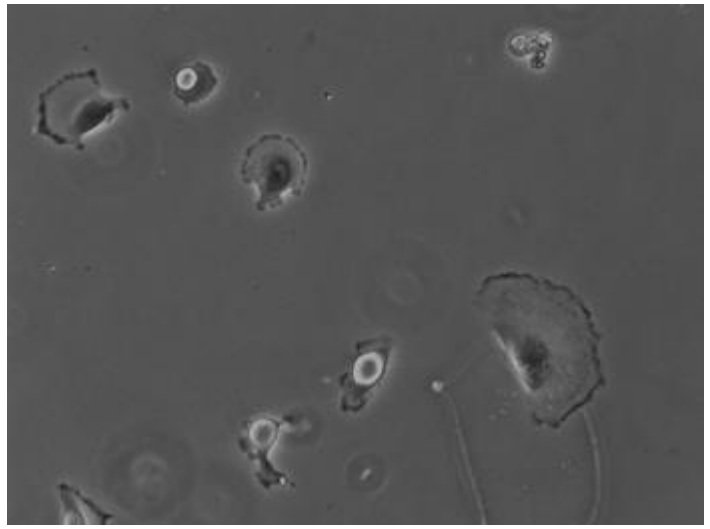
- **ImageJ** is a public software for processing and analyzing images, developed in Java.
- **ImageJ2** is a new version of ImageJ1 that focuses on multidimensional image data.
- **Fiji** is another upgrade that includes both ImageJ1, ImageJ2, and more useful features.

→ Fiji is the best option to use all features from ImageJ1 and ImageJ2.

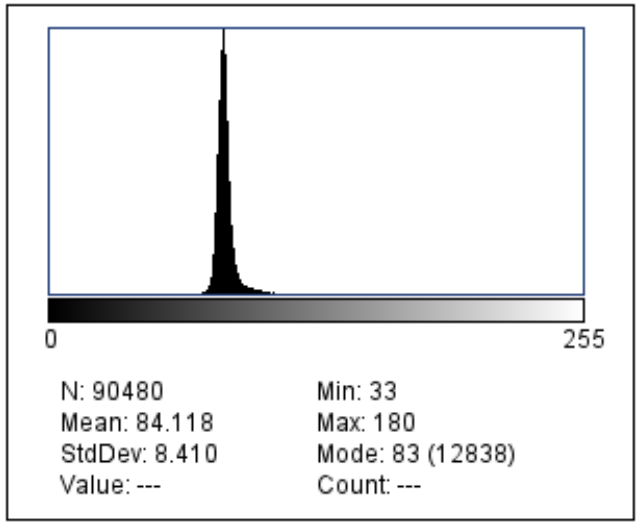
ImageJ, ImageJ2 and Fiji

- Fiji/ImageJ provide
  - User interface with useful functions.
  - Tasks: image processing, colocalization, deconvolution, registration, segmentation, tracking, visualization and more.
  - Plugins: extended plugins for more image analyzing tasks, i.e. DeepImageJ is an extension in Fiji/ImageJ.
  - Scripts and macros: reproducible workflow.
  - Community: very active forum for questions and issues.
- Easy-to-install and easy-to-use
- Good documentation

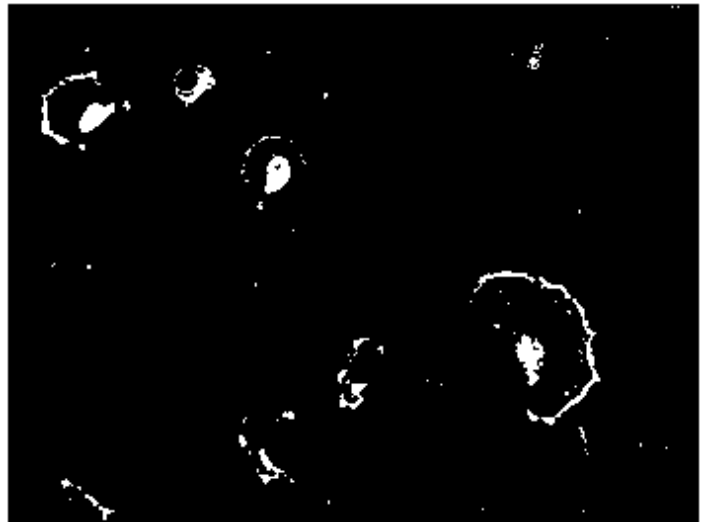
Examples



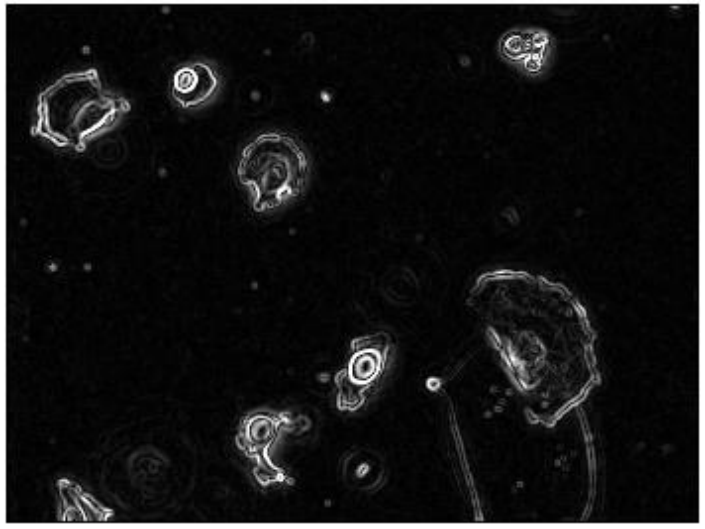
Original image (cell in glioblastoma phase)



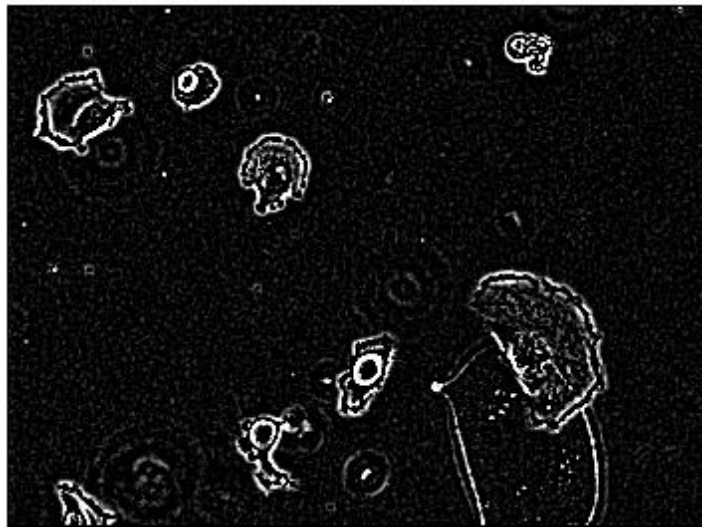
Histogram



Binary image (threshold 73)



Edges detector (Sobel's method)



Convolution operator

$$\begin{bmatrix} -1 & -1 & -1 & -1 & -1 \\ -1 & -1 & -1 & -1 & -1 \\ -1 & -1 & 24 & -1 & -1 \\ -1 & -1 & -1 & -1 & -1 \\ -1 & -1 & -1 & -1 & -1 \end{bmatrix}$$

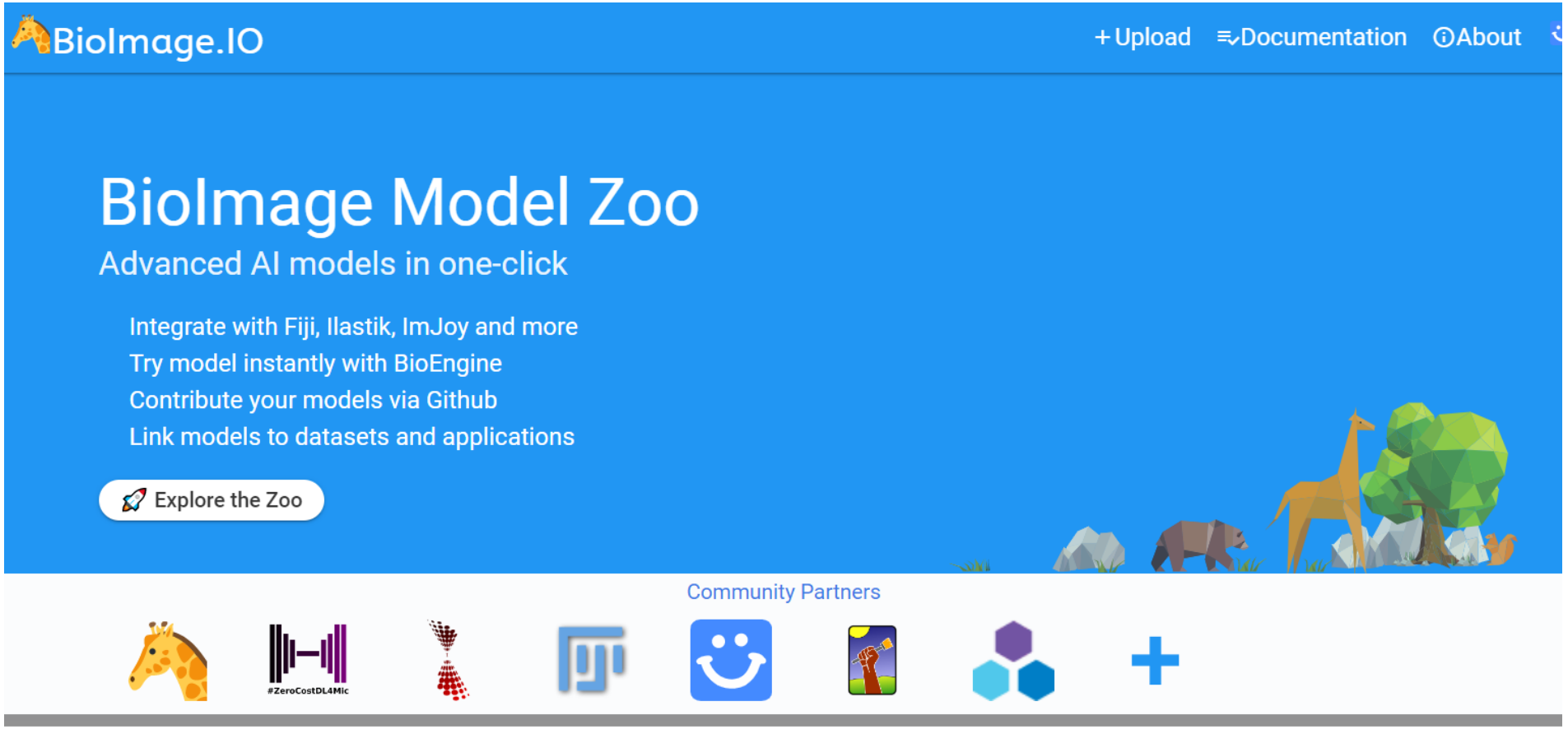
Kernel for the convolution operator

DeepImageJ

- A plugin in Fiji/ImageJ that supports pre-trained DL models for image data.
- An easy way for biomedical researchers to try and explore DL models without any prerequisites for programming skills or DL knowledge.
- Currently supports
  - TensorFlow 1 & 2, Pytorch
  - GPU support (for Linux machine only)
- Pros
  - Deep Learning models with GUI interactions
  - Can incorporate models from BioImage Model Zoo or trained by users.
- Cons
  - ~~One test image at a time~~

DeepImageJ: A user-friendly environment to run deep learning models in ImageJ, Nature Methods, 2021

BioImage Model Zoo

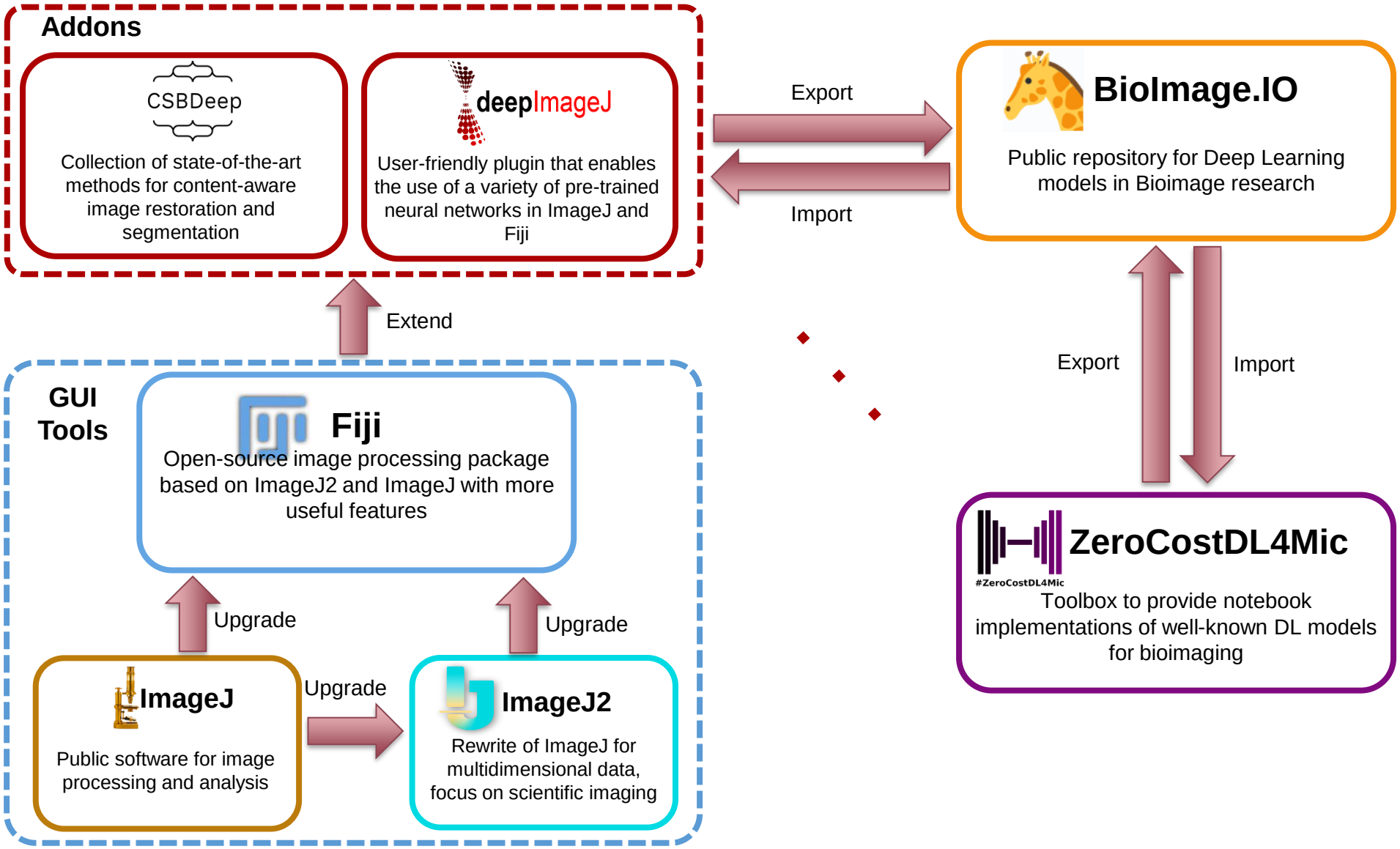


BioImage Model Zoo

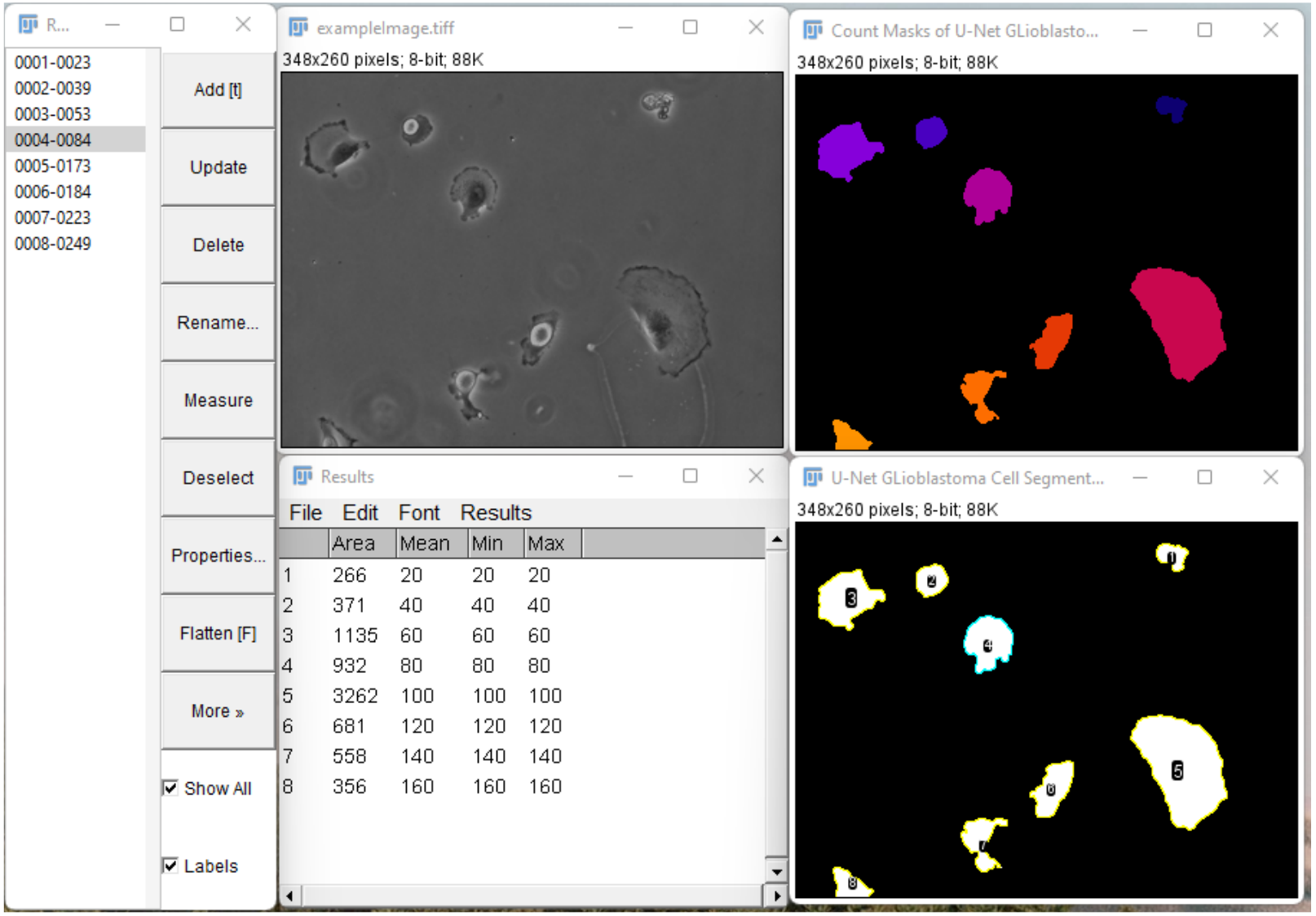
- A public repository for DL models in Bioimage research.
- Contribution from community
  - ImageJ/Fiji
  - DeepImageJ
  - ZeroCostDL4Mic
  - …
- BioImage Model Zoo provides
  - Pre-trained model
  - Training data
  - Description for each model: related paper, model's architecture, …

https://bioimage.io/#/

Software Tools



DeepImageJ



Model: [Glioblastoma Phase Contrast Cell Segmentation (2D U-NET)]

Pre-trained from BioImage Model Zoo

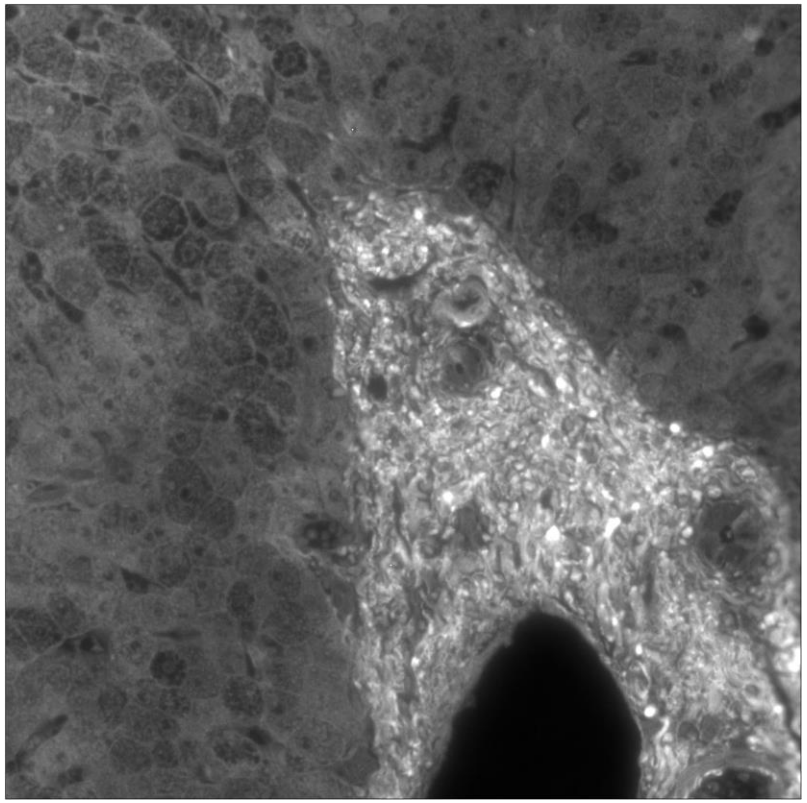
Dataset

- Microscopy modality: 2D Phase contrast
- 24 images for training, 10 images for testing
- Cell type: Glioblastoma-astrocytoma (U373)
- Source: Cell tracking challenge

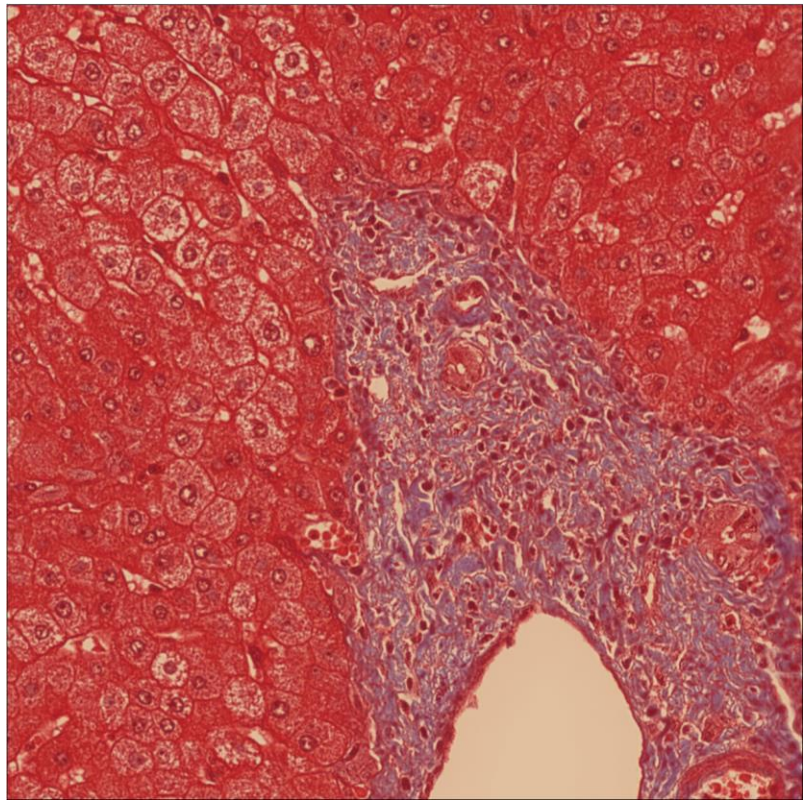
Training procedure

- 10 epochs, 500 steps per epoch
- Learning rate: 0.0001

DeepImageJ



Original Image



Output Image

Model: Masson's Trichrome Virtual Staining (GAN)

Pre-trained from BioImage Model Zoo
Training dataset: N/A.
Dataset is collected by authors.
Training procedure

- 3 epochs
- Learning rate: 0.0001 (generator) and 0.00001 (discriminator)

Exercise 2 - **FIJI**

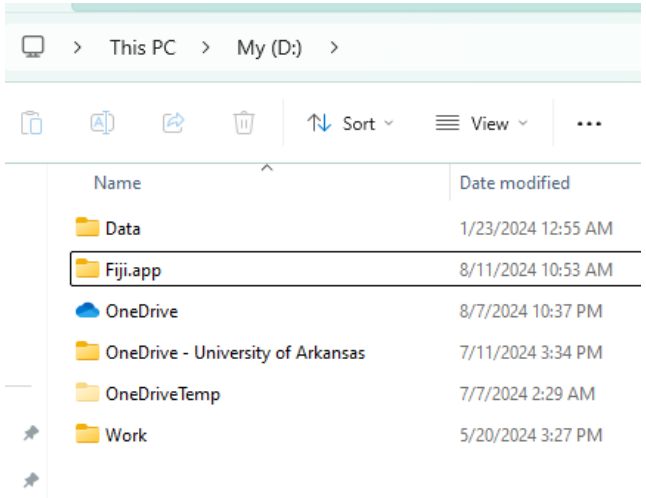
1. Download Fiji here:
https://imagej.net/software/fiji/downloads

Fiji Downloads

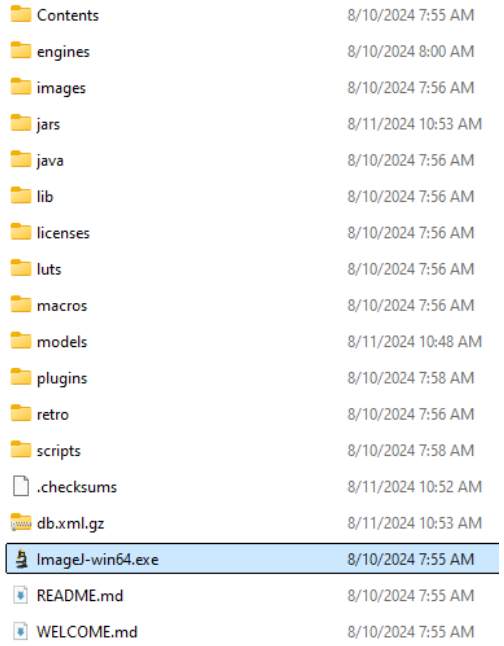
Fiji is a distribution of ImageJ which includes many useful plugins contributed by the community.

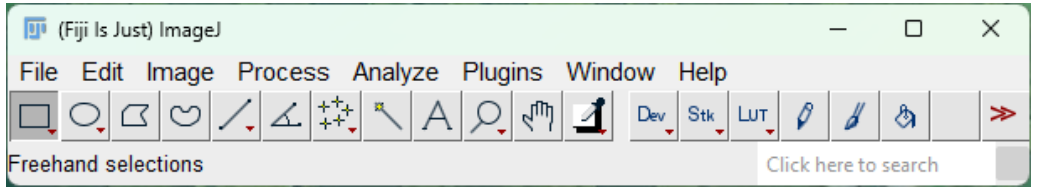
~ Download Fiji for your OS ~		
	Windows 64-bit	imagej.net (USA), micron.ox.ac.uk (European mirror)
	Windows 32-bit	imagej.net (USA), micron.ox.ac.uk (European mirror)
	macOS (x86_64)	imagej.net (USA), micron.ox.ac.uk (European mirror)
	Linux (64-bit)	imagej.net (USA), micron.ox.ac.uk (European mirror)
	No JRE	imagej.net (USA), micron.ox.ac.uk (European mirror)

2. Extract Fiji



3. Run Fiji

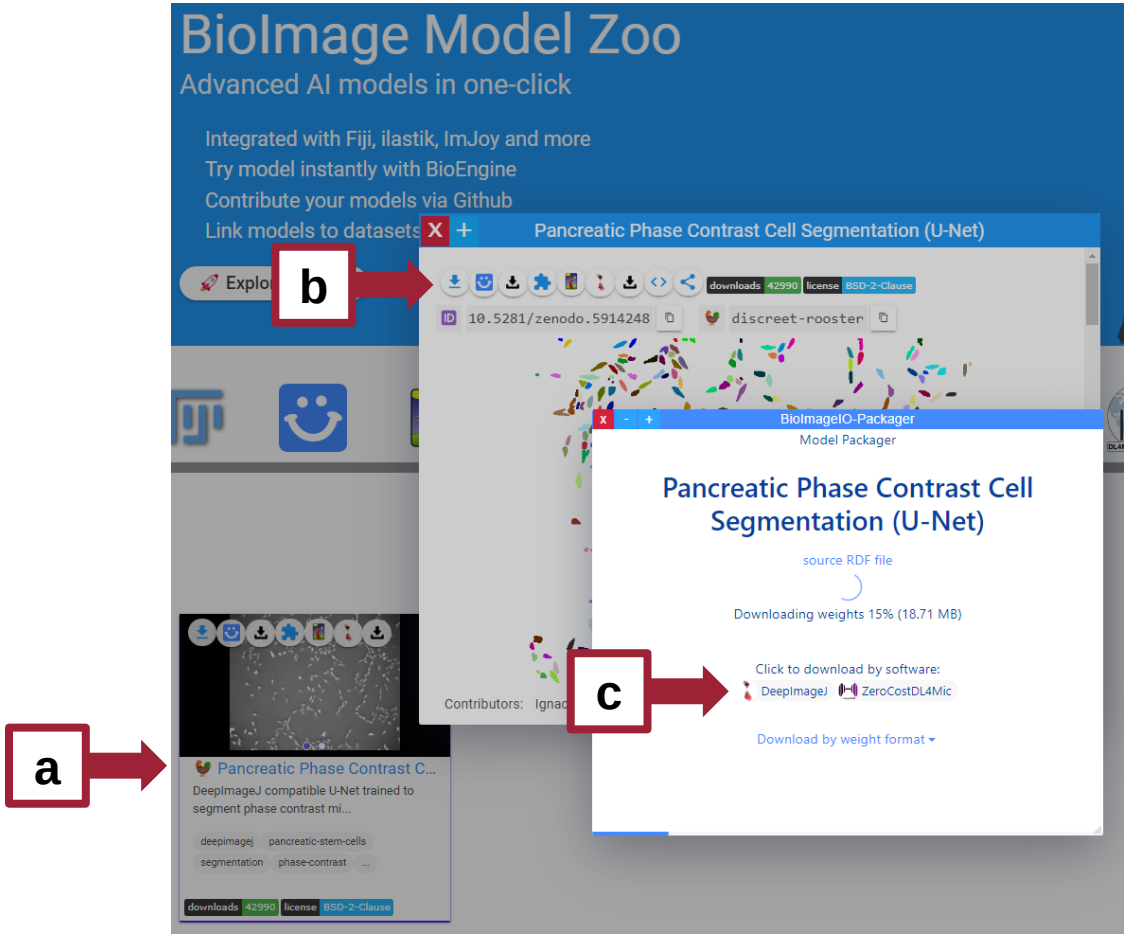




Exercise 2 - **FIJI**

4. Download a pretrained model from Bioimage.io

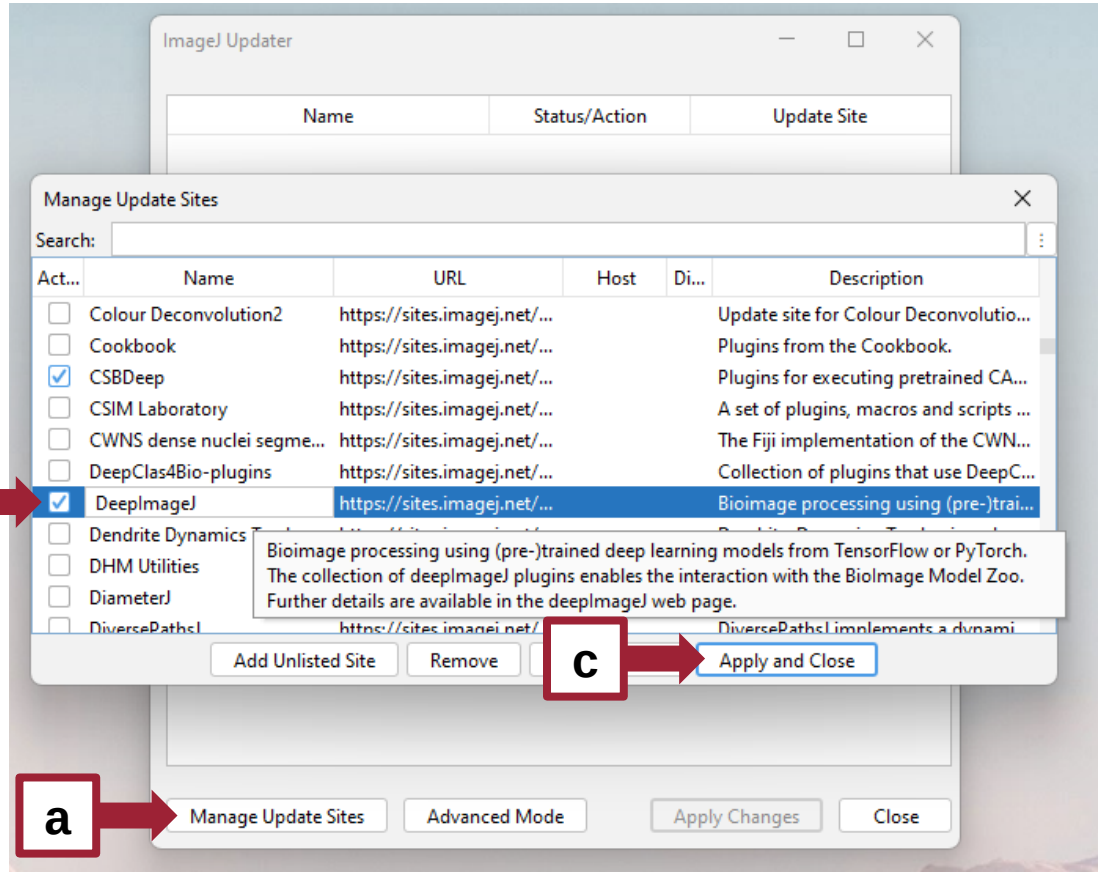
Download the “Pancreatic Phase…” model zip file for DeepImageJ.



Exercise 2 - **FIJI**

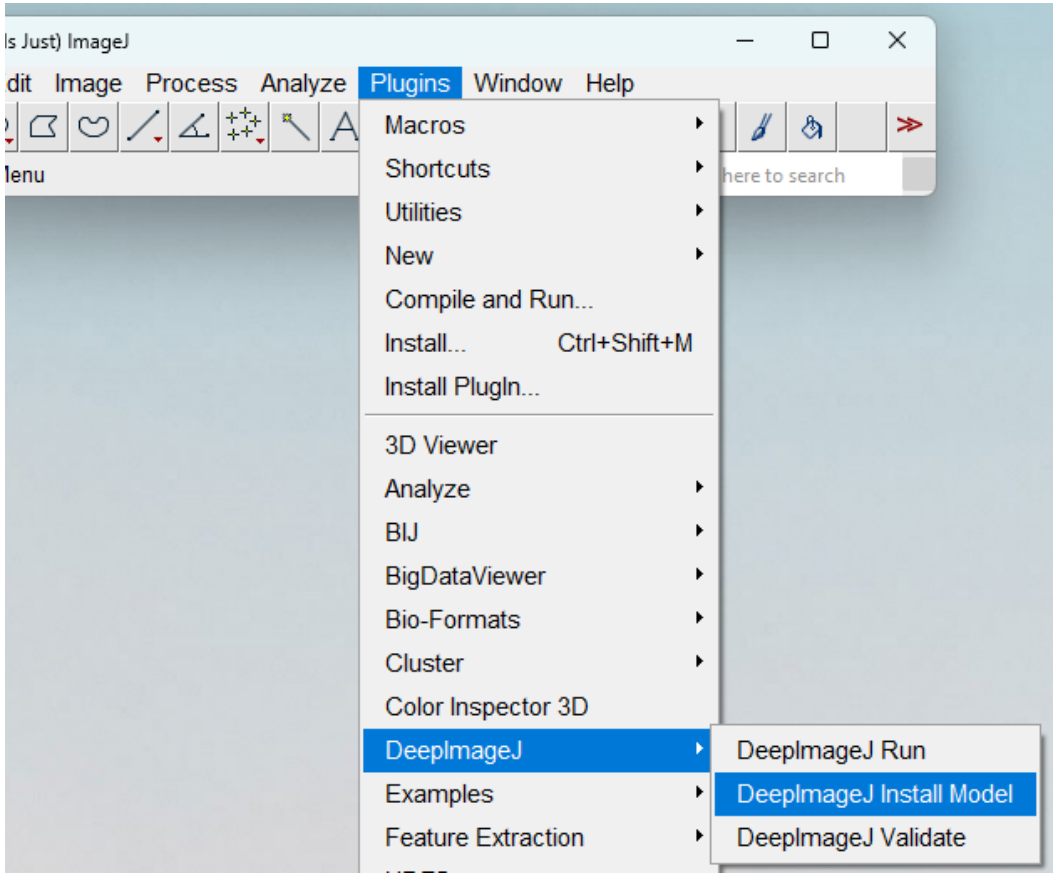
5. Install DeepImageJ Plugin

a. Help > Update. Click on Manage Update Sites.
b. Check DeepImageJ and click on Apply and Close.
c. Restart Fiji.



6. Install a model in DeepImageJ

a. Plugins > DeepImageJ > Install Model



Exercise 2 - **FIJI**

6. Install a model in DeepImageJ (contd.)

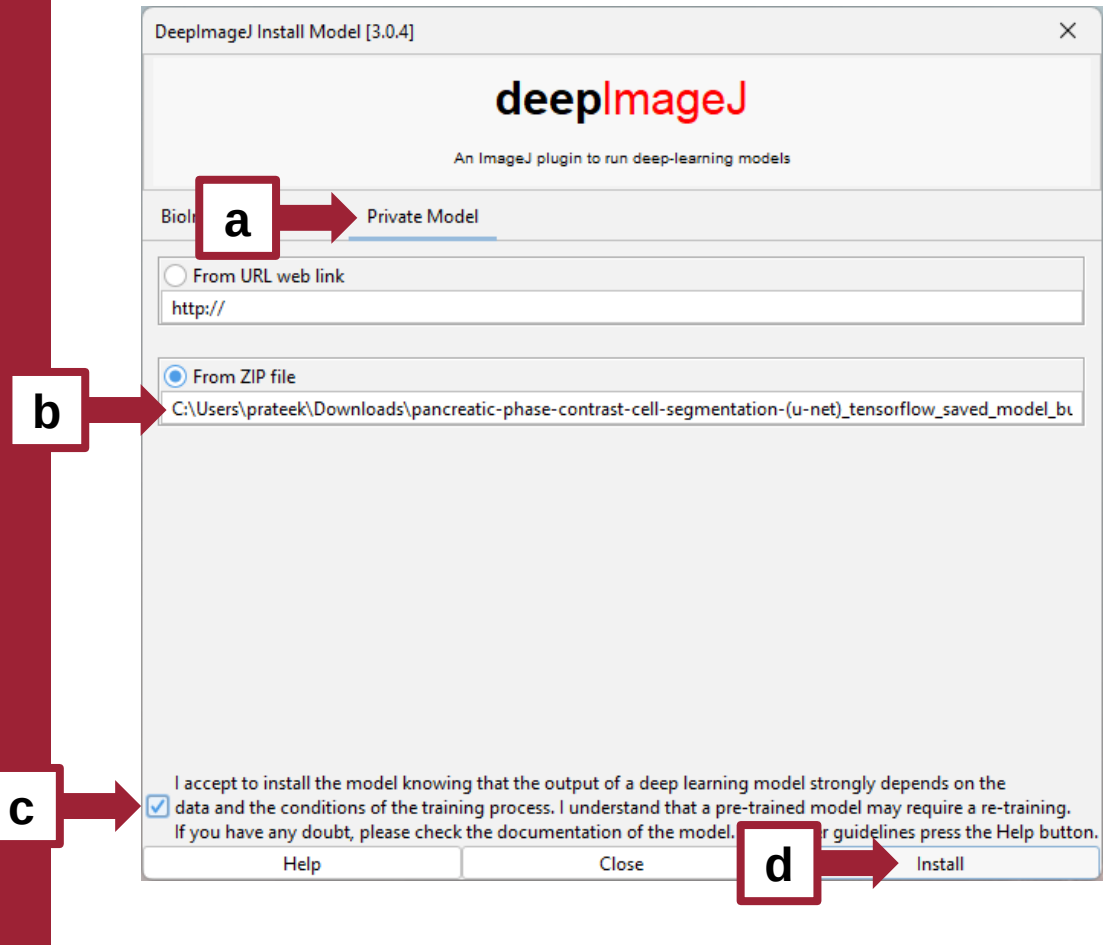
b. Select “Private Model” tab, “From ZIP file”, enter path

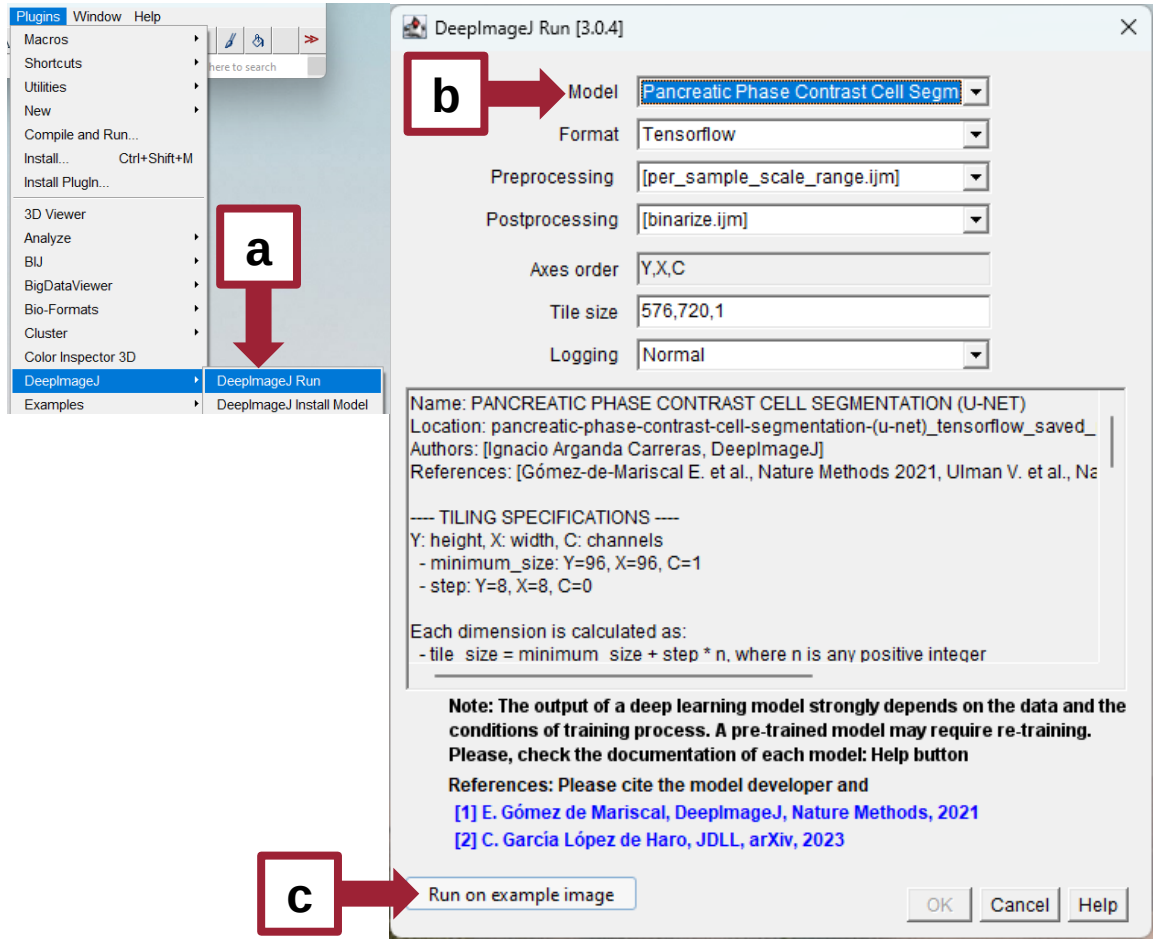
c. Check “I accept”, click Install. Restart Fiji.

7. Run model on example image

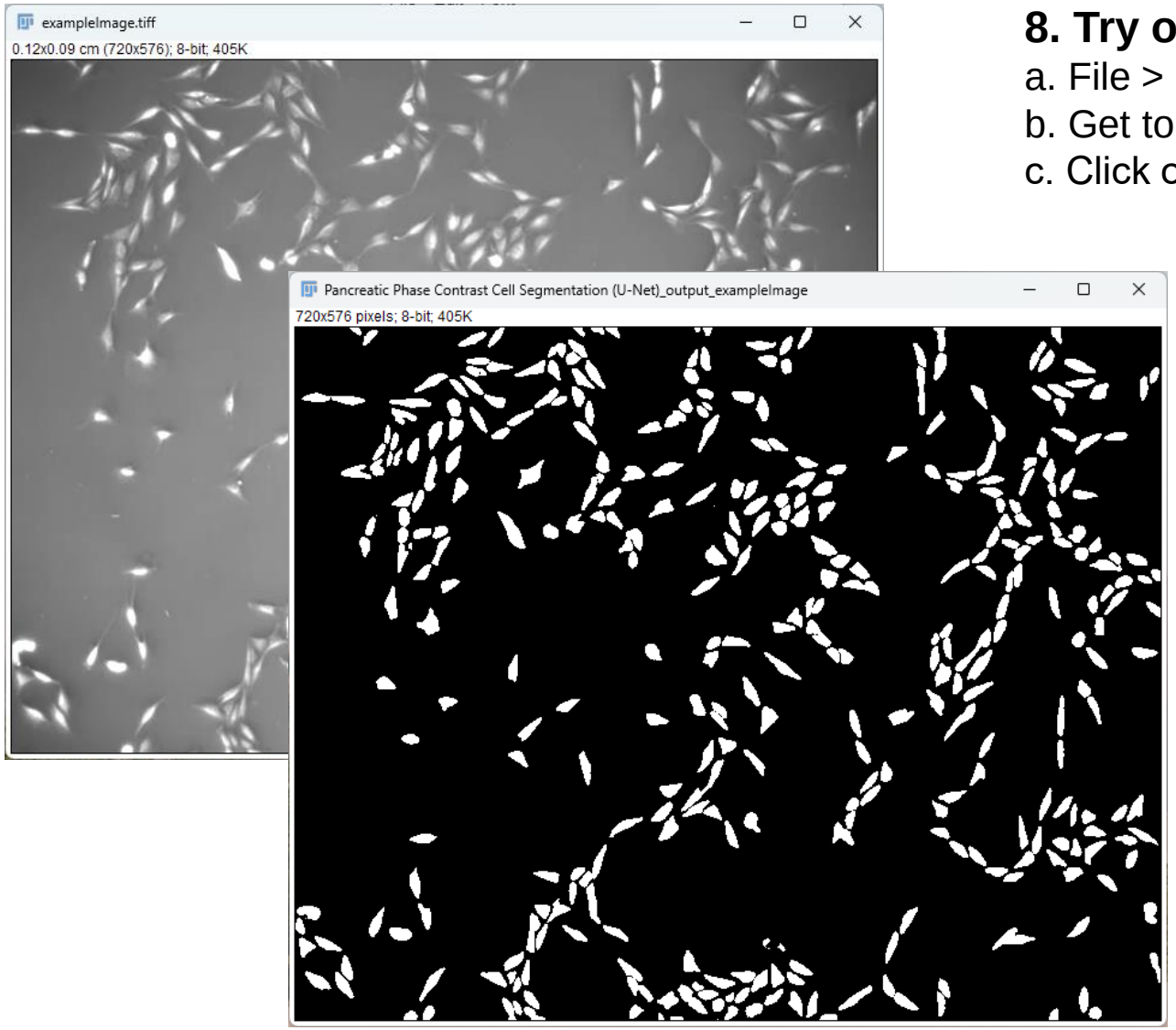
a. Plugins > DeepImageJ > “…Run”. Wait for loading.

b. Select your model. Click “Run on example image”.





Exercise 2 - **FIJI**

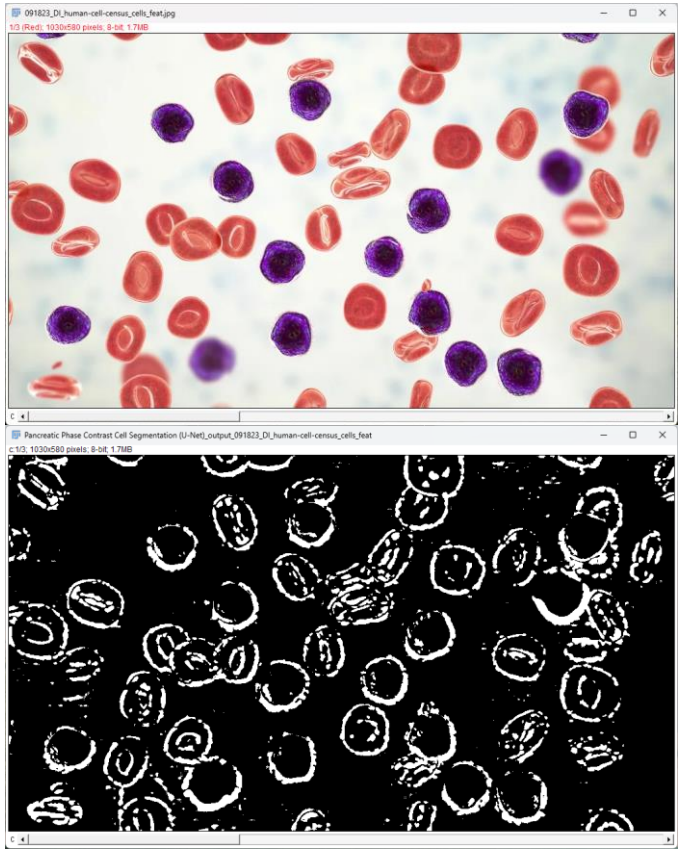


8. Try on your own image (or images/stack)

a. File > Open. Select image(s)

b. Get to the run model dialogue box as described in 7.

c. Click on OK, instead of “Run on example image”

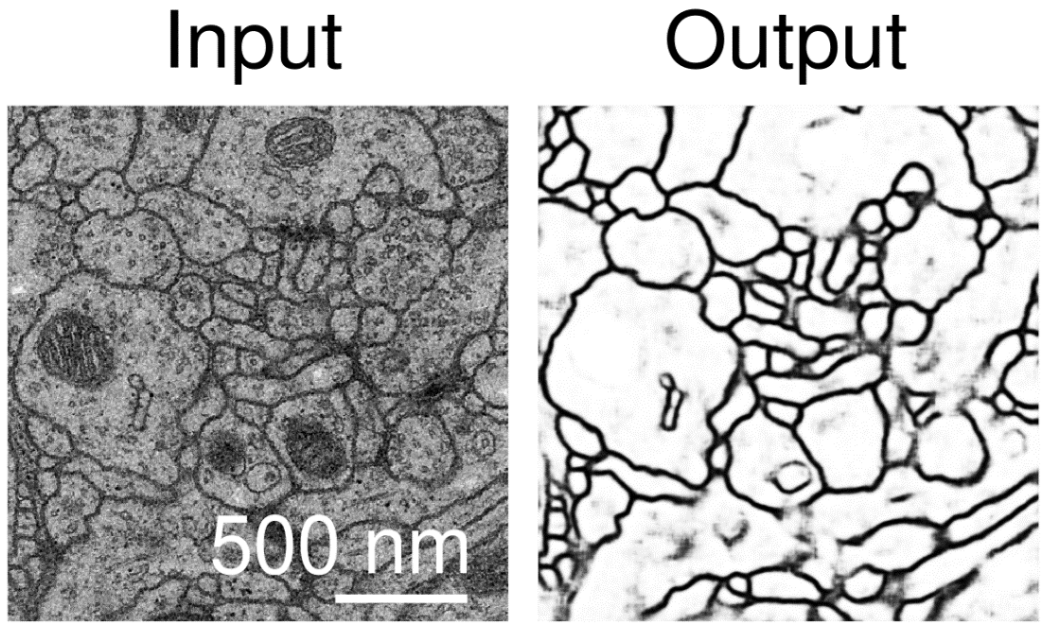


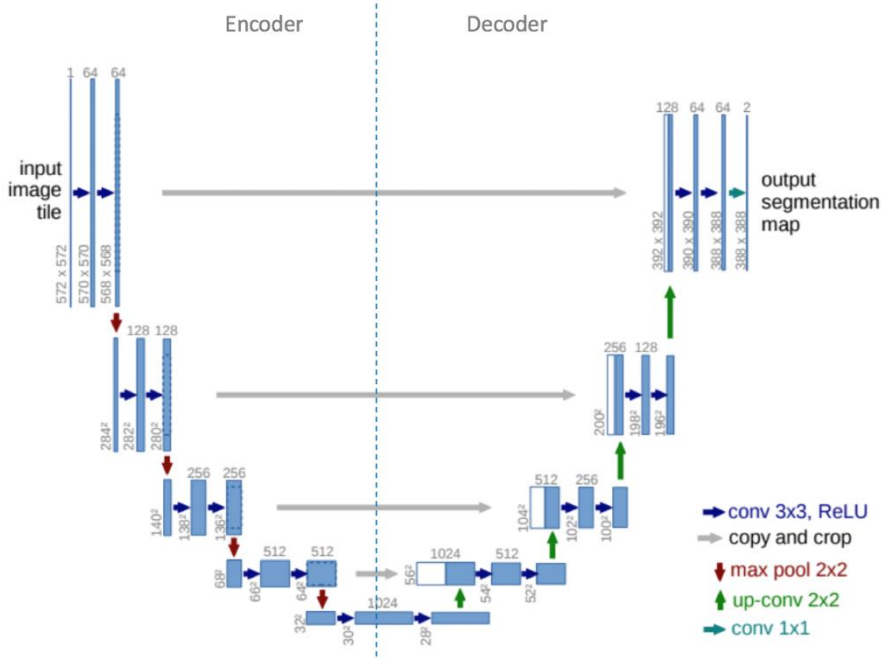
UNIVERSITY OF ARKANSAS

Popular Deep Learning based Bioimaging Models

U-Net (2D)

- An encoder-decoder network architecture originally used for image segmentation.
- The first half of the U-Net architecture applies convolution blocks followed by a maxpool downsampling to encode the input image into feature representations at different levels.
- The other half consists of upsampling and concatenation followed by regular convolution operations and project lower resolution features onto the higher resolution to get a dense classification.

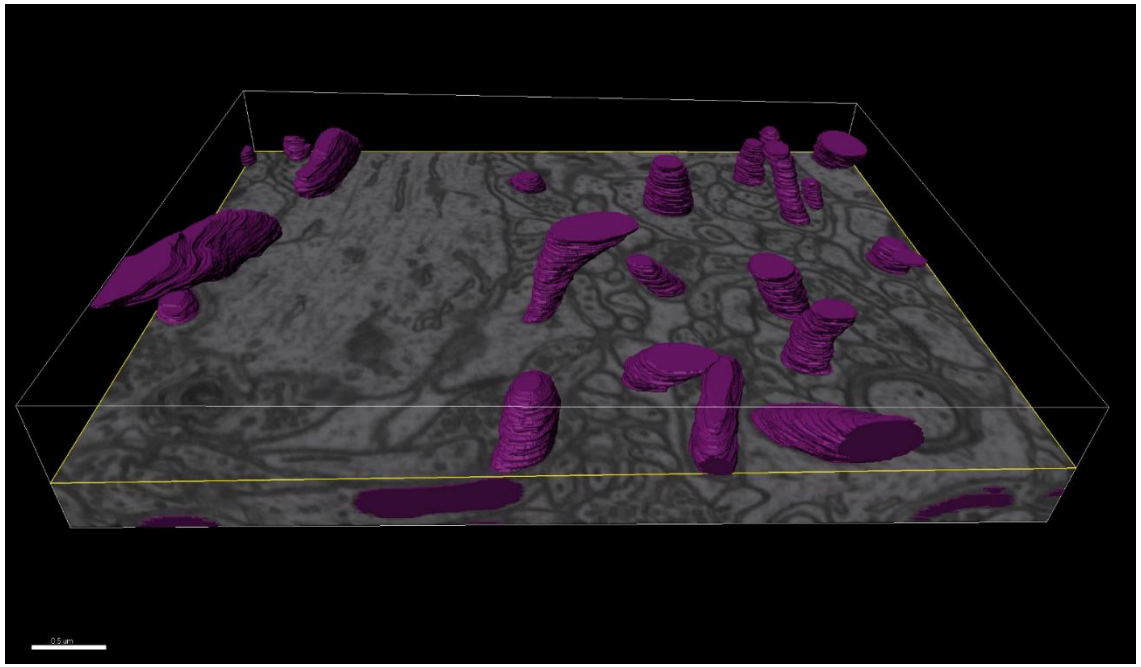




U-Net: deep learning for cell counting, detection, and morphometry, Nature Method, 2018
https://arxiv.org/abs/1505.04597

U-Net (3D)

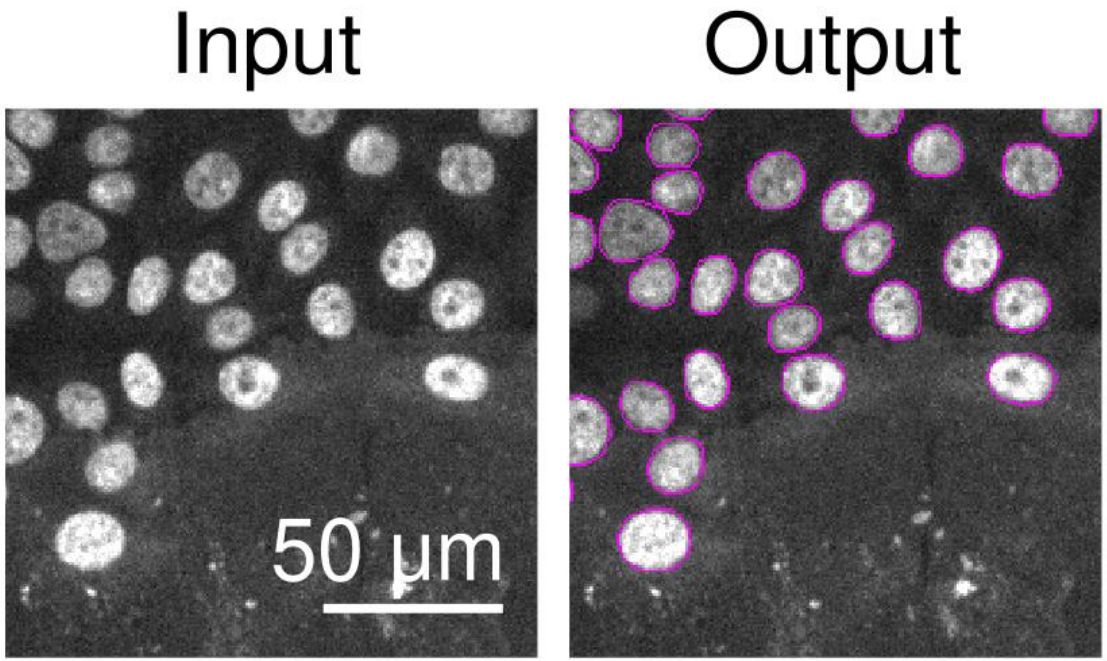
- A network for volumetric segmentation that learns from sparsely annotated volumetric images.
- Extends the U-Net architecture by replacing all 2D operations with 3D counterparts.



3D U-Net: Learning Dense Volumetric Segmentation from Sparse Annotation, MICCAI 2016
https://arxiv.org/abs/1505.04597

StarDist (2D)

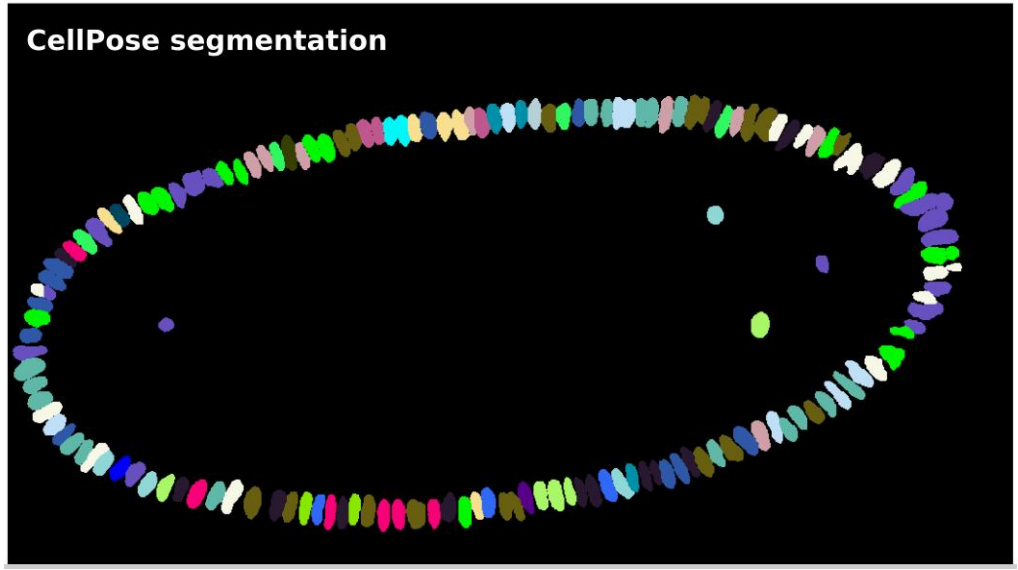
- A DL method that can be used to segment cell nuclei from bioimages.
- Uses a shape representation based on star-convex polygons for nuclei in an image to predict the presence and the shape of these nuclei.
- Improves the approaches of using bounding boxes and does not need shape refinement.

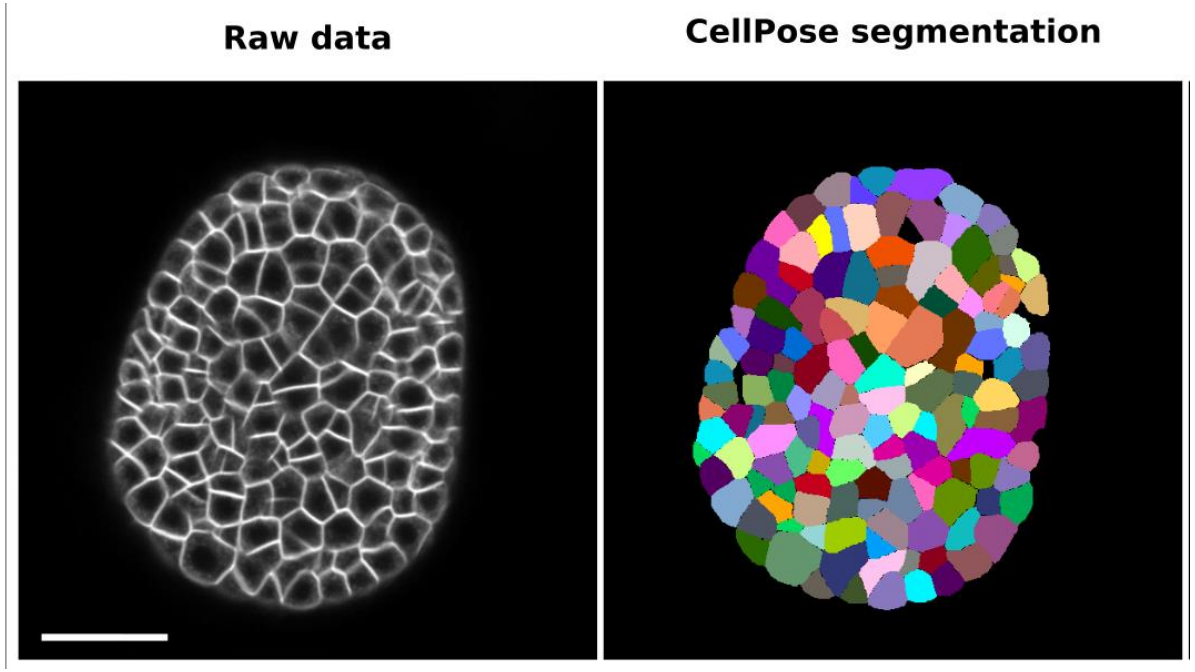


Cell Detection with Star-convex Polygons, MICCAI 2018. https://arxiv.org/abs/1806.03535

Cellpose

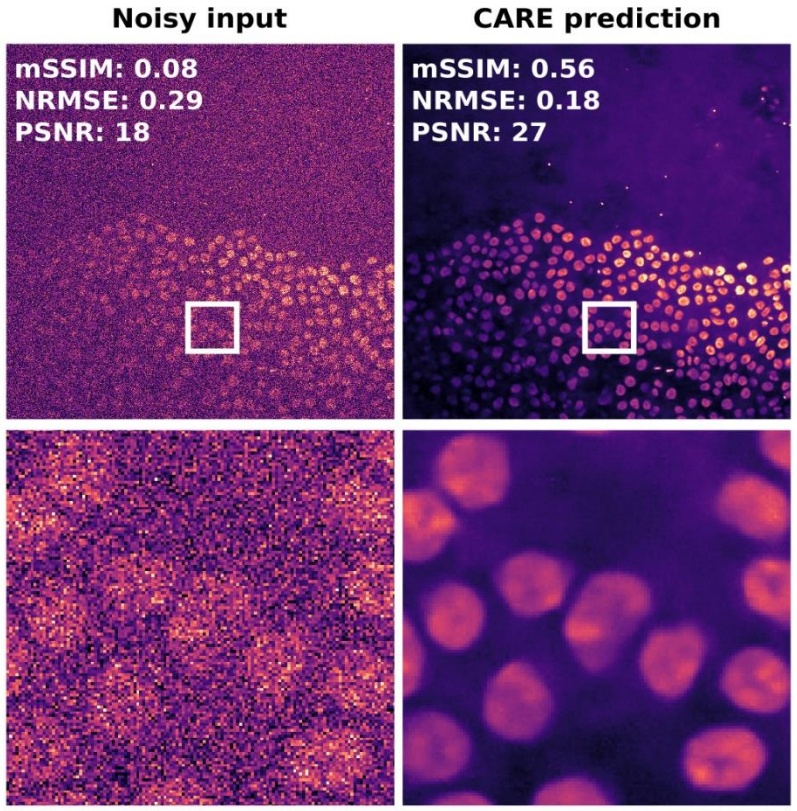
- Segment cell and/or nuclei from a wide range of image types and does not require model pretraining or parameter adjustments.
- Cellpose: a generalist algorithm for cellular segmentation, Nature Methods, 2020 https://www.nature.com/articles/s41592-020-01018-x

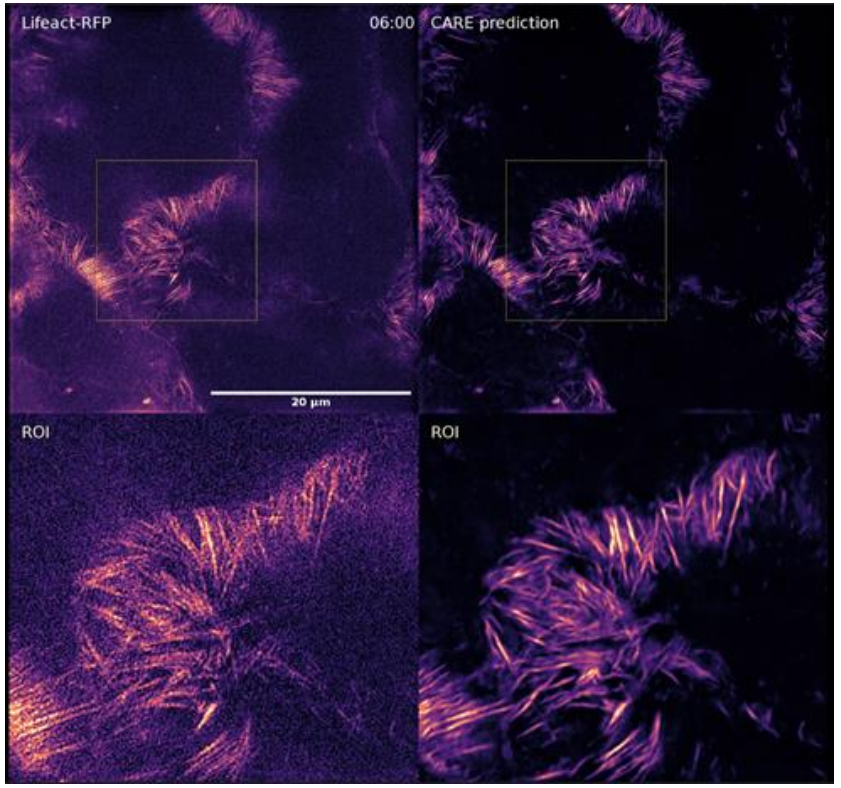




CARE

- Content-aware image restoration with matching pairs.
- Uses a U-Net network architecture.

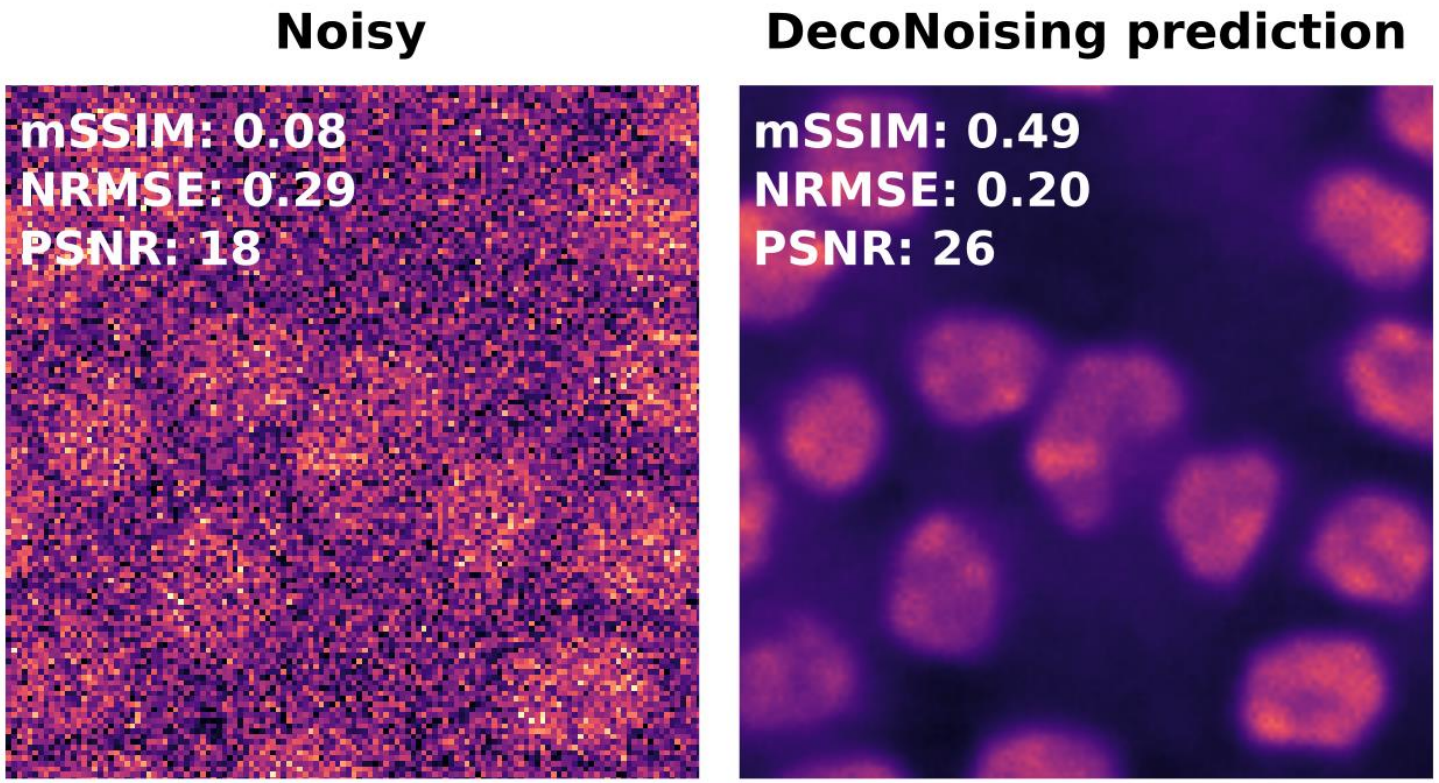




Content-aware image restoration: pushing the limits of fluorescence microscopy, Nature Methods 2018
https://www.nature.com/articles/s41592-018-0216-7

DecoNoising (2D)

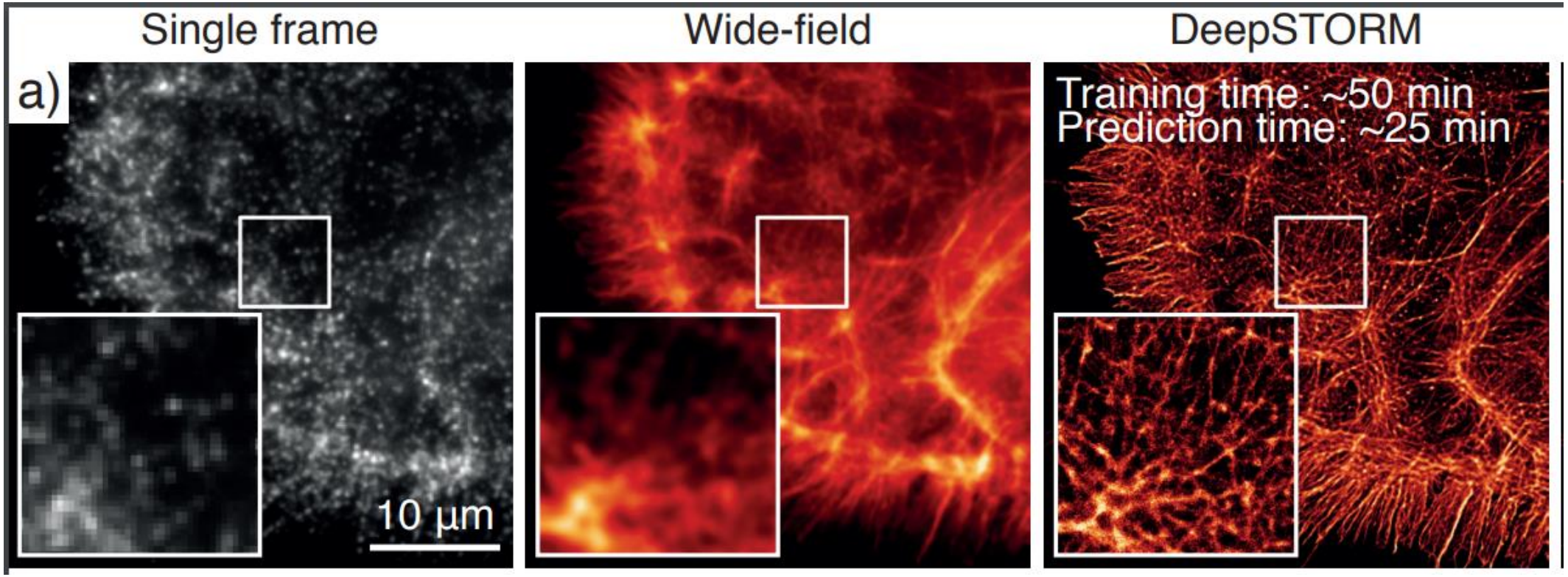
- A self-supervised denoising method that does not need paired training data.
- Applies the point spread function to denoise and improve the predictions.



Improving Blind Spot Denoising for Microscopy, ECCV Workshops 2020

Deep-STORM (2D)

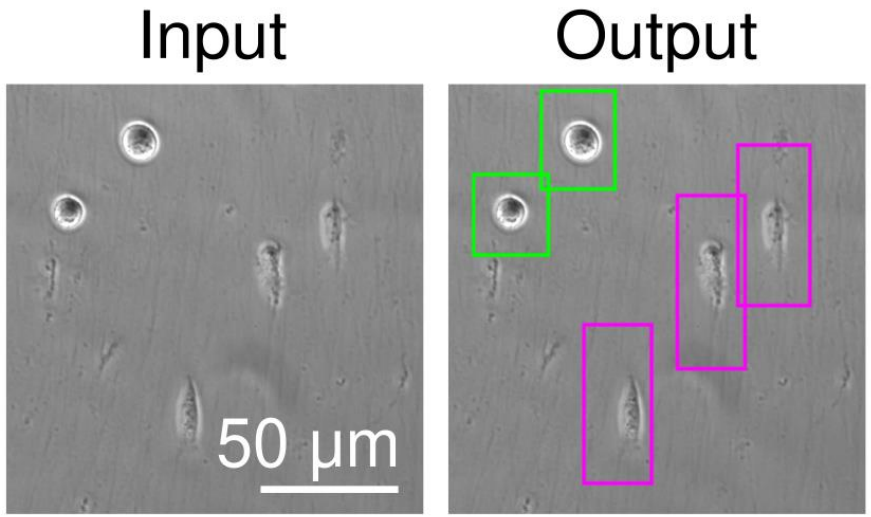
– A deep neural network capable of obtaining super-resolution images from fluorescent molecules used for localization microscopy.

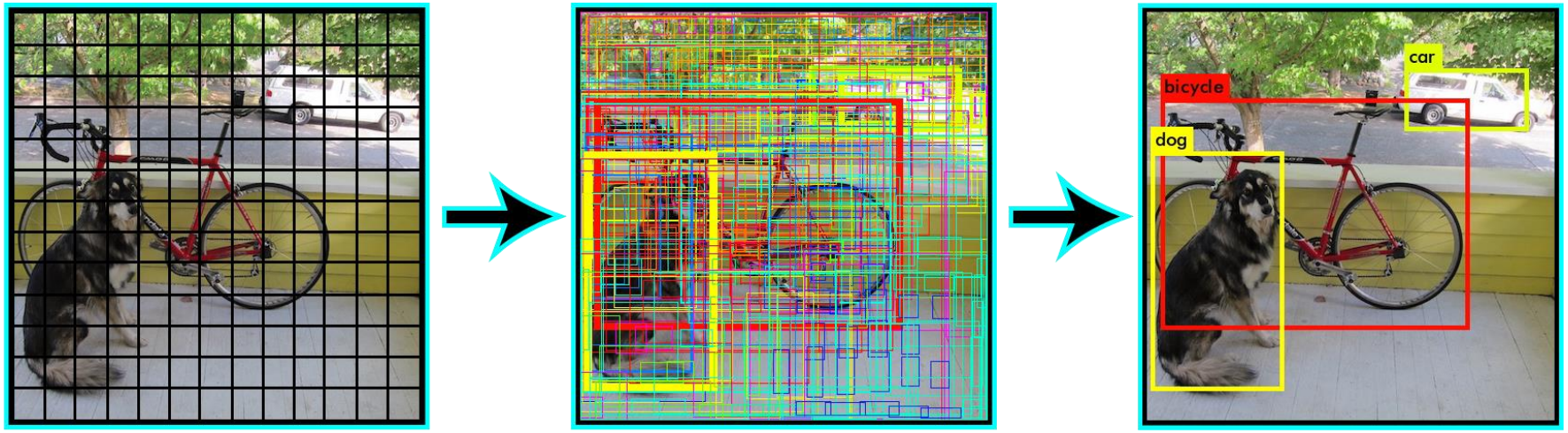


Deep-STORM: super-resolution single-molecule microscopy by deep learning, Optica 2018

YOLO

- A real time object detection system that can detect over 9000 object categories.
- The network divides the image into regions and predicts bounding boxes and prob. for each region.

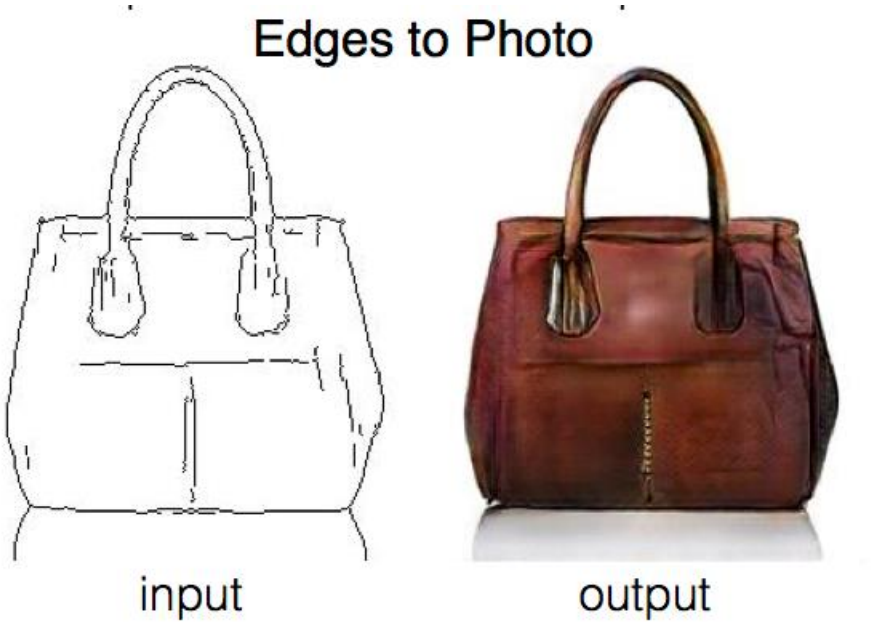




YOLO9000: Better, Faster, Stronger, 2016. https://arxiv.org/abs/1612.08242

Pix2Pix

- A DL method allowing image-to-image translation from one image domain type to another image domain type.
- Requires paired images in training.



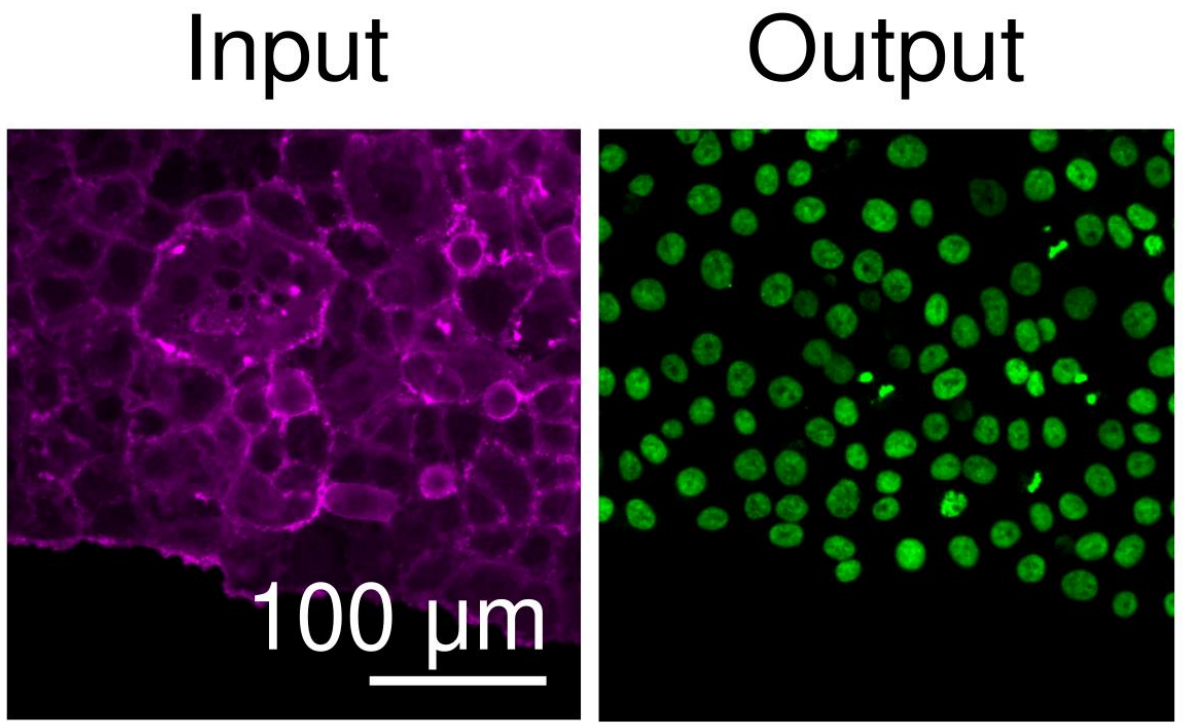
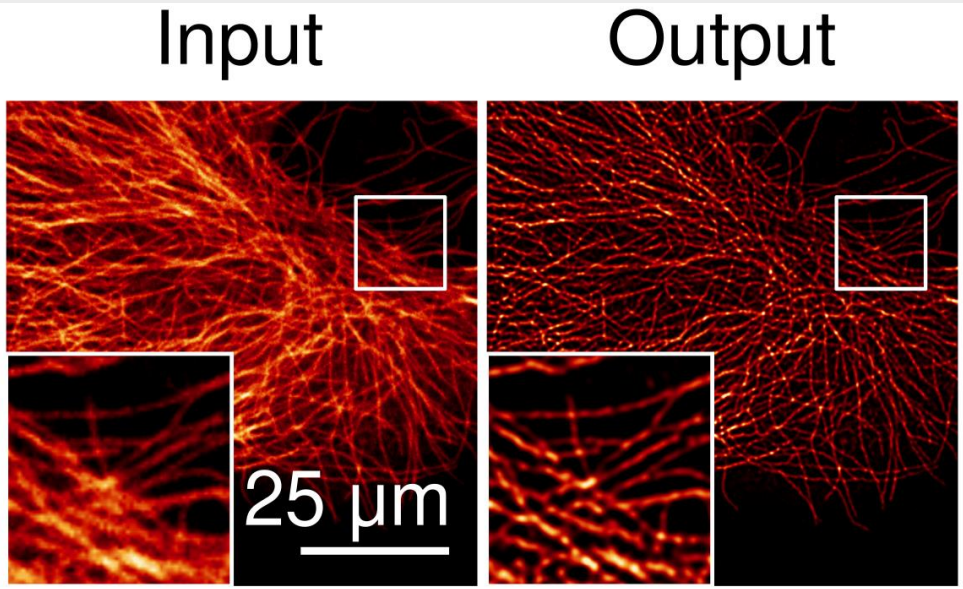


Image-to-Image Translation with Conditional Adversarial Nets, CVPR 2017.

CycleGAN

- Introduces the cycle consistency loss.
- Does not require paired examples in the training.
- Achieves great success in collection style transfer, object transfiguration, season transfer, photo enhancement, etc.

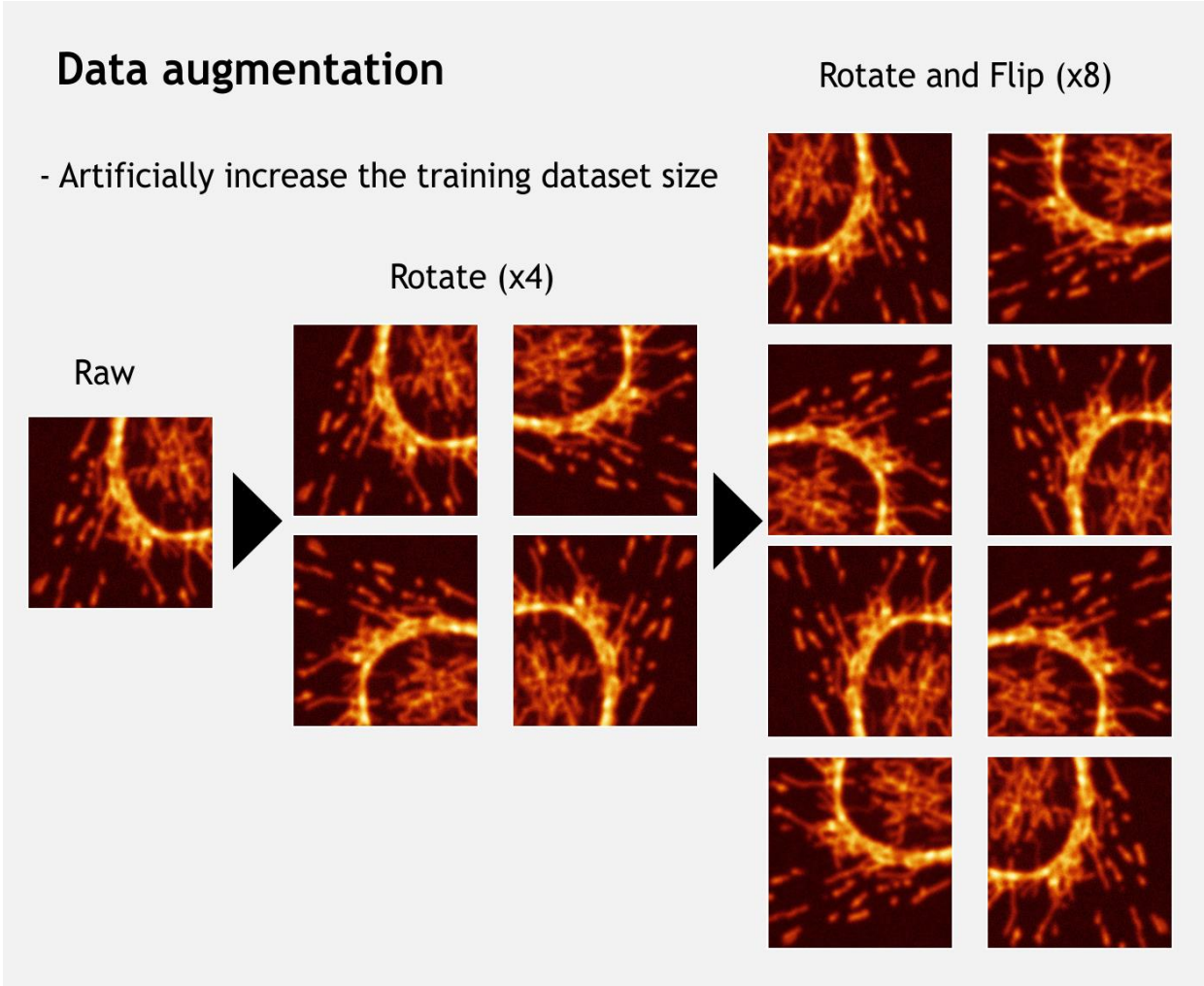


winter Yosemite → summer Yosemite

Unpaired Image-to-Image Translation using Cycle-Consistent Adversarial Networks, 2017.
https://arxiv.org/abs/1703.10593

Augmentor

- Data augmentation can improve training progress by amplifying differences in the dataset.
- Useful if the available dataset is small.
- Supports z-stack augmentation and randomized elastic distortions.



Biomedical image augmentation using Augmentor, Bioinformatics, 2019

Exercise 3 – **U-Net**

Please follow the instructions in the Jupyter Notebook.

UNIVERSITY OF ARKANSAS

Open source Jupyter Notebooks **ZeroCostDL4Mic**

ZeroCostDL4Mic

- A toolbox to provide notebook implementations of well-known models and deploy them in Google Colab.
- Two use cases
  - Online: No need to handle Python environments or GPU setup when using Google Colab. However, the resources are limited with free subscription.
  - Offline: Using local machines/servers to run the notebook.
- Tasks: image segmentation, denoising, restoration, object detection
- Pros
  - Users can run it online with Colab or offline with local machines
  - Users have more control in defining the model, train/test procedure, dataset
  - ZeroCostDL4Mic can help export to BioImage Model Zoo format
- Cons
  - Users need to have programming skills
  - This is a code-based toolbox, not an interactive software

ZeroCostDL4Mic

Segmentation networks

Network	Paper(s)	Tasks	Status	Last test	Link to example training and test dataset	Direct link to the notebook in Colab
U-Net (2D)	here and here	Binary segmentation	Fully supported	⚠ broken (no GPU) (GJ)	here	Open in Colab
U-Net (3D)	here	Binary segmentation	Fully supported	⚠ broken (no GPU) (GJ)	EPFL dataset	Open in Colab
U-Net (2D) multilabel	here and here	Semantic segmentation	Under beta-testing	⚠ broken (no GPU) (GJ)	here	Open in Colab
DenoiSeg	here	Joint denoising and binary segmentation	Fully supported	⚠ broken (no GPU) (GJ)	Available soon	Open in Colab
StarDist (2D)	here and here	Instance segmentation	Fully supported	08/10/22 ✅ working (GJ)	here	Open in Colab
StarDist (3D)	here and here	Instance segmentation	Fully supported	07/10/22 ✅ working (GJ)	from Stardist github	Open in Colab
Cellpose (2D and 3D)	here	Instance segmentation (Cells or Nuclei)	Fully supported	08/10/22 ✅ working (GJ)	Coming soon!	Open in Colab
SplineDist (2D)	here	Instance segmentation	Fully supported	07/10/22 ✅ working (GJ)	here	Open in Colab
EmbedSeg (2D)	here	Instance segmentation	Under beta-testing	⚠ broken by recent updates	here	Open in Colab
MaskRCNN (2D)	here	Instance segmentation	Under beta-testing		Coming soon!	Open in Colab
Interactive Segmentation - Kaibu (2D)	here	Interactive instance segmentation	Under beta-testing		Coming soon!	Open in Colab

Denoising and image restoration networks

Network	Paper(s)	Tasks	Status	Last test	Link to example training and test dataset	Direct link to the notebook in Colab
Noise2Void (2D)	here	Self-supervised denoising	Fully supported	06/10/22 ✅ working (EGM)	here or here	Open in Colab
Noise2Void (3D)	here	Self-supervised denoising	Fully supported	07/10/22 ✅ working (GJ)	here	Open in Colab
CARE (2D)	here	Supervised denoising	Fully supported	07/10/22 ✅ working (GJ)	here or here	Open in Colab
CARE (3D)	here	Supervised denoising	Fully supported	07/10/22 ✅ working (GJ)	here	Open in Colab
3D-RCAN	here	Supervised denoising	Under beta-testing	⚠ broken (no GPU)	here	Open in Colab
DecoNoising (2D)	here	Self-supervised denoising	Under beta-testing	07/10/22 ✅ working (GJ)	here or here	Open in Colab

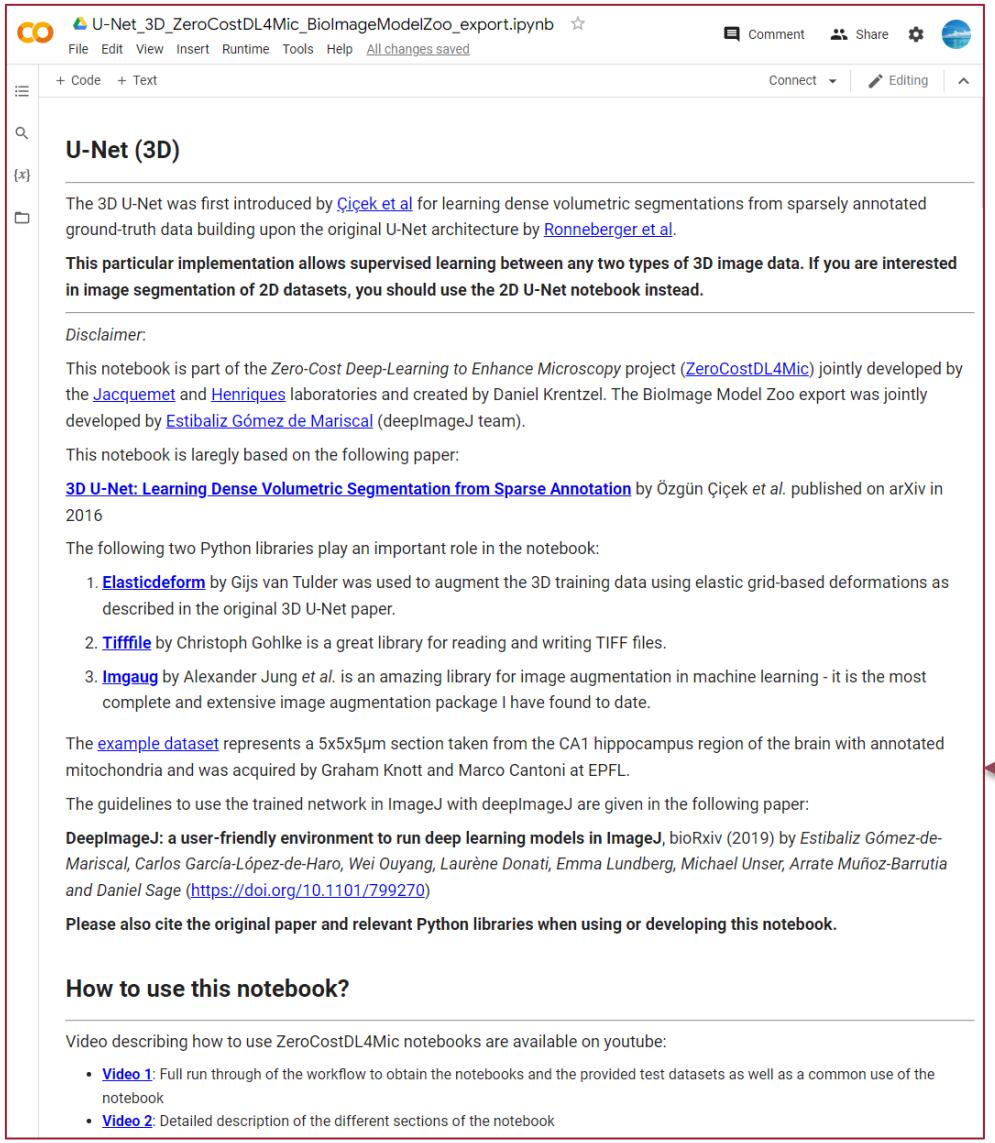
Super-resolution microscopy networks

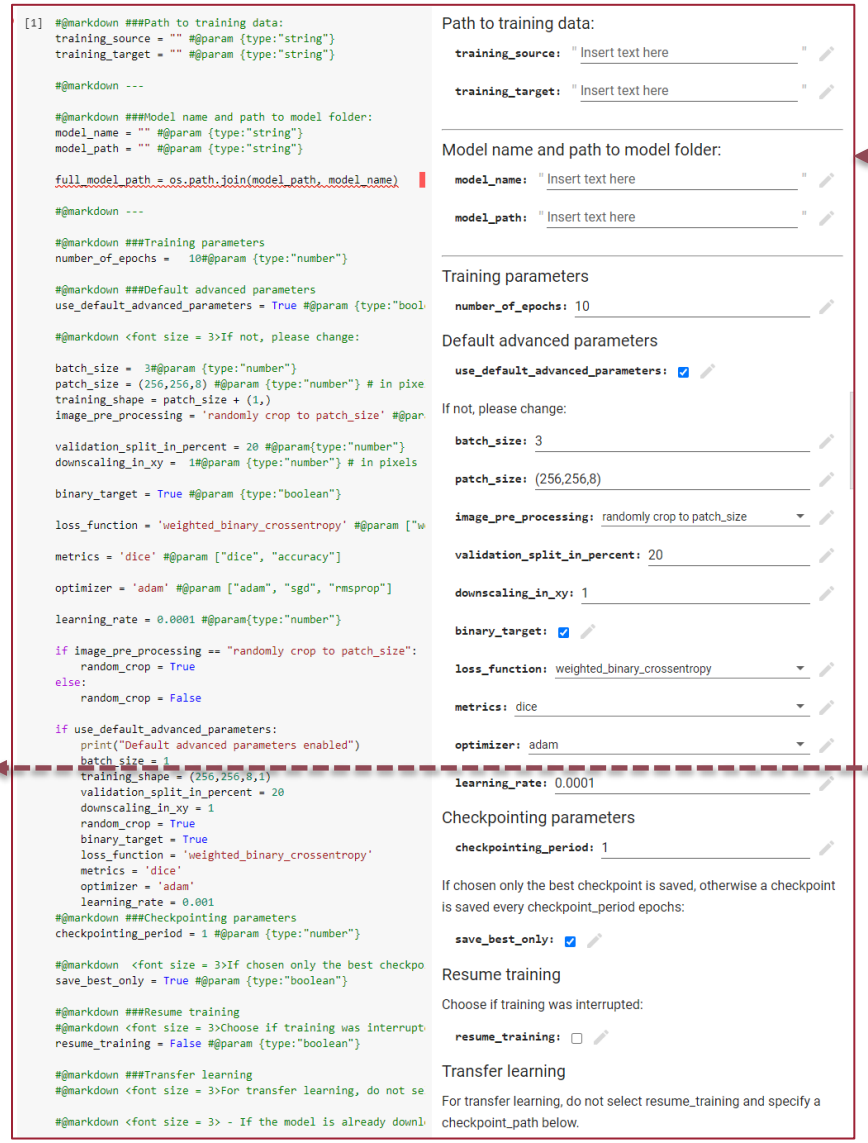
Network	Paper(s)	Tasks	Status	Last test	Link to example training and test dataset	Direct link to the notebook in Colab
Deep-STORM	here	Single Molecule Localization Microscopy (SMLM) image reconstruction from high-density emitter data	Fully supported	08/10/22 ✅ working (GJ)	Training data simulated in the notebook or available from here	Open in Colab
DFCAN	here	image upsampling	Under beta-testing	08/10/22 ✅ working (GJ)	here	Open in Colab
WGAN	here	image upsampling	Under beta-testing	22/09/22 ✅ working (IvanHidalgo & EGM)	here	Open in Colab

Start notebook with Google Colab

Details about model and data

ZeroCostDL4Mic





Either **input parameters to input form** or **modify the code**

ZeroCostDL4 Mic provides very **detailed instruction**

Segmentation

Network	Paper(s)	Tasks	Status	Last test	Link to example training and test dataset	Direct link to the notebook in Colab
U-Net (2D)	here and here	Binary segmentation	Fully supported	⚠ broken (no GPU) (GJ)	here	Open in Colab
U-Net (3D)	here	Binary segmentation	Fully supported	⚠ broken (no GPU) (GJ)	EPFL dataset	Open in Colab
U-Net (2D) multilabel	here and here	Semantic segmentation	Under beta-testing	⚠ broken (no GPU) (GJ)	here	Open in Colab
DenoiSeg	here	Joint denoising and binary segmentation	Fully supported	⚠ broken (no GPU) (GJ)	Available soon	Open in Colab
StarDist (2D)	here and here	Instance segmentation	Fully supported	08/10/22 ✅ working (GJ)	here	Open in Colab
StarDist (3D)	here and here	Instance segmentation	Fully supported	07/10/22 ✅ working (GJ)	from Stardist github	Open in Colab

https://github.com/HenriquesLab/ZeroCostDL4Mic/wiki

Segmentation (Cont'd)

Cellpose (2D and 3D)	here	Instance segmentation (Cells or Nuclei)	Fully supported	08/10/22 ✅ working (GJ)	Coming soon!	Open in Colab
SplineDist (2D)	here	Instance segmentation	Fully supported	07/10/22 ✅ working (GJ)	here	Open in Colab
EmbedSeg (2D)	here	Instance segmentation	Under beta-testing	⚠ broken by recent updates	here	Open in Colab
MaskRCNN (2D)	here	Instance segmentation	Under beta-testing		Coming soon!	Open in Colab
Interactive Segmentation - Kaibu (2D)	here	Interactive instance segmentation	Under beta-testing		Coming soon!	Open in Colab

Denoising and Image Restoration

Network	Paper(s)	Tasks	Status	Last test	Link to example training and test dataset	Direct link to the notebook in Colab
Noise2Void (2D)	here	Self-supervised denoising	Fully supported	06/10/22 ✅ working (EGM)	here or here	Open in Colab
Noise2Void (3D)	here	Self-supervised denoising	Fully supported	07/10/22 ✅ working (GJ)	here	Open in Colab
CARE (2D)	here	Supervised denoising	Fully supported	07/10/22 ✅ working (GJ)	here or here	Open in Colab
CARE (3D)	here	Supervised denoising	Fully supported	07/10/22 ✅ working (GJ)	here	Open in Colab
3D-RCAN	here	Supervised denoising	Under beta-testing	⚠ broken (no GPU)	here	Open in Colab
DecoNoising (2D)	here	Self-supervised denoising	Under beta-testing	07/10/22 ✅ working (GJ)	here or here	Open in Colab

Super-resolution Microscopy

Network	Paper(s)	Tasks	Status	Last test	Link to example training and test dataset	Direct link to the notebook in Colab
Deep-STORM	here	Single Molecule Localization Microscopy (SMLM) image reconstruction from high-density emitter data	Fully supported	08/10/22 ✅ working (GJ)	Training data simulated in the notebook or available from here	Open in Colab
DFCAN	here	image upsampling	Under beta-testing	08/10/22 ✅ working (GJ)	here	Open in Colab
WGAN	here	image upsampling	Under beta-testing	22/09/22 ✅ working (IvanHidalgo & EGM)	here	Open in Colab

Object Detection

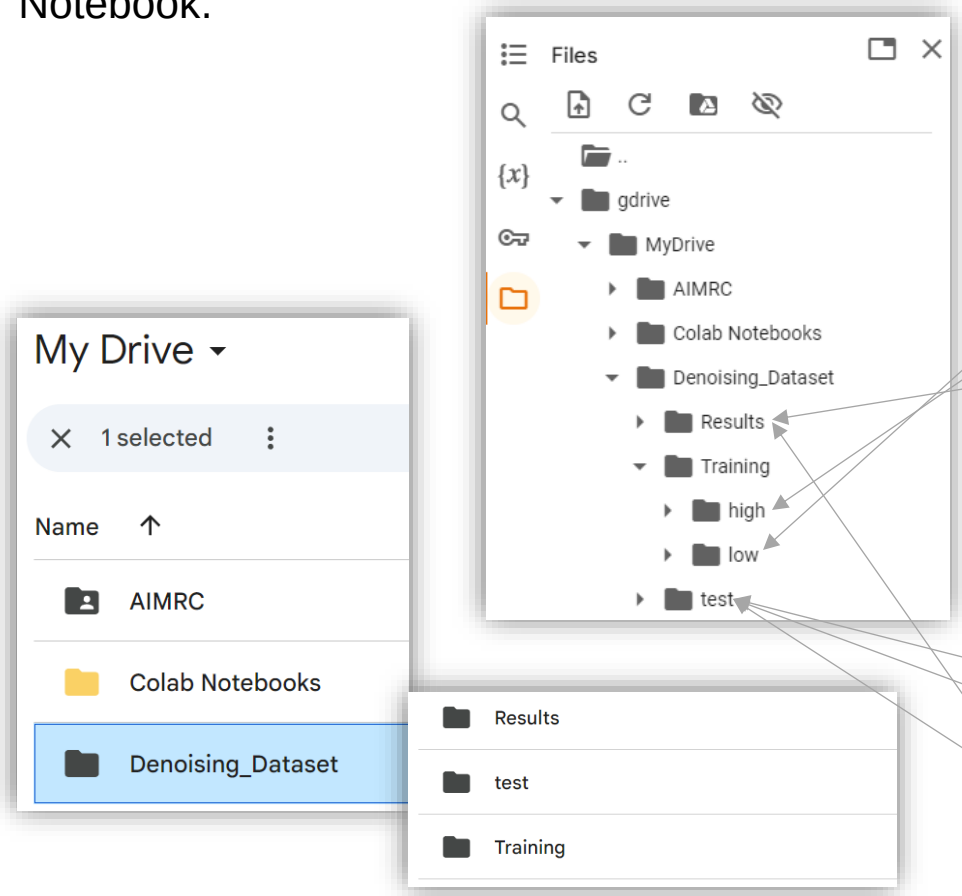
Network	Paper(s)	Tasks	Status	Last test	Link to example training and test dataset	Direct link to the notebook in Colab
YOLOv2	here	Object detection (bounding boxes)	Fully supported		here	Open in Colab
Detectron2	here	Object detection (bounding boxes)	Under beta-testing		here	Open in Colab
RetinaNet	here	Object detection (bounding boxes)	Under beta-testing		here	Open in Colab

Image-to-image Translation

Network	Paper(s)	Tasks	Status	Last test	Link to example training and test dataset	Direct link to the notebook in Colab
Label-free prediction (fnet) 2D	here	Artificial labelling	Under beta-testing	11/08/22 ✅ working (EGM)	Coming soon	Open in Colab
Label-free prediction (fnet) 3D	here	Artificial labelling	Fully supported		here	Open in Colab
CycleGAN	here	Unpaired Image-to-Image Translation	Fully supported		here	Open in Colab
pix2pix	here	Paired Image-to-Image Translation	Fully supported	04/08/22 ✅ working (Sujan Ghimire)	here	Open in Colab

Exercise 4 – ZeroCostDL4Mic

CARE denoising
(Content Aware Image Restoration)
Please follow the instructions in the Jupyter Notebook.



1. Upload denoising dataset to your Google Drive
2. Open ZeroCostDLC4Mic Notebook in a new tab. Save it as a copy to your Drive.
3. Connect to a T4 GPU session.

4. Edit following in the notebook file

Section 3.1
Training_source: `/content/gdrive/MyDrive/Denoising_Dataset/Training/low`
Training_target: `/content/gdrive/MyDrive/Denoising_Dataset/Training/high`
model_name: `denoising_model`
model_path: `/content/gdrive/MyDrive/Denoising_Dataset/Results`
number_of_epochs: `5` you can increase/decrease this number depending on whether or not you are connected to a GPU kernel and/or if you have time

Section 4.2 (Important)
Change lr to learning_rate in this line

```
writer.writerow([history.history['loss'][i], history.history['val_loss'][i],
history.history['learning_rate'][i]])
```

Section 5.2
Source_QC_folder: `/content/gdrive/MyDrive/Denoising_Dataset/test/Low`
Target_QC_folder: `/content/gdrive/MyDrive/Denoising_Dataset/test/High`

Section 6.1
Data_folder: `/content/gdrive/MyDrive/Denoising_Dataset/test/Low`
Result_folder: `/content/gdrive/MyDrive/Denoising_Dataset/Results`

Even more resources

PyDeepImageJ

CSBDeep

PydeepImageJ

- Q: How can we use DeepImageJ with our own model?
- PydeepImageJ is a Python package and can help convert Python models to BioImage Model Zoo format.
- Steps:
  - Training the proposed model in Python.
  - Exporting the trained model to DeepImageJ's format using PydeepImageJ.
  - Loading exported model to DeepImageJ.
  - Continuing exploration with DeepImageJ.

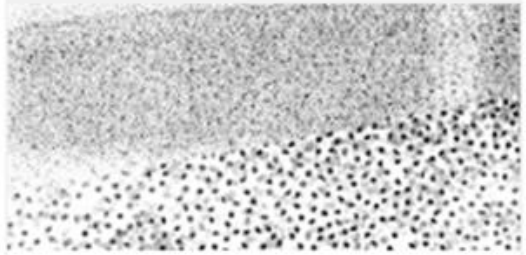
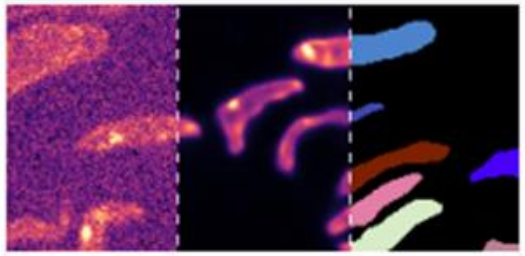
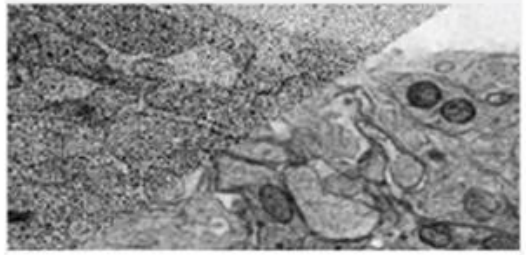
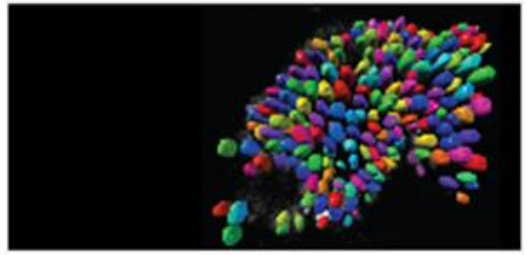
https://github.com/deepimagej/pydeepimagej

CSBDeep

- A DL-based toolbox for microscopy image restoration and segmentation.
- Two use cases
  - CSBDeep for Python: users can create and train model with their own data from scratch.
  - CSBDeep for Fiji: pre-trained model with CSBDeep can be loaded to Fiji for training or testing.
- Pros
  - Provides users an ability to train model from scratch.
  - Allows users to train model with their own data directly on Fiji.
  - Allows users to export their models to Fiji's format.
- Cons
  - Supports Tensorflow 1 and 2, but not Pytorch

https://csbdeep.bioimagecomputing.com/

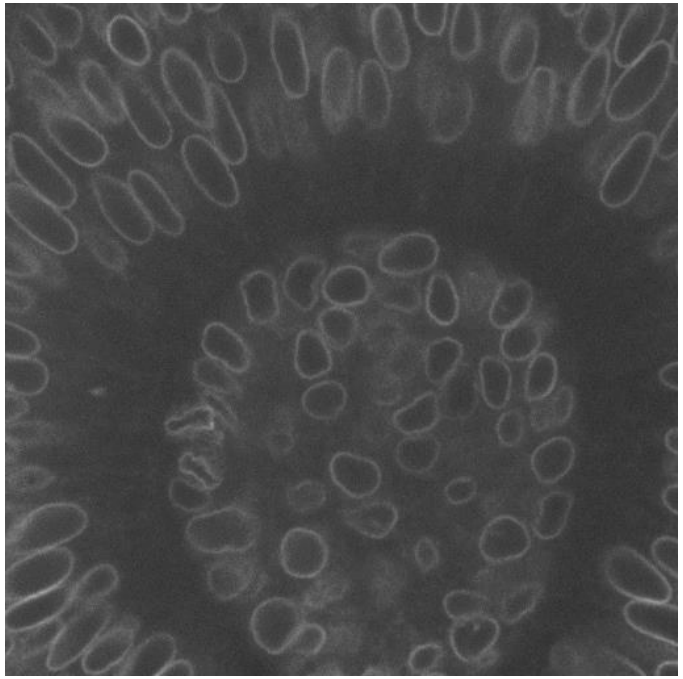
Tools in CSBDeep

CARE	DenoiSeg	Noise2Void	StarDist
			
Training data: Matching image pairs.	**Training data:** Noisy images, some of them with existing segmentation.	**Training data:** Only noisy images.	**Training data:** Matching pairs of raw data and segmented data.
Purpose: Image restoration.	**Purpose:** Image restoration and object detection.	**Purpose:** Image restoration.	**Purpose:** Object detection.

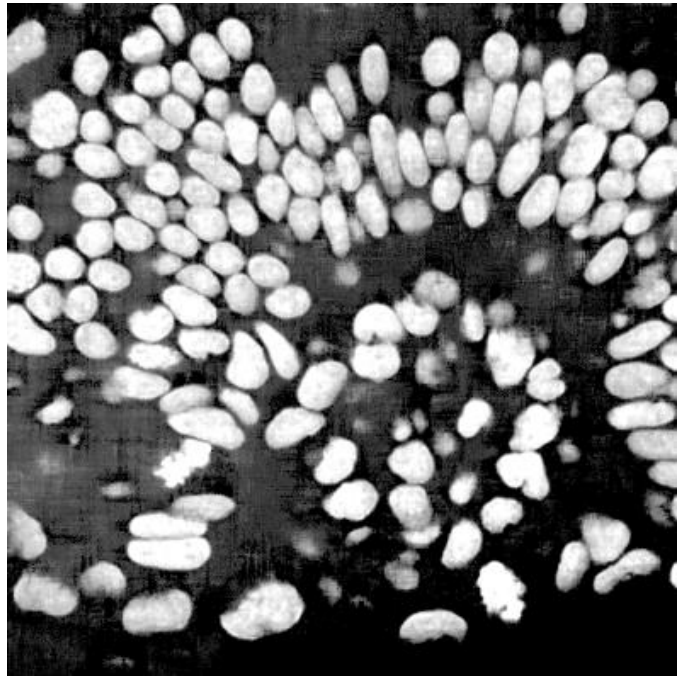
Support image restoration (with paired of noisy and clean images, paired of noisy images, and single noisy images) and object detection

https://csbdeep.bioimagecomputing.com/

CSBDeep



Original Image



Output image

Model: Isotropic Reconstruction (Zebrafish retina)

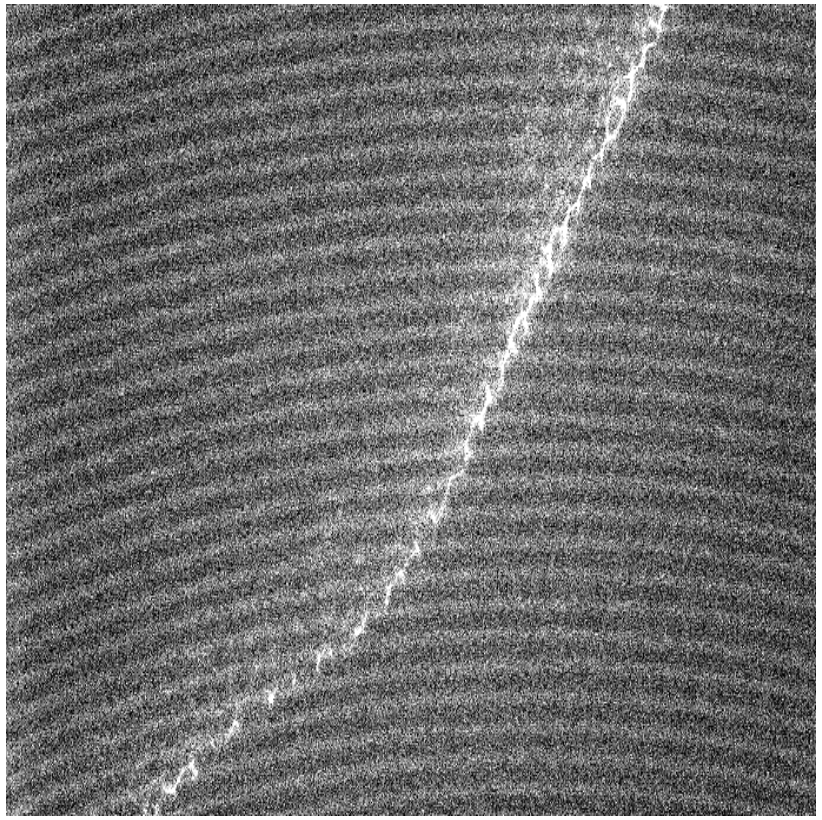
Pre-trained from CSBDeep
Training dataset: dataset is collected by authors

- 24121 images with 36x128x128x2 for training

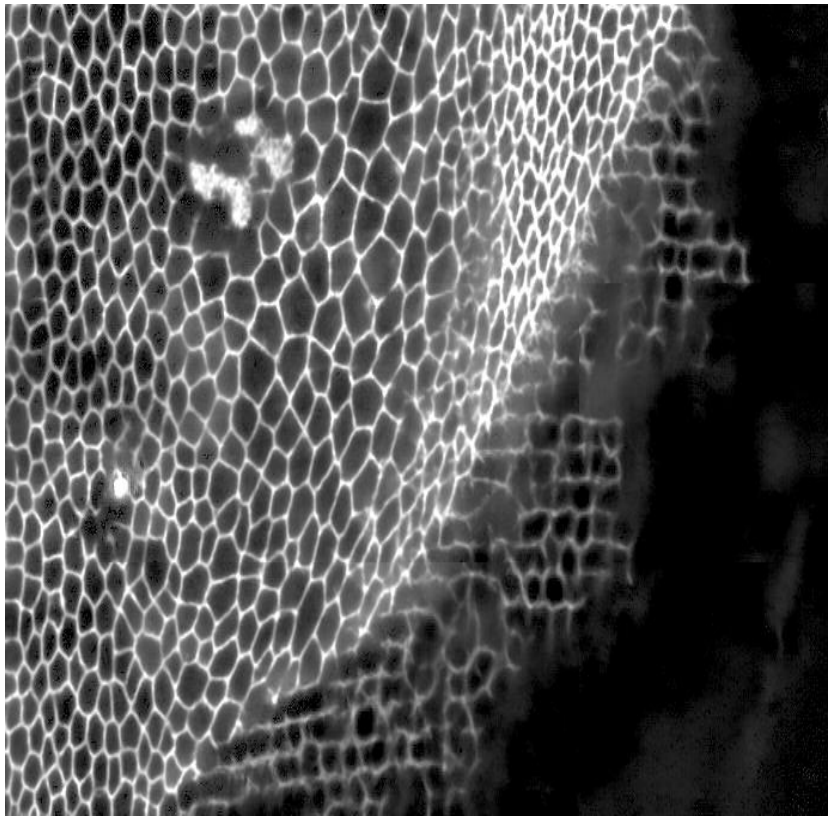
Training procedure

- 100 epochs
- Learning rate: 0.0004

CSBDeep



Original Image



Output image

Model: Surface Projection (Drosophila wing, e-cadherin)

Pre-trained from CSBDeep
Training dataset: dataset is collected by another research group

- 16891 images with 50x64x64x1 for training

Training procedure

- 100 epochs
- Learning rate: 0.0004

Guide

- I want to learn current DL-based methods, datasets or tools. How could I start?
  - Start with BioImage Model Zoo
- I want to try some traditional image processing first. What is the best way?
  - Fiji/ImageJ
- Without programming skills or technical details. What is the best way to try DL methods?
  - DeepImageJ and CSBDeep in Fiji
- Without programming skills. How to train some existing models with my own data?
  - CSBDeep in Fiji and ZeroCostDL4Mic
- I am okay with Python, PyTorch or TensorFlow. What is the best way to try DL methods?
  - CSBDeep in Python and ZeroCostDL4Mic