StarDist Course

Sept 6th, 2024

Todd Fallesen

Chris Hadjigeorgiou

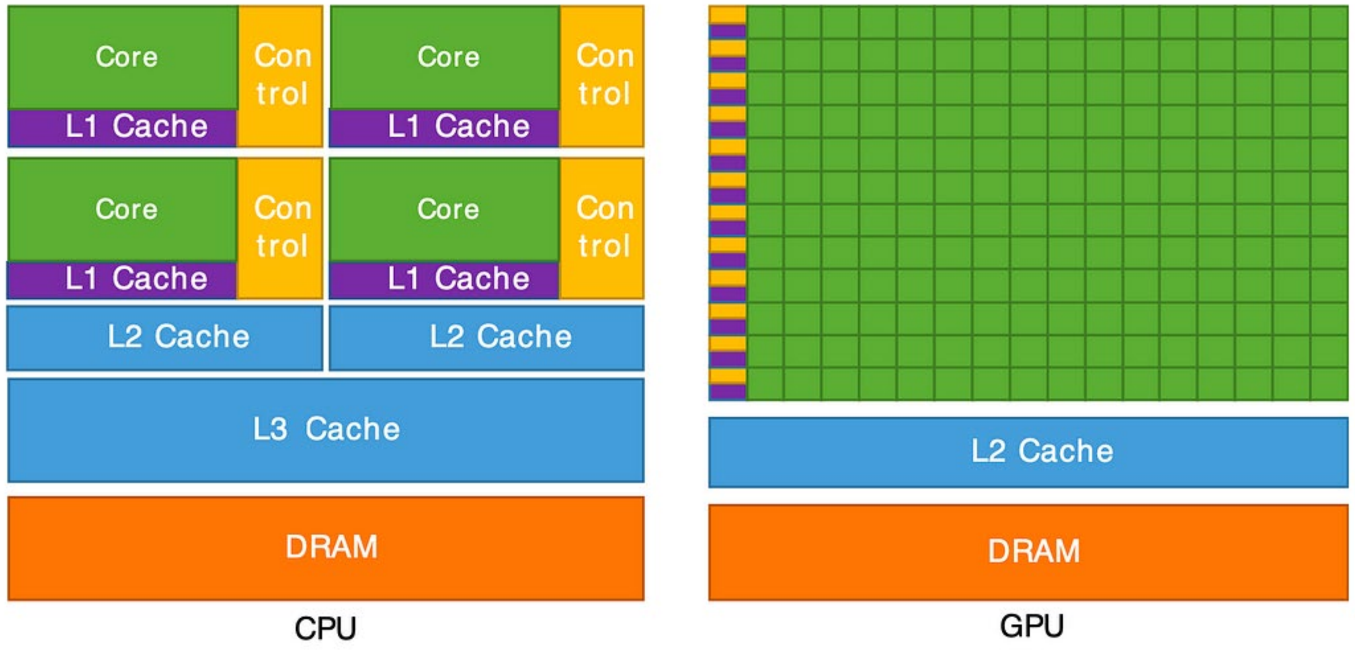
Course Files

FileName	Use	Folder
StarDist_bash_Sept24.sh	Installs StarDist into your working directory on camp	Setup_Scripts
StarDist_Crick_Sept24.yml	The environment file for StarDist, called by StarDist_bash_Sept24.sh	Setup_Scripts
Install_FIJI_w_download.sh	Downloads and installs FIJI into your working directory	Setup_Scripts
GPU_Test.ipynb	Test installation of StarDist	Notebooks
StarDist_Demo_Data_Download.ipynb	Download the test dataset, see what labels look like	Notebooks
StarDist_Training_Demo.ipynb	Demo of how the training works, including augmentation and number of rays. Works on demo data	Notebooks
StarDist_Prediction_Demo.ipynb	Demo of how prediction works. Includes a task on changing models	Notebooks
StarDist_Training_Task.ipynb	Notebook for training your own model.	Notebooks
StarDist_Prediction_Workshop_Model	Notebook for doing prediction and saving predictions and FIJI ROI’s	Notebooks

GPU vs CPU

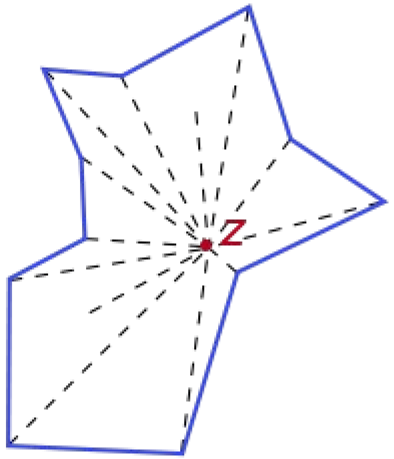
CPUs are the normal processors in your computer. They act in a serial way, doing one task, and then another. They do a variety of tasks incredibly quickly and well, but don't do multiple tasks well at the same time.

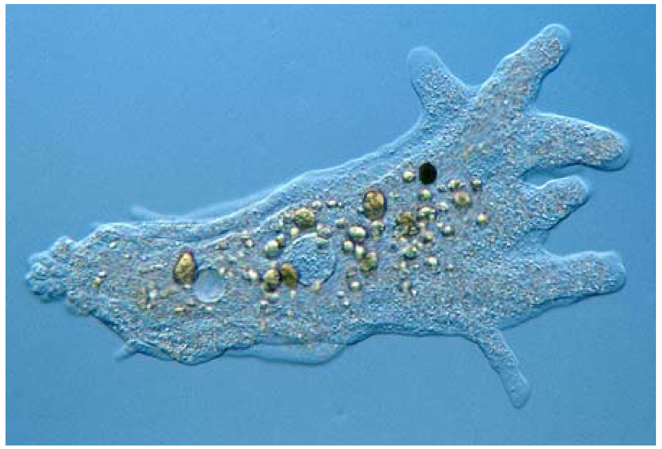
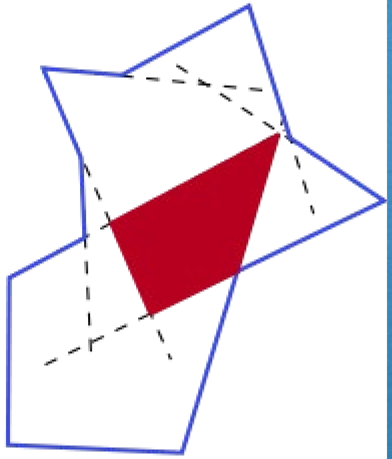
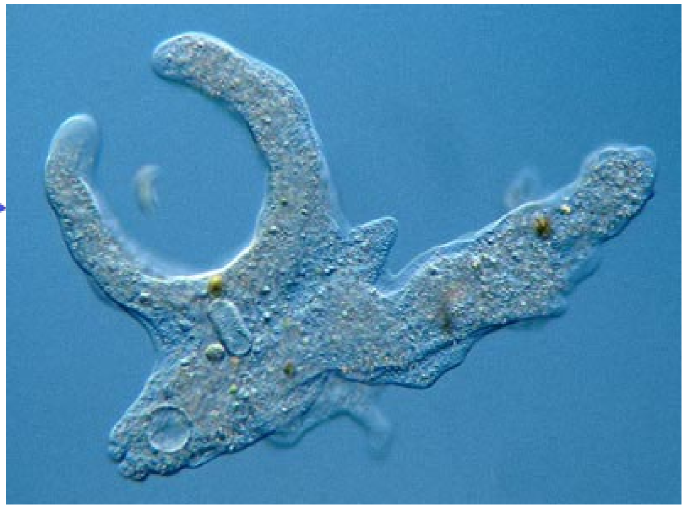
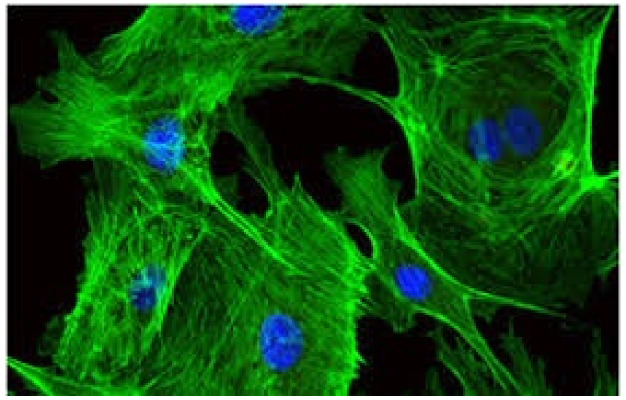
GPUs are dumber and slower, but have many many many processors, so can be faster for simple parallelizable tasks. For things like image analysis, where we can easily parallelize a task (i.e. the same operation on every slice of a Z-stack) GPUs are much faster.



Star-convex?

Stardist can only detect objects that are Star-convex: all the points of the perimeter can be reached by rays emanating from a point somewhere in the object.



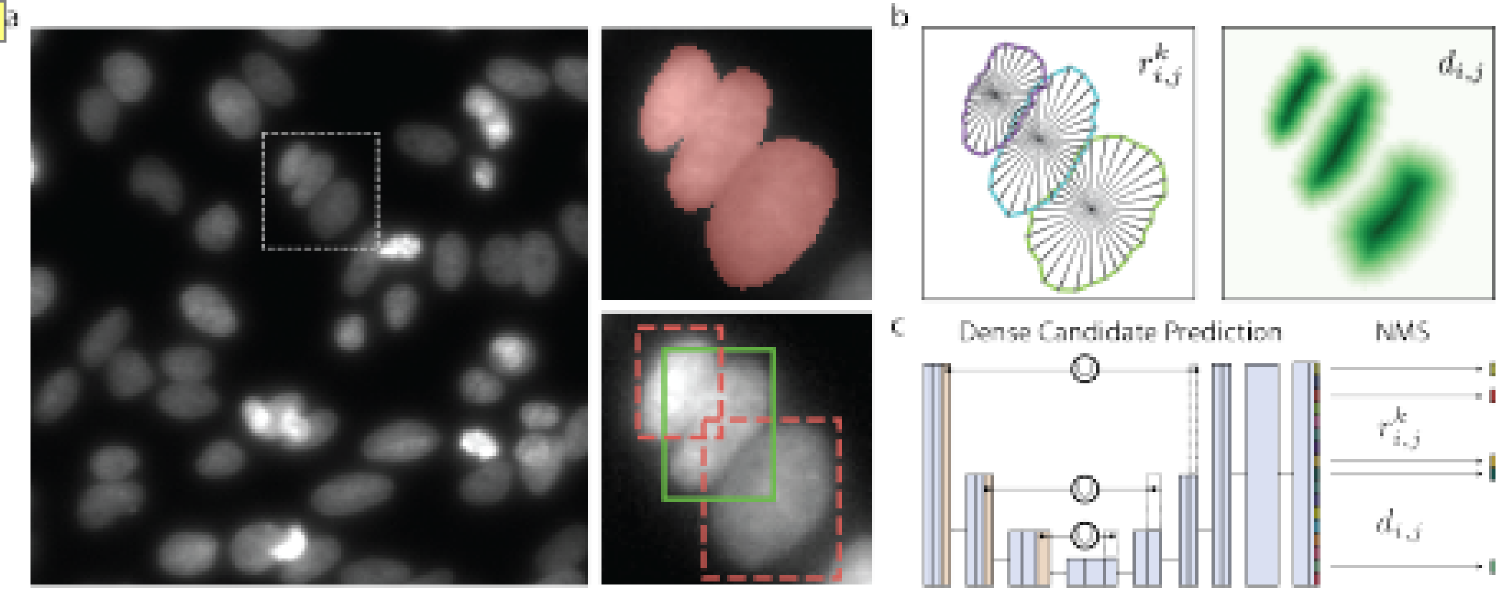


Fig. 1: *(a)* Potential segmentation errors for images with crowded nuclei: Merging of touching cells (upper right) or suppression of valid cell instances due to large overlap of bounding box localization (lower right). *(b)* The proposed StarDist method predicts object probabilities $d_{i,j}$ and star-convex polygons parameterized by the radial distances $r_{i,j}^k$. *(c)* We densely predict $r_{i,j}^k$ and $d_{i,j}$ using a simple U-Net architecture [15] and then select the final instances via non-maximum suppression (NMS).

- Is a pixel in an object
- If pixel is in an object, how far to the boundary of the object in which the pixel belongs
- Pixels are weighted higher towards the middle of objects

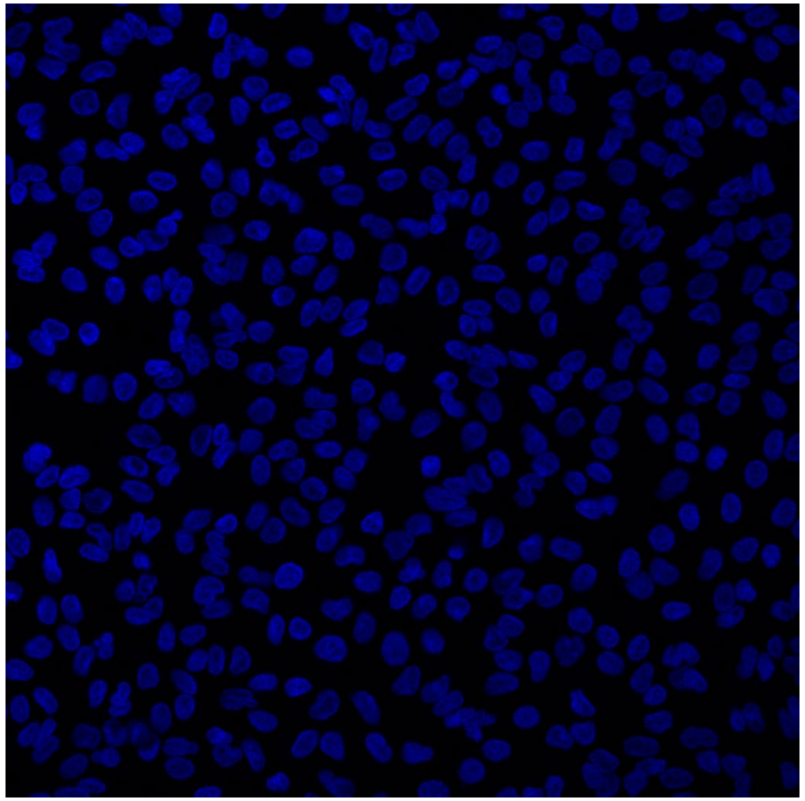
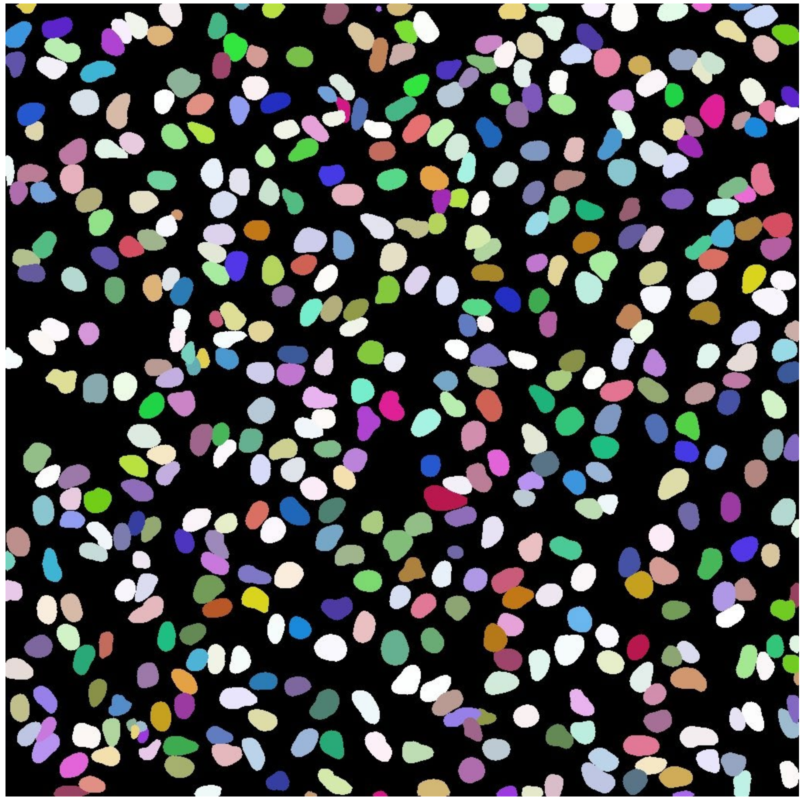
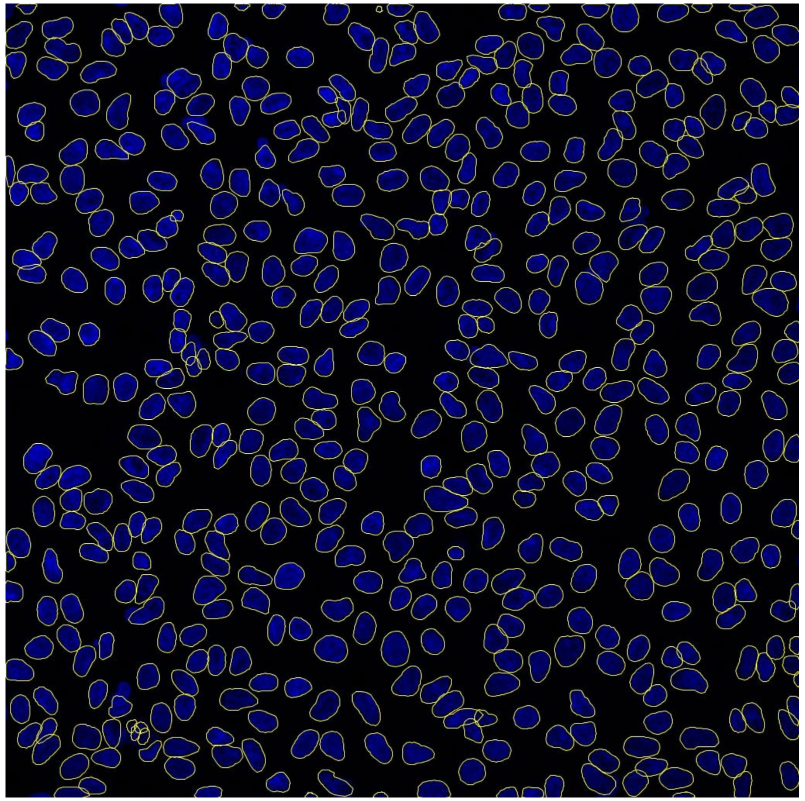
1.) ($d_{i,j}$) The object probability that a pixel is in an object is calculated for every pixel. The object probability is defined as the normalized distance to the nearest background pixel.

2.) For every pixel the distance to the edge of the object of which it belongs is calculated, along n-rays. This forms a polygon

3.) Polygon proposals are generated from each pixel, but only pixels with a high enough "d" value are considered.

4.) Non-maximum suppression is used to find the final polygons

Quick-use FIJI

https://imagej.net/plugins/stardist

To Install FIJI

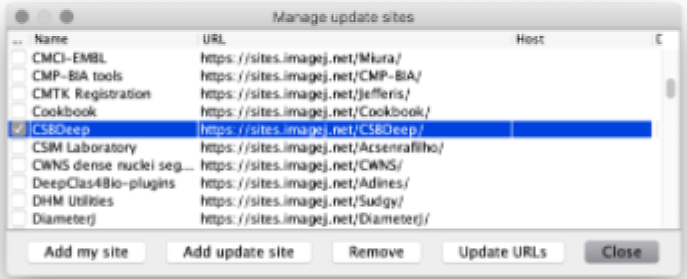
- Save the Install_FIJI_w_download.sh file to a folder where you want to put Fiji on Nemo
- Run Install_FIJI_w_download.sh ("./ Install_FIJI_w_download.sh")
- At home directory prompt (~) run "./Run_Fiji.sh"

- Other option, download Fiji linux from website, move into a directory on nemo and unzip it

To Install in FIJI

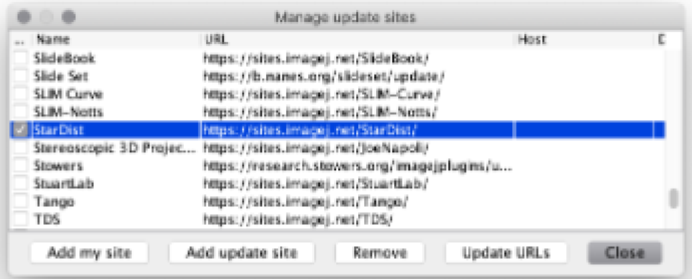
Installation

1. Start Fiji (or download and install it from here first).
2. Select `Help > Update...` from the menu bar.
3. Click on the button `Manage update sites`.
4. Scroll down the list and tick the checkboxes for update sites `CSBDeep` and `StarDist`, then click the `Close` button.
   (If `StarDist` is missing, click `Update URLs` to refresh the list of update sites.)



Manage update sites

...	Name	URL	Host
☐	CMCI-EMBL	https://sites.imagej.net/Miura/	
☐	CMP-BIA tools	https://sites.imagej.net/CMP-BIA/	
☐	CMTK Registration	https://sites.imagej.net/Jefferis/	
☐	Cookbook	https://sites.imagej.net/Cookbook/	
✓	CSBDeep	https://sites.imagej.net/CSBDeep/	
☐	CSIM Laboratory	https://sites.imagej.net/Acsenrafilho/	
☐	CWNS dense nuclei seg...	https://sites.imagej.net/CWNS/	
☐	DeepClas4Bio-plugins	https://sites.imagej.net/Adines/	
☐	DHM Utilities	https://sites.imagej.net/Sudgy/	
☐	DiameterJ	https://sites.imagej.net/DiameterJ/	



Manage update sites

...	Name	URL	Host
☐	SlideBook	https://sites.imagej.net/SlideBook/	
☐	Slide Set	https://b.nanes.org/slideset/update/	
☐	SLIM Curve	https://sites.imagej.net/SLIM-Curve/	
☐	SLIM-Notts	https://sites.imagej.net/SLIM-Notts/	
✓	StarDist	https://sites.imagej.net/StarDist/	
☐	Stereoscopic 3D Projec...	https://sites.imagej.net/JoeNapoli/	
☐	Stowers	https://research.stowers.org/imagejplugins/u...	
☐	StuartLab	https://sites.imagej.net/StuartLab/	
☐	Tango	https://sites.imagej.net/Tango/	
☐	TDS	https://sites.imagej.net/TDS/	

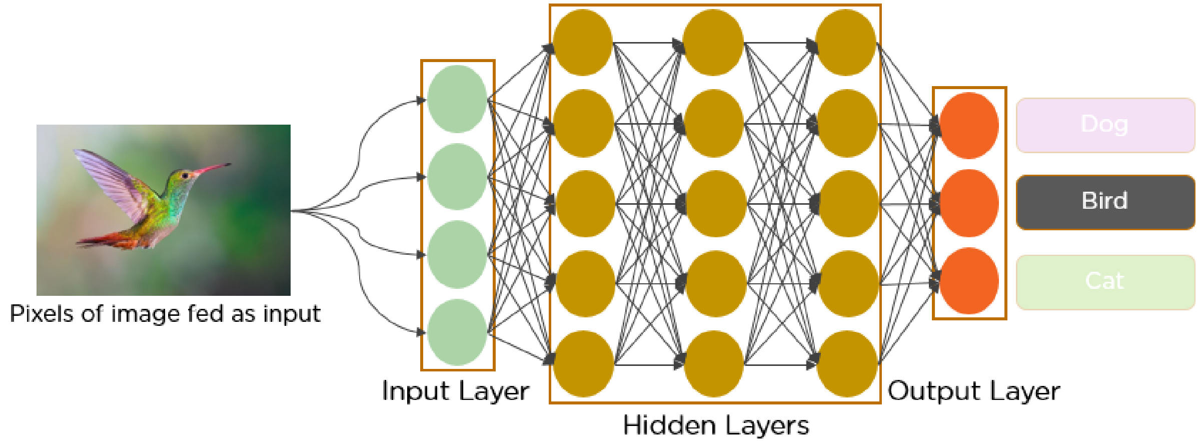
5. Click on `Apply changes` to install the plugin.
6. Restart Fiji.

FIJI Demo

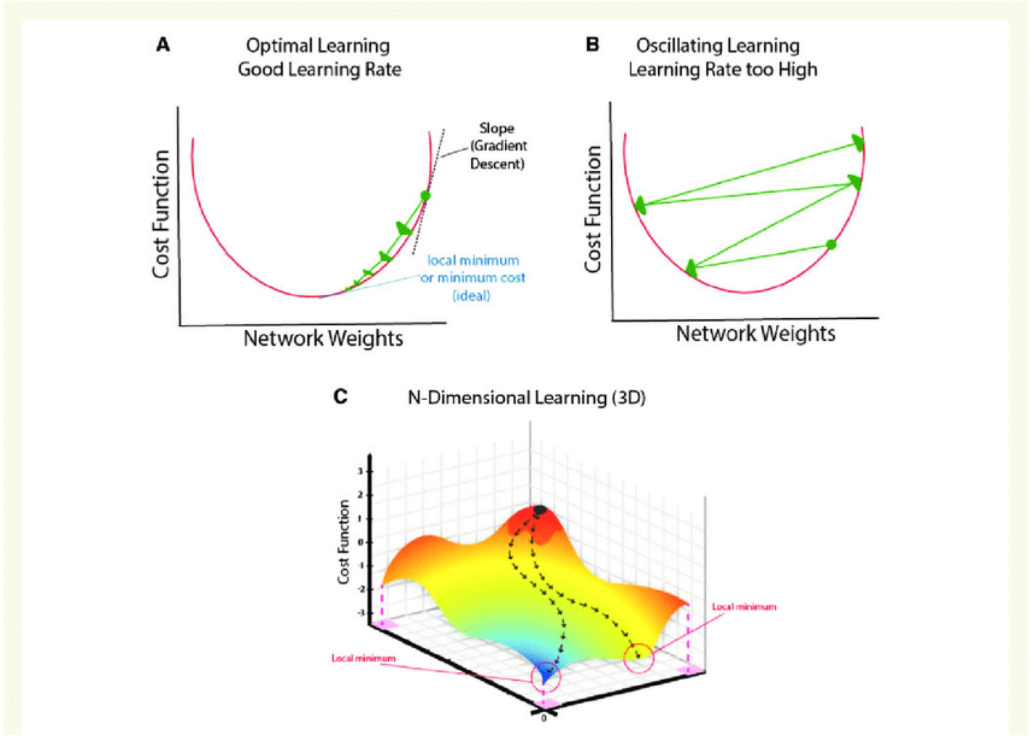
- Open up FIJI, install StarDist
    - Also, please install Labkit
- Open the file: StarDist_Course_Materials\Example_Data\Crick_Dapi\Full_Size_Unseen\Image_Set_001.tif
- Run Stardist on this image (plugins → StarDist→Run)

StarDist in Jupyter Notebooks

ML Basics



- Broadly we want to lower the ‘loss function’. This could be classification error, Mean Squared Error, etc.
- The ‘neural network’ is made up of many interconnected nodes, and the data that passes through them is each weighted.
- The algorithm evaluates the input data, takes a guess at what the result should be, and compares it to the ‘ground truth’
- The algorithm then changes the weights between the neurons of the neural network, and tries again. The rate at which the weights change is the ‘learning rate.’ You want a learning rate that is high enough to converge, but not to high that you get caught in a local minima. Start with the defaults, and change only if needed.



ML Basics Continued

- Overfitting is when we have too many of the same types of objects. We can only then find that type of object, but really well.
  - Imagine you are training to identify fruit and vegetables. You make a training set of 99 apples and a tomato. You'll never properly identify tomatoes, and definitely not bananas.
- Transfer learning: using another data set as a starting point.
  - StarDist supports this, but it's not documented, it's complicated, and it's not recommended.

Data set Training:

Data set Training: Making our own models

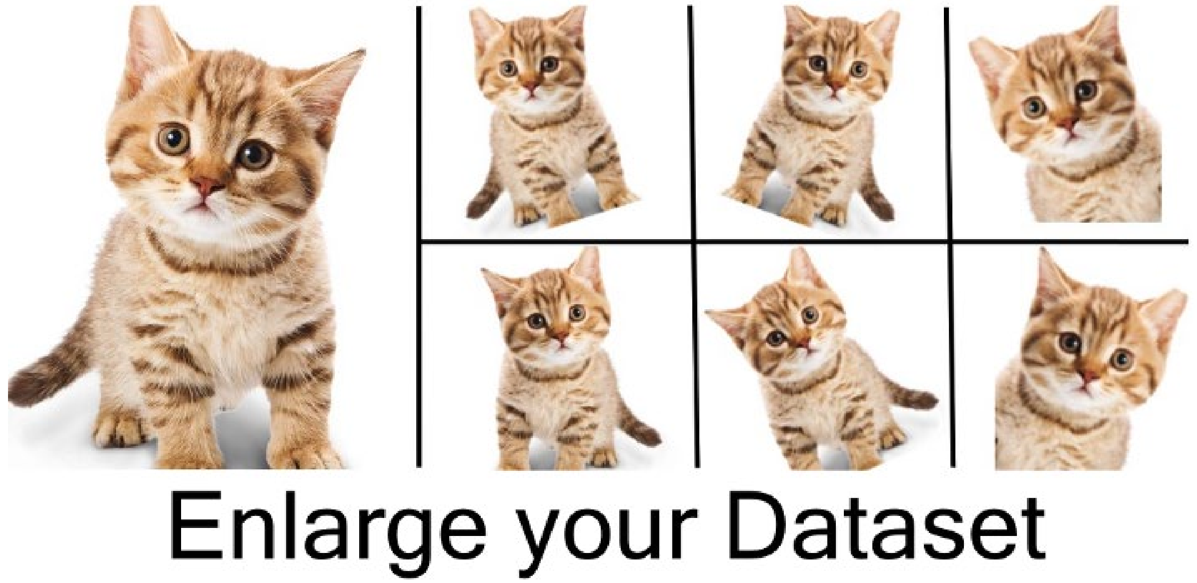
- Jupyter Notebooks/Camp On Demand
- Need a GPU to do this, and the notebooks that are downloadable
- For 2D data, at least 5-10 image crops with 20 annotated objects
- For 3D, less crops, but you’re going through far more slices
- Want edge case and weird data, more than once

Data set Training: Data Augmentation

Data augmentation is a way to get more training data out of what you have

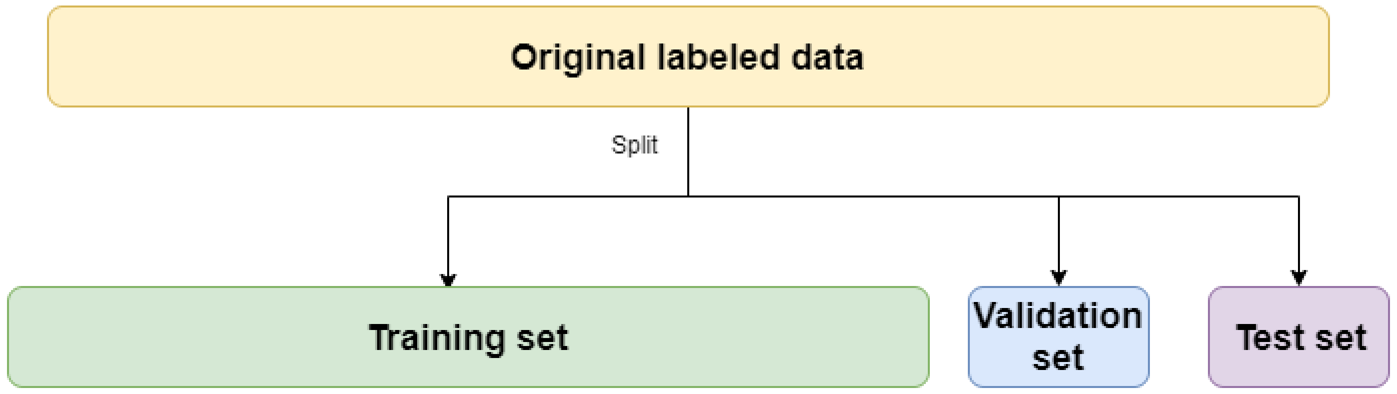
Augmentation is just rotating, flipping, skewing, transforming the image and the labels in the exact same way.

In our datasets, we always do this. It's free data.



https://nanonets.com/blog/data-augmentation-how-to-use-deep-learning-when-you-have-limited-data-part-2/

Data set Training: Train/Test Split



https://www.brainstobytes.com/test-training-and-validation-sets/

As mentioned earlier, it is generally better to annotate a variety of image crops as your training data. However, those crops must be big enough to contain entire fully visible objects and provide some context around them. Also make sure that not too many of the annotated objects are touching the border (it’s fine if some do, but it should not be the majority). Example: if you have small cells with a diameter of 20 pixels, it might be sufficient to have annotated images of size 160x160, whereas if your objects have a diameter of 80 pixels, you would need to use larger annotated images e.g. of size 512x512.
The “patch size” is an important parameter for training StarDist, and the size of images used for training affects what an appropriate value for the patch size should be (to maintain compatibility with the neural network architecture). For example, the patch size used for training StarDist must be smaller or equal than the size of the smallest annotated training image. To be on the safe side, ensure that the patch size is divisible by 16 along all dimensions. For example, you can annotate image crops of 300x300 pixels and then use a patch size of 256x256 pixels for training.

Data set Training: Training data for our test

- https://github.com/stardist/stardist#annotating-images

- The data we are using is right here from the Crick
- Input data is 1200x1200
- Using 600x600 crops
- Everyone needs to label one image for labelling
- Data is in StarDist_Course_Materials/Example_Data/Crick_Dapi/Crops

Data set Training: Labelling Images using labkit

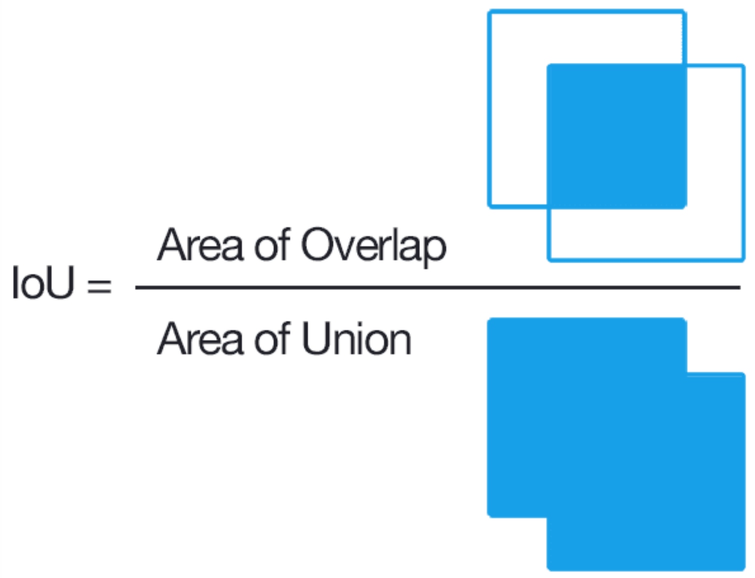
- Open the cropped image
- Open labkit (type in labkit in search bar)
- Make sure to NOT “allow overlapping labels”
- Save often!
- Remove background and foreground labels
- Add a new label for each object
- Save often!
- Typing “s” brings up contrast control
- To save the labels, go to “Labelling→ Save Labelling→ label” while in progress.
- When finished, save as Tif “Labelling→Save Labelling→Tif”. DO NOT SAVE IN THE SAME FOLDER!!! You’ll overwrite the cropped image

Data set Training: Setting up training data

- Training images need to be in a folder called : *train*
- ~~Test images need to be in a folder called : *test*~~
    - *This used to be the case, you'll see the data still labelled like that, but the training function will automatically do a train test split.*
- In both folders, there need to be folders called : *images* and *masks*
- Images, and their corresponding masks, have to have EXACTLY the same name.
- Remember the test/train split. You might want about 20-30% of your data for testing/validation

Accuracy

- Detected object is considered a true positive (Tp) if the intersection over union (IoU) is higher than some threshold (IoU is the ratio of total overlap of detected object and ground-truth and the total union of area between detected and ground truth)



Unmatched detected objects are false positives (FP)
Unmatched ground truth are false negatives (FN)

$$AP_\tau = \frac{TP_\tau}{TP_\tau + FN_\tau + FP_\tau}$$

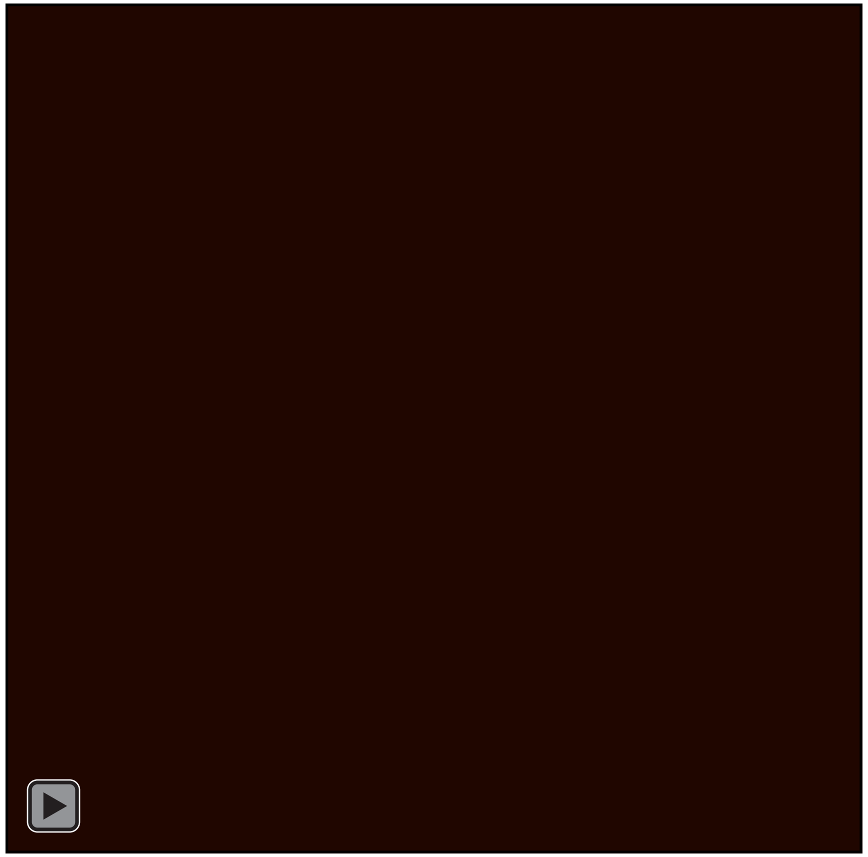
Data set Training: Set up training data and run on cluster

- We will run the training data on the cluster now to build our own model to run predictions
- First we will run through StarDist_Training_Demo.ipynb
- You will then change the variables needed to run StarDist_Training_Task.ipynb

  - You'll want to change the train_image_path, train_mask_path, model name, number of rays

Training over epochs:

- An epoch is one complete pass through the training data and updating the model.



Prediction:

Prediction:

- Prediction is much easier than training
- You can predict on a CPU, but it’s still faster on the GPU
- Make sure that the images that you want to predict on are similar to those you trained on
- You may have to rescale your images to fit the diameter of the objects you trained on.

Prediction:

- Run through the StarDist_Prediction_Demo.ipynb notebook
- Switch to the H&E model and predict an H&E image
- H&E data is in the StarDist_Course_Materials/Example_Data/HnE_Data folder

Prediction:

- Open up StarDist_Prediction_Workshop_Model.ipynb
- Use our newly trained model to predict some data from the folder:
  - StarDist_Course_Materials/Example_Data/Crick_Dapi/Full_Size_Unseen

Other models

- Multi-class
- H&E
- 3D