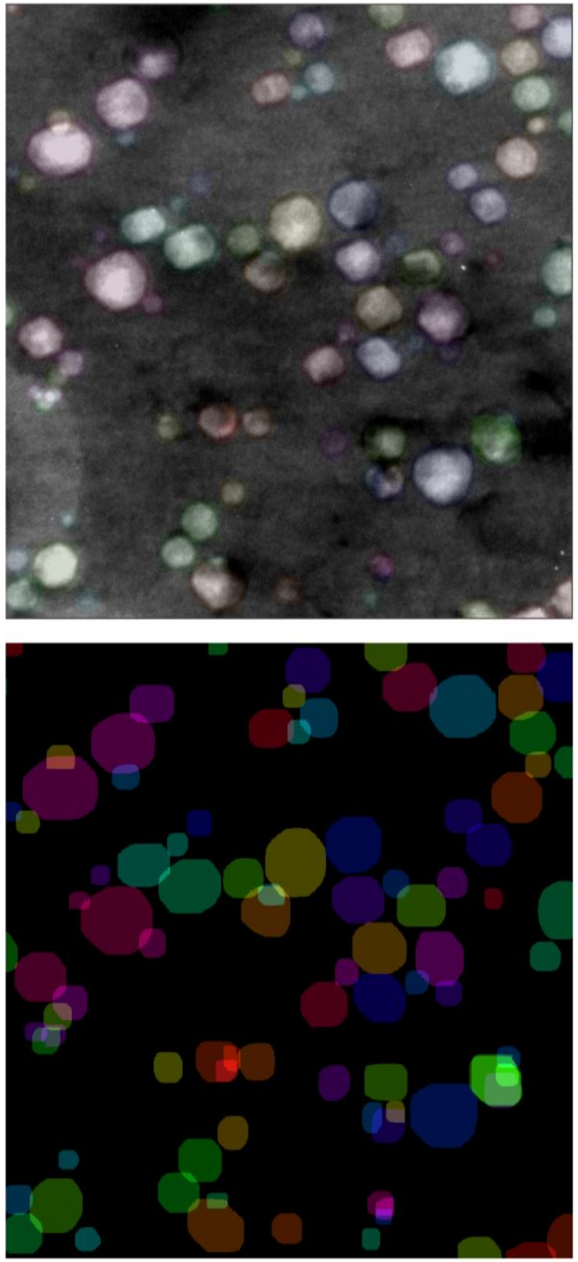
Virtual Summer School: Machine Learning in Electron Microscopy

Examples of DCNN in Electron Microscopy

Tommy Wong

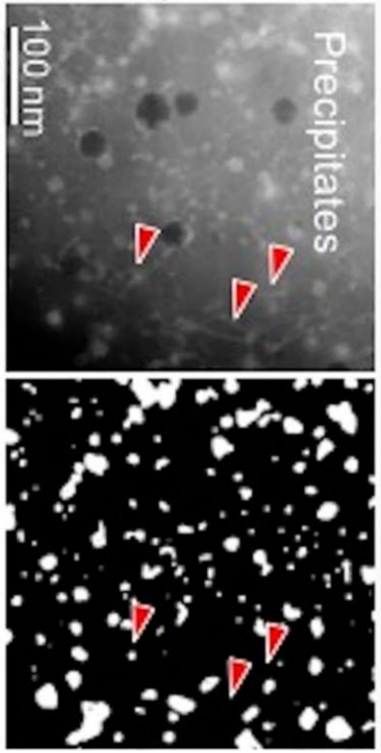
Github: https://github.com/SergeiVKalinin/ML-ElectronMicroscopy-2023/tree/main/Lecture%2013



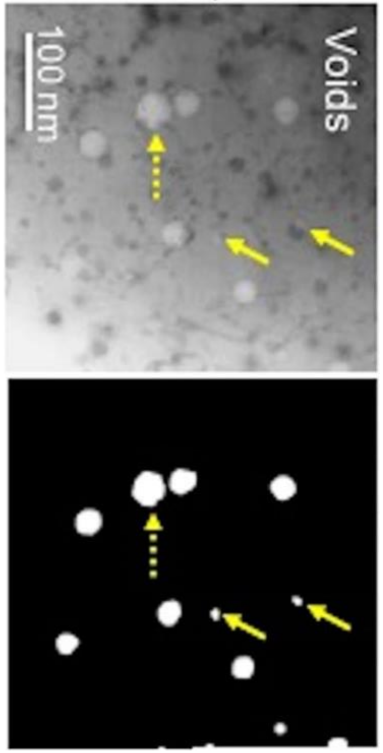
Labeling this image took ~ 1 hr, DL prediction took < 1 min.

Microstructural defects can be labeled for pixel-wise segmentation

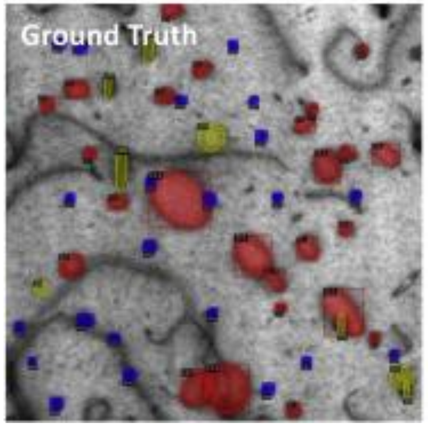
- Segmentation: associating each pixel in an image with a class



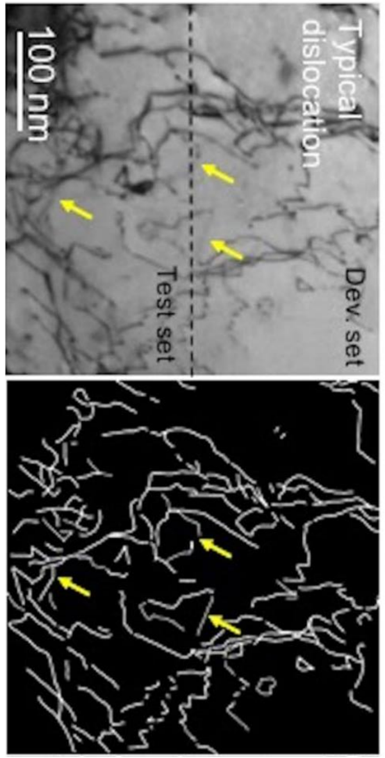
Cavities (bubbles & voids)



Precipitates



Dislocation loops

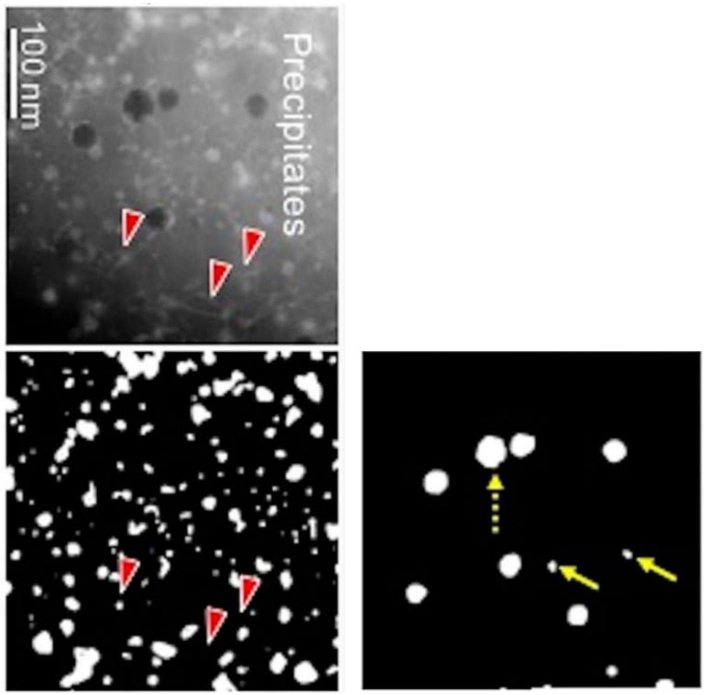


Dislocation lines

Adapted from Roberts et al. 2019 10.1038/s41598-019-49105-0, Jacobs et al. 2022 10.1016/j.xcrp.2022.100876

Different labeling systems are required for different segmentation algorithms

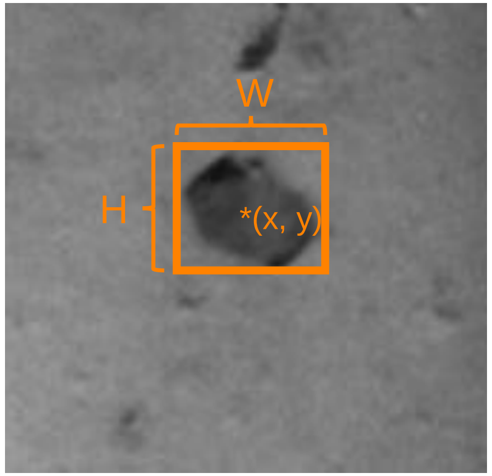
- **Important: labels are usually either 0 or 1**



Precipitates

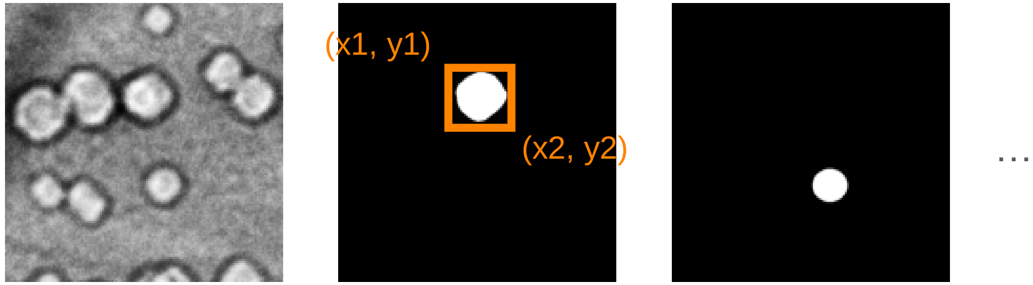
Voids

Semantic segmentation: one-hot encoding (U-Net)



[Class x y W H]

Object identification: bounding box (YOLO)

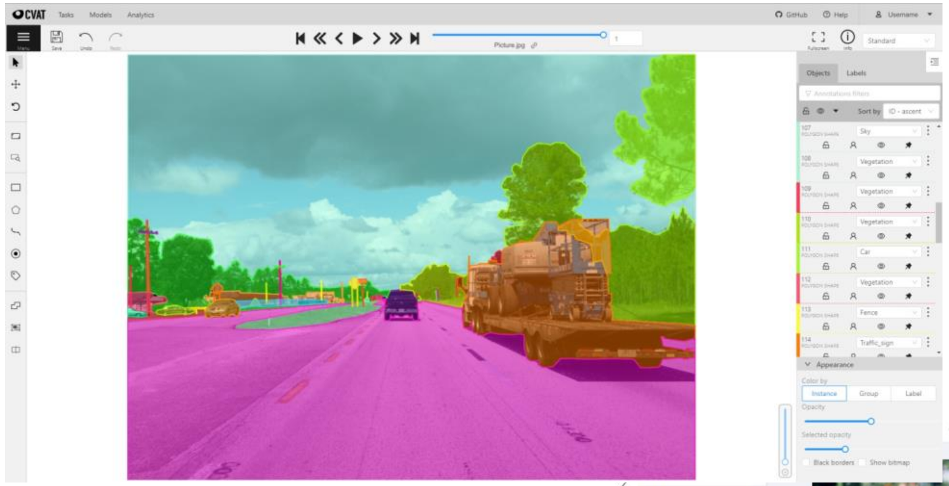


Dictionary{
‘boxes’: [x1 y1 x2 y2]
‘labels’: class
‘masks’: feature mask
}

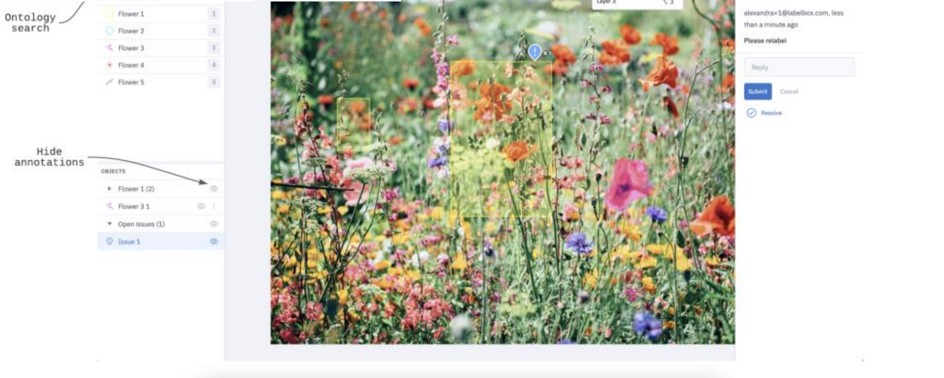
Instance segmentation: label encoding (Mask R-CNN)

Adapted from Roberts et al. 2019 10.1038/s41598-019-49105-0, Yang et al. 2011 10.1038/srep00190

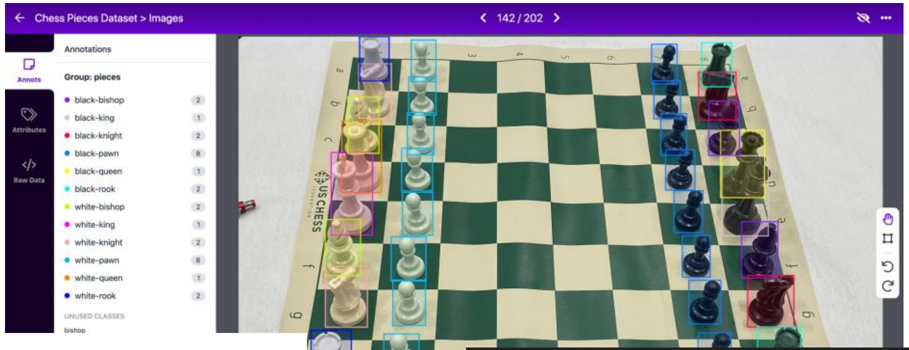
Web-based GUI tools are used for labeling



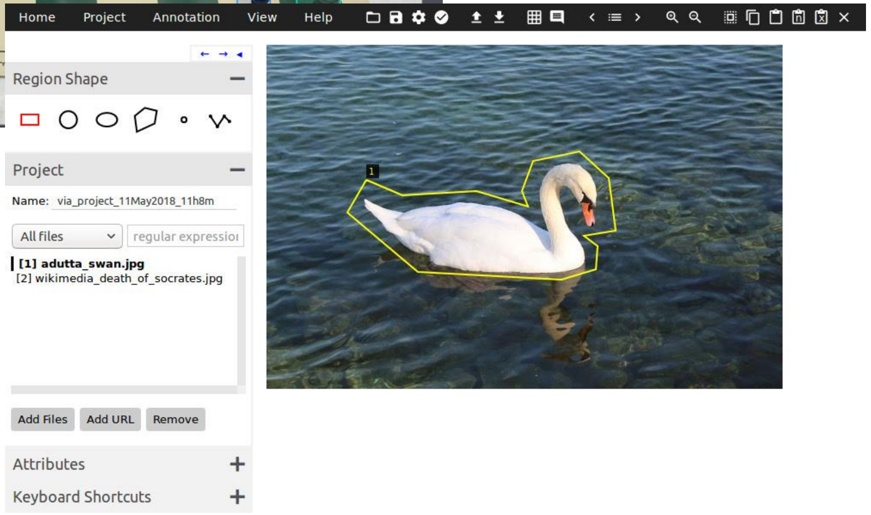
CVAT


Zoom
Brightness controls
Keyboard shortcuts
Toggle image layers
Show annotation names
Ontology search
Hide annotations

Labelbox



Roboflow



VIA

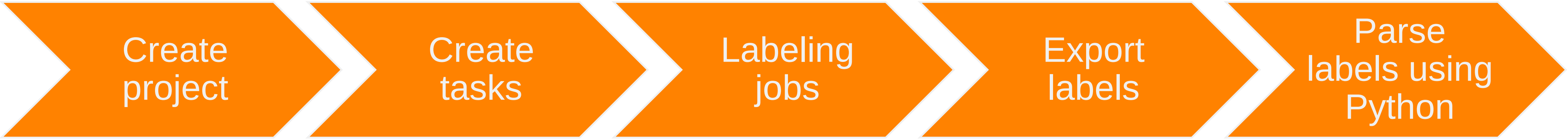
Labeling using Computer Vision Annotation Tool (CVAT)

cvat.ai

Documentation:
github.com/TaSeeMba/cvat/blob/master/cvat/apps/documentation/user_guide.md

CVAT labeling workflow

Important: before labeling, ensure all image data have the same dimensions e.g. 1024x1024



Create project

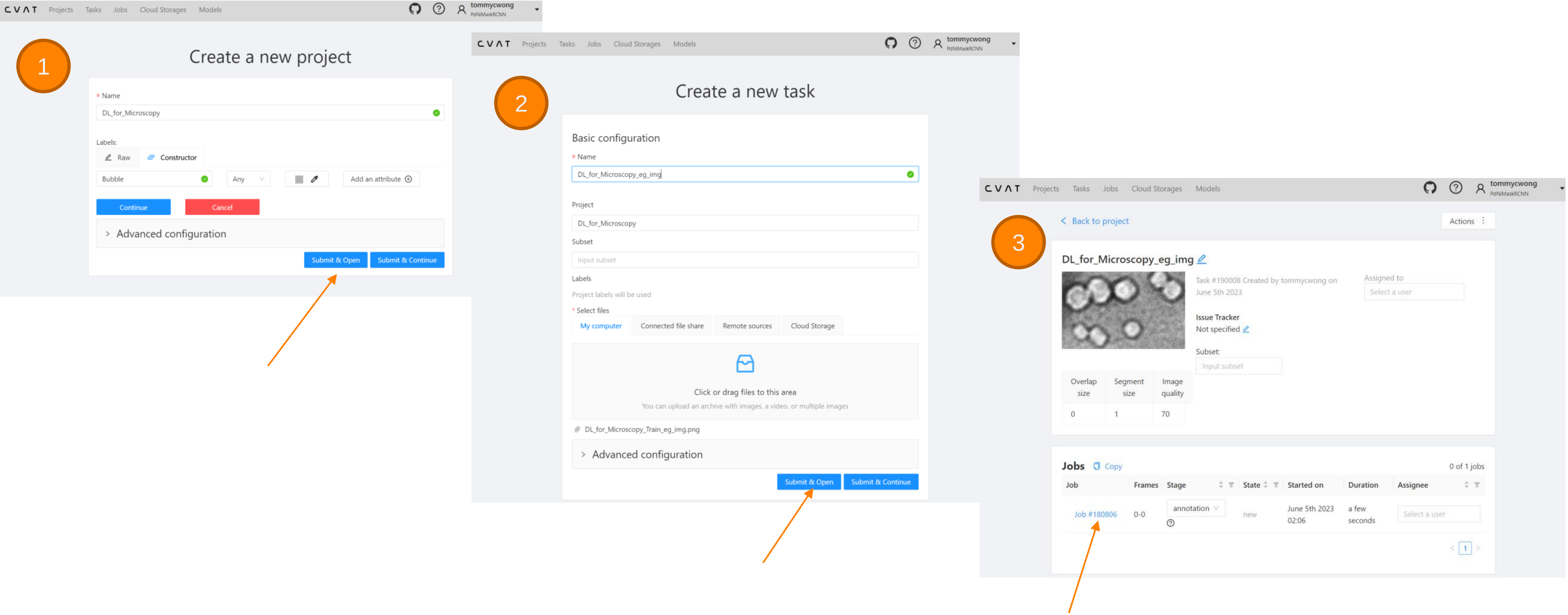
Create tasks

Labeling jobs

Export labels

Parse labels using Python

Creating a project and labeling tasks



1

Create a new project

* Name

DL_for_Microscopy

Labels:

Raw | Constructor

Bubble | Any | Add an attribute

2

Create a new task

Basic configuration

* Name

DL_for_Microscopy_eg_img

Project

DL_for_Microscopy

Subset

Input subset

Labels

Project labels will be used

* Select files

My computer | Connected file share | Remote sources | Cloud Storage

Click or drag files to this area

You can upload an archive with images, a video, or multiple images

DL_for_Microscopy_Train_eg_img.png

3

DL_for_Microscopy_eg_img

Task #190008 Created by tommycwong on June 5th 2023

Assigned to

Select a user

Issue Tracker

Not specified

Subset:

Input subset

Overlap size	Segment size	Image quality
0	1	70

Jobs Copy

0 of 1 jobs

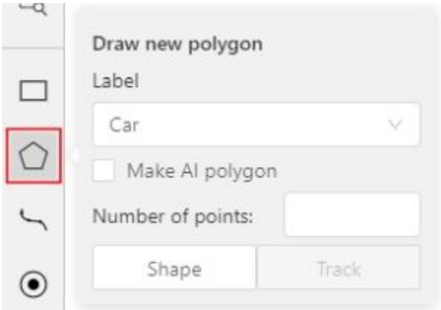
Job	Frames	Stage	State	Started on	Duration	Assignee
Job #180806	0-0	annotation	new	June 5th 2023 02:06	a few seconds	Select a user

Labeling using polygon and ellipse tools

Remember to click **Save**

Polygon tool

- Hold **Shift** to draw

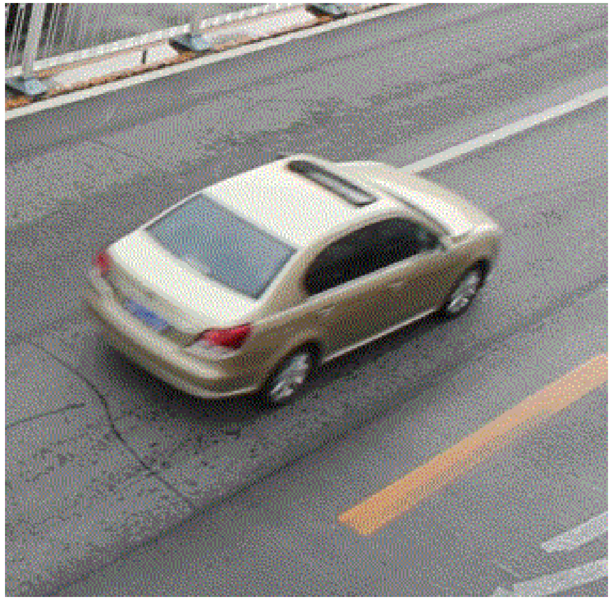


Draw new polygon

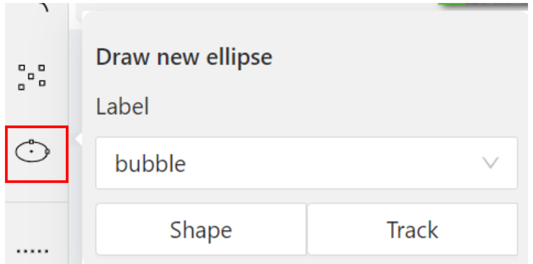
Label: Car

□ Make AI polygon

Number of points:

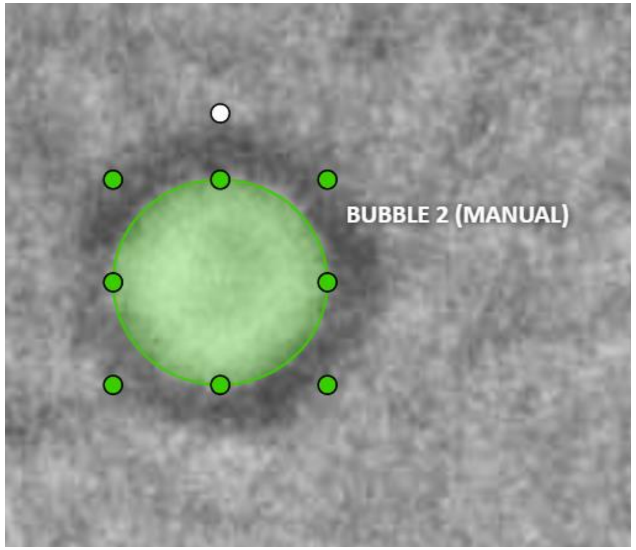


Ellipse tool



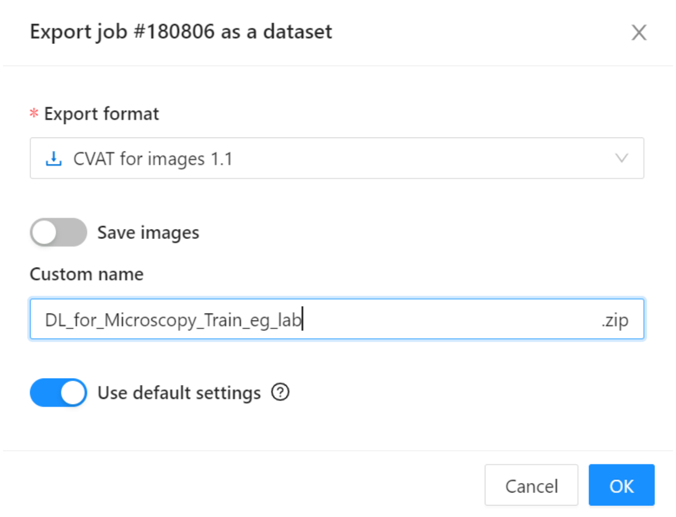
Draw new ellipse

Label: bubble



Exporting and parsing labels

Export as .xml



Export job #180806 as a dataset

* Export format: CVAT for images 1.1

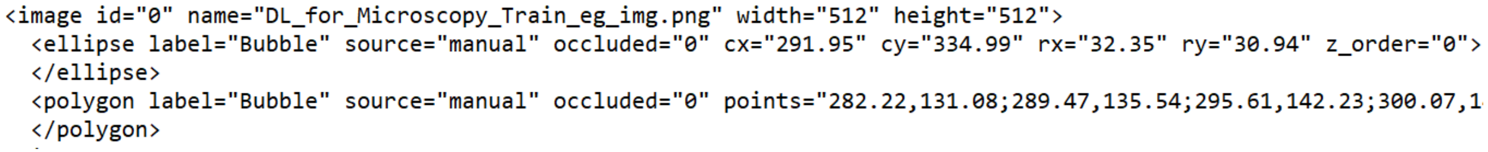
○ Save images

Custom name: DL_for_Microscopy_Train_eg_lab .zip

● Use default settings

Parsing labels using Python

```
get_imgs(train_img_names)
parse_anno_file(xml, train_img_filename)
get_unet_mask(annos)
get_maskrcnn_mask(annos)
get_maskrcnn_dataset(images=train_imgs,
labels=maskRcnn_masks)
```



```
<image id="0" name="DL_for_Microscopy_Train_eg_img.png" width="512" height="512">
  <ellipse label="Bubble" source="manual" occluded="0" cx="291.95" cy="334.99" rx="32.35" ry="30.94" z_order="0">
  </ellipse>
  <polygon label="Bubble" source="manual" occluded="0" points="282.22,131.08;289.47,135.54;295.61,142.23;300.07,1
  </polygon>
```

Additional notes on labeling

- Features typically should have a convex mask
  - Concave masks are likely occluded convex masks
- Don’t leave holes between multiple overlapping masks
- Keep in mind output files: different parsing scripts needed for .xml, .json, etc.

Data Augmentation Techniques

Notebook: https://github.com/tommycwong/ML-ElectronMicroscopy-2023/blob/main/Lecture%2013/ML4EM_Summer2023_Data_Augmentation_supp.ipynb

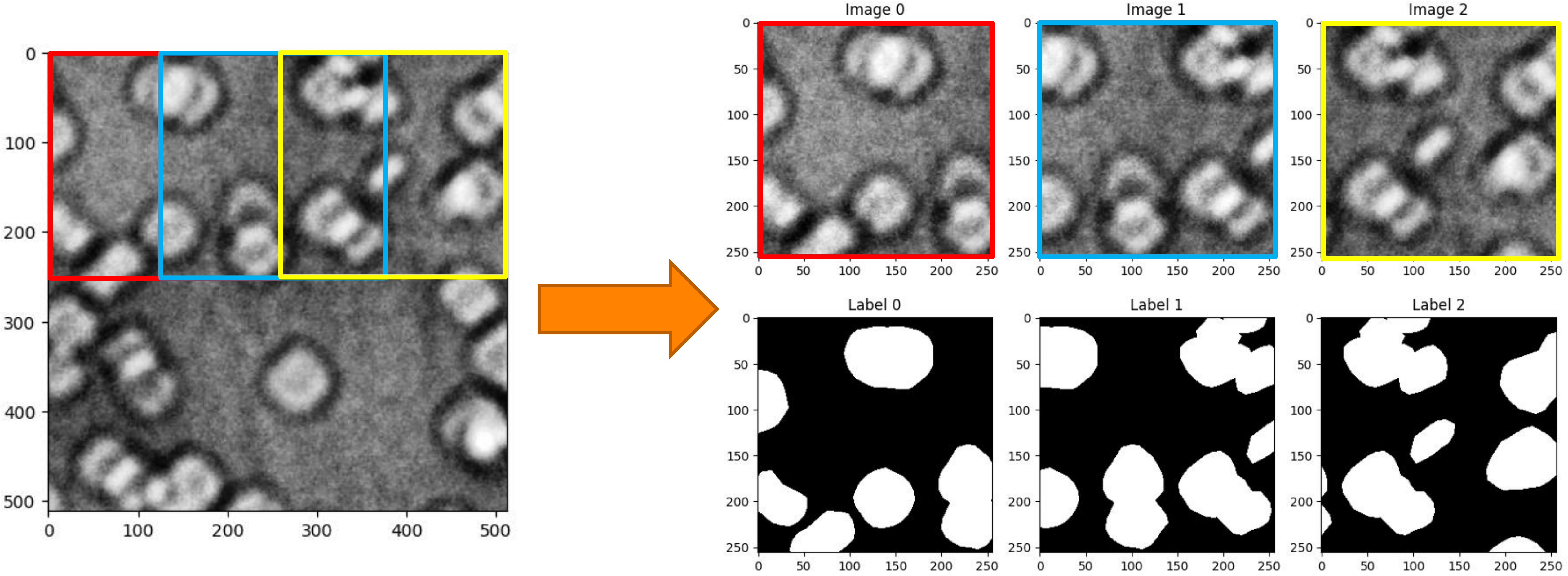
THE UNIVERSITY OF
TENNESSEE
KNOXVILLE

Rationale: more varied data → less overfitting

Electron microscopy experiments yield (relatively) small datasets

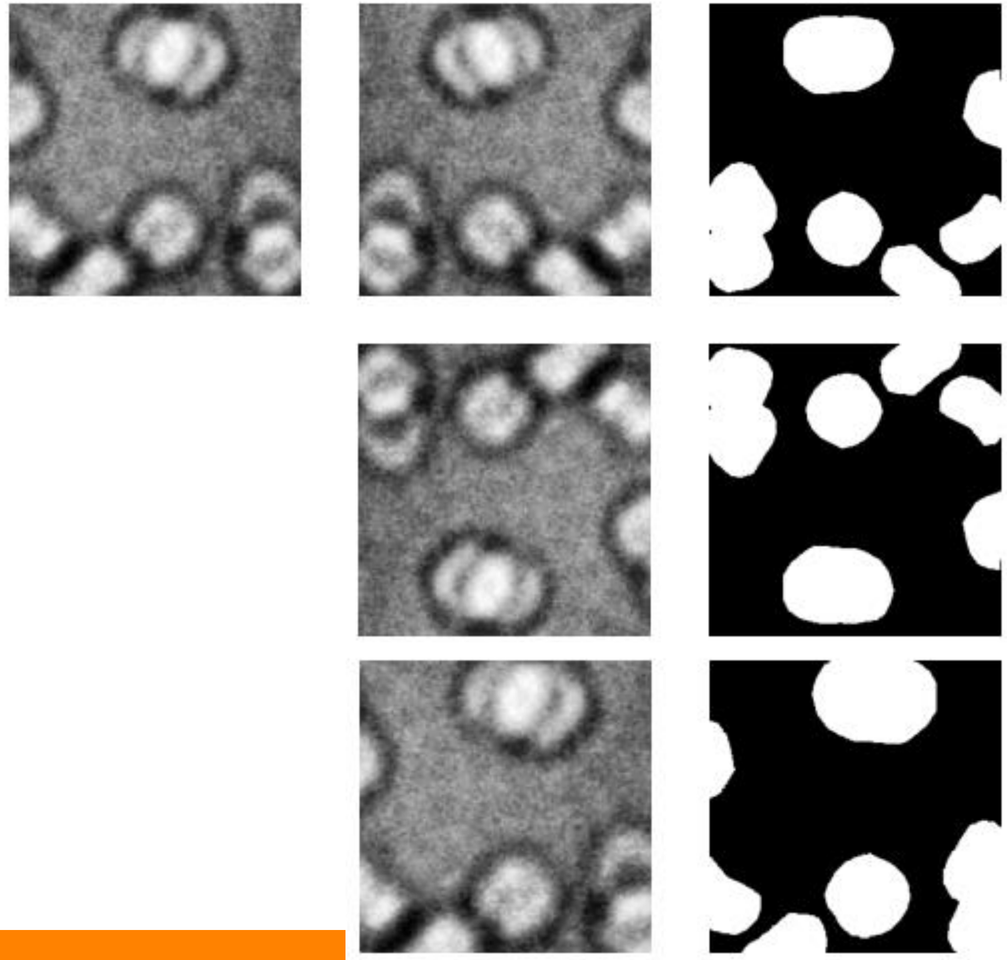
∴ requires augmentation

Use a sliding window cropper to enhance the dataset



Rotate, flip, resize the image to increase variability

Original



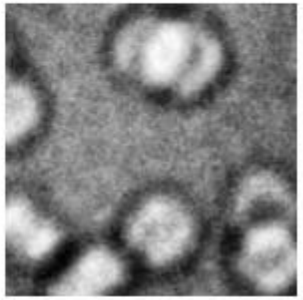
Flipped/Mirrored

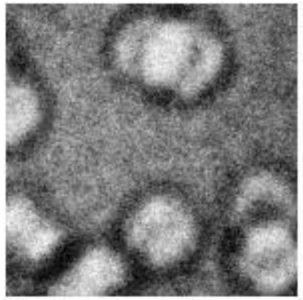
Rotated 180 degrees

Zoomed in

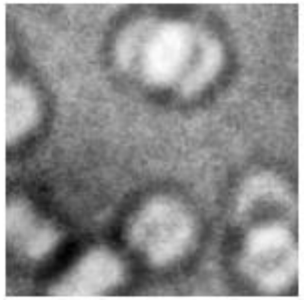
Adding noise to simulate noises during imaging

Original

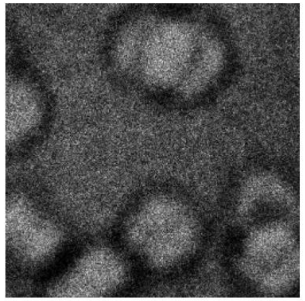




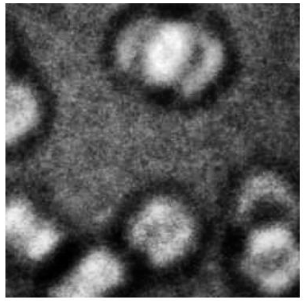
Gaussian



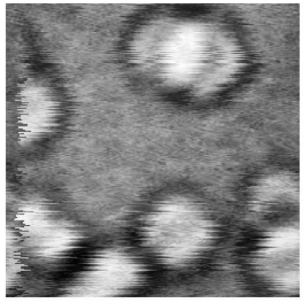
Background noise



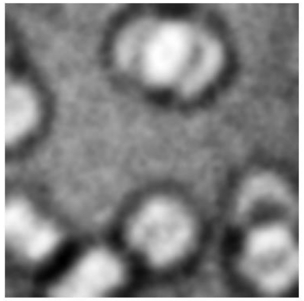
Poisson



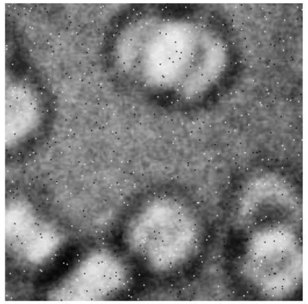
Contrast



Jitter



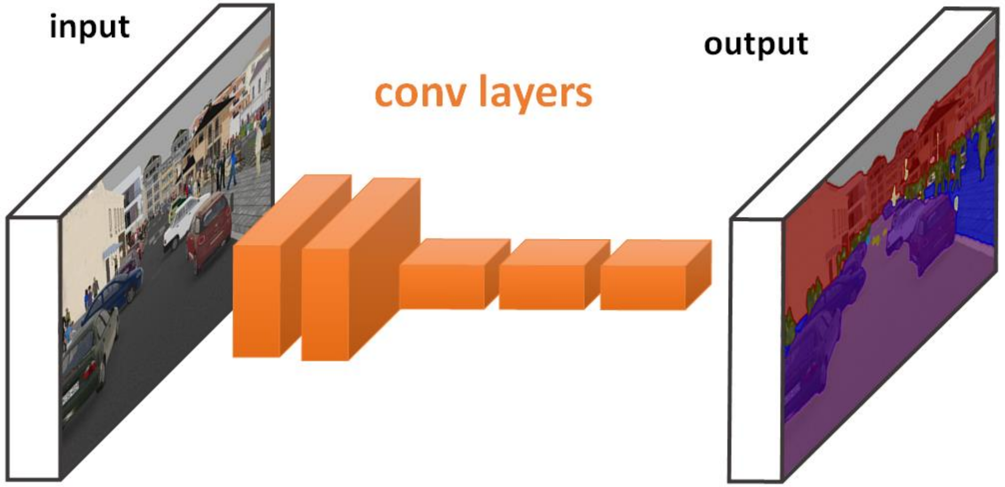
Blur



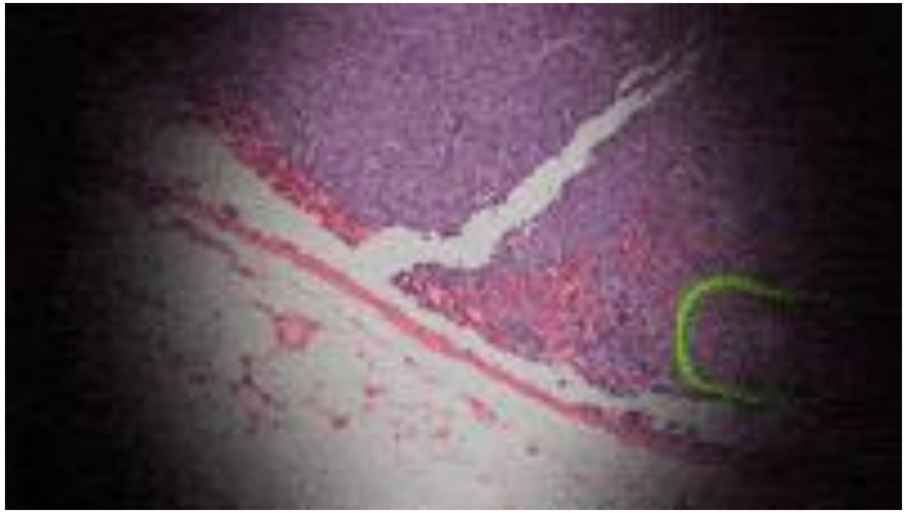
Salt & pepper

Convolutional Neural Networks (CNN)

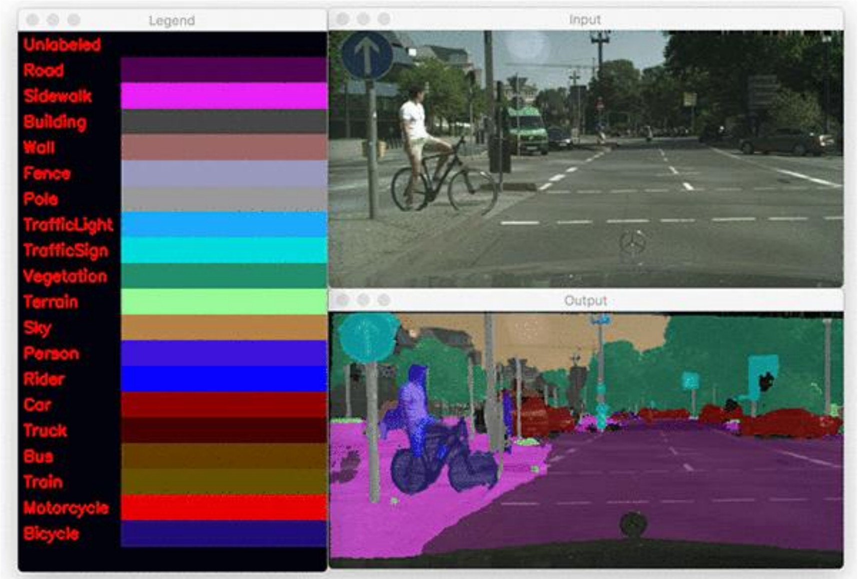
THE UNIVERSITY OF
TENNESSEE
KNOXVILLE



Semantic segmentation assigns class labels to each pixel

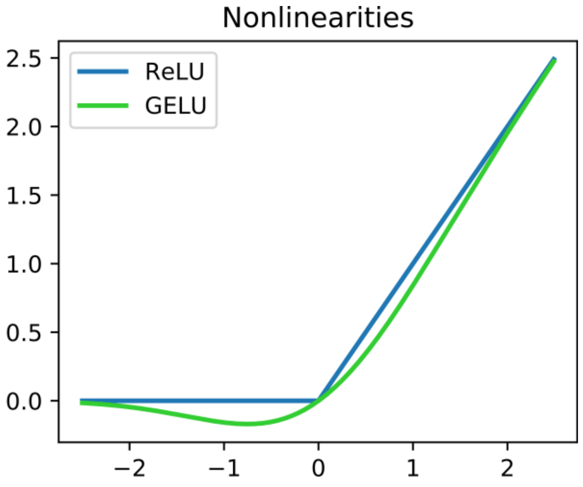


Augmented reality microscope *[4]*

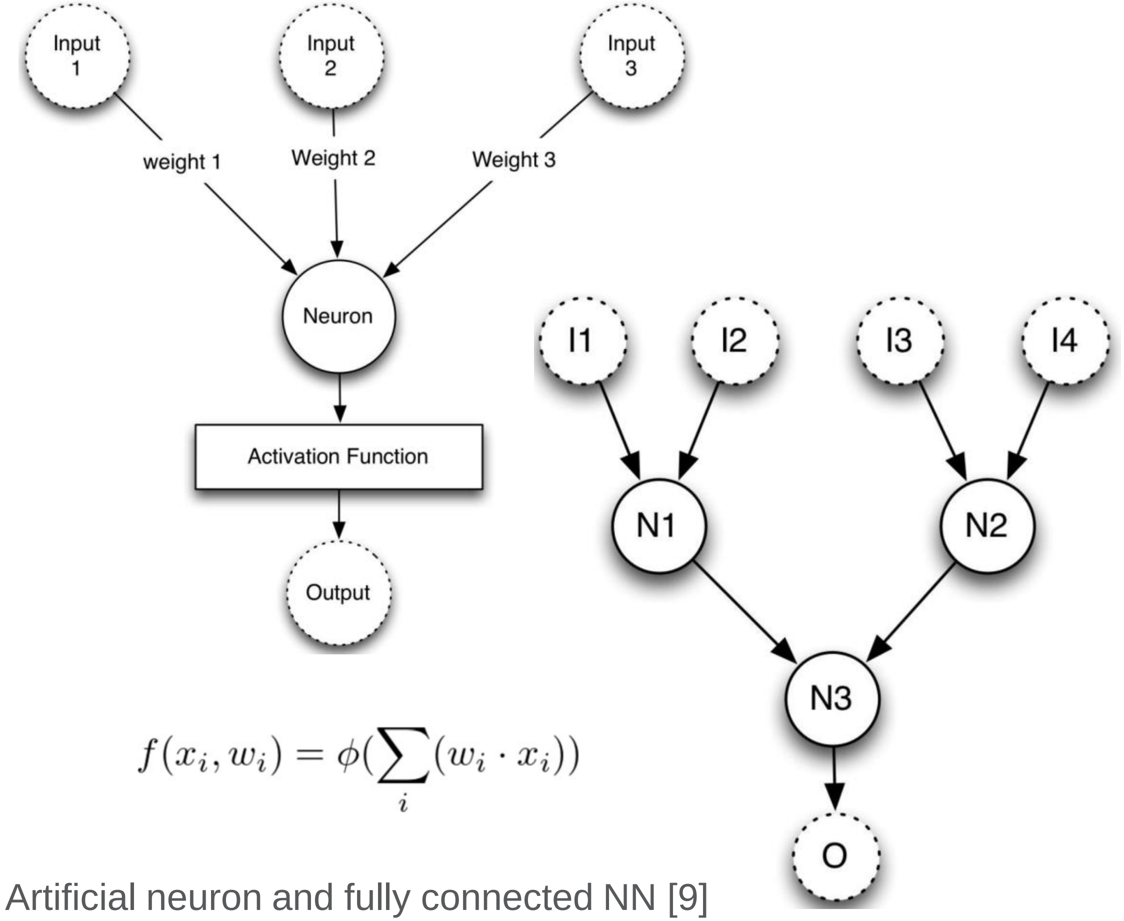


Semantic segmentation of street views *[2], [3]*

Convolutional Neural Network (CNN)



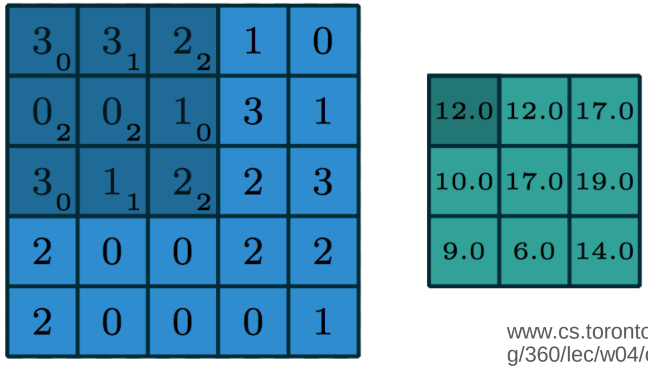
ReLU activation (Wikipedia)



$$f(x_i, w_i) = \phi(\sum_i (w_i \cdot x_i))$$

Artificial neuron and fully connected NN [9]

Convolutional Neural Network (CNN)



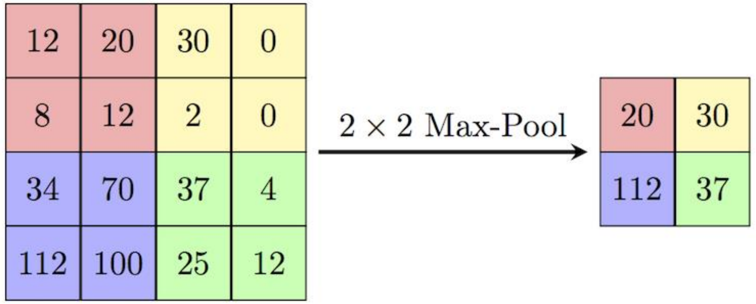
3_0	3_1	2_2	1	0
0_2	0_2	1_0	3	1
3_0	1_1	2_2	2	3
2	0	0	2	2
2	0	0	0	1

12.0	12.0	17.0
10.0	17.0	19.0
9.0	6.0	14.0

www.cs.toronto.edu/~lczhang/360/lec/w04/convnet.html

Normal convolution kernel

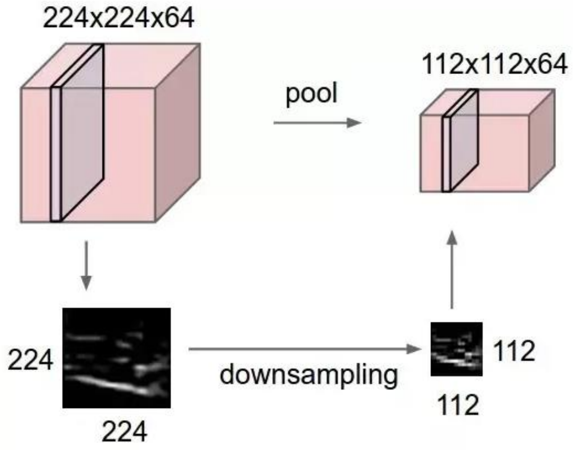
$$\begin{gathered}(3 * 0 + 3 * 1 + 2 * 2) \\ + (0 * 2 + 0 * 2 + 1 * 0) \\ + (3 * 0 + 1 * 1 + 2 * 2) \\ = 12\end{gathered}$$



12	20	30	0
8	12	2	0
34	70	37	4
112	100	25	12

2×2 Max-Pool

20	30
112	37



Maxpooling

https://computersciencewiki.org/index.php/Max-pooling_/_Pooling

Deep learning lingo

- Training data
- Validation data
- Testing data
- Epochs
- Batch size
- Hidden layer

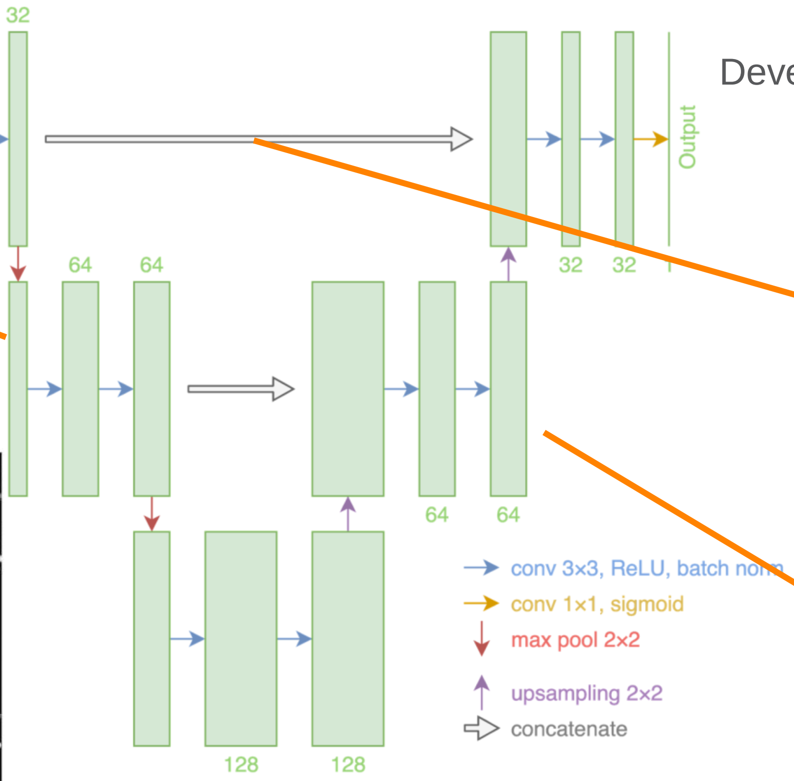
Semantic segmentation w/ U-Net

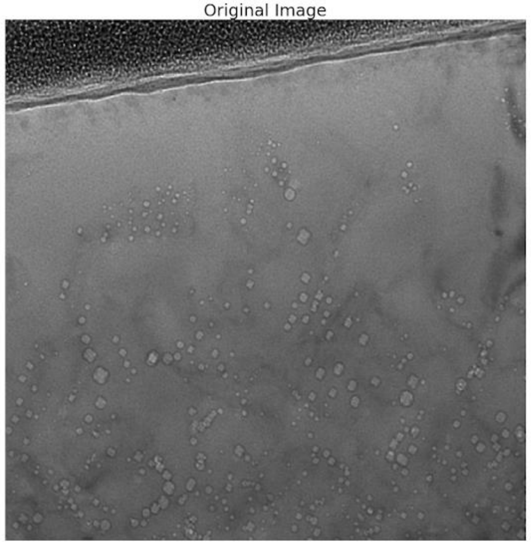
Developed by Ronneberger et al. (2015) [11]

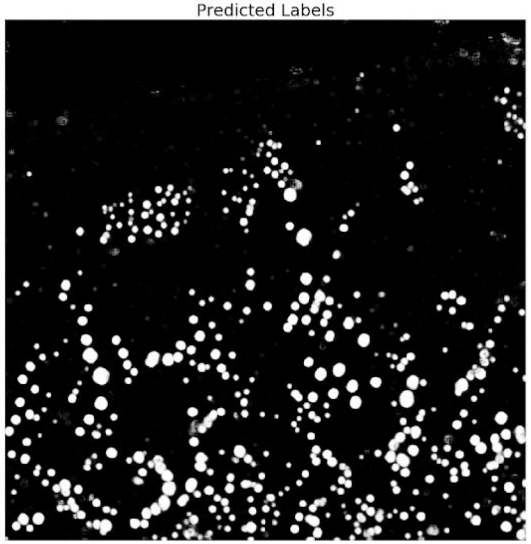
Look for salient features

Preserve “big picture”

Expand back to original size



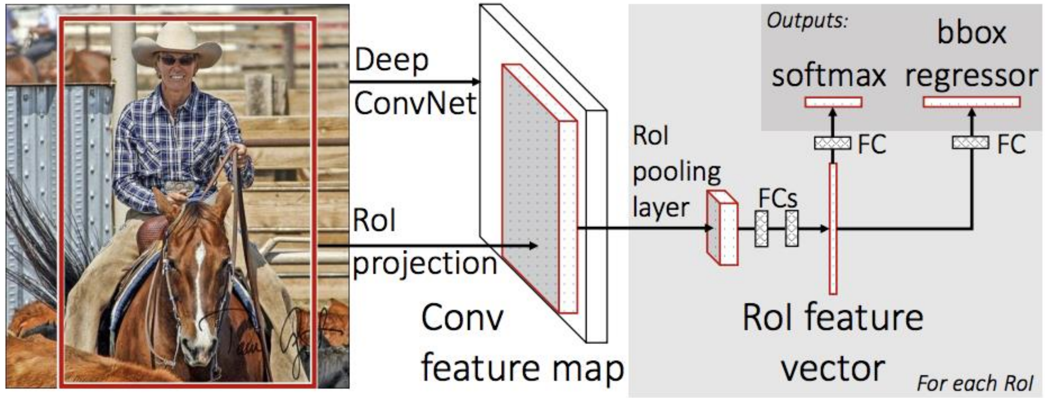


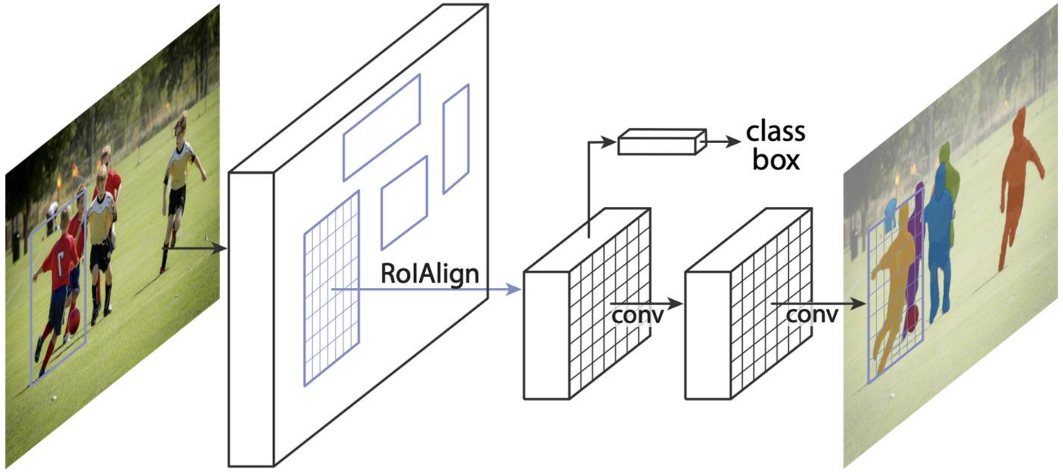


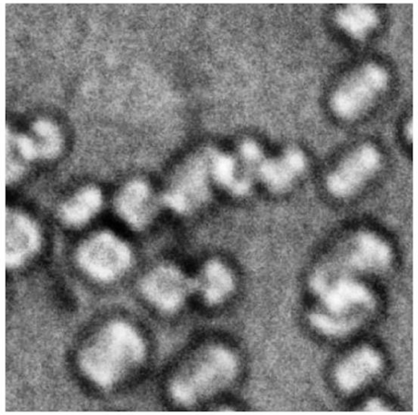
U-Net demonstration

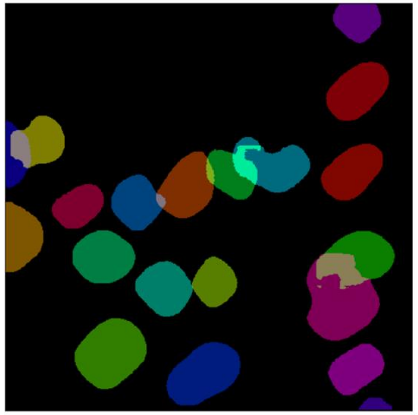
https://github.com/tommycwong/ML-ElectronMicroscopy-2023/blob/main/Lecture%2013/ML4EM_Summer2023_U_Net.ipynb

Instance segmentation w/ Mask R-CNN









Mask R-CNN demonstration

https://github.com/tommycwong/ML-ElectronMicroscopy-2023/blob/main/Lecture%2013/ML4EM_Summer2023_Mask_R_CNN.ipynb