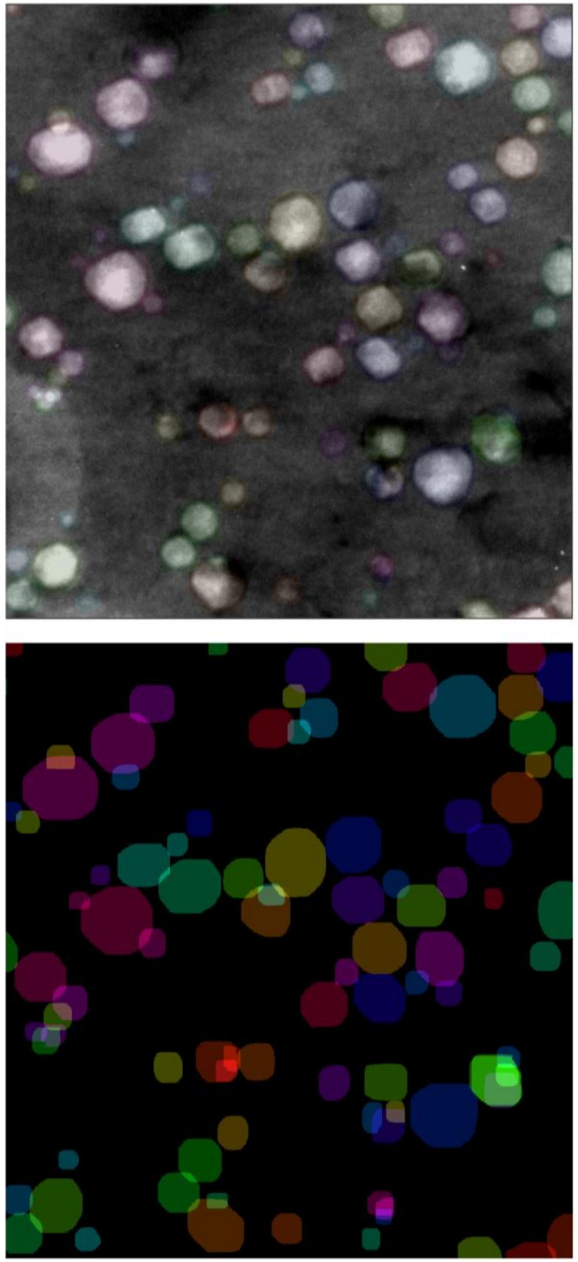
Virtual Summer School: Machine Learning in Electron Microscopy Lecture 13

Examples of DCNN in Electron Microscopy

Tommy Wong

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



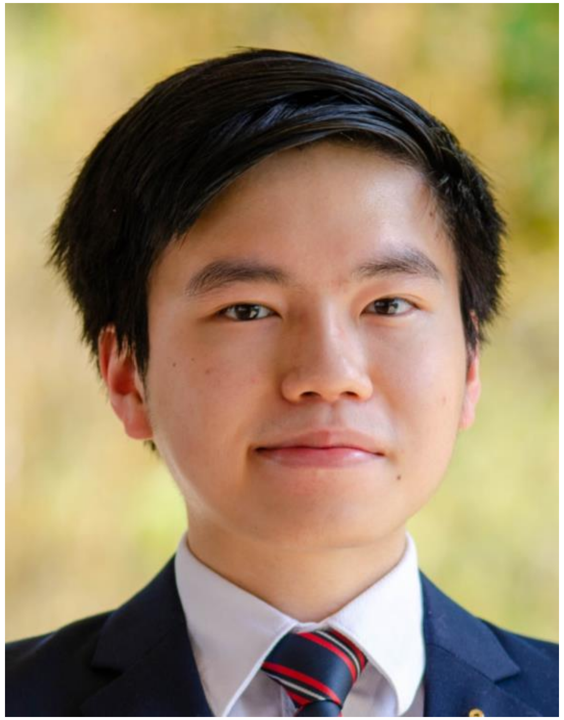
Labeling this image took ~ 1 hr, DL prediction took < 1 min.

About me

Tommy (Chun Yin) Wong

Advisors: Dr. Sergei Kalinin, Dr. Maxim Ziatdinov

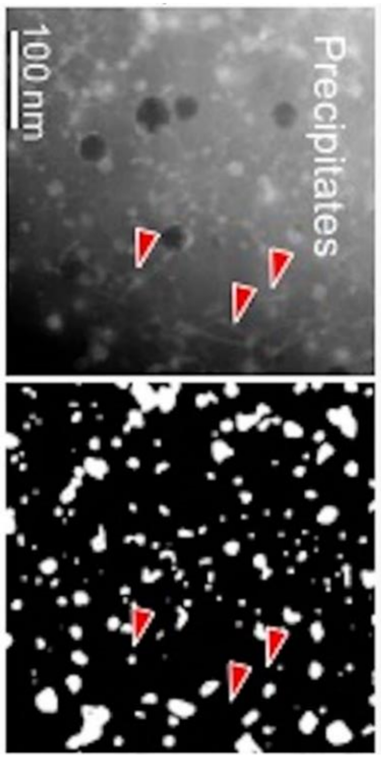
- Research interests: deep learning & computer vision, electron microscopy, radiation damage
- 3rd year Ph.D. student in Energy Science & Engineering, Bredesen Center, University of Tennessee (joint program with ORNL)
- M.S. in Materials Science & Engineering ‘22, University of Tennessee
- B.S.E. in Nuclear Engineering & Radiological Sciences ‘20, University of Michigan



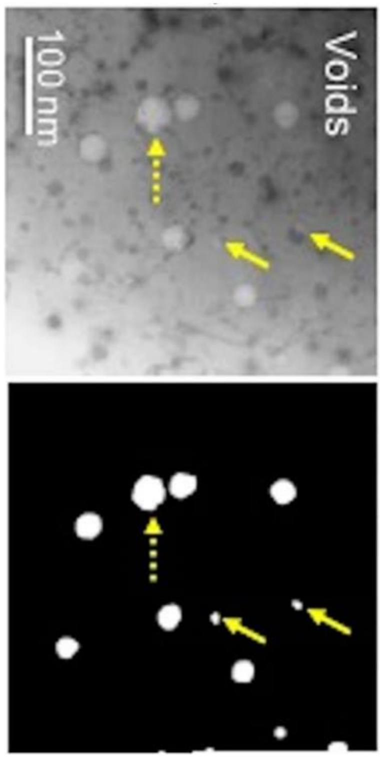
- cwong13@vols.utk.edu
- My Github: github.com/tommycwong

Microstructural defects can be labeled for pixel-wise segmentation

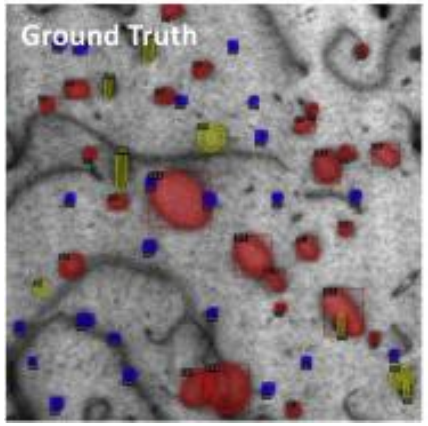
- Segmentation: associating each pixel in an image with a class



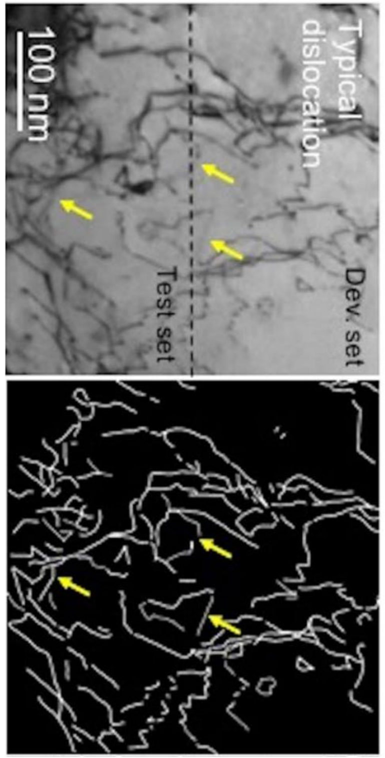
Cavities (bubbles & voids)



Precipitates



Dislocation loops

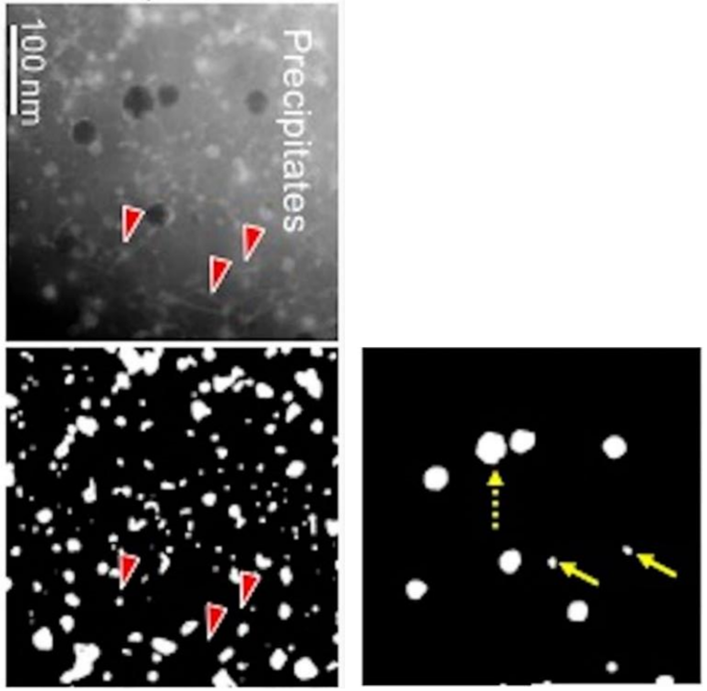


Dislocation lines

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

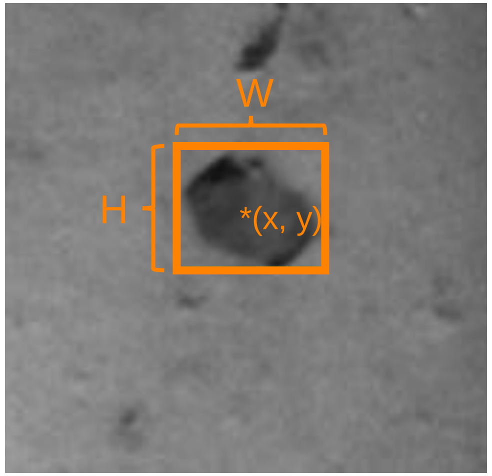
- **Important: labels are usually either 0 or 1**



Precipitates

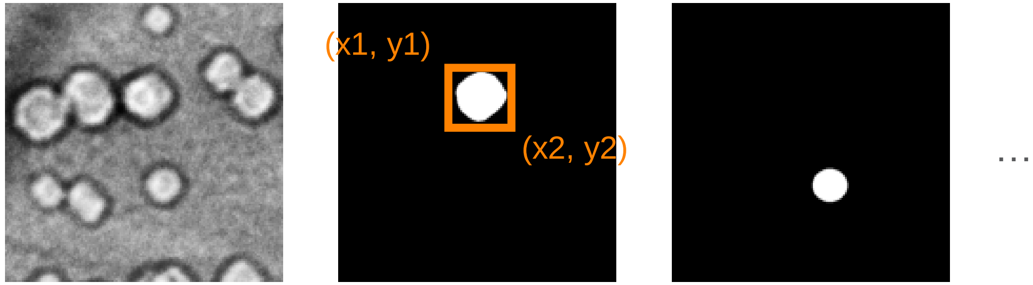
Voids

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



[Class x y W H]

Object identification: bounding box (YOLO)

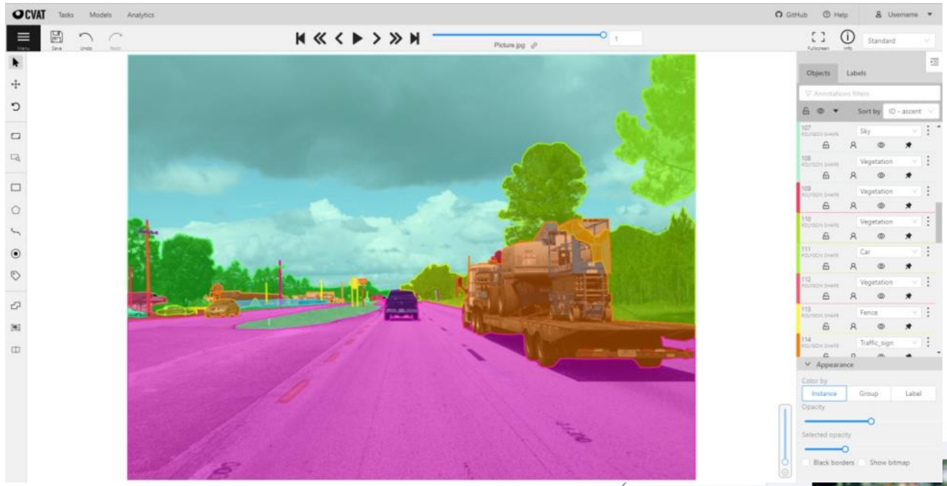


Dictionary{
'boxes': [x1 y1 x2 y2]
'labels': class
'masks': feature mask
}

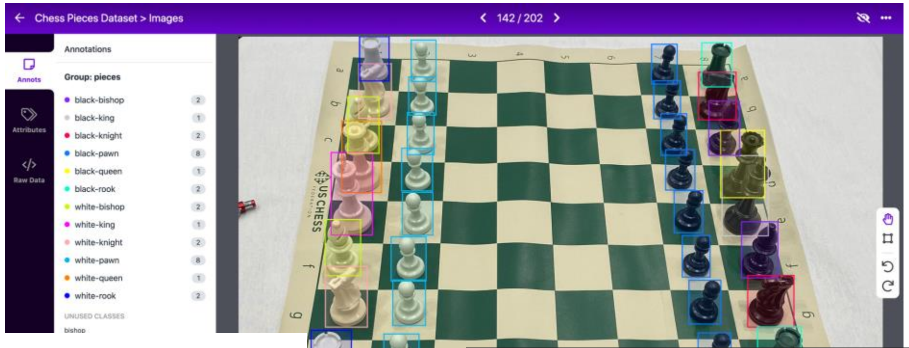
Instance segmentation: label encoding (Mask R-CNN)

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

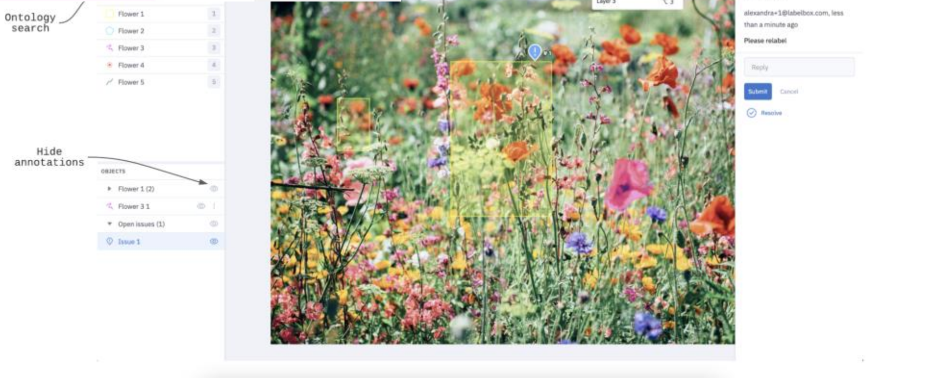
Web-based GUI tools are used for labeling



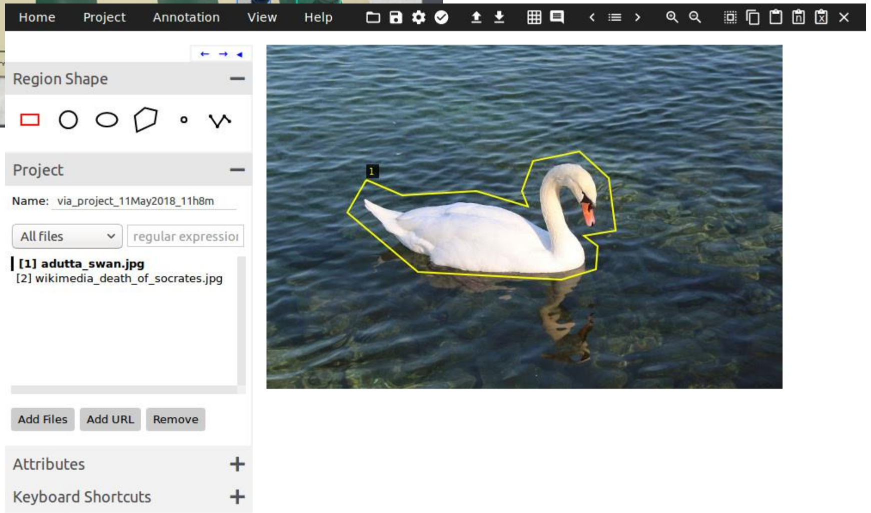
CVAT



Roboflow



Labelbox



Name: via_project_11May2018_11h8m

All files | regular expression

[1] adutta_swan.jpg
[2] wikimedia_death_of_socrates.jpg

VIA

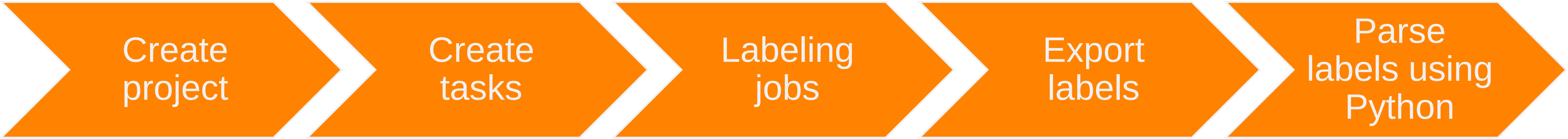
Labeling using Computer Vision Annotation Tool (CVAT)

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

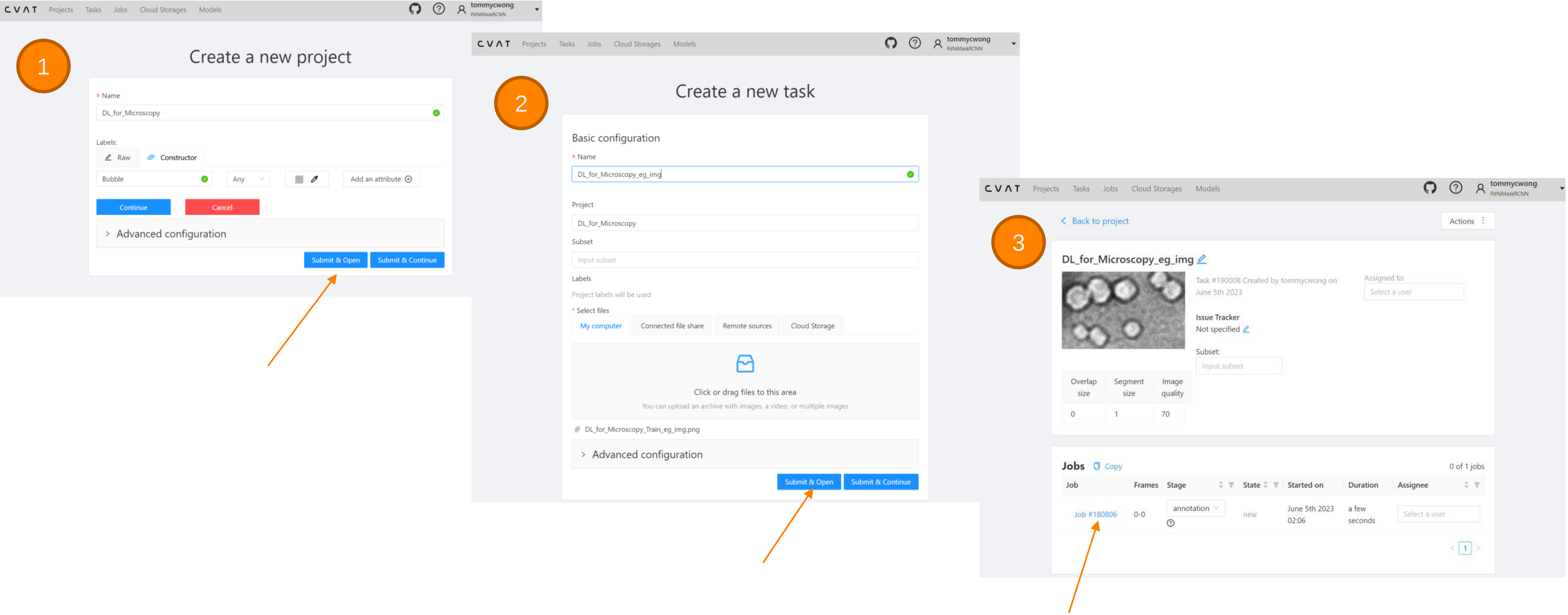
Create tasks

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 — 0 of 1 jobs

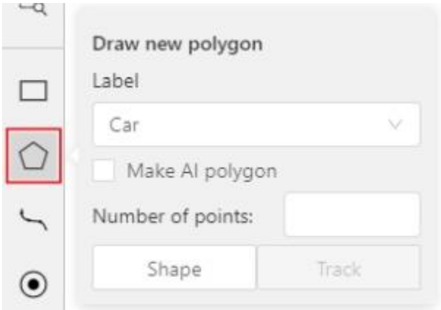
Job	Frames	Stage	State	Started on	Duration	Assignee
Job #180806	0-0	annotation	new	June 5th 2023 02:06	a few seconds	Select a user

Labeling using polygon and ellipse tools

Remember to click **Save**

Polygon tool

- Hold **Shift** to draw

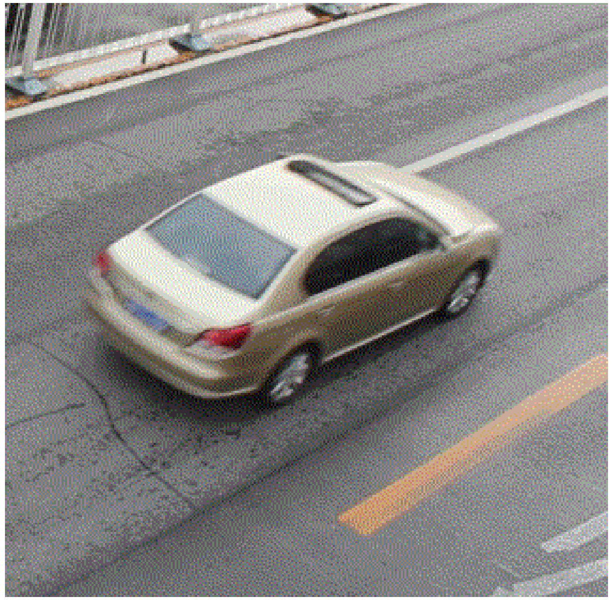


Draw new polygon

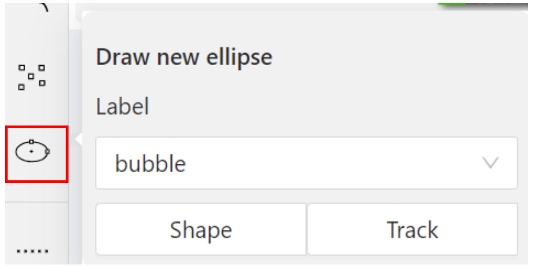
Label: Car

□ Make AI polygon

Number of points:

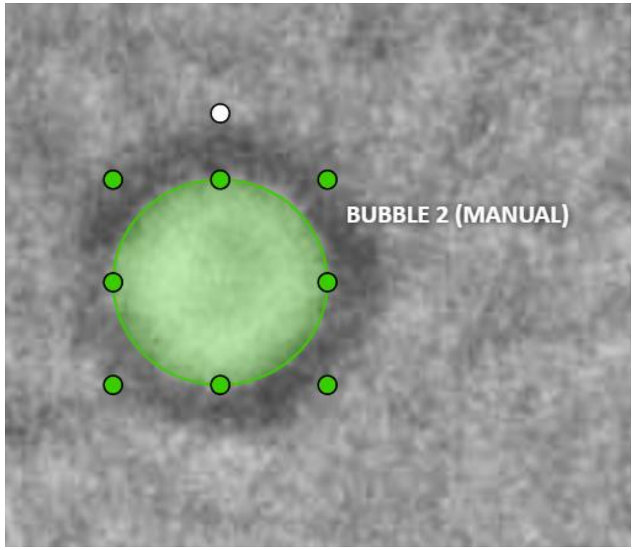


Ellipse tool



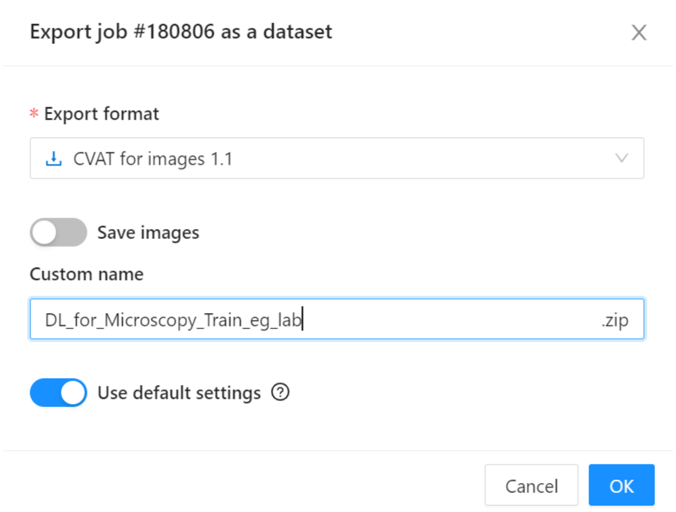
Draw new ellipse

Label: bubble



Exporting and parsing labels

Export as .xml



Export job #180806 as a dataset

* Export format

CVAT for images 1.1

Save images

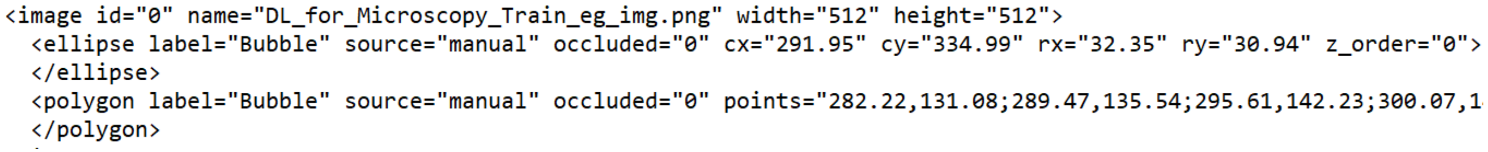
Custom name

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/SergeiVKalinin/ML-ElectronMicroscopy-2023/blob/main/Lecture%2013/ML4EM_Summer2023_Data_Augmentation_supp.ipynb

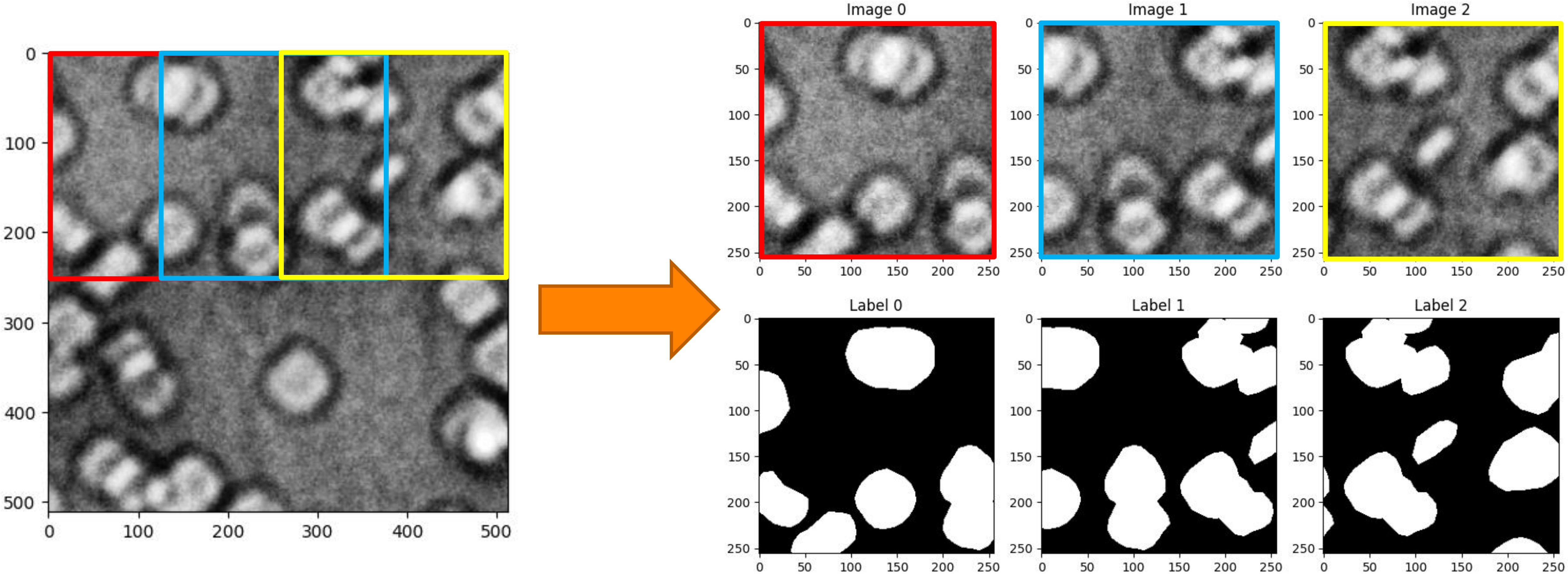
THE UNIVERSITY OF
TENNESSEE
KNOXVILLE

Rationale: more varied data → less overfitting

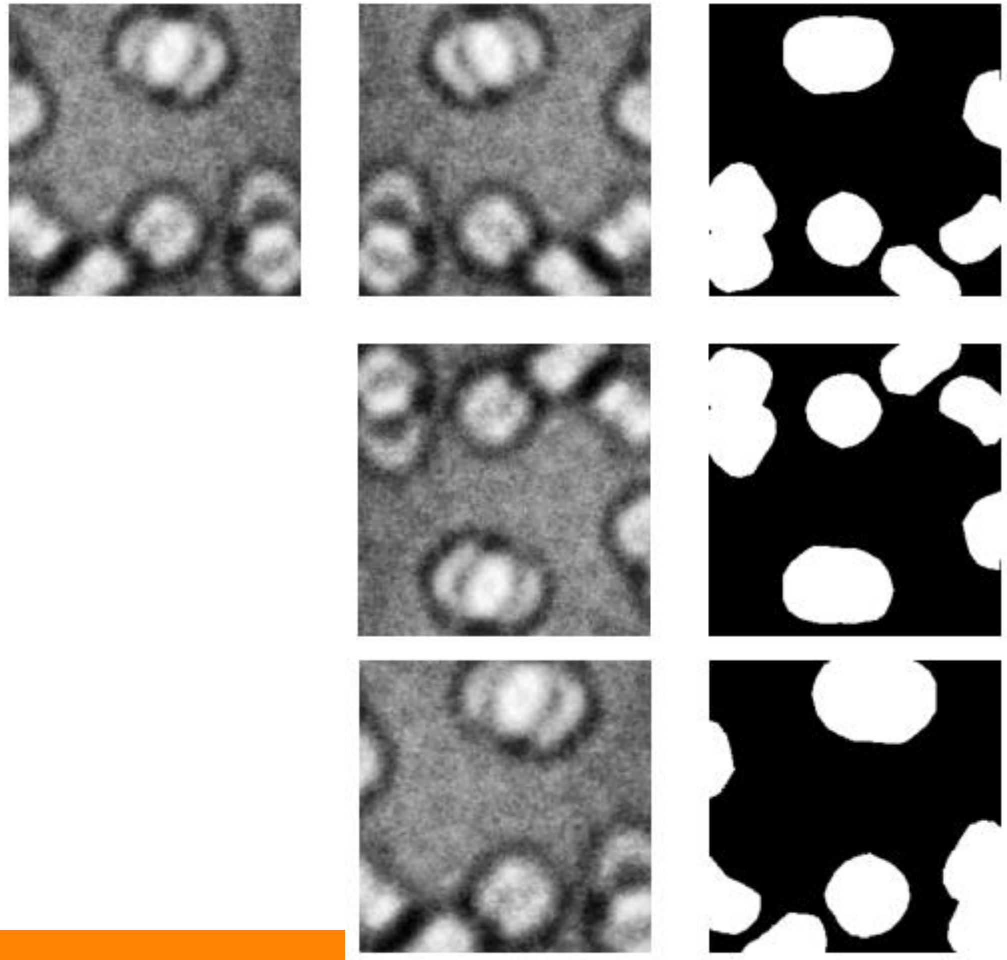
Electron microscopy experiments yield (relatively) small datasets

∴ requires augmentation

Use a sliding window cropper to enhance the dataset



Rotate, flip, resize the image to increase variability



Original

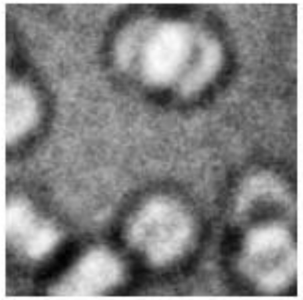
Flipped/Mirrored

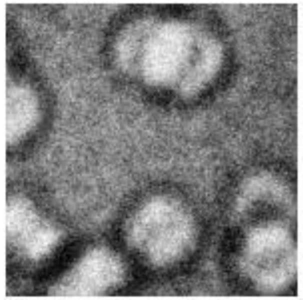
Rotated 180 degrees

Zoomed in

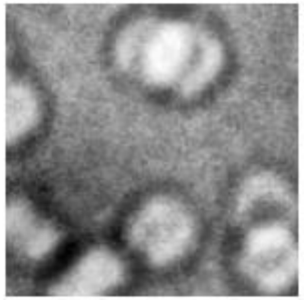
Adding noise to simulate noises during imaging

Original

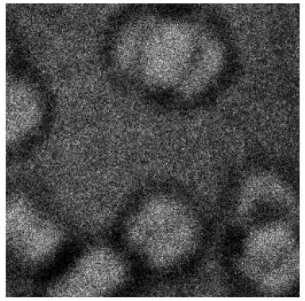




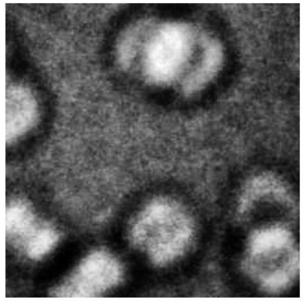
Gaussian



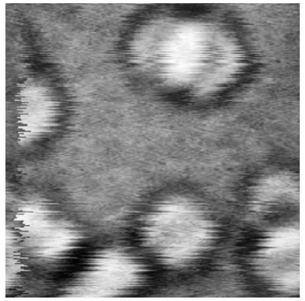
Background noise



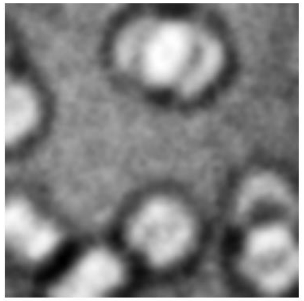
Poisson



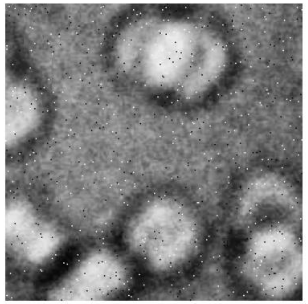
Contrast



Jitter



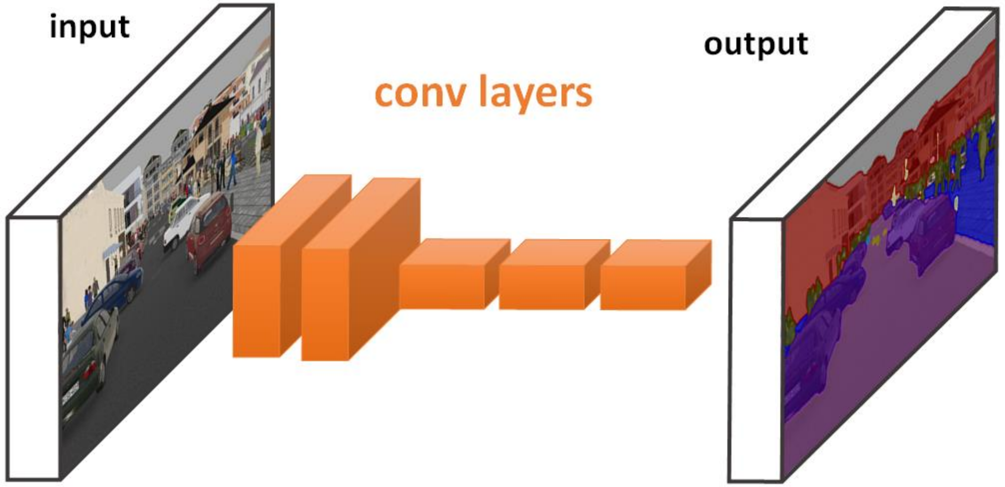
Blur



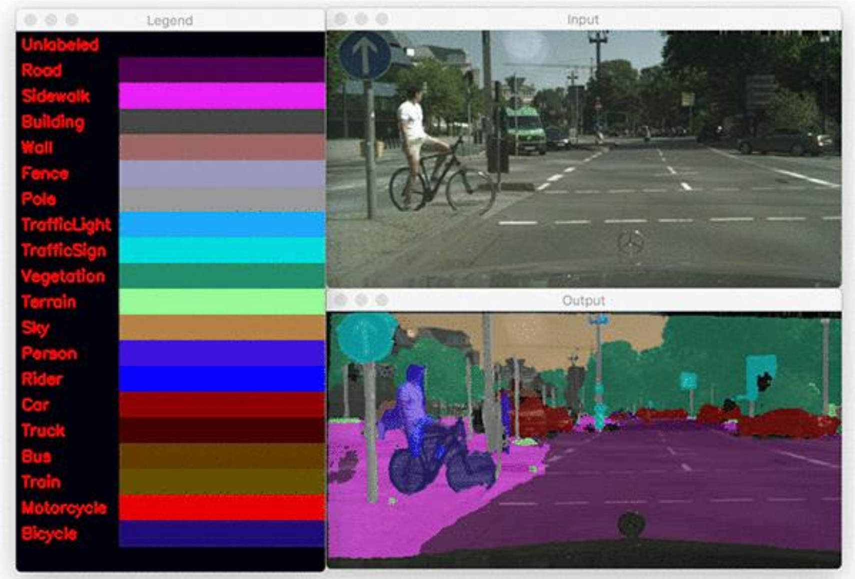
Salt & pepper

Convolutional Neural Networks (CNN)

THE UNIVERSITY OF
TENNESSEE
KNOXVILLE

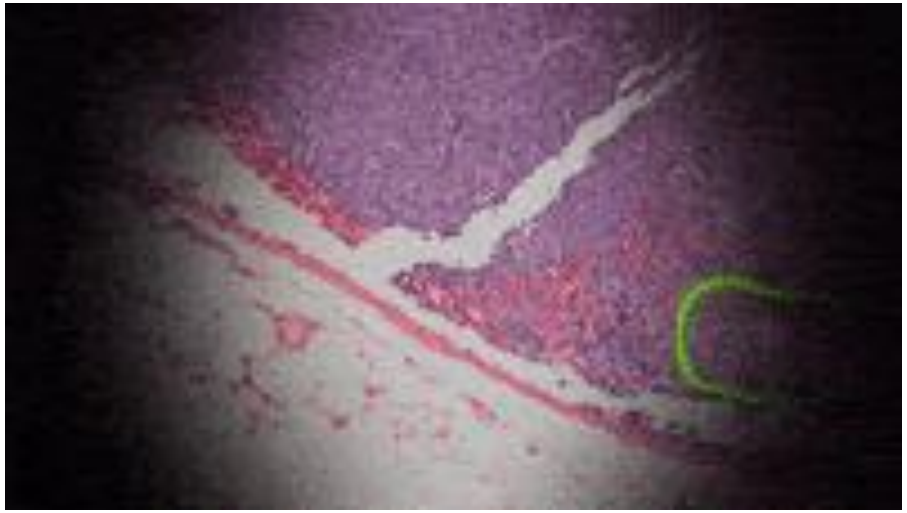


Semantic segmentation assigns class labels to each pixel



Unlabeled
Road
Sidewalk
Building
Wall
Fence
Pole
TrafficLight
TrafficSign
Vegetation
Terrain
Sky
Person
Rider
Car
Truck
Bus
Train
Motorcycle
Bicycle

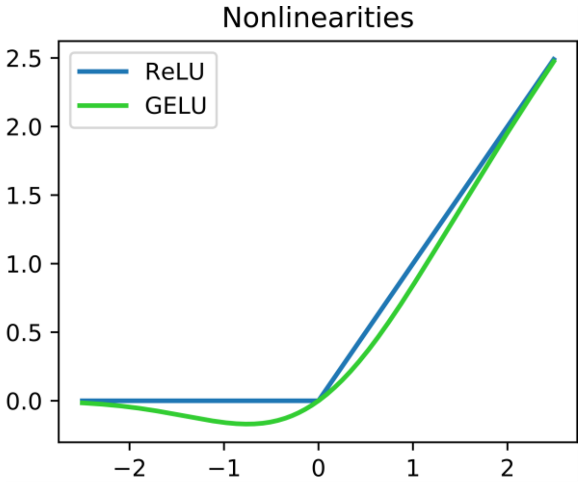
Semantic segmentation of street views *[1], [2]*



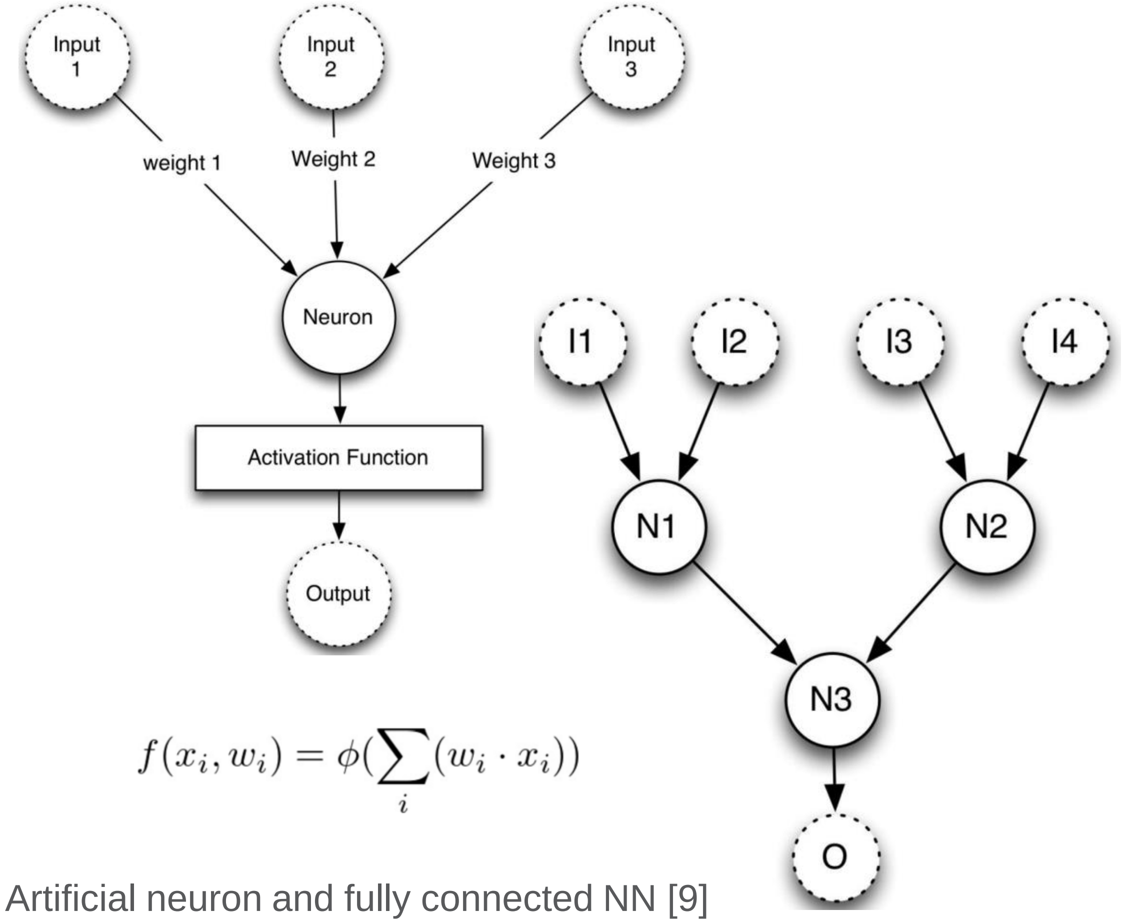
Augmented reality microscope *[3]*

[1] G. Heinrich, “Image Segmentation Using DIGITS 5,” NVIDIA Developer Blog, Nov. 10, 2016. https://developer.nvidia.com/blog/image-segmentation-using-digits-5/ (accessed Feb. 25, 2021).
[2] M. Cordts et al., “The Cityscapes Dataset for Semantic Urban Scene Understanding,” in 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, Jun. 2016, pp. 3213–3223, doi: 10.1109/CVPR.2016.350.
[3] P.-H. C. Chen et al., “Microscope 2.0: An Augmented Reality Microscope with Real-time Artificial Intelligence Integration,” arXiv:1812.00825 [cs], Dec. 2018, doi: 10.1038/s41591-019-0539-7.

Convolutional Neural Network (CNN)



Activation function: ReLU (Wikipedia)

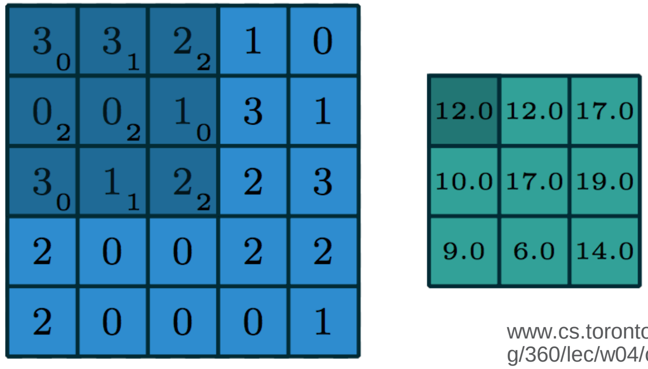


$$f(x_i, w_i) = \phi(\sum_i (w_i \cdot x_i))$$

Artificial neuron and fully connected NN [9]

Adapted from Heaton (2015) ISBN:1505714346

Convolutional Neural Network (CNN)



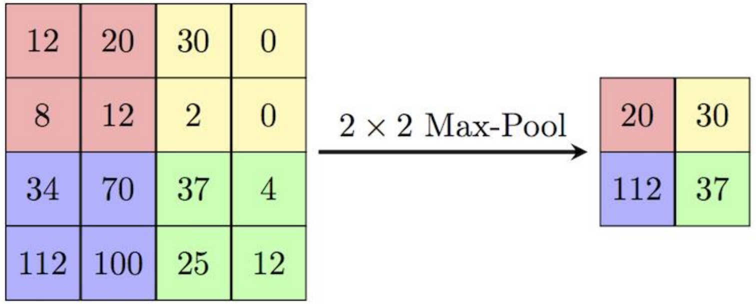
3_0	3_1	2_2	1	0
0_2	0_2	1_0	3	1
3_0	1_1	2_2	2	3
2	0	0	2	2
2	0	0	0	1

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

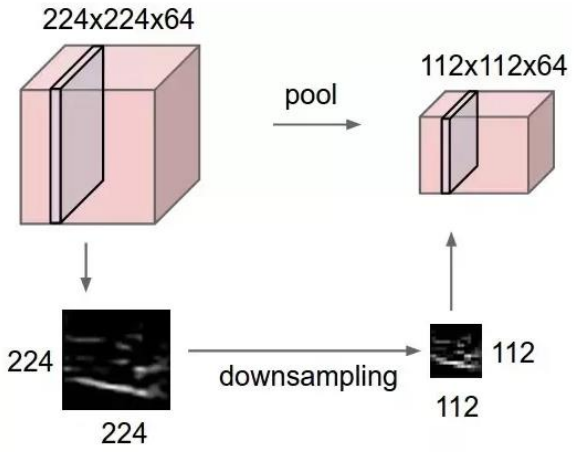
Normal convolution kernel

$$\begin{aligned}&(3 \ast 0 + 3 \ast 1 + 2 \ast 2)\\ &+ (0 \ast 2 + 0 \ast 2 + 1 \ast 0)\\ &+ (3 \ast 0 + 1 \ast 1 + 2 \ast 2)\\ &= 12\end{aligned}$$



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

20	30
112	37



Maxpooling

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

Deep learning lingo

- Training data—used for training
- Validation data—for predictions *during* training
- Testing data—for predictions *after* training
- Ground truth—actual true labels
- Epoch—1 training cycle
- Batch size—no. of training images each epoch
- Hidden layer—layers between input and output layers
- Learning rate—how big a step an optimizer goes each epoch

Deep learning lingo

Loss function

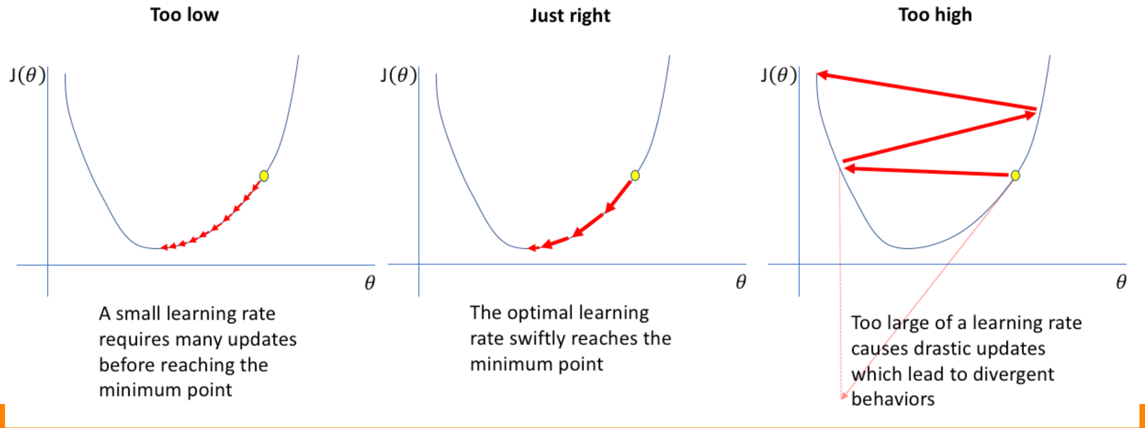
- Evaluates how close the predictions are to the ground truth → Minimize
- Binary cross entropy: (dis)similarity between prediction and GT

$$H_p(q) = -\frac{1}{N}\sum_{i=1}^{N} y_i \cdot log(p(y_i)) + (1 - y_i) \cdot log(1 - p(y_i))$$

https://towardsdatascience.com/understanding-binary-cross-entropy-log-loss-a-visual-explanation-a3ac6025181a

Optimizer

- Controls the *convergence* of a model (by minimizing loss func.)
- Step controlled by learning rate
- Adam optimizer



A small learning rate requires many updates before reaching the minimum point

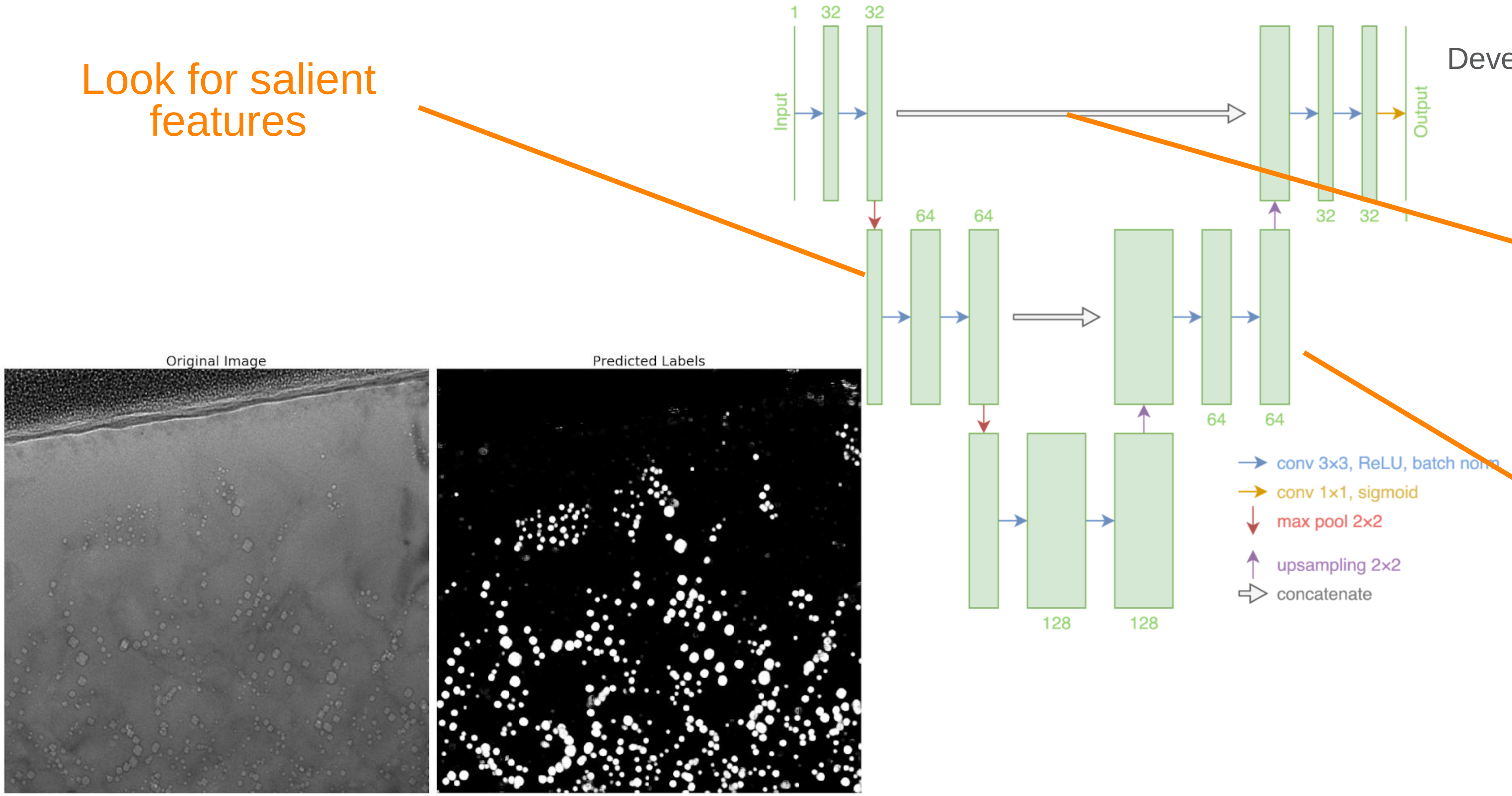
The optimal learning rate swiftly reaches the minimum point

Too large of a learning rate causes drastic updates which lead to divergent behaviors

https://www.jeremyjordan.me/nn-learning-rate/

Semantic segmentation w/ U-Net

Look for salient features



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

Preserve “big picture”

Expand back to original size

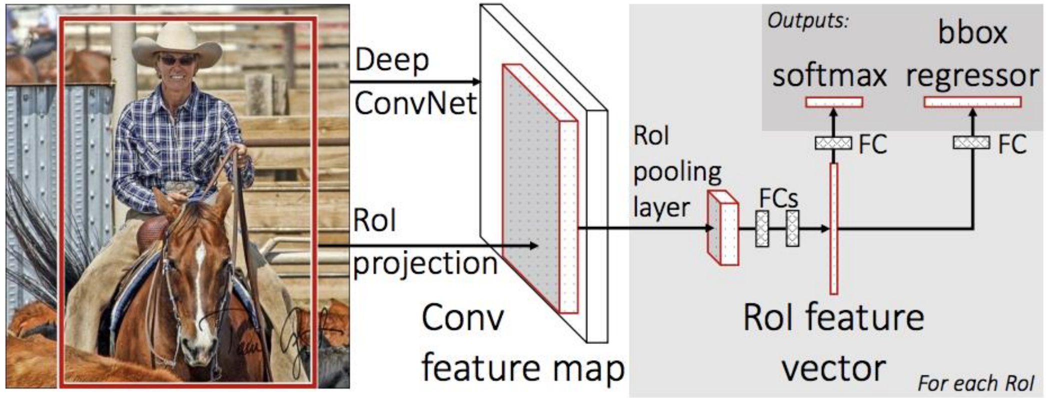
Adapted from Wong et al. (2021) https://doi.org/10.1017/S1431927621007686

U-Net demonstration

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

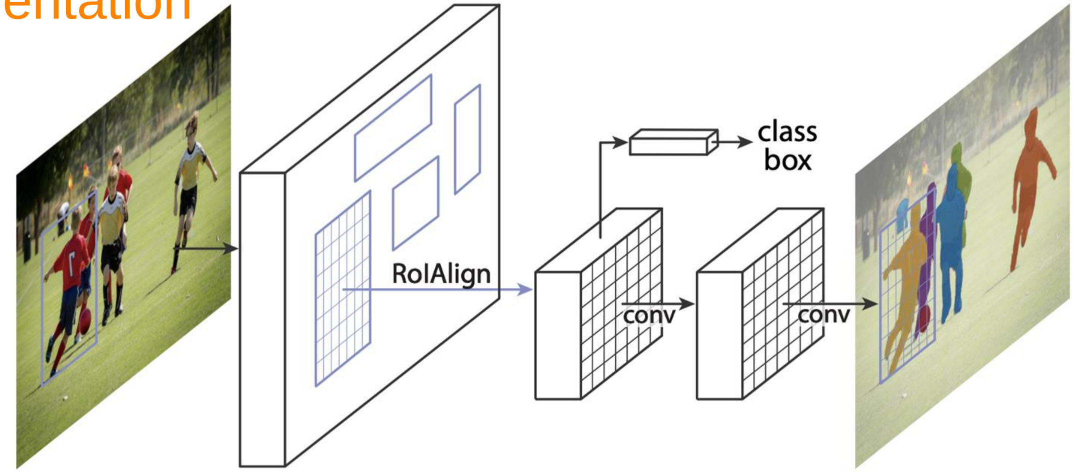
Instance segmentation w/ Mask R-CNN

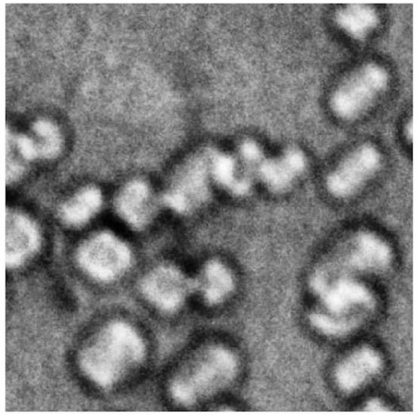
Fast R-CNN

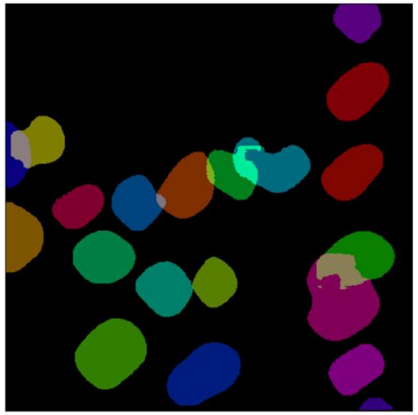


+ segmentation

Mask R-CNN







$$Loss = L_{class} + L_{box} + L_{mask}$$

Adapted from Girschick (2015) https://arxiv.org/abs/1504.08083, He et al. (2018) https://arxiv.org/abs/1703.06870v3

Mask R-CNN demonstration

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