

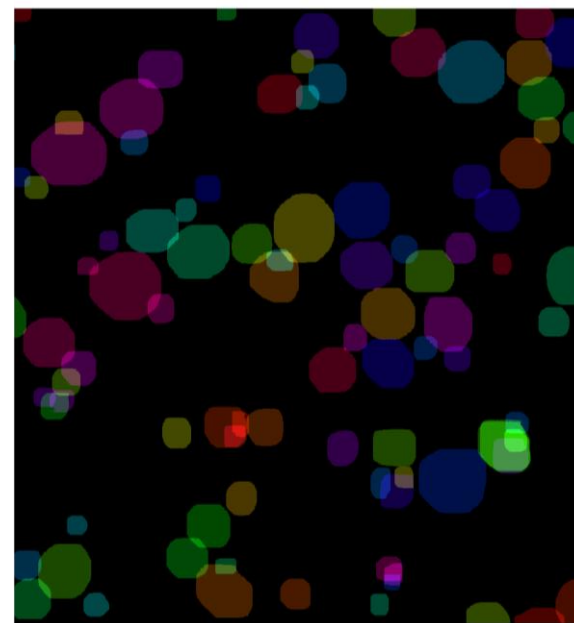
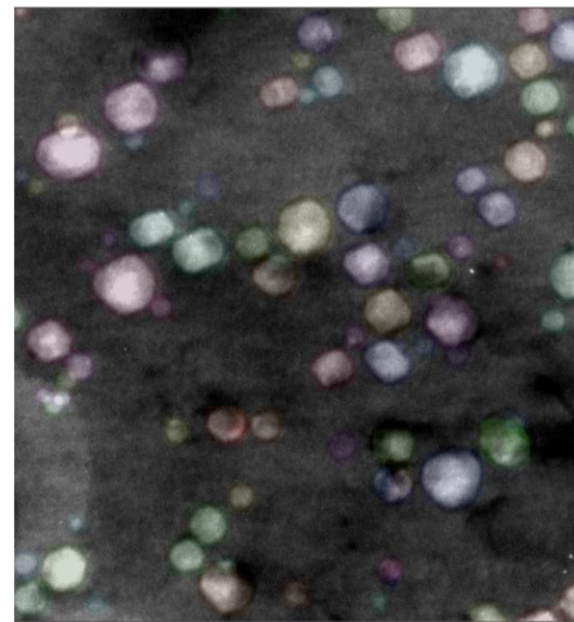
*Virtual Summer School: Machine Learning in
Electron Microscopy*

Examples of DCNN in Electron Microscopy

Tommy Wong

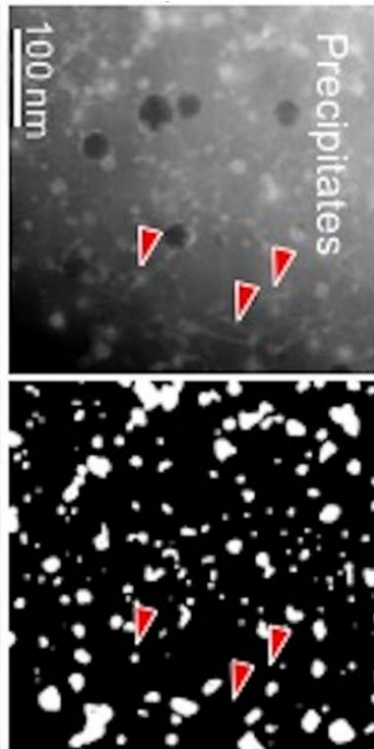
Github:

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

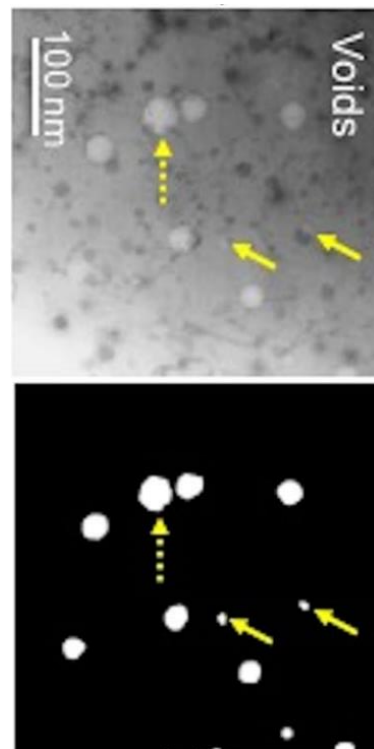


Microstructural defects can be labeled for pixel-wise segmentation

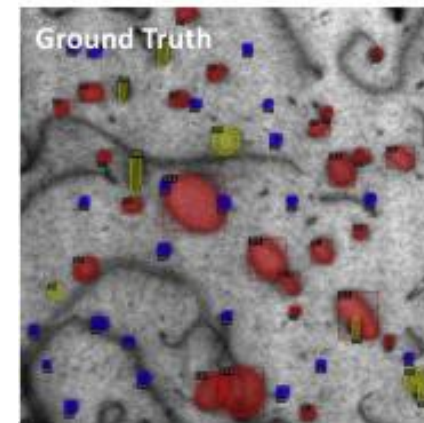
- Segmentation: associating each pixel in an image with a class



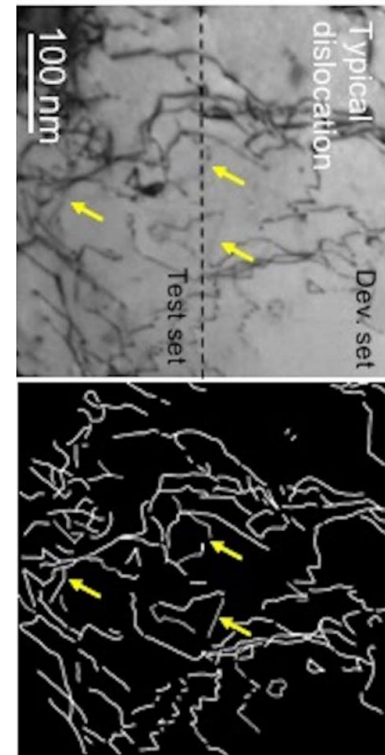
Cavities (bubbles & voids)



Precipitates



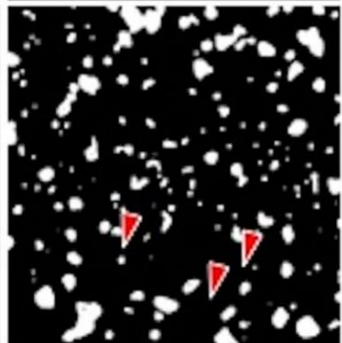
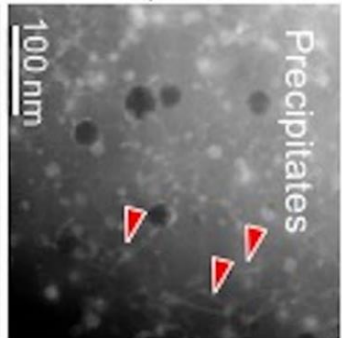
Dislocation loops



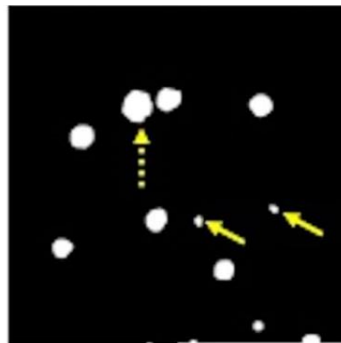
Dislocation lines

Different labeling systems are required for different segmentation algorithms

- Important: labels are usually either 0 or 1

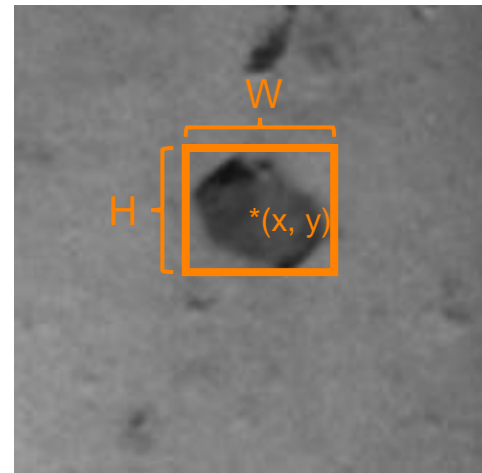


Precipitates



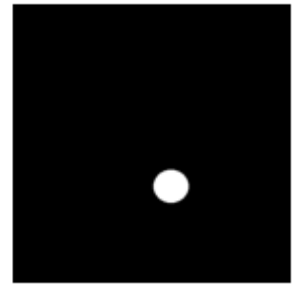
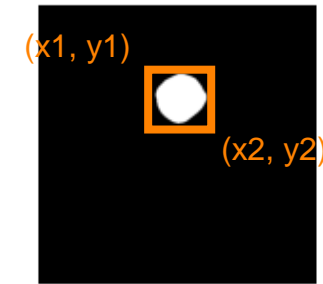
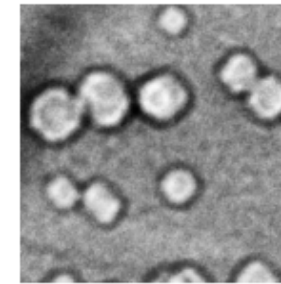
Voids

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



[Class x y W H]

Object identification: bounding box (YOLO)

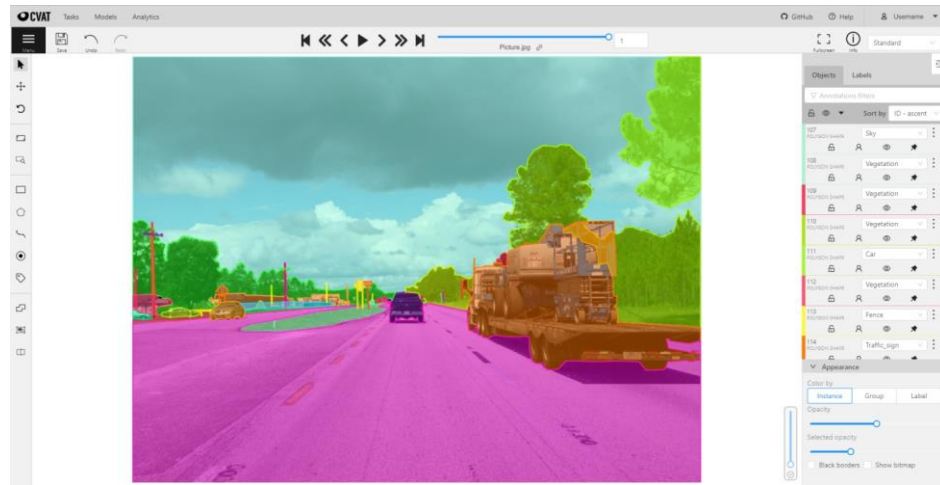


...

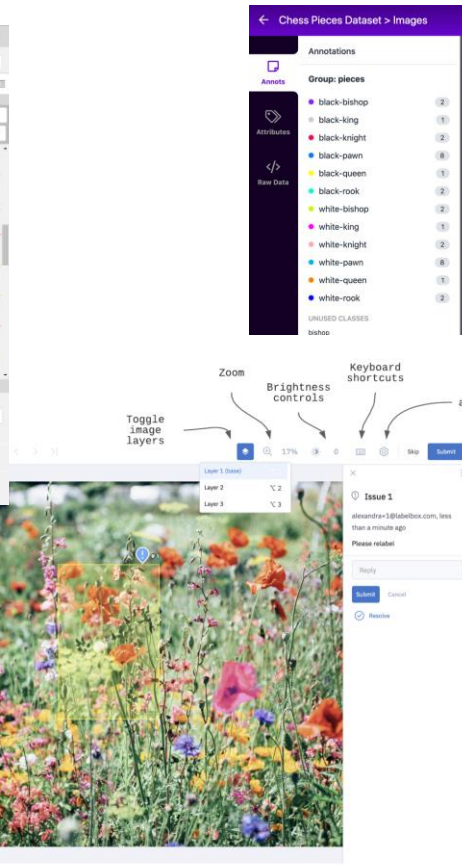
```
Dictionary{  
  'boxes': [x1 y1 x2 y2]  
  'labels': class  
  'masks': feature mask  
}
```

Instance segmentation: label encoding (Mask R-CNN)

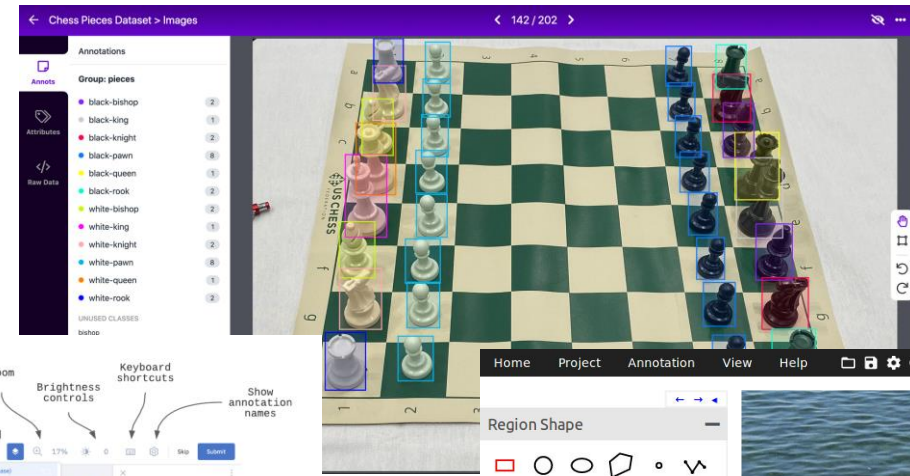
Web-based GUI tools are used for labeling



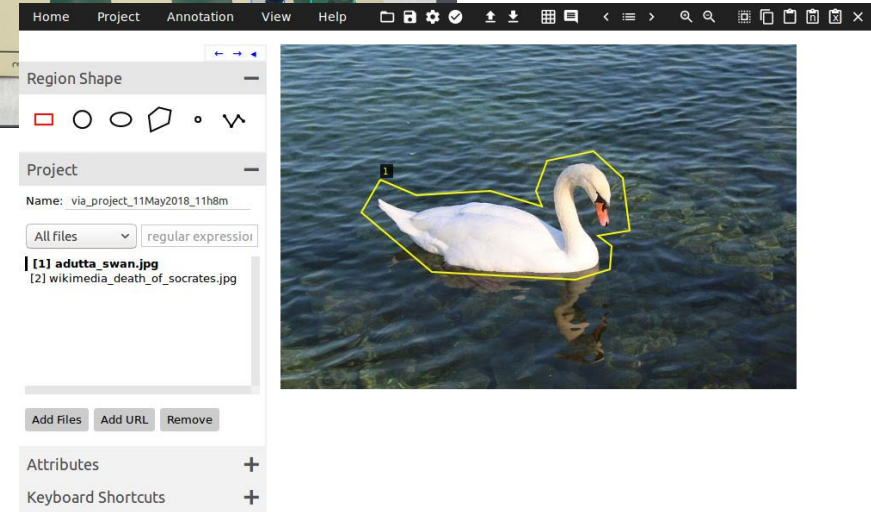
CVAT



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



Creating a project and labeling tasks

The image displays three sequential screenshots of the CVAT (Computer Vision Annotation Tool) web interface, illustrating the process of creating a project and labeling tasks. Each screenshot is marked with a numbered orange circle (1, 2, and 3) and an orange arrow pointing to a specific action.

Screenshot 1: Create a new project
The user is on the 'Create a new project' page. The 'Name' field is filled with 'DL_for_Microscopy'. Under the 'Labels' section, the 'Raw' tab is selected, and 'Bubble' is chosen as the label type. The 'Submit & Open' button is highlighted with an orange arrow.

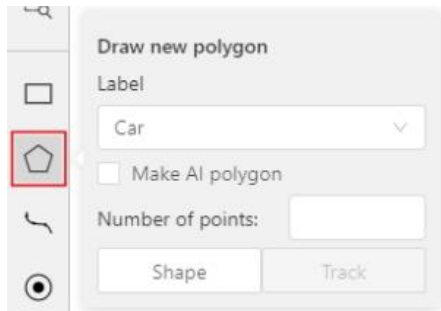
Screenshot 2: Create a new task
The user is on the 'Create a new task' page. The 'Name' field is filled with 'DL_for_Microscopy_eg_img'. The 'Project' field is set to 'DL_for_Microscopy'. The 'Subset' field is set to 'Input subset'. The 'Labels' section shows 'Project labels will be used'. The 'Select files' section has 'My computer' selected. A file upload area is visible with the text 'Click or drag files to this area'. The 'Submit & Open' button is highlighted with an orange arrow.

Screenshot 3: Task details and jobs
The user is on the task details page for 'DL_for_Microscopy_eg_img'. The task is identified as 'Task #190008 Created by tommycwong on June 5th 2023'. The 'Issue Tracker' is 'Not specified'. The 'Subset' is 'Input subset'. Below the task details, there is a table of jobs. The 'Jobs' section shows '0 of 1 jobs'. The table has columns: Job, Frames, Stage, State, Started on, Duration, and Assignee. One job is listed: 'Job #180806' with 0-0 frames, in the 'annotation' stage, with a 'new' state, started on 'June 5th 2023 02:06', with a duration of 'a few seconds', and an assignee of 'Select a user'. The 'Job #180806' is highlighted with an orange arrow.

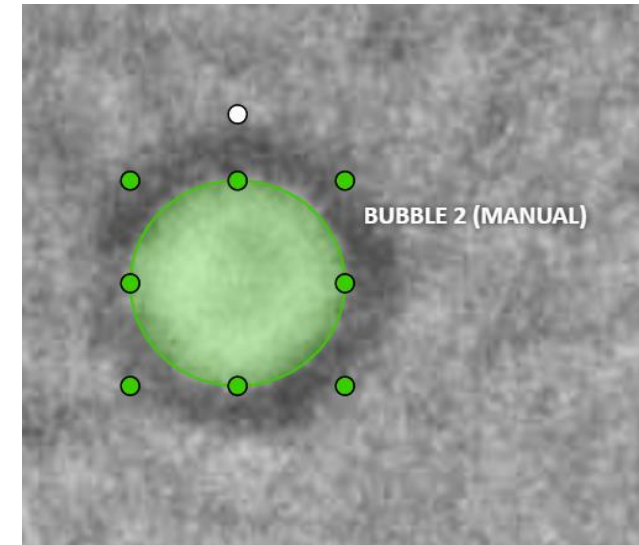
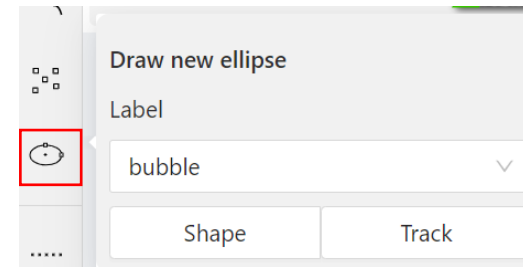
Labeling using polygon and ellipse tools

Remember to click **Save**
Polygon tool

- Hold **Shift** to draw



Ellipse tool




Exporting and parsing labels

Export as .xml

Export job #180806 as a dataset ×

* Export format

 CVAT for images 1.1 ▼

☐ Save images

Custom name

.zip

☒ Use default settings ⓘ

Cancel OK

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

Github:

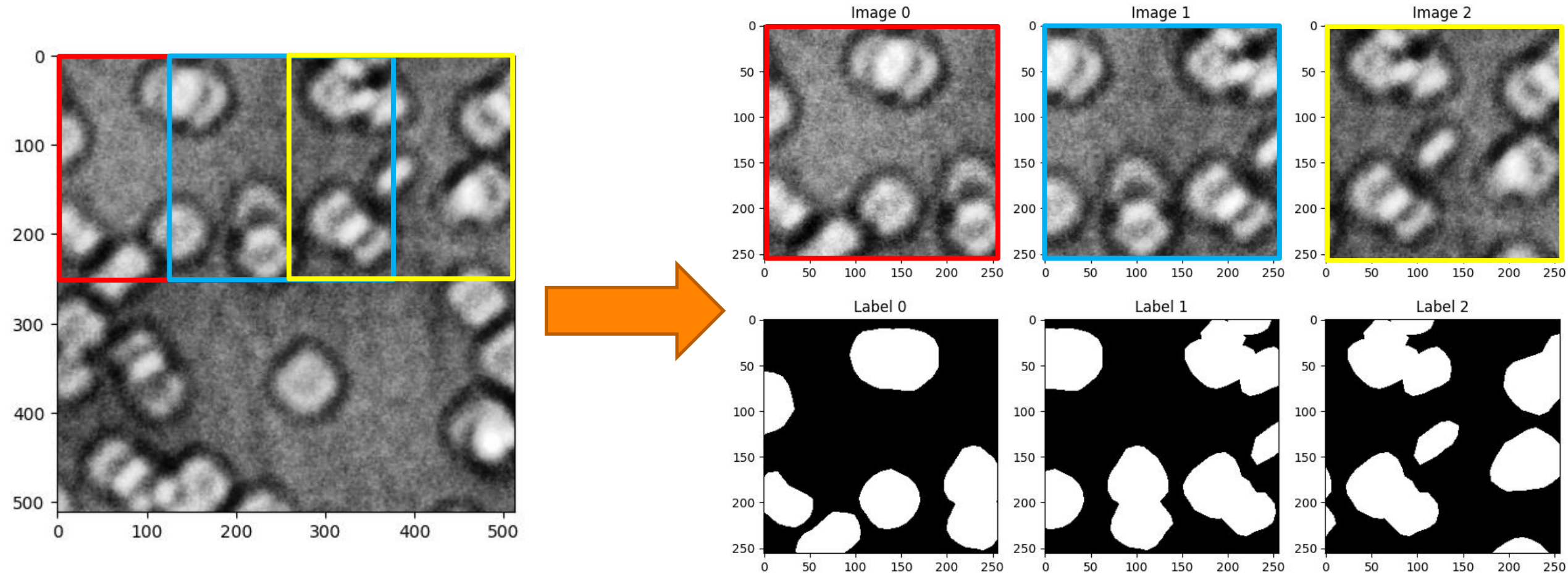


THE UNIVERSITY OF
TENNESSEE
KNOXVILLE

Rationale: more varied data → less overfitting

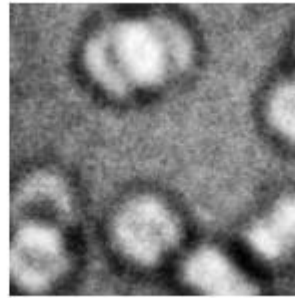
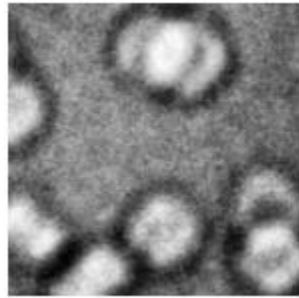
Electron microscopy experiments yield (relatively) small datasets
∴ requires augmentation

Use a sliding window cropper to enhance the dataset

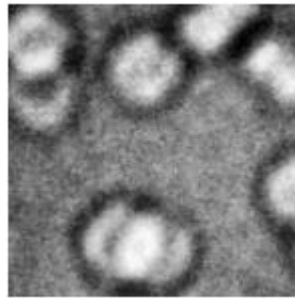


Rotate, flip, resize the image to increase variability

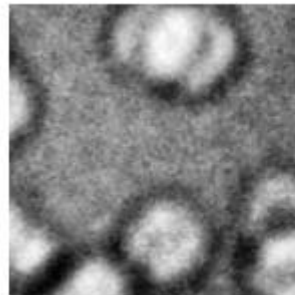
Original



Flipped/Mirrored



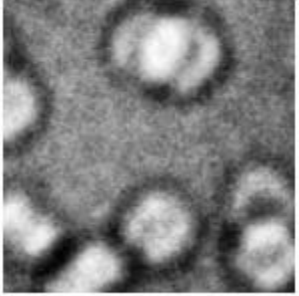
Rotated 180 degrees



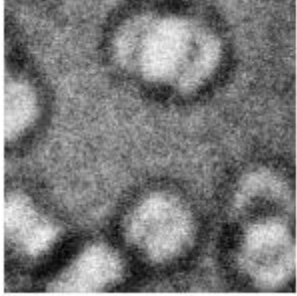
Zoomed in

Adding noise to simulate noises during imaging

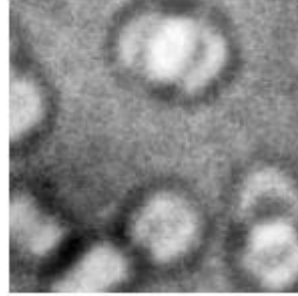
Original



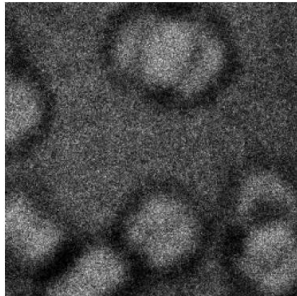
Gaussian



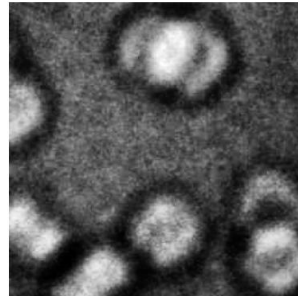
Background noise



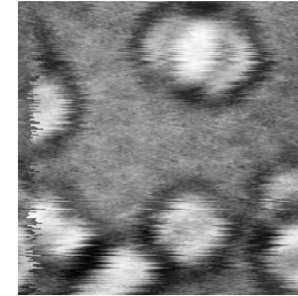
Poisson



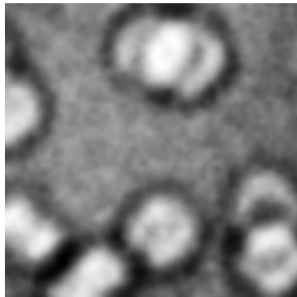
Contrast



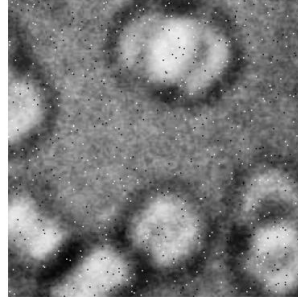
Jitter



Blur



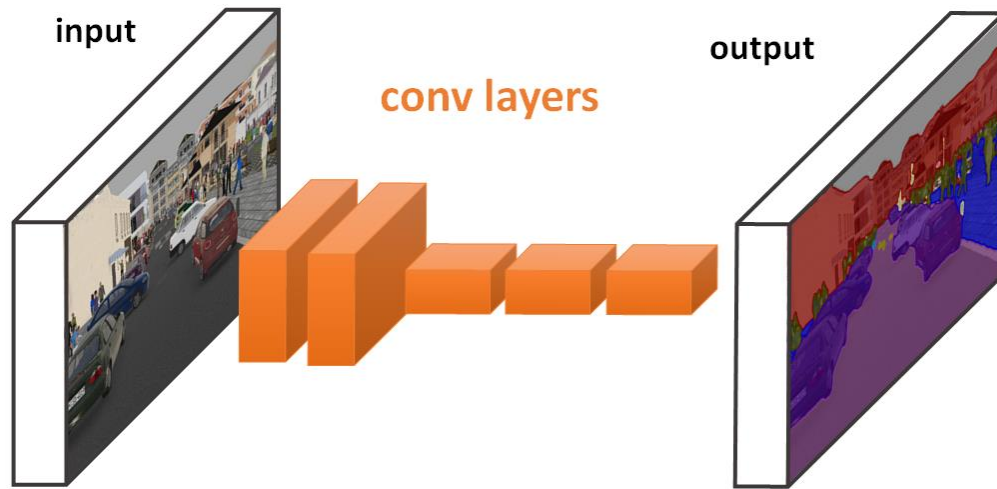
Salt & pepper



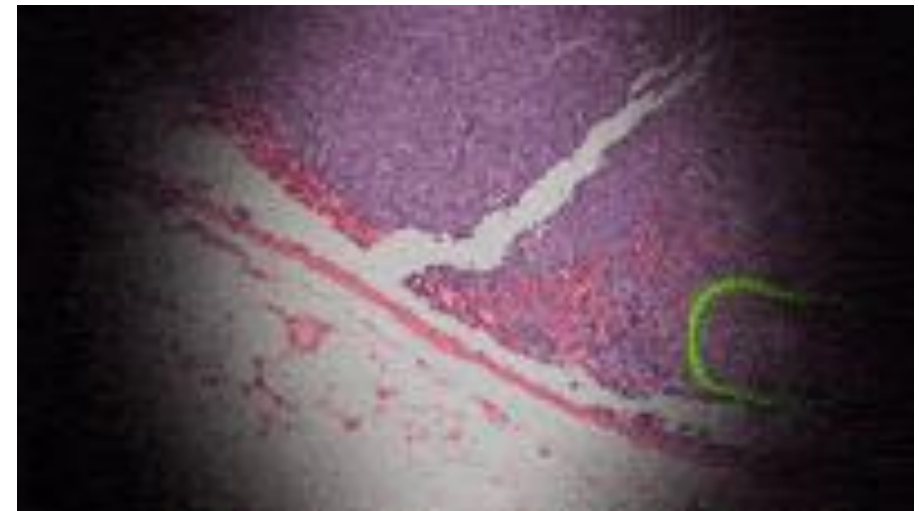
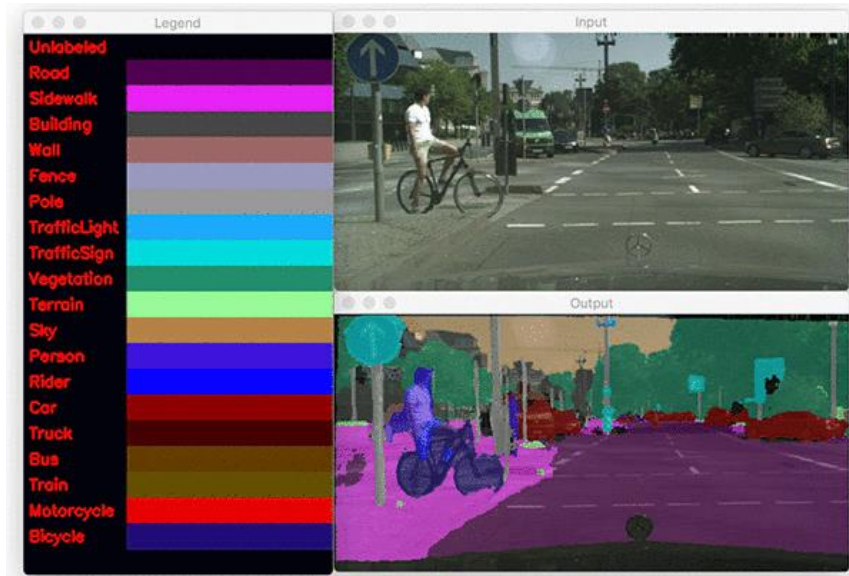
Convolutional Neural Networks (CNN)



THE UNIVERSITY OF
TENNESSEE
KNOXVILLE



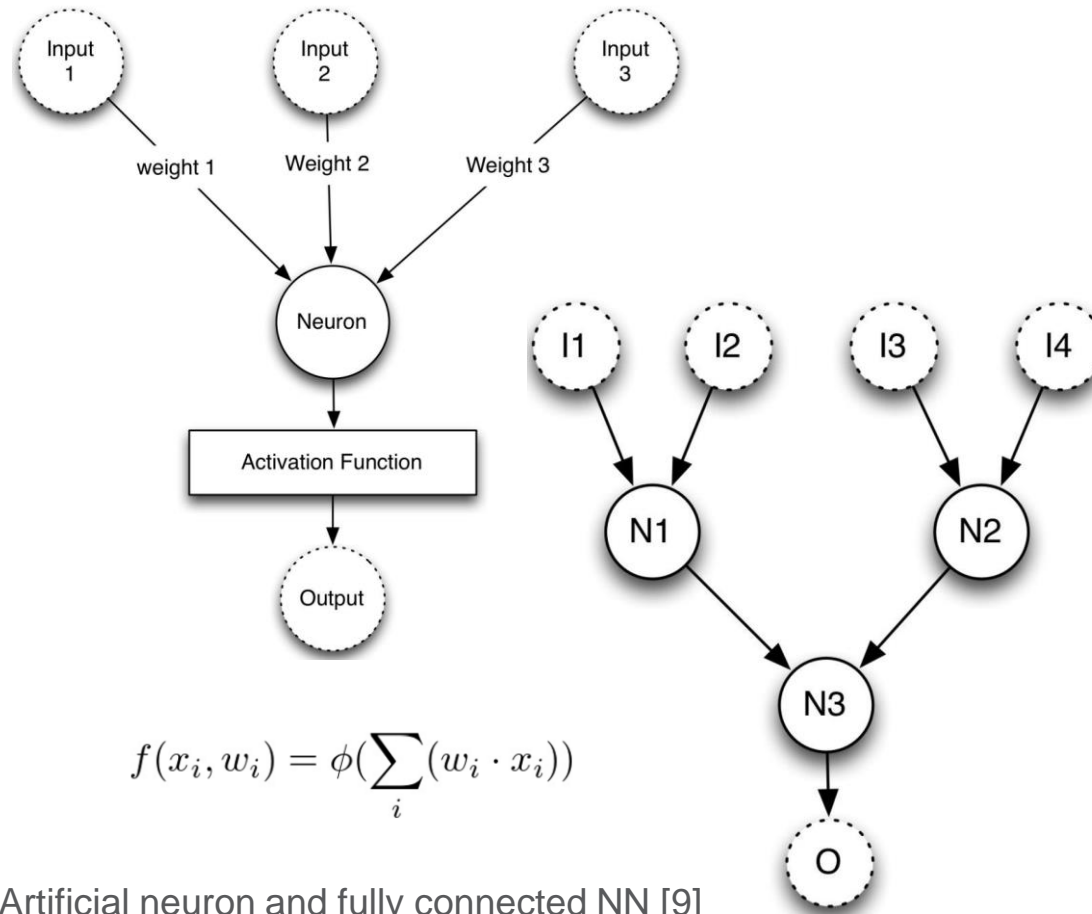
**Semantic segmentation
assigns class labels to
each pixel**



Augmented reality microscope
[4]

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

Convolutional Neural Network (CNN)



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

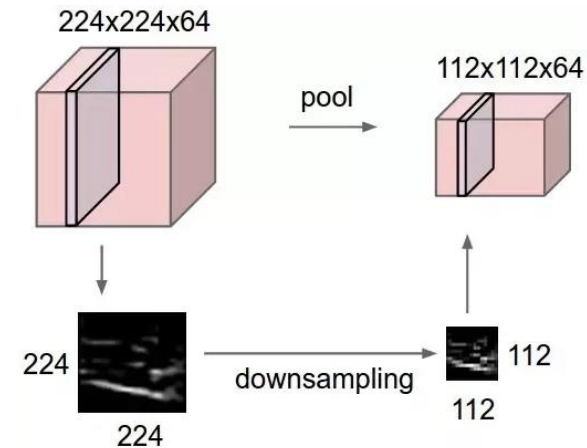
Normal convolution kernel

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

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

2×2 Max-Pool

20	30
112	37

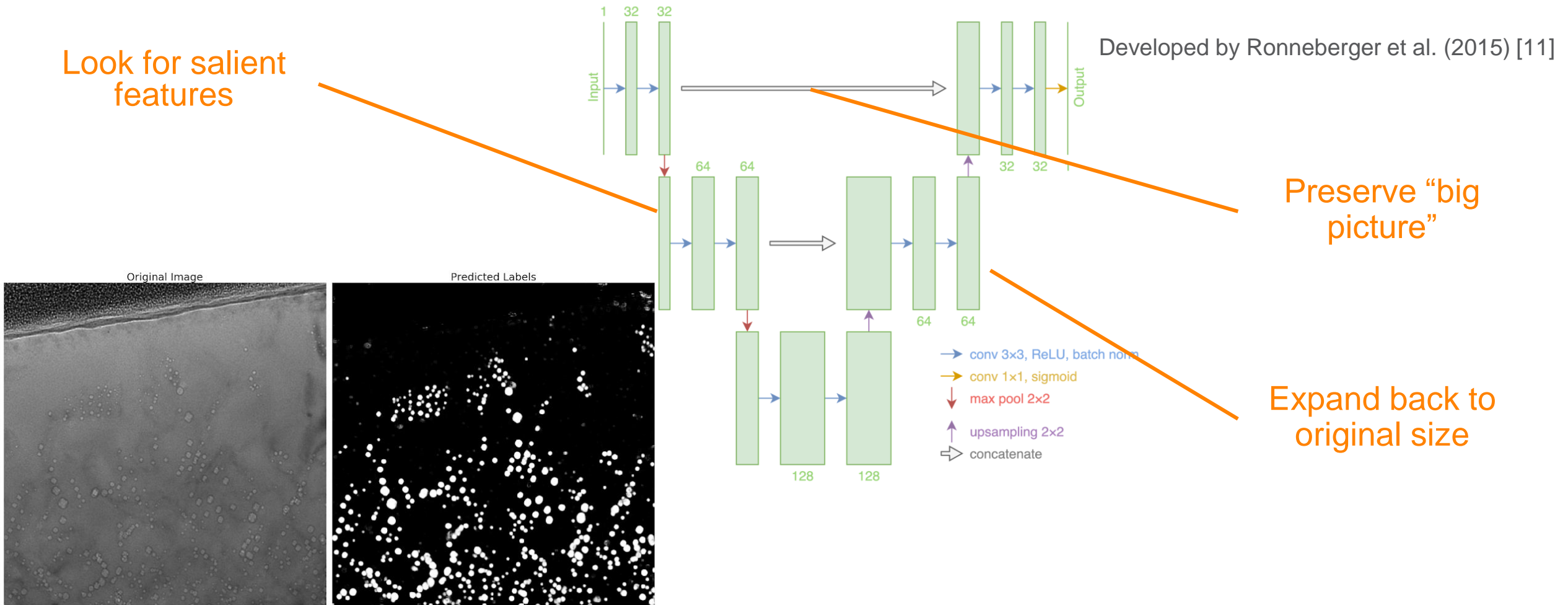


Maxpooling

Semantic segmentation w/ U-Net

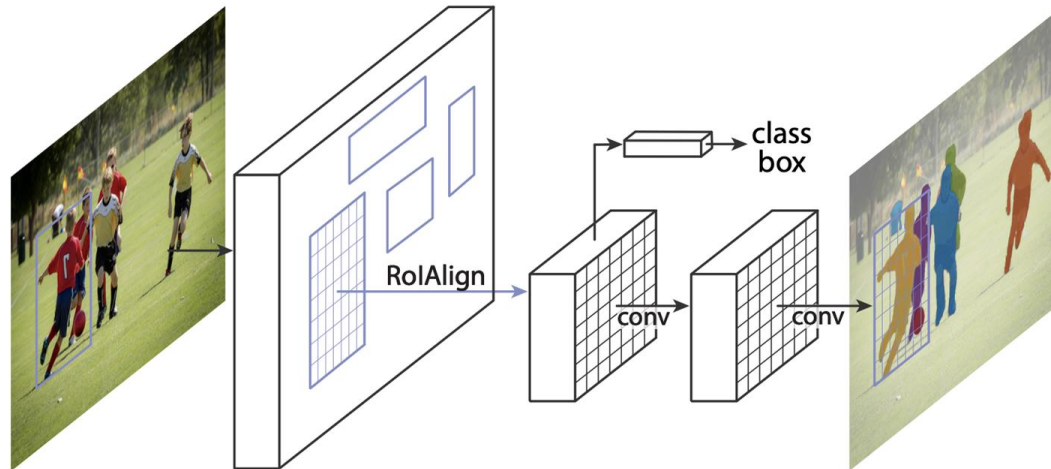
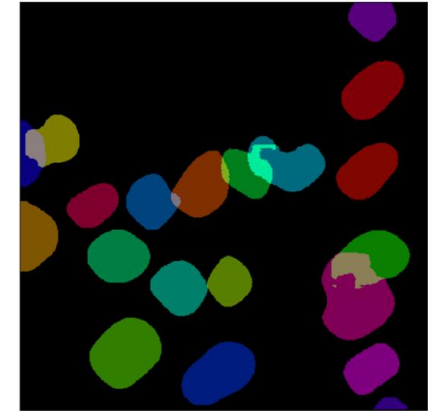
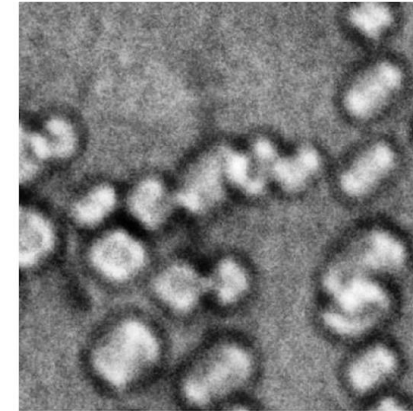
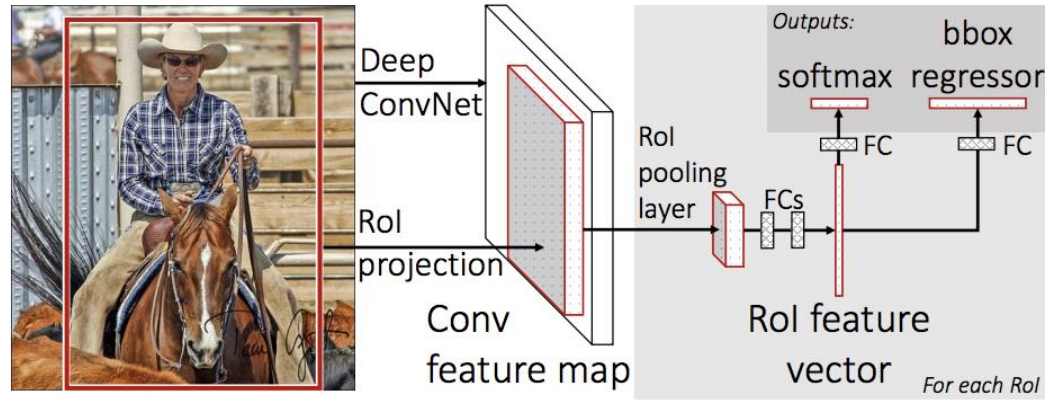
Look for salient features

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



U-Net demonstration

Instance segmentation w/ Mask R-CNN



Mask R-CNN demonstration