Image Masking and Cell Segmentation

BY JOSHUA PAIK

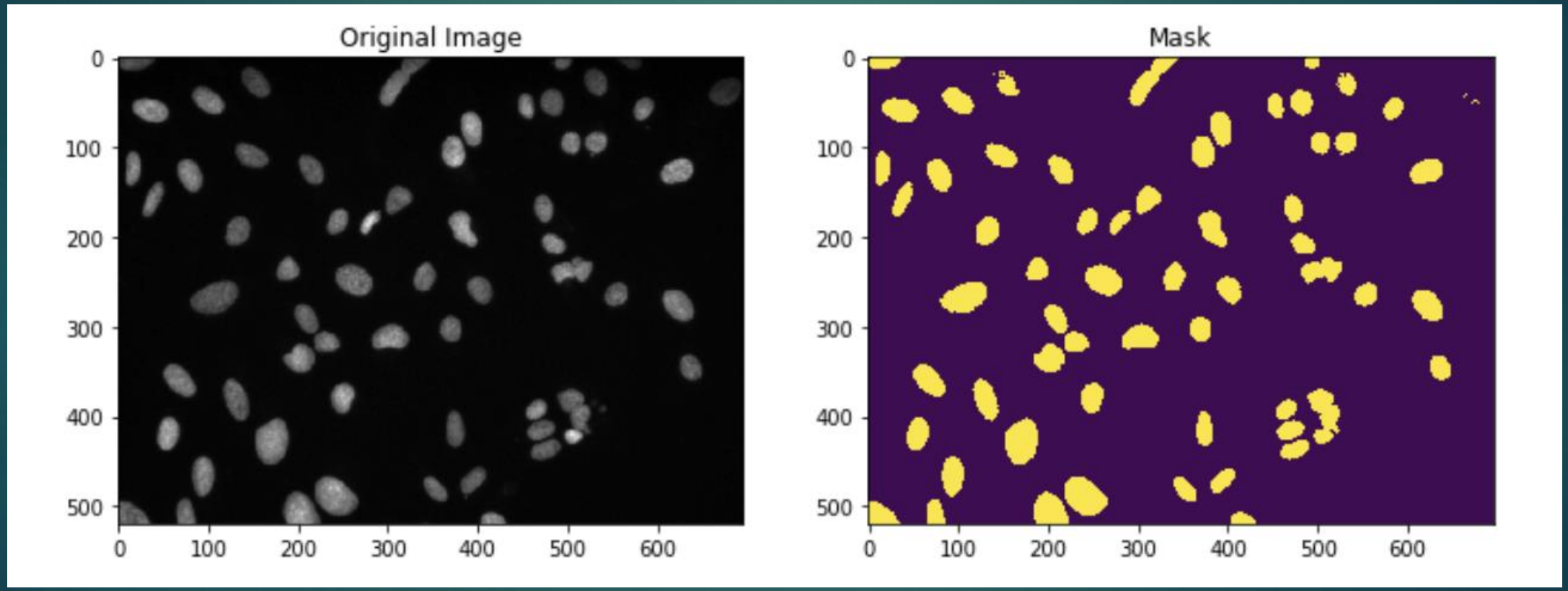
Outline of Today's Talk

- Introduction to the Kaggle Data Science Bowl 2018
- Exploratory Data Analysis
- Explanation of UNET
- Data Augmentation
- My Solution
- Winning Solutions

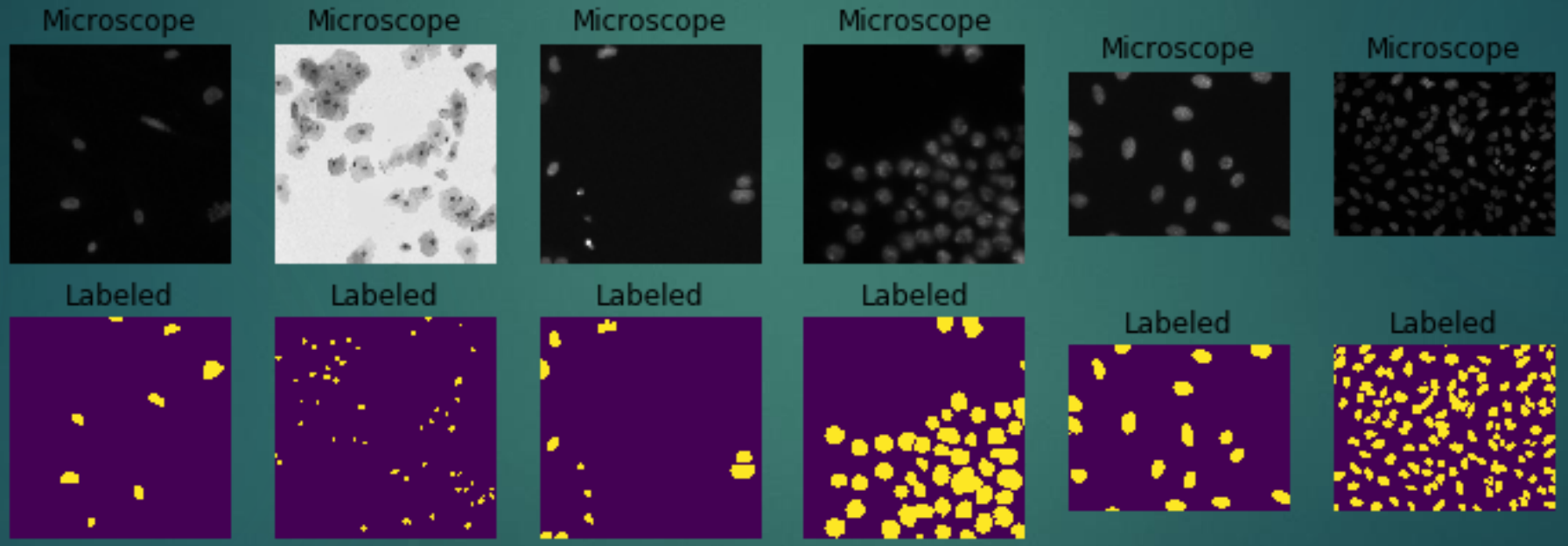
Introduction to the Kaggle Data Science Bowl 2018

- GOAL? Identify cell nuclei from microscopy data.
- WHY? To advance research. As the nuclei contains DNA, it is important to locate it in genetic and biological research.
- HOW? Use existing neural network architecture platforms and image processing techniques

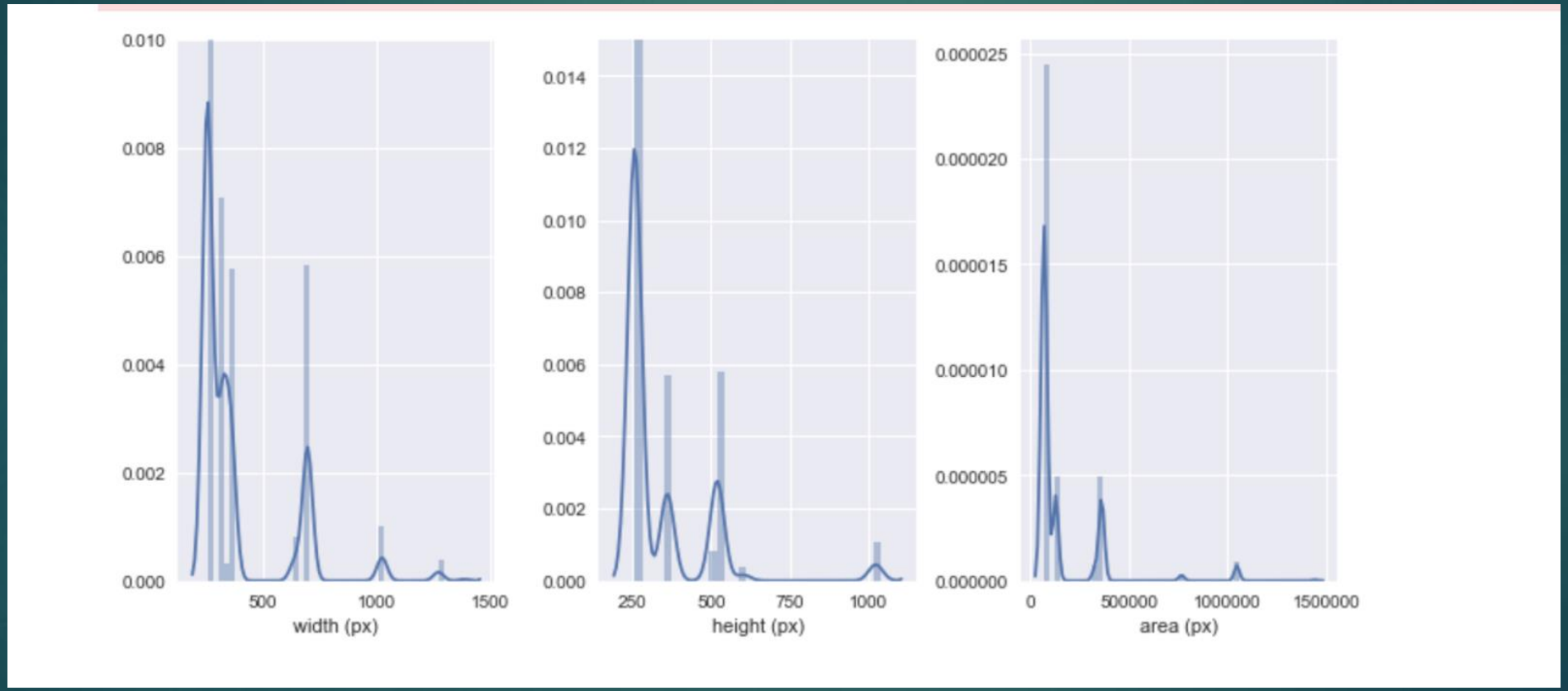
Goal: Create image masks for each cells



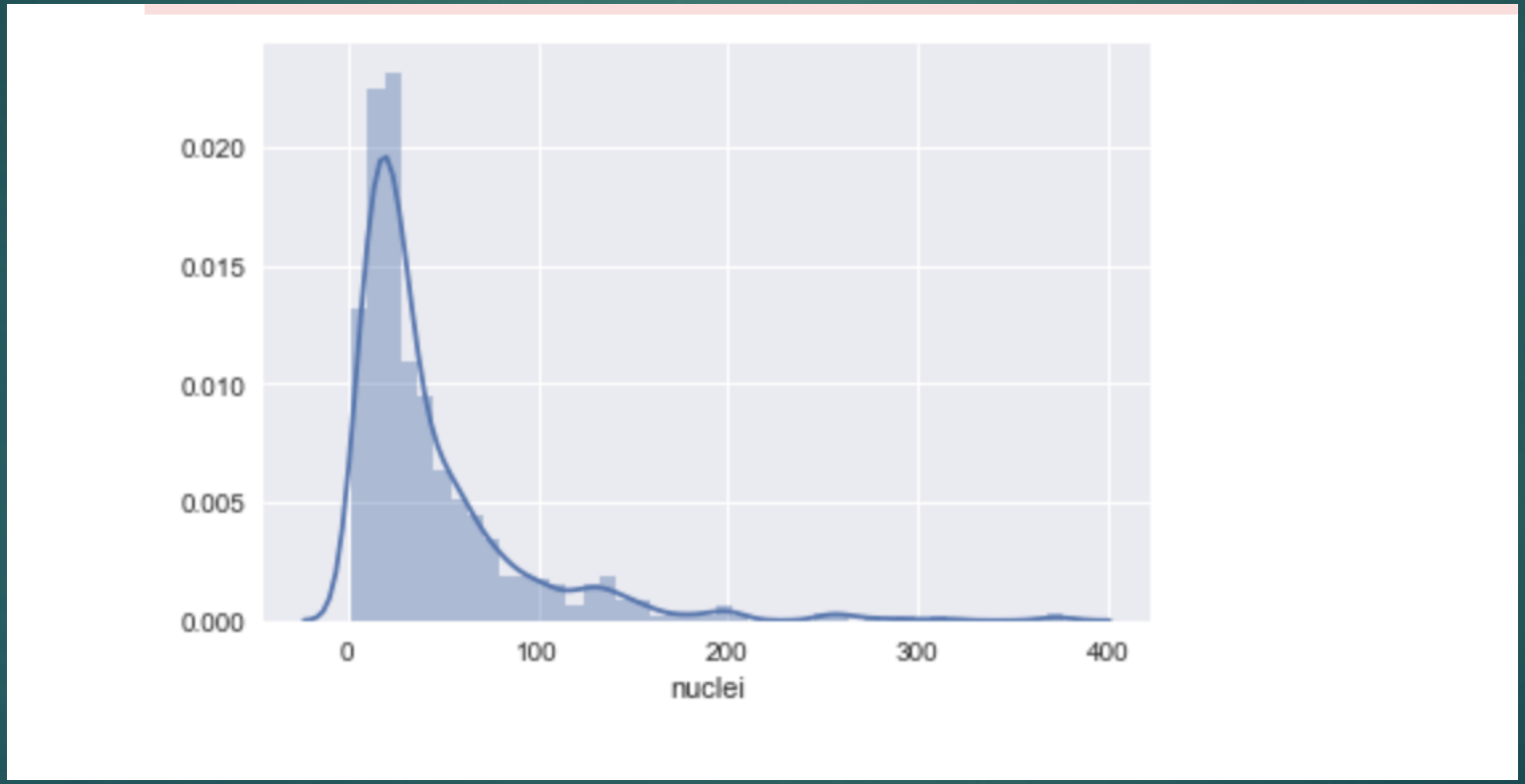
More Sample Data



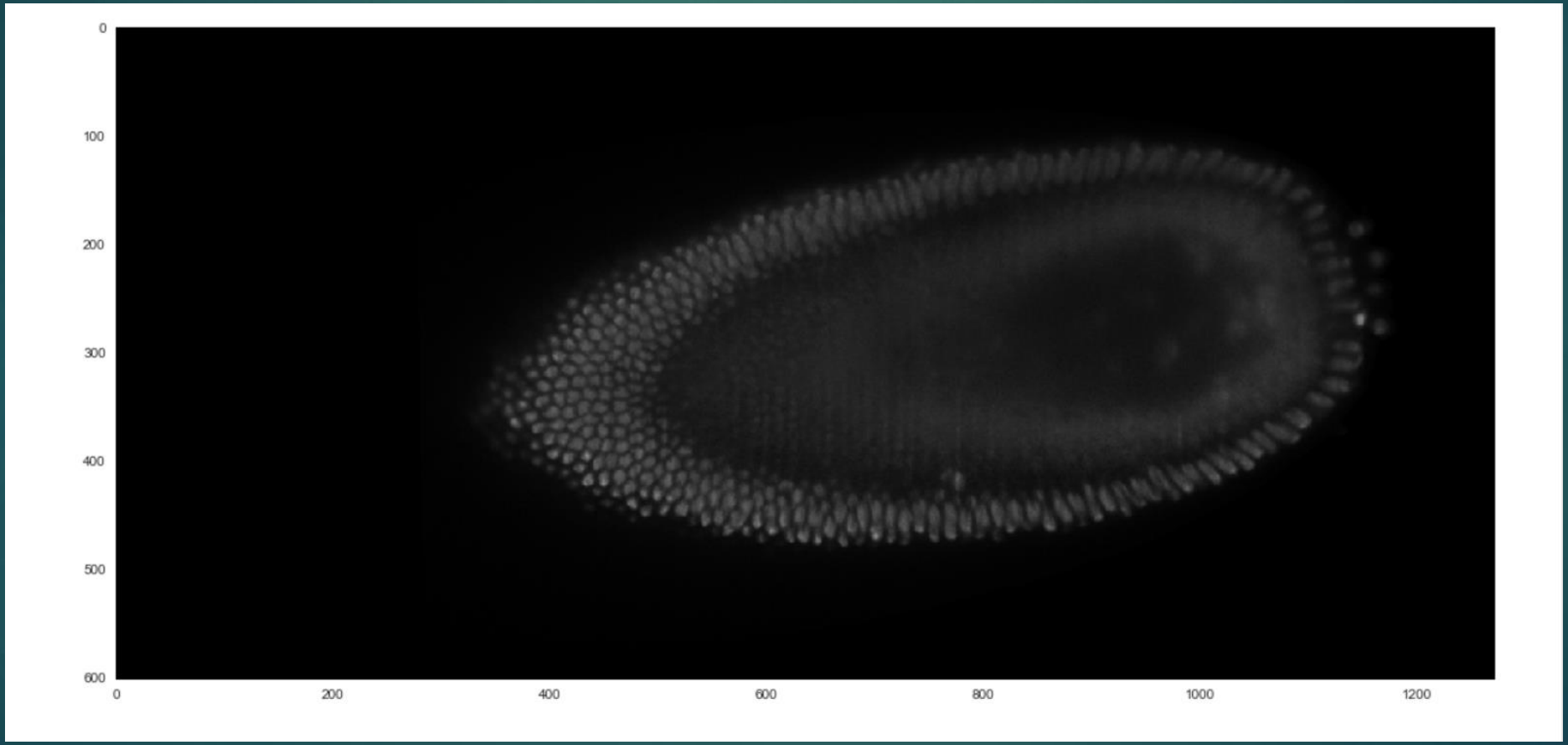
EDA: Distribution of image masks width, height, area



EDA: Distribution of number of nuclei in each images

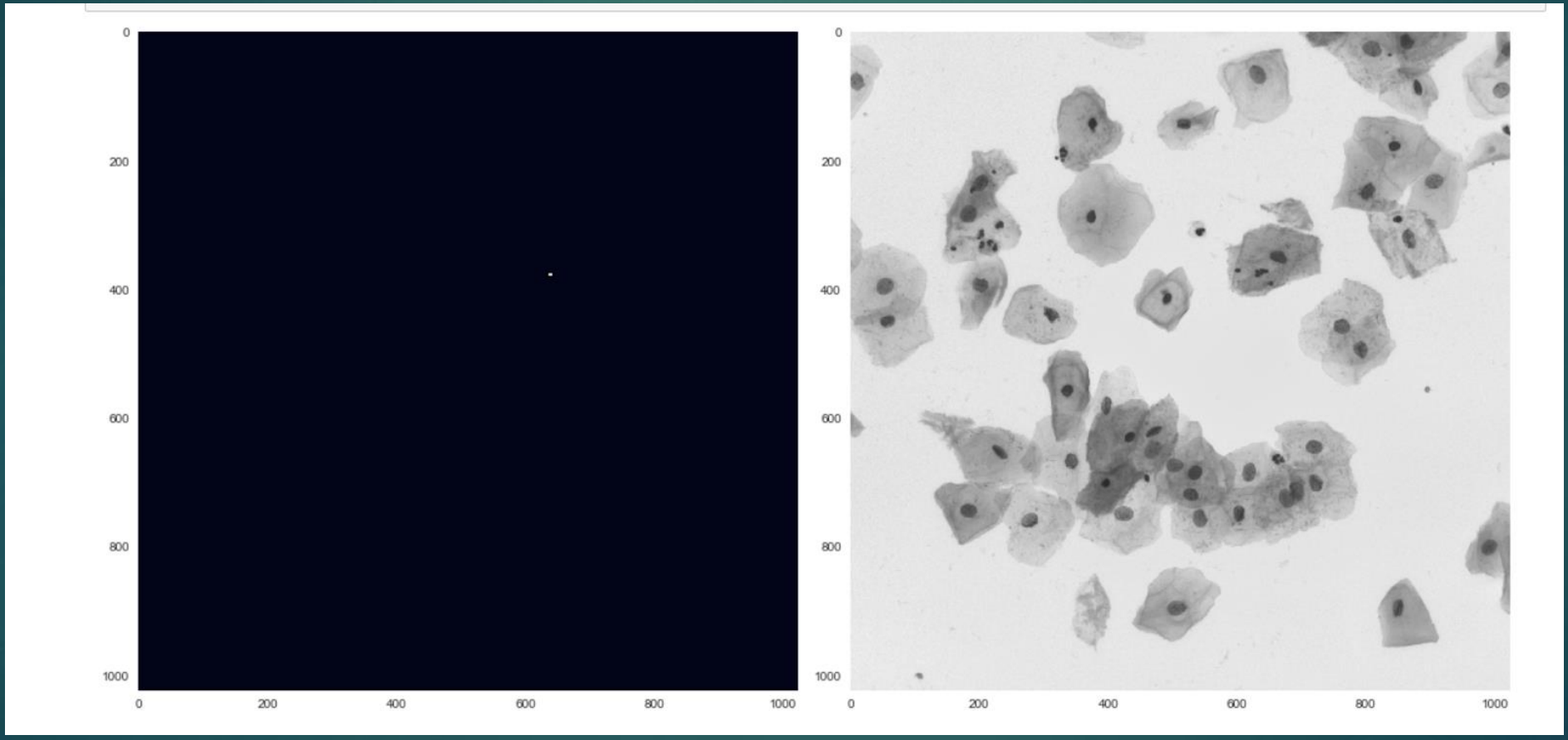


EDA: Image with most number of nuclei

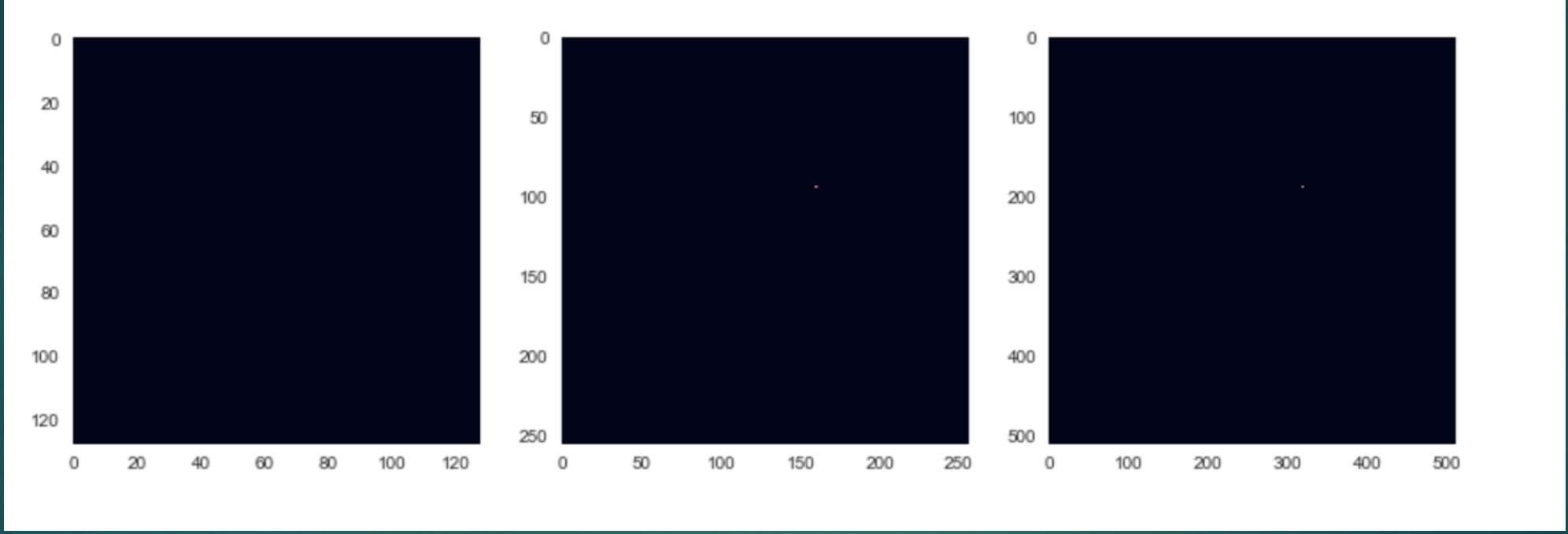


EDA: Image with Least number of nuclei

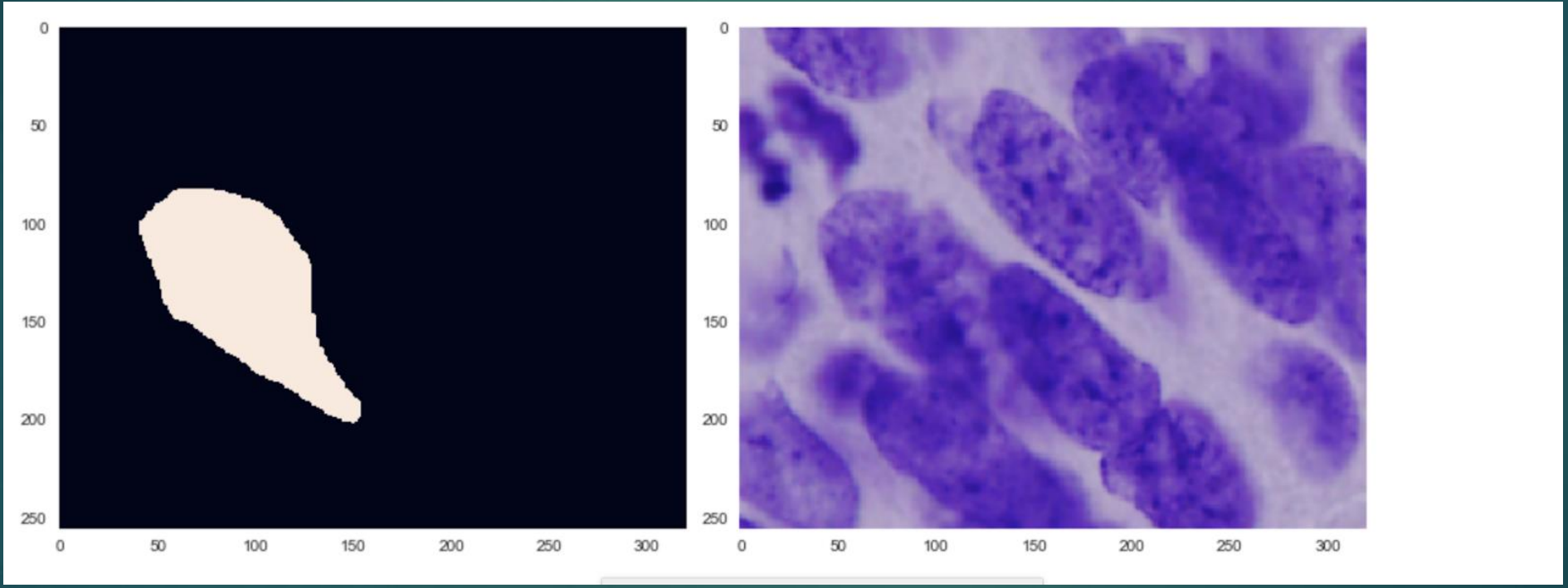
EDA: Smallest nuclei problem



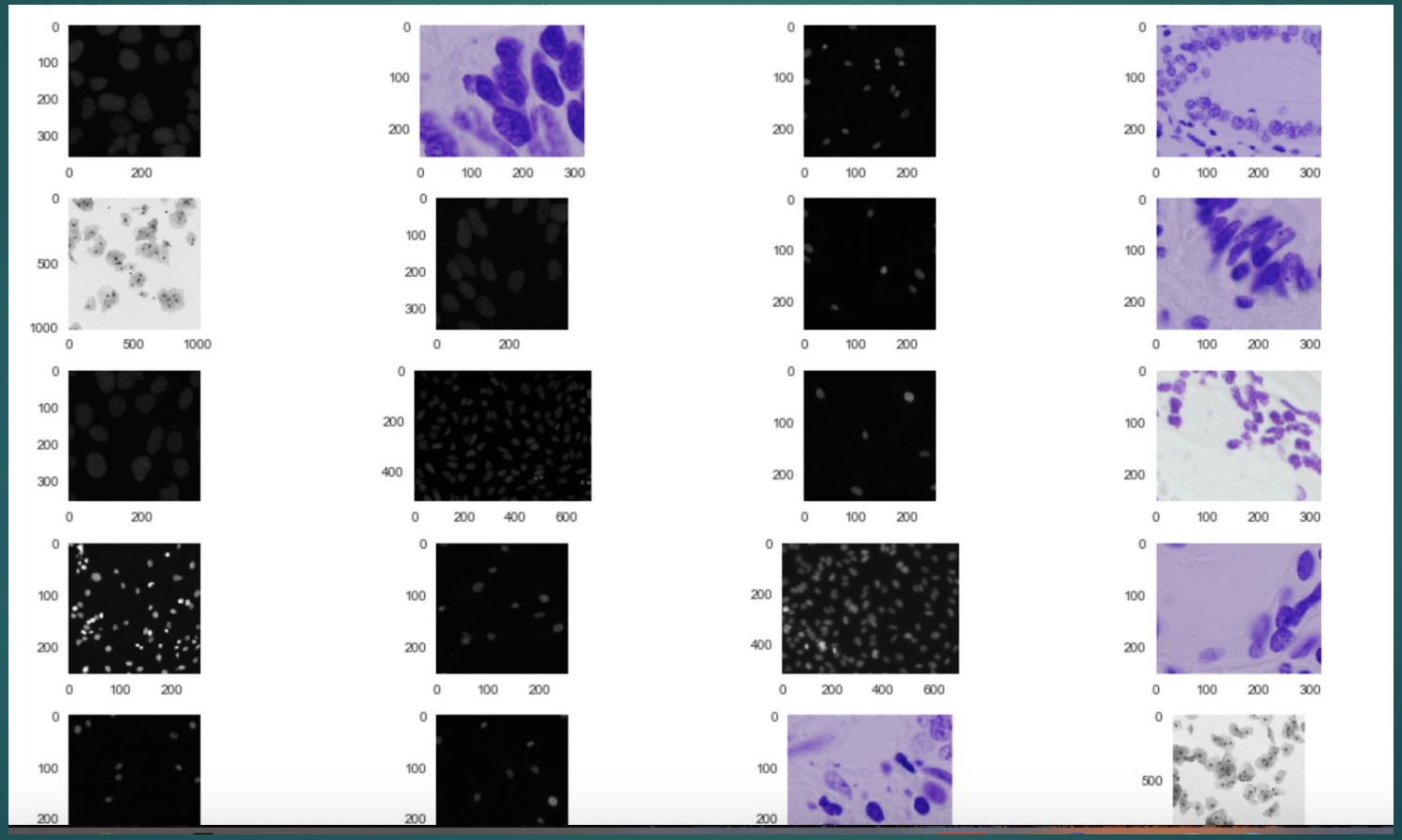
EDA: Smallest Nuclei Problem: Dependent on Scaling



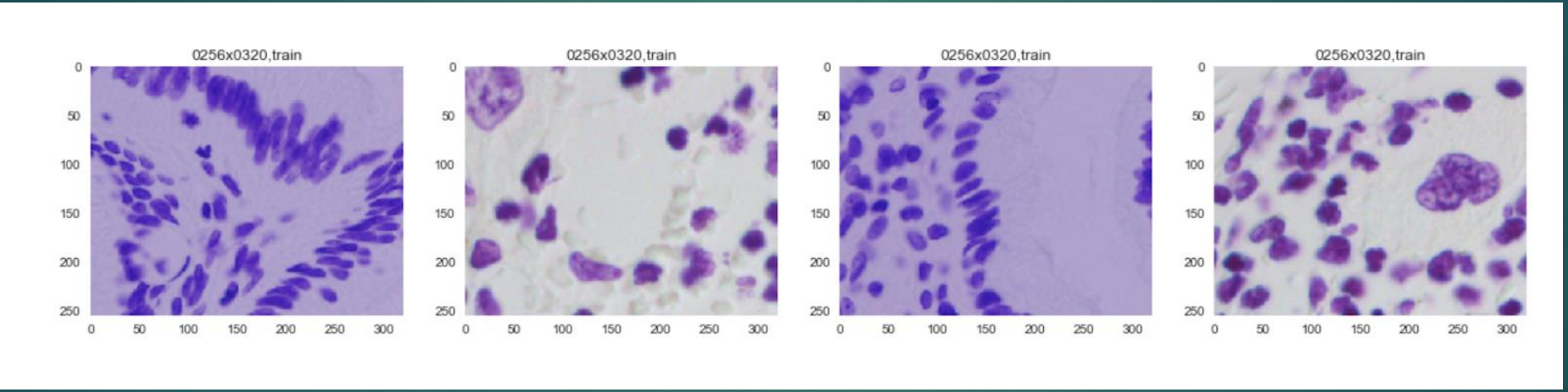
EDA: Overlapping cell problem



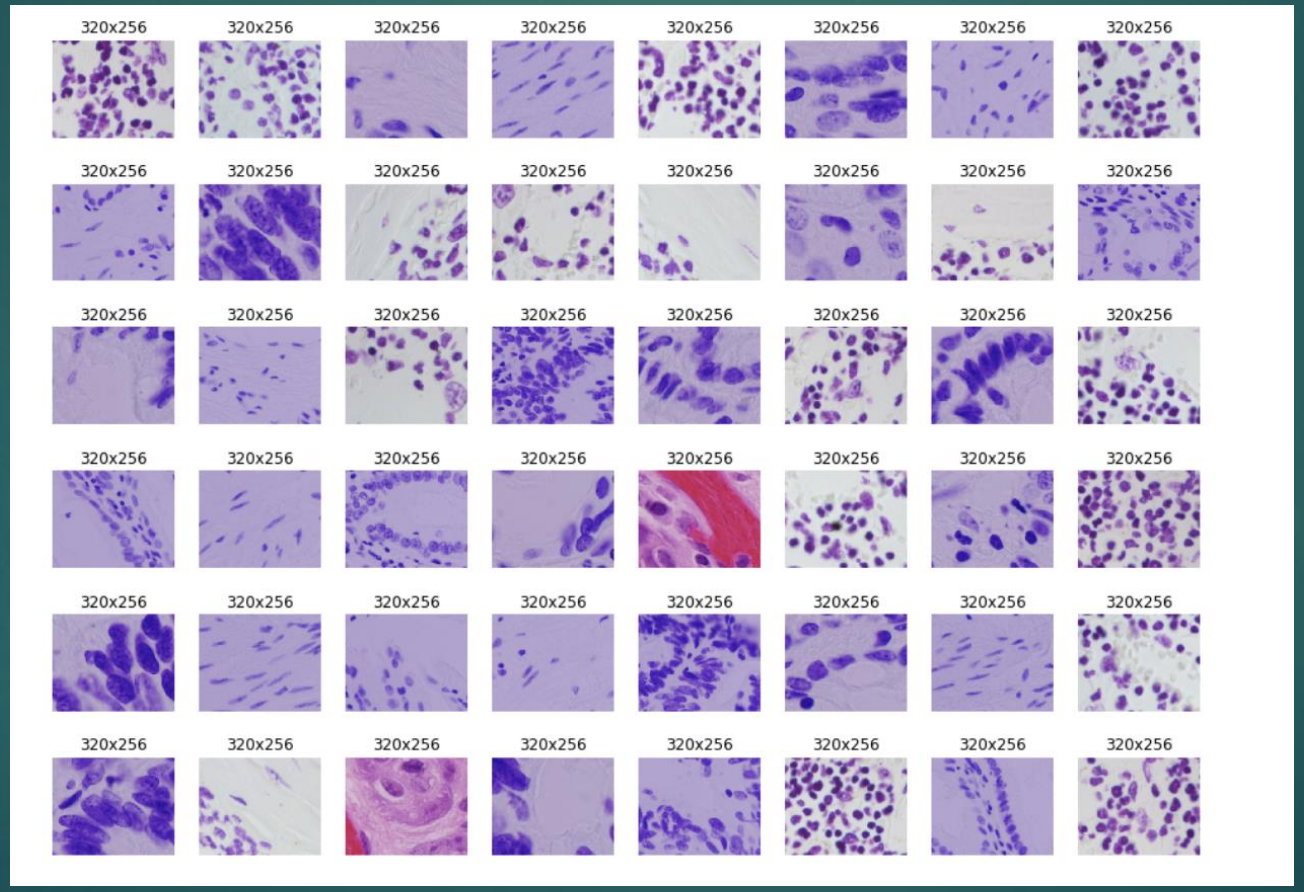
There are a lot of overlapping cells



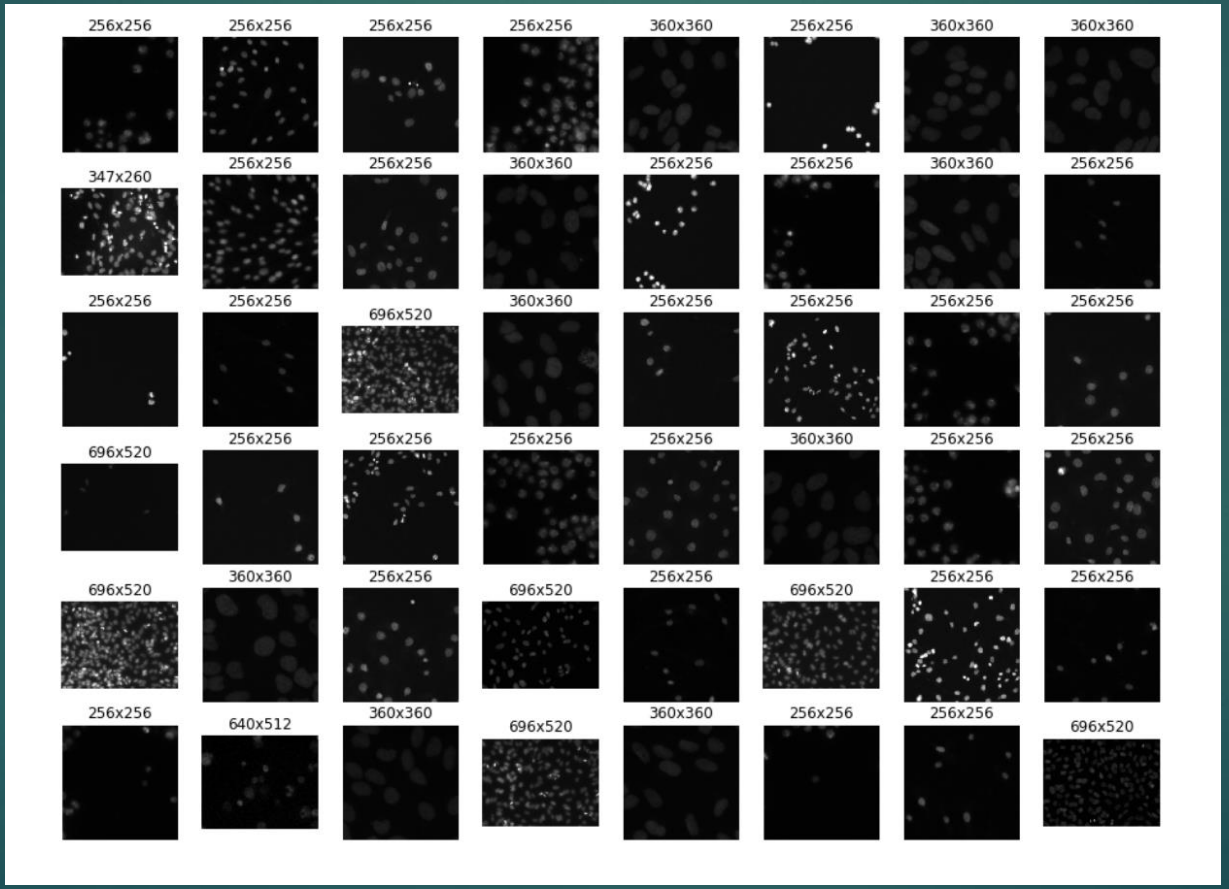
More Overlapping Cells



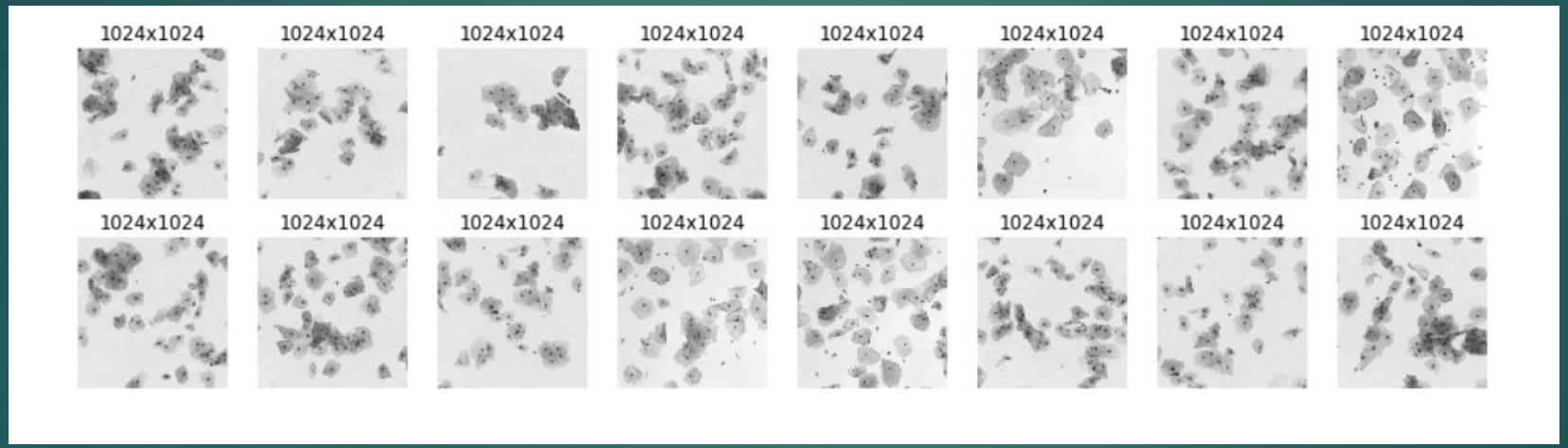
EDA: K Means Classification Fluorescent: 81.5%



EDA: K Means Classification Histological: 16.1%



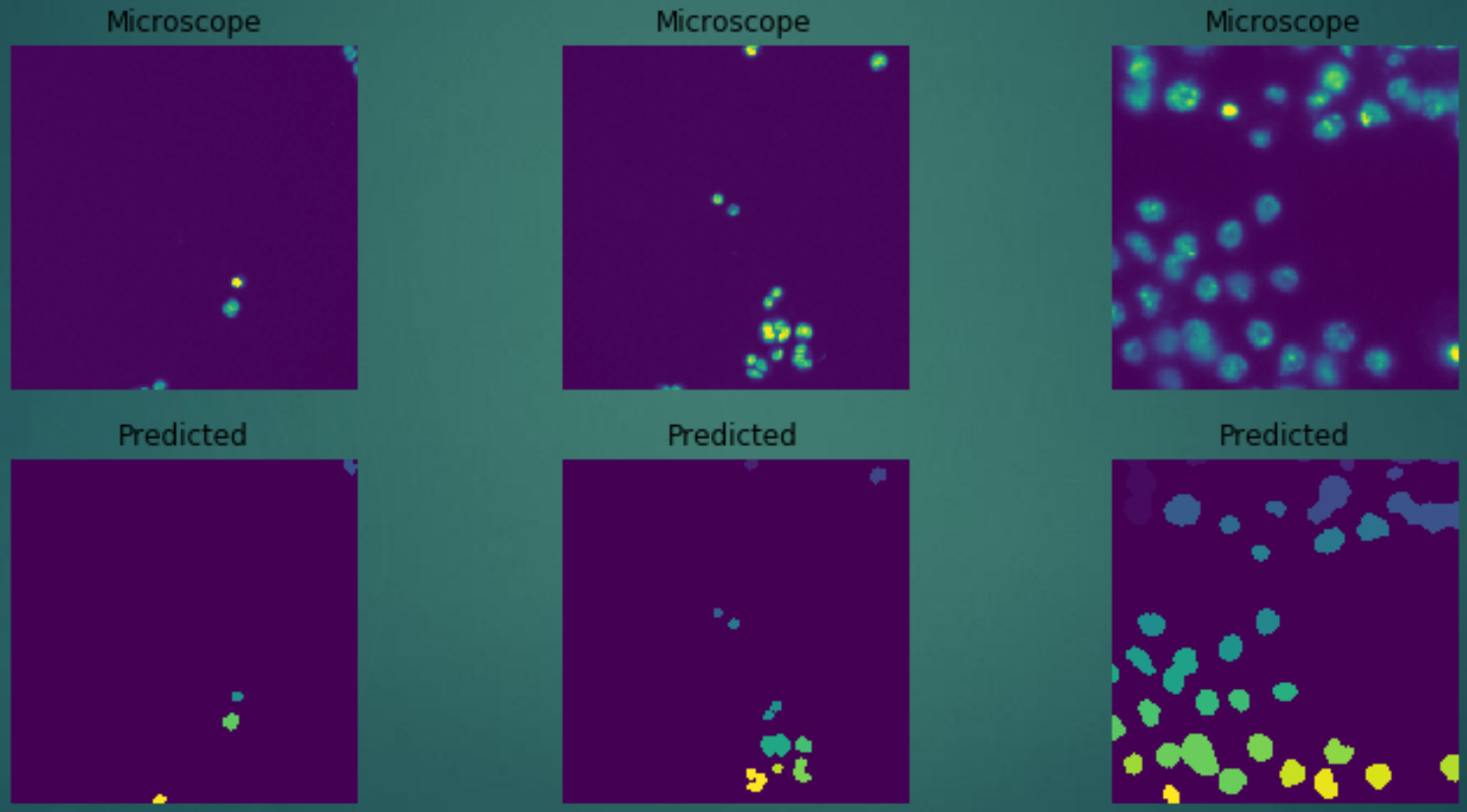
EDA: K Means Classification Bright-field: 2.4%



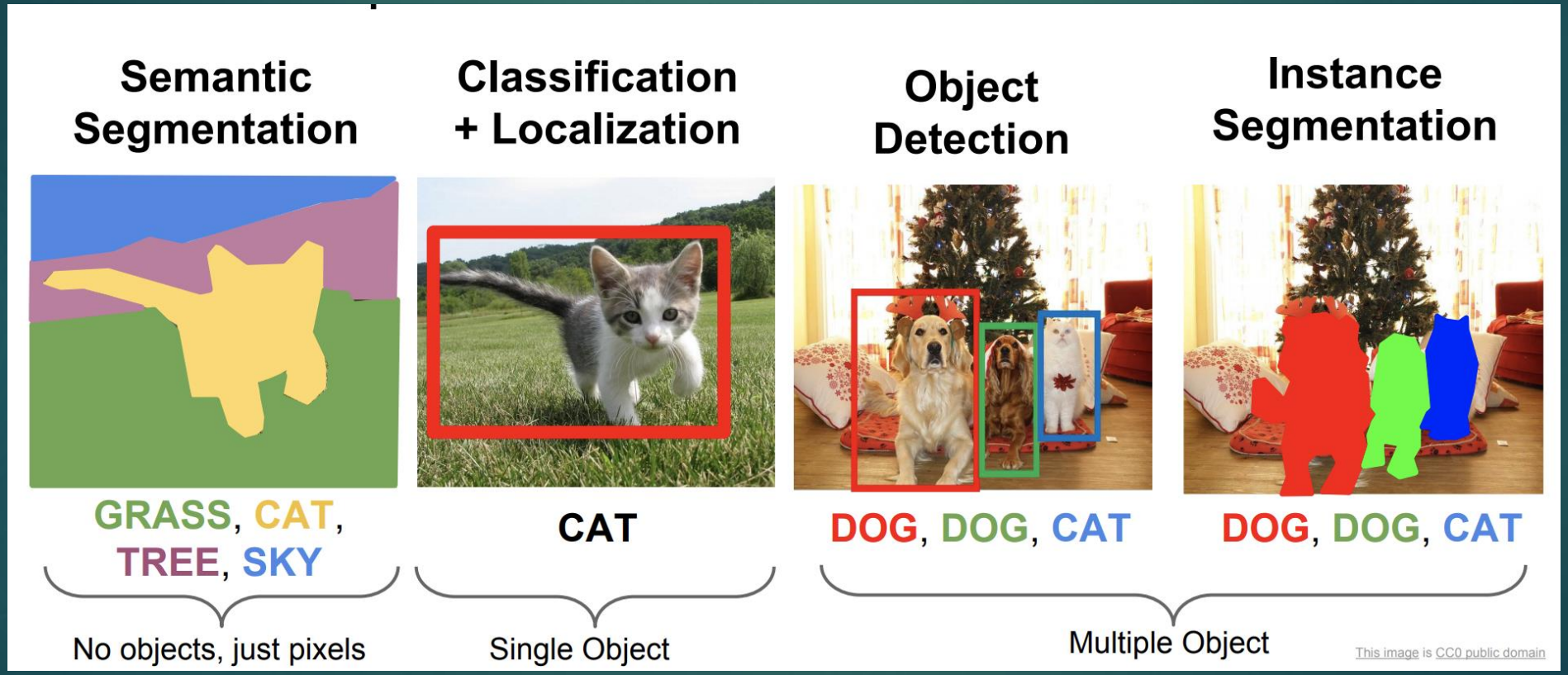
```python
def parametric_pipeline(img_green,
                invert_thresh_pd = 10,
                circle_size_x = 7,
                circle_size_y = 7,
                ):
    circle_size_x = np.clip(int(circle_size_x), 1, 30)
    circle_size_y = np.clip(int(circle_size_y), 1, 30)

    #green channel happends to produce slightly better results
    #than the grayscale image and other channels
    #morphological opening (size tuned on training data)
    circle7=cv2.getStructuringElement(cv2.MORPH_ELLIPSE,(circle_size_x, circle_size_y))
    img_open=cv2.morphologyEx(img_green, cv2.MORPH_OPEN, circle7)
    #Otsu thresholding
    img_th=cv2.threshold(img_open,0,255,cv2.THRESH_OTSU)[1]
    #Invert the image in case the objects of interest are in the dark side
    if(np.sum(img_th==255)>((invert_thresh_pd/10.0)*np.sum(img_th==0))):
        img_th=cv2.bitwise_not(img_th)
    #second morphological opening (on binary image this time)
    bin_open=cv2.morphologyEx(img_th, cv2.MORPH_OPEN, circle7)
    #connected components
    cc=cv2.connectedComponents(bin_open)[1]
    #cc=segment_on_dt(bin_open,20)
    return cc
```

Non Machine Learning Approach

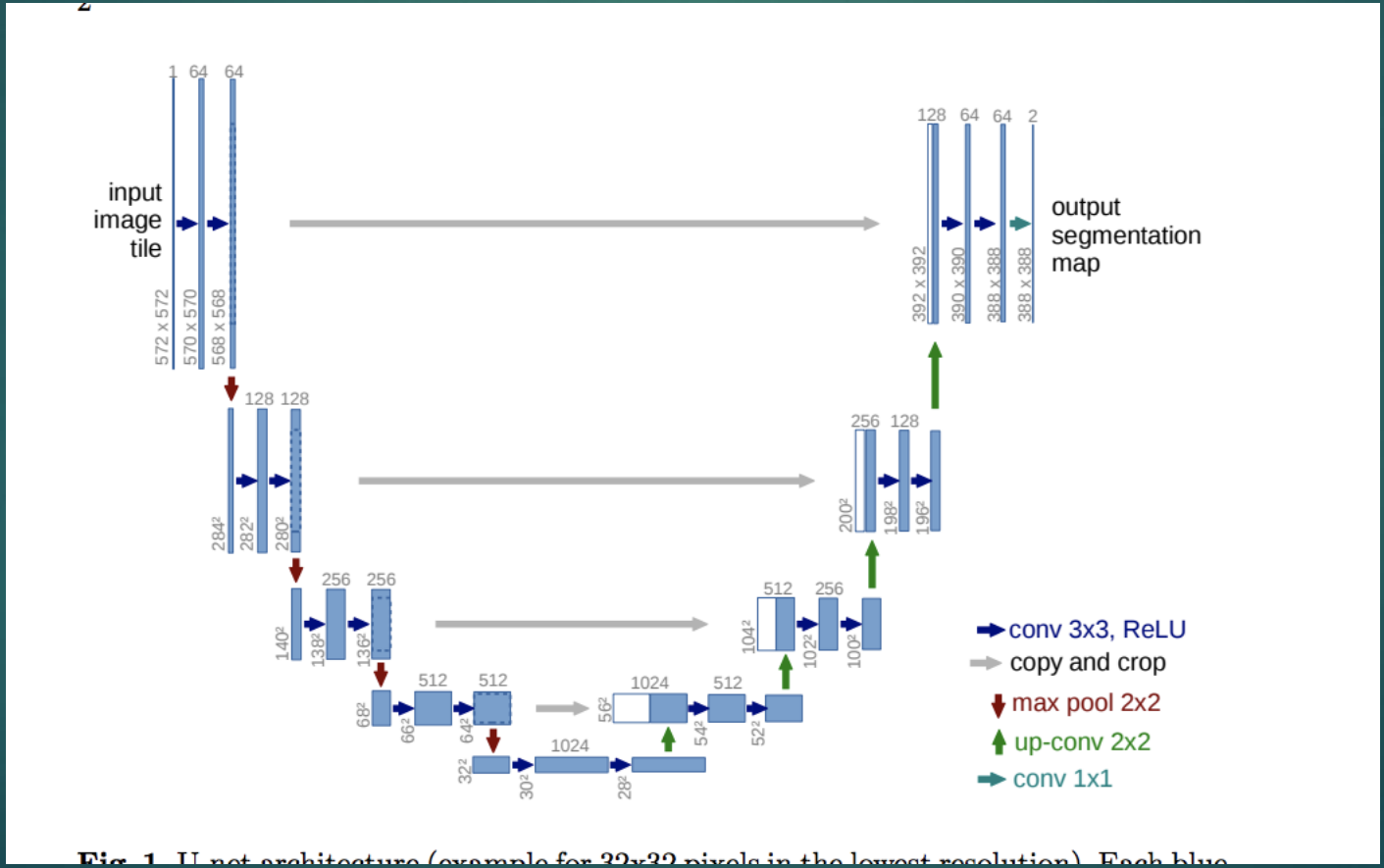


What type of problem is the Kaggle DSBowl

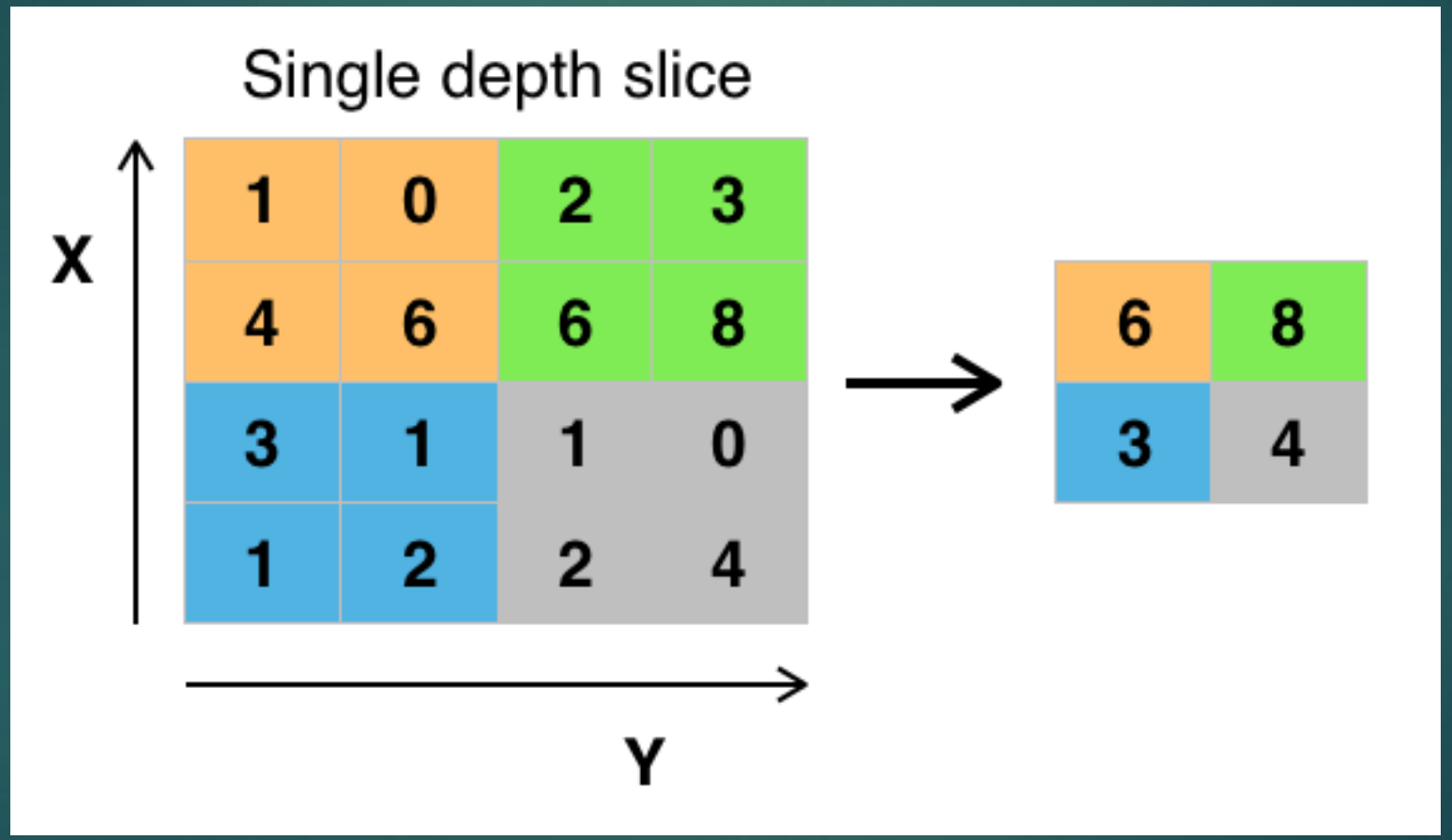


Explanation of UNET

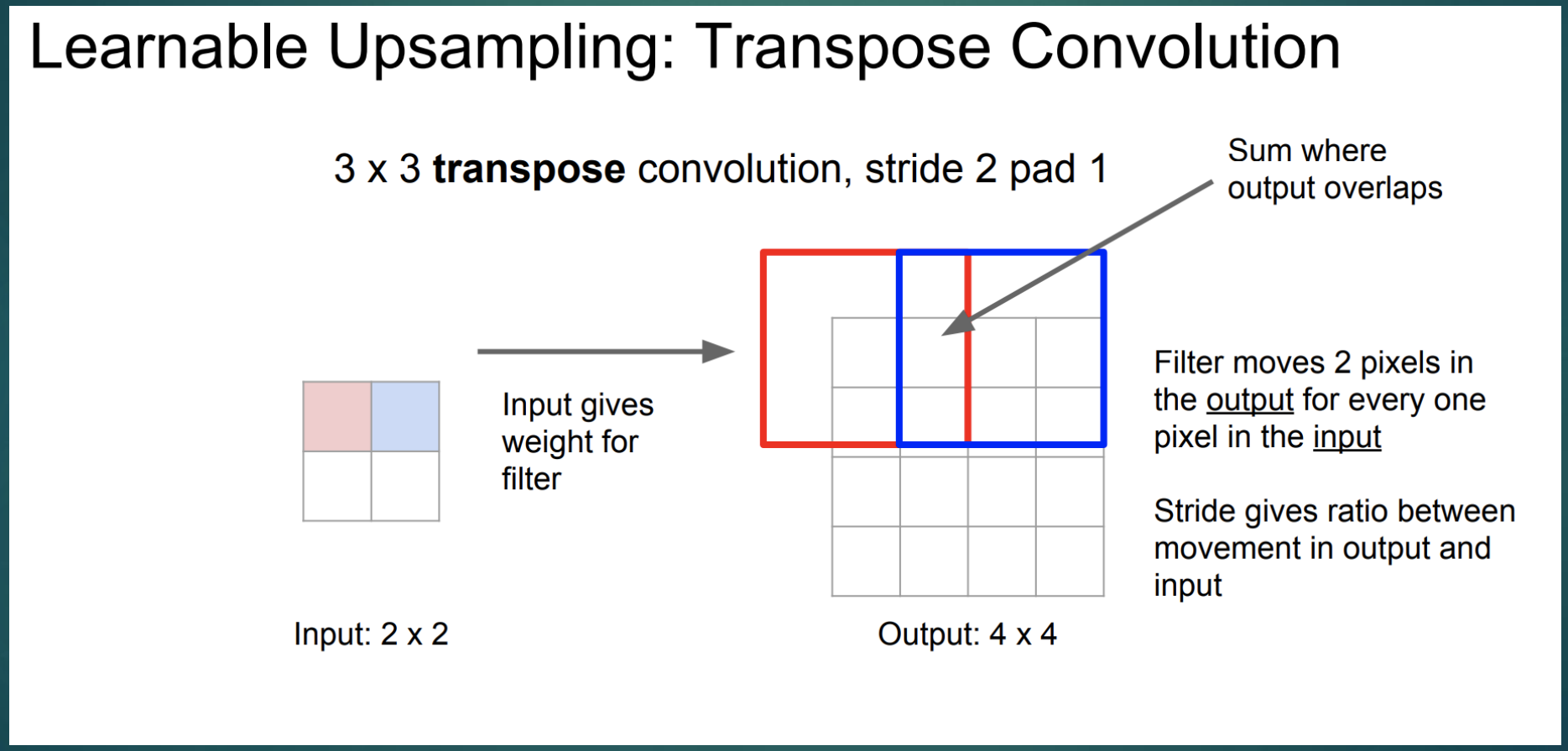
https://arxiv.org/pdf/1505.04597.pdf



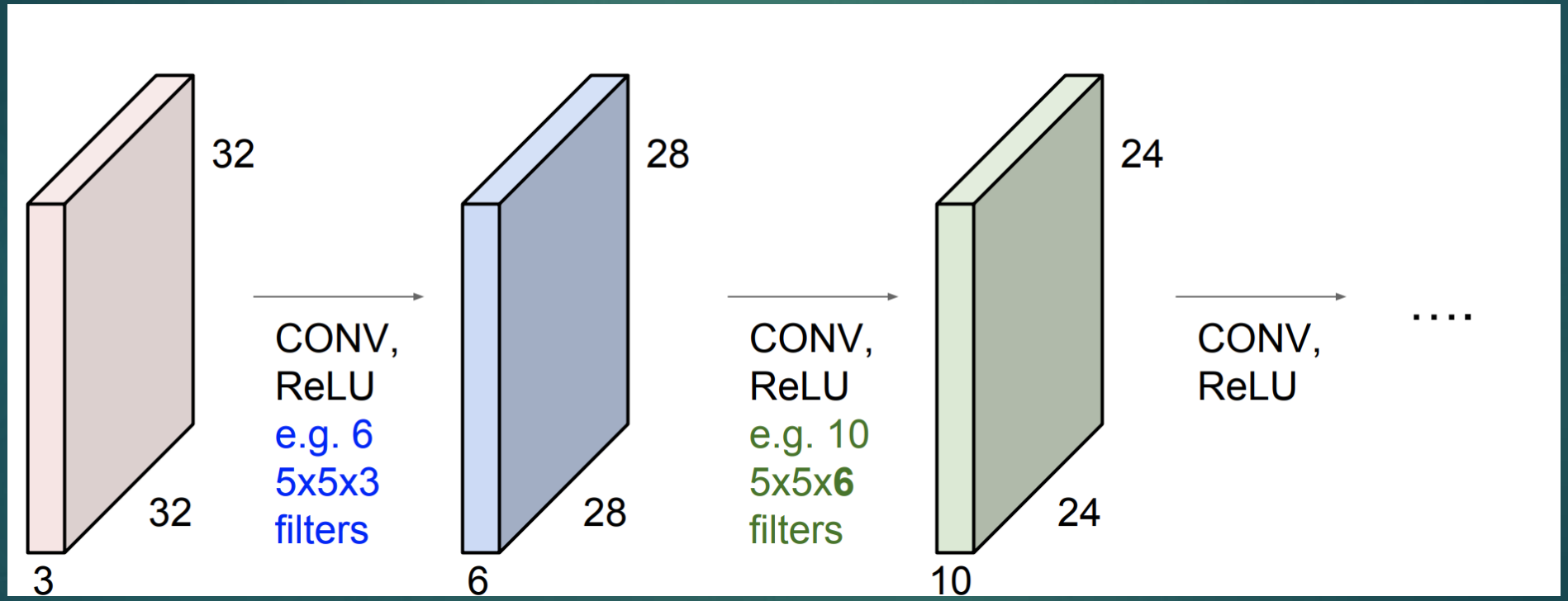
Max Pool



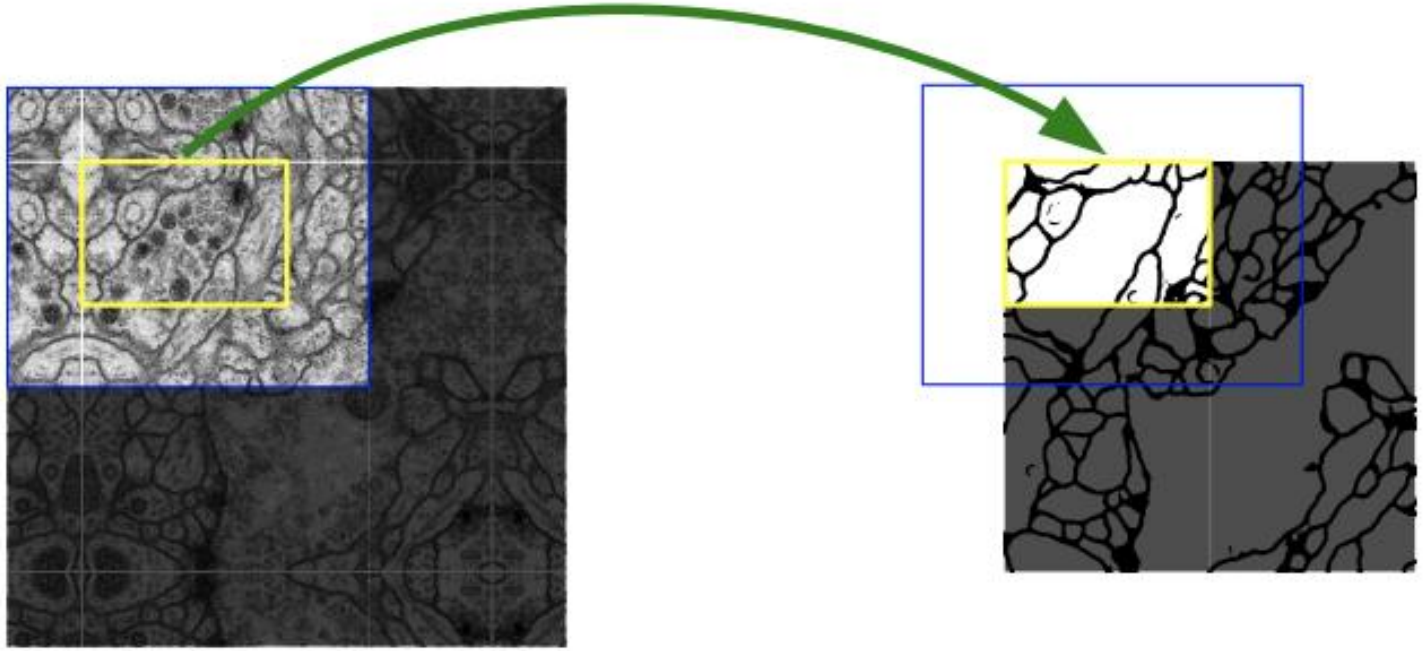
Up Convolution or Transpose Convolution



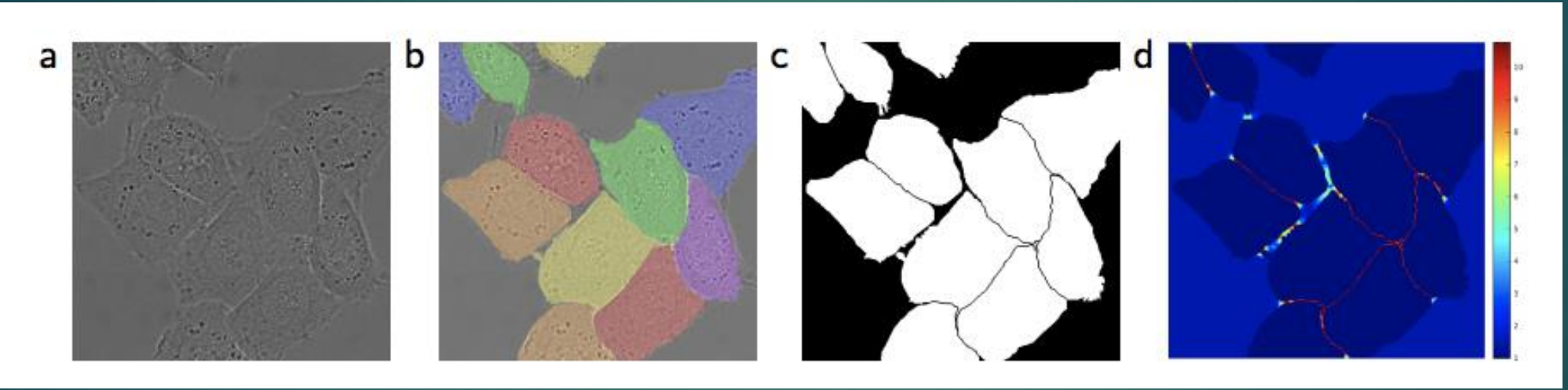
Conv 3x3 ReLU



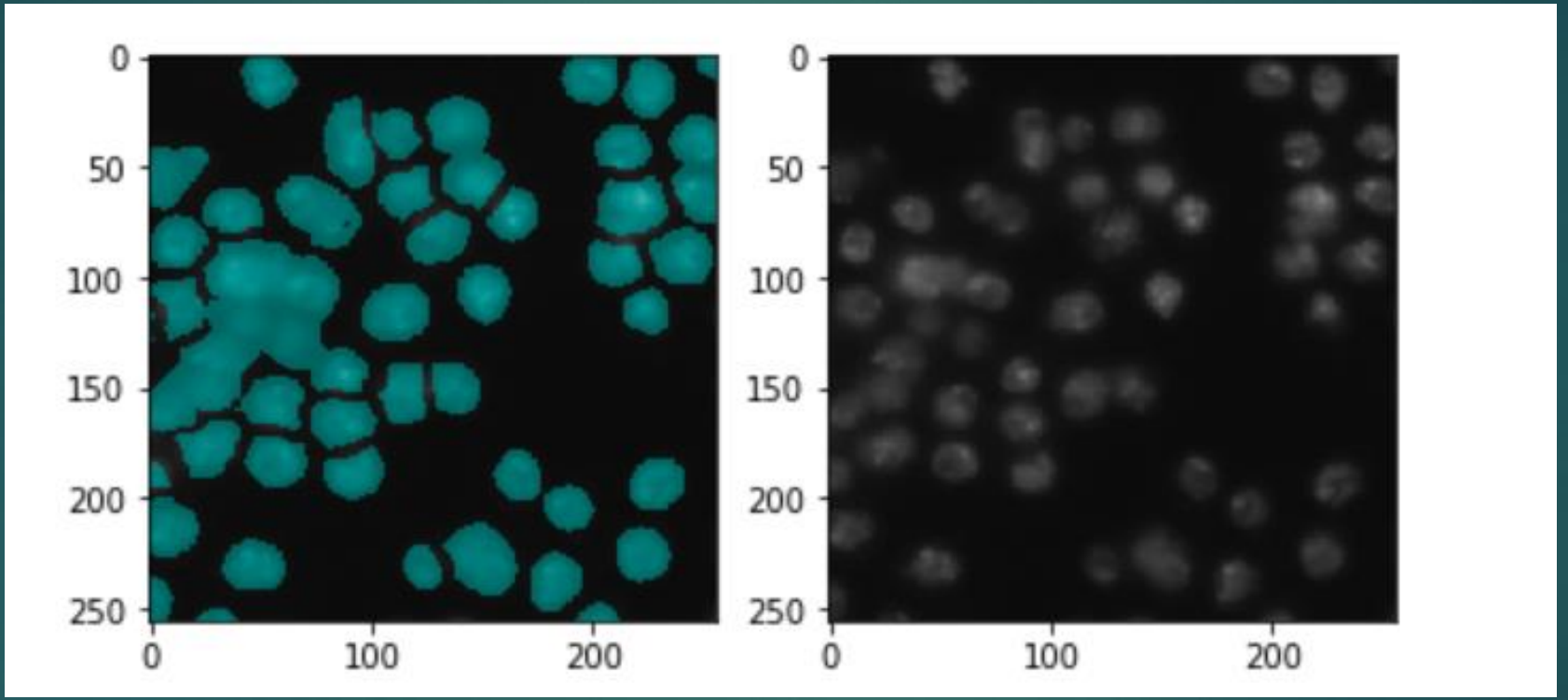
UNET - Overlapping Tile Strategy



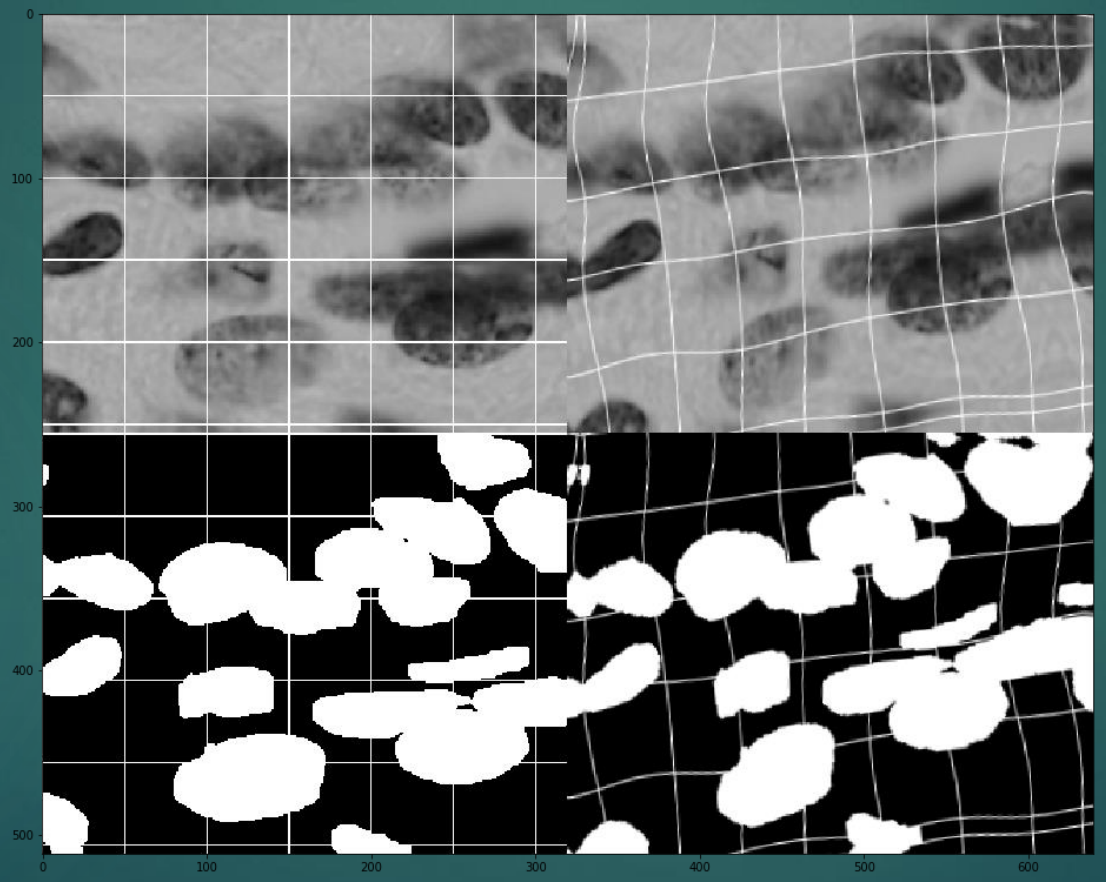
UNET – Accuracy and Method



First Implementation



Pre Process: Data Augmentation Elastic Deformations

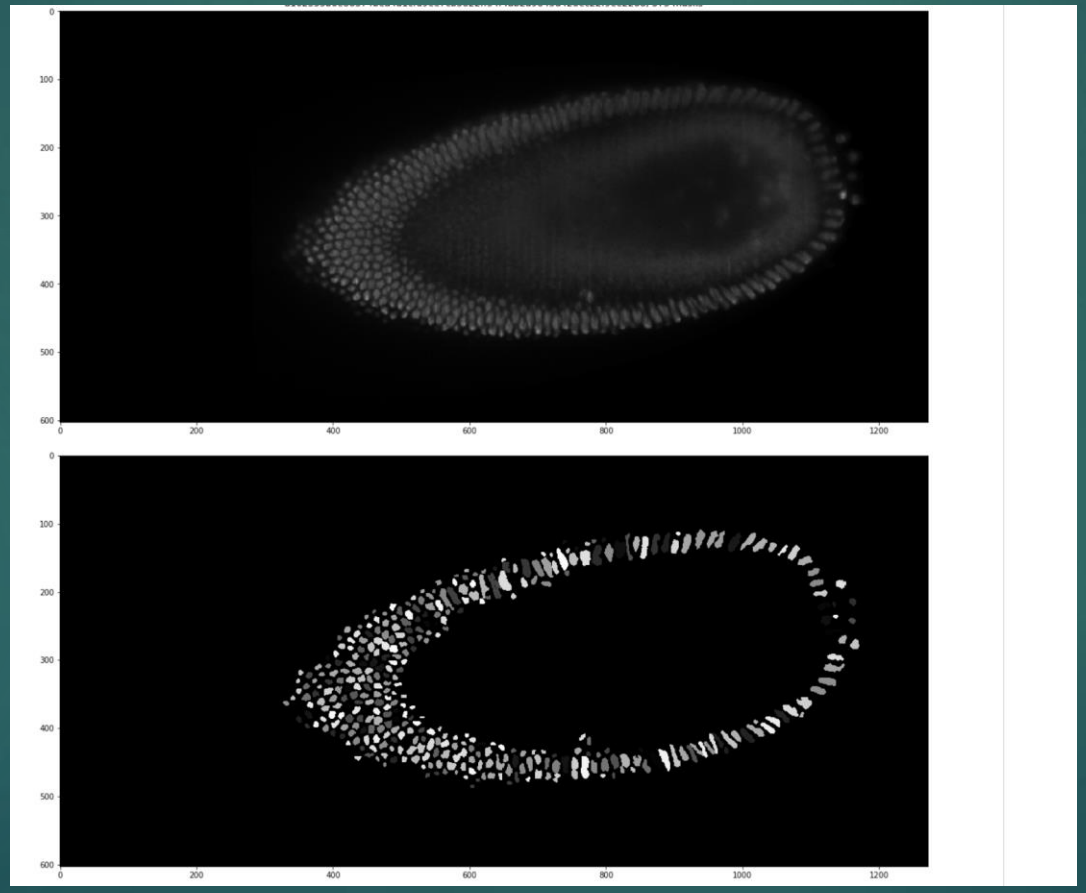


What did the winners do?

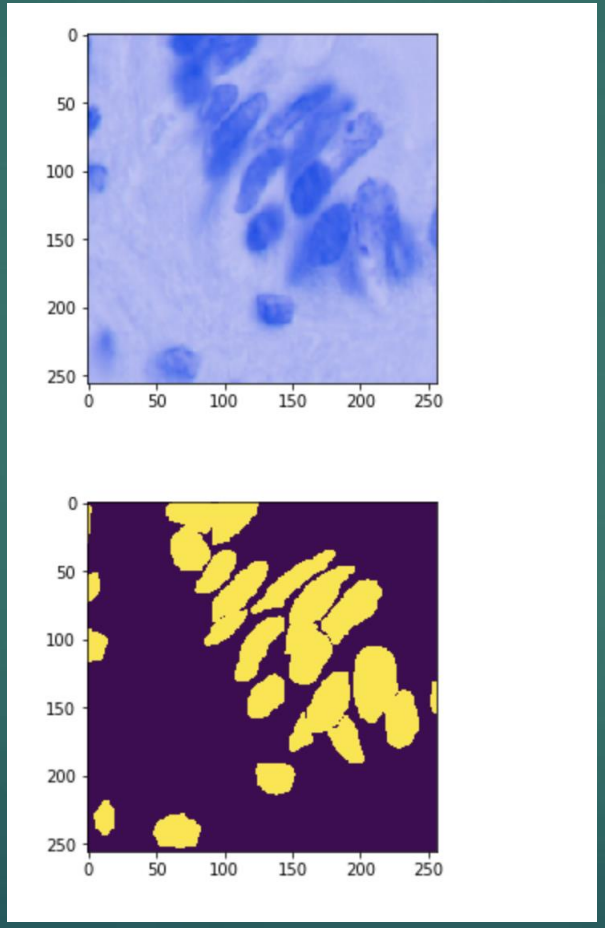
- Clache, Sharpen, Emboss
- Gaussian Noise
- Color to Gray
- Inverting
- Remapping grayscale images to random color images
- Blur, Median Blur, Motion Blur
- contrast and brightness
- random scale, rotates and flips
- Heavy geometric transformations: Elastic Transform, Perspective Transform, Piecewise Affine transforms, pincushion distortion
- Random HSV
- Channel shuffle - I guess this one was very important due to the nature of the data
- Nucleus copying on images. That created a lot of overlapping nuclei. It seemed to help networks to learn better borders for overlapping nuclei.

Post Processing: Watershed Algorithm

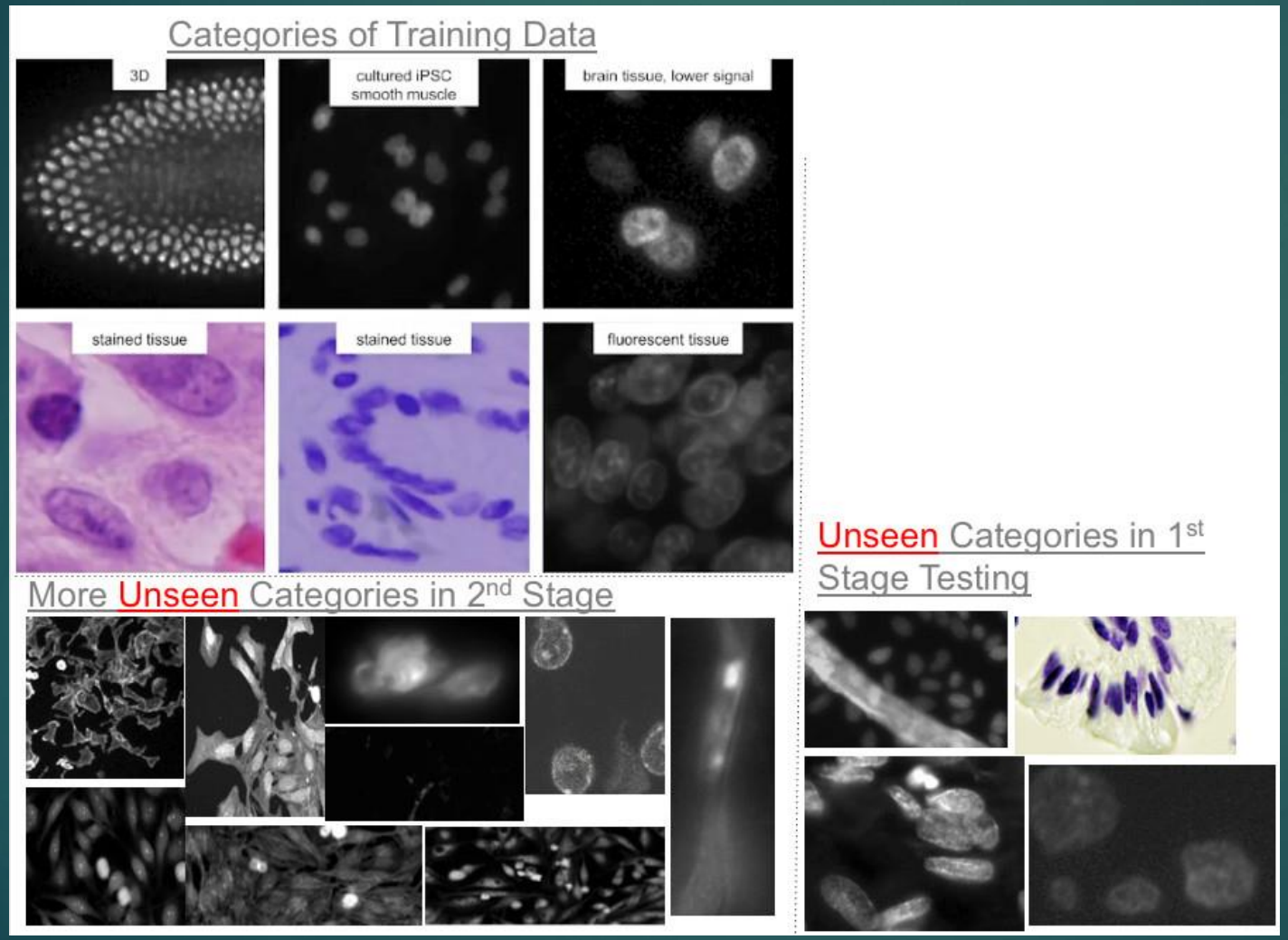
Analysis of Results



Analysis of Results



Data Variation in Stage 2!!!



Video of predictions

- https://drive.google.com/open?id=1ZdNzQOTR83vrqNPeynUYuelWFRxJ-35i