

# Level 4 Project

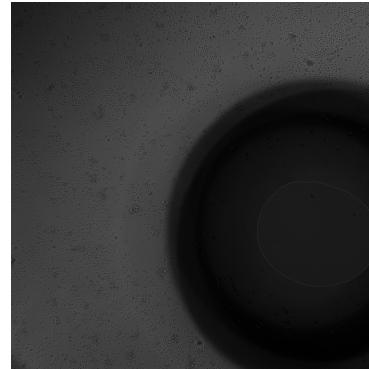
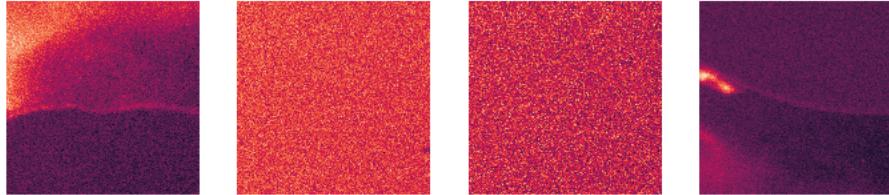
Week 2/15

(Week 1/14 Recap)

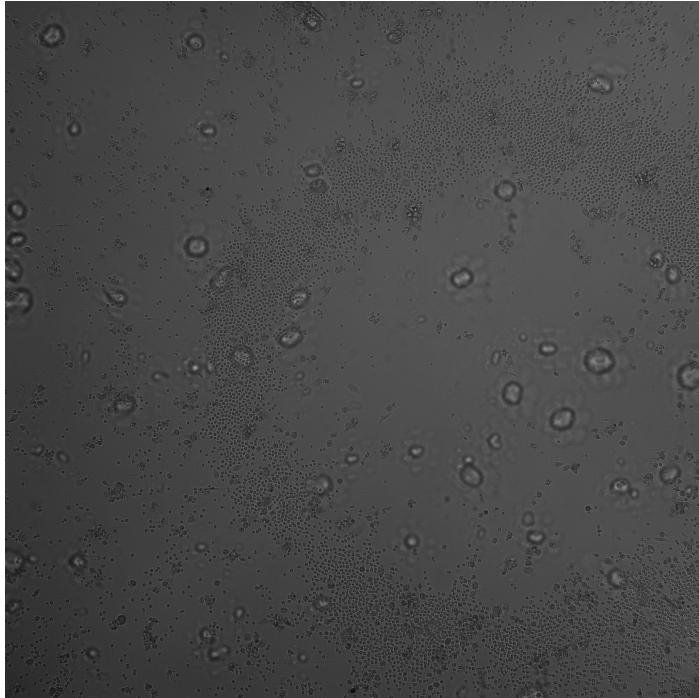
# Schedule for weeks 1-2 (14-15)

- Complete autoencoder tuning and image pre-processing.
- Deliverables:
  - Code for an autoencoder model
  - Code for pre-processing
  - Explanation for the choices made

# Outlier images



- This occurs when wells are impacted by a big blob (of water?)
  - Should they be eliminated?
    - Wasn't sure if I would have to manually eliminate them,
    - But I was having problems with NaN and it came from pictures like this



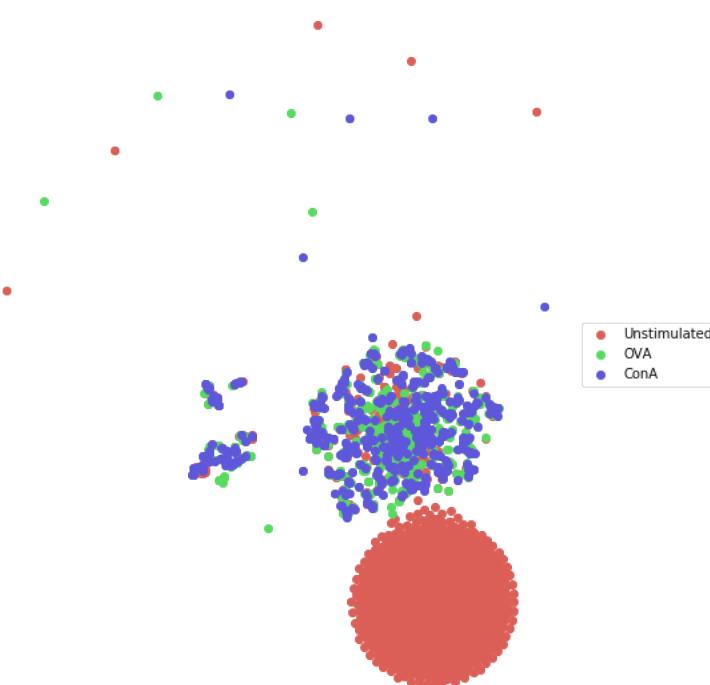
- Is this an outlier image?

# Processing method code

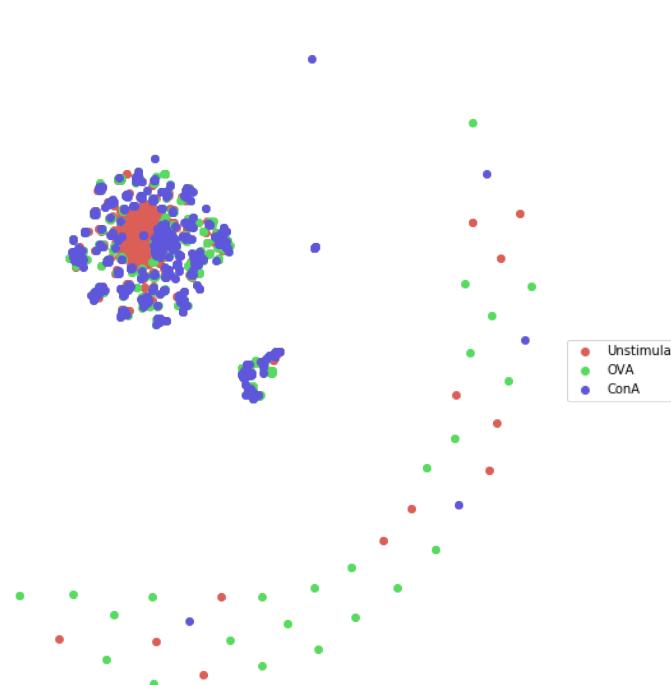
- Fixed problems
  - False assumption that minimum = 0.0 for all images
  - There's images where this was not the case
  - Found this while looking through outlier images
  - Proper min-max normalisation fixes this
    - This combined with the 255-clipping was causing the NaN problems

# Autoencoder results

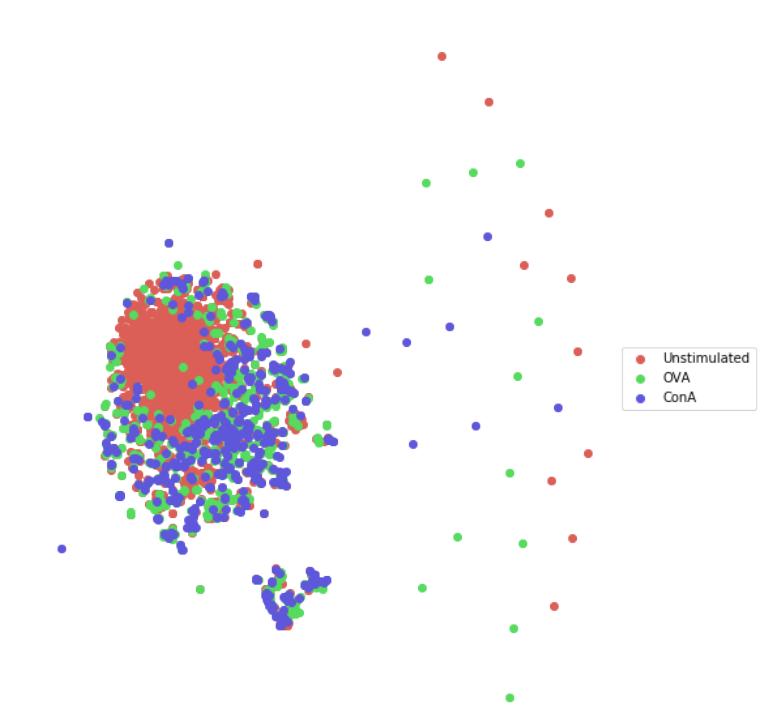
- Images masked out
- Basic autoencoder structure
- Final encoded dimensions:  
 $24*24*8=4,608$



- Images masked out
- Basic autoencoder structure
- Final encoded dimensions:  
 $12*12*8=1,152$



- Images masked out
- Using PReLU over ReLU
- Final encoded dimensions:  
 $12*12*8=1,152$



# Autoencoder structure

- Simpler convolution parameters is better
- PReLU performs better
  - Keeping reference of the papers I read to back my structure

# Thresholding v. K-means (1/2)

- Recall: I tested an image processing method where the main cells stood out

- Discovered in an odd way

```
def mean_clip(x):  
    mean = np.mean(x)  
    return np.clip(x, mean-126, mean+127)
```

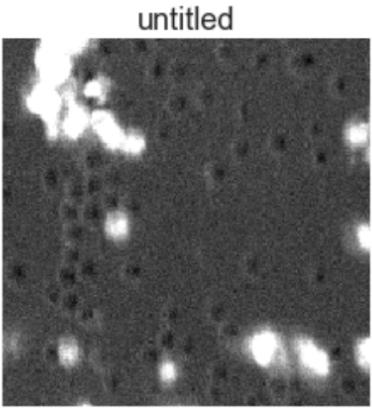
- It worked well, but why?

- 126/127 could represent some variation of noise
    - Standard deviation is a good measure of noise
    - Use that:

```
if pixel > mean+x.std():  
    mask[idx] = 1
```

# Thresholding v. K-means (2/2)

```
In [32]: show_image(mean_clip(img))
```



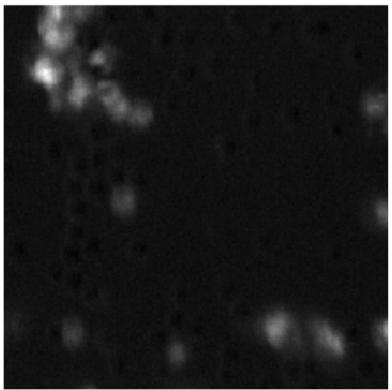
```
show_image(threshold(img))
```



Masks are very similar.

- Both satisfactory?

```
show_image(get_mask(img))
```



# Next steps for this week

- I want to try and see if I can get the blue/red clusters to separate (OVA/ConA)
  - Remove outlier images: maybe they impact learning
  - Otherwise this clustering obtained will have to do
- Finalise code as needed for the deliverables
- With Hannah:
  - Get overlap values
  - Organise session to sit at GE software and take notes?
  - Read provided articles

# Next steps for the coming week

- Use thresholding method to compare performance with K-means when masking out parts of the images.