Level 4 Project

Week 13

(Week 12/13 Recap)

Original plan for weeks 12 and 13

- **Week 12**
  - Finish evaluating best pre-processing method
  - Tune autoencoder for dimensionality reduction
  - Evaluate t-sne clustering

- **Week 13**
  - Visualise t-sne → changed visualisation function + gif function
  - Explore methods of segmentation:
    - k-means clustering (2 clusters)
    - contour finding
    - structure similarity
    - blob detection
    - Top-hat transformation?
  - Submit status report

T-SNE visualisation in GIF

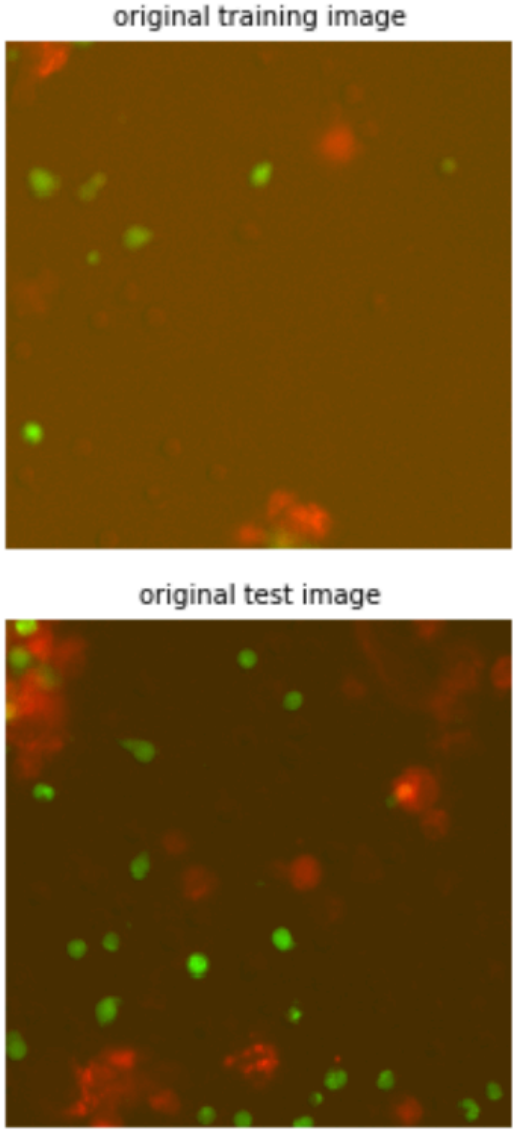
- Not pictured here because of the size of the GIFs
- https://drive.google.com/file/d/1SKd2M9Kv6xKsrUxS2M0NC2KuRyV2Ebf6/view?usp=sharing

Clipping DMSO dataset values under 255

- low_clip function transforms <255 values to 0
- max_normalise function normalises according to max pixel value of the image
- Performance is very similar to previous, if not slightly worse in terms of loss

Combining tcell, dc DMSO images into RGB image

- Red channel = dendritic cell
- Green channel = tcell
- Blue channel = empty



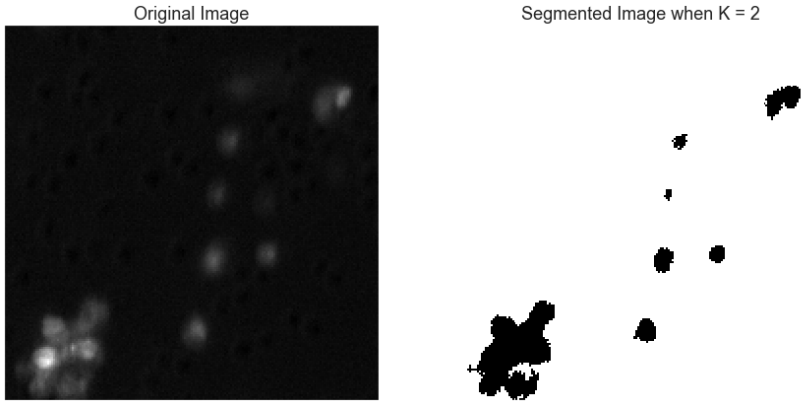
Summary of all models and outputs

- Available at: https://github.com/leonore/l4-project/blob/master/src/cell_autoencoder.ipynb
  - The important things are the titles and the images
- While doing this I found out that the issue I had been having with poor reconstructions all this time was because unclear documentation in Numpy made me think I was making copies of the dataset, but I was really reusing the array every time.
  - So normalisation over normalisation etc. resulted in a lot of loss.
  - So even 16-bit normalisation gets *some* kind of result but max normalisation still works a lot better.

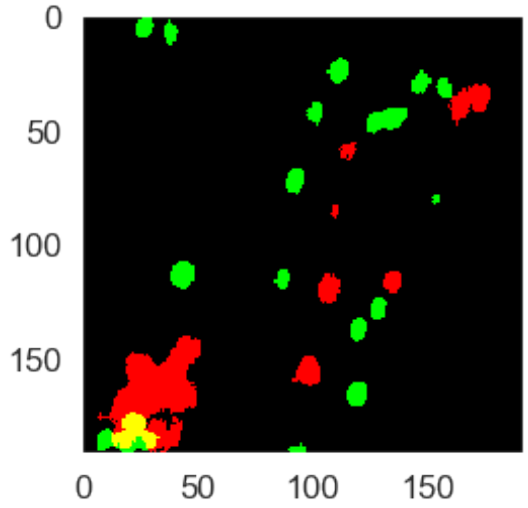
What I learned from the summary

- Always make a new copy of the data
- Combining the images in a RGB image seems to work better for clustering.

Image segmentation with K-means

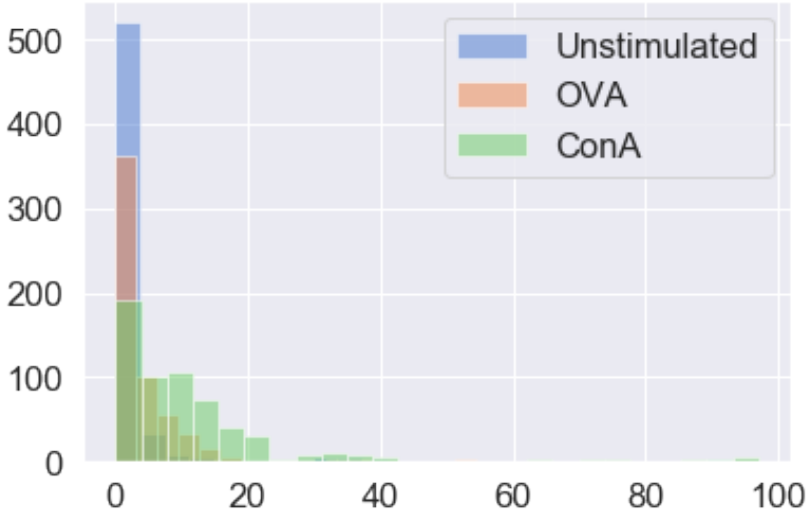


- Works very well for getting masks and direct overlap
- Overlap is shown in yellow below, calculated with intersection over union

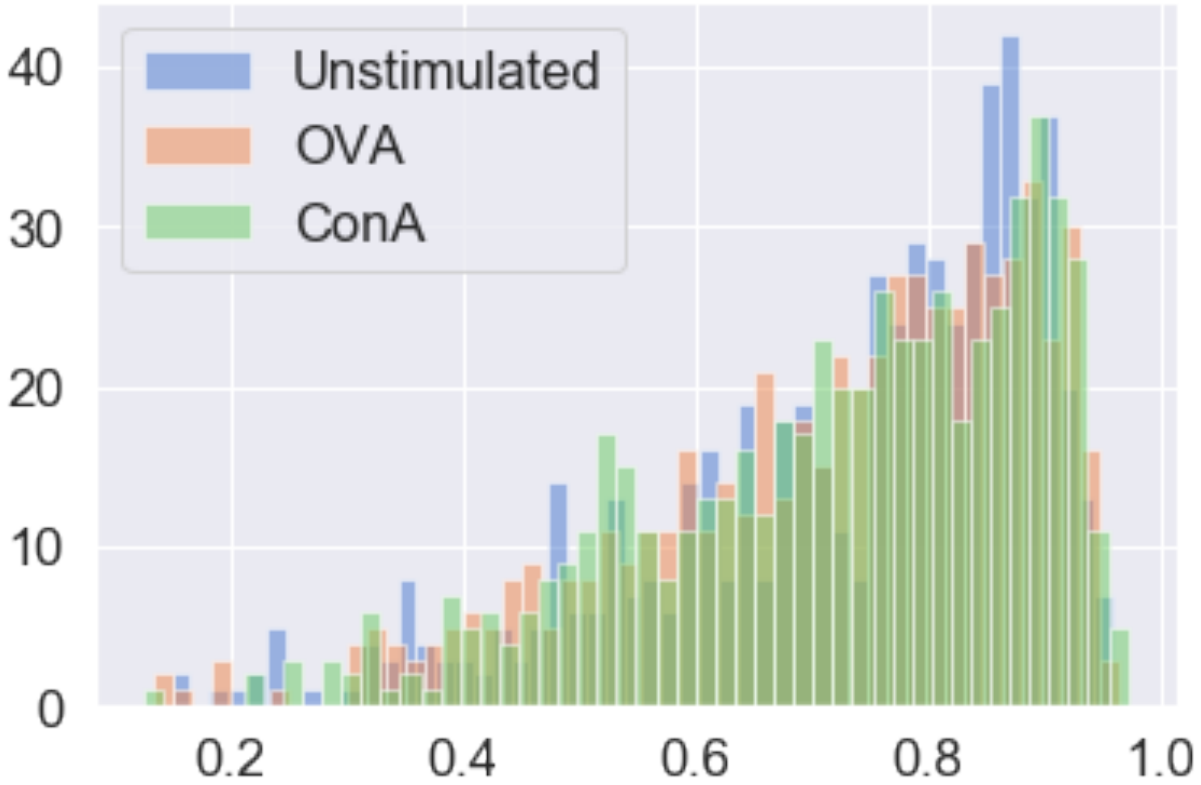


Overlap results with K-means

- Quite spread out, but the mean is maximum for ConA
  - This is in line with what Hannah has told me about ConA driving the most interaction at 60-80% overlap
  - Calculated % is much lower here though
- Sample size issues: both unstimulated and ConA have 9400 samples, but OVA has 1200
  - Results are consistent with a same-sized subsamples:
  - We get more 0% interaction with no stimulation, then OVA, then ConA

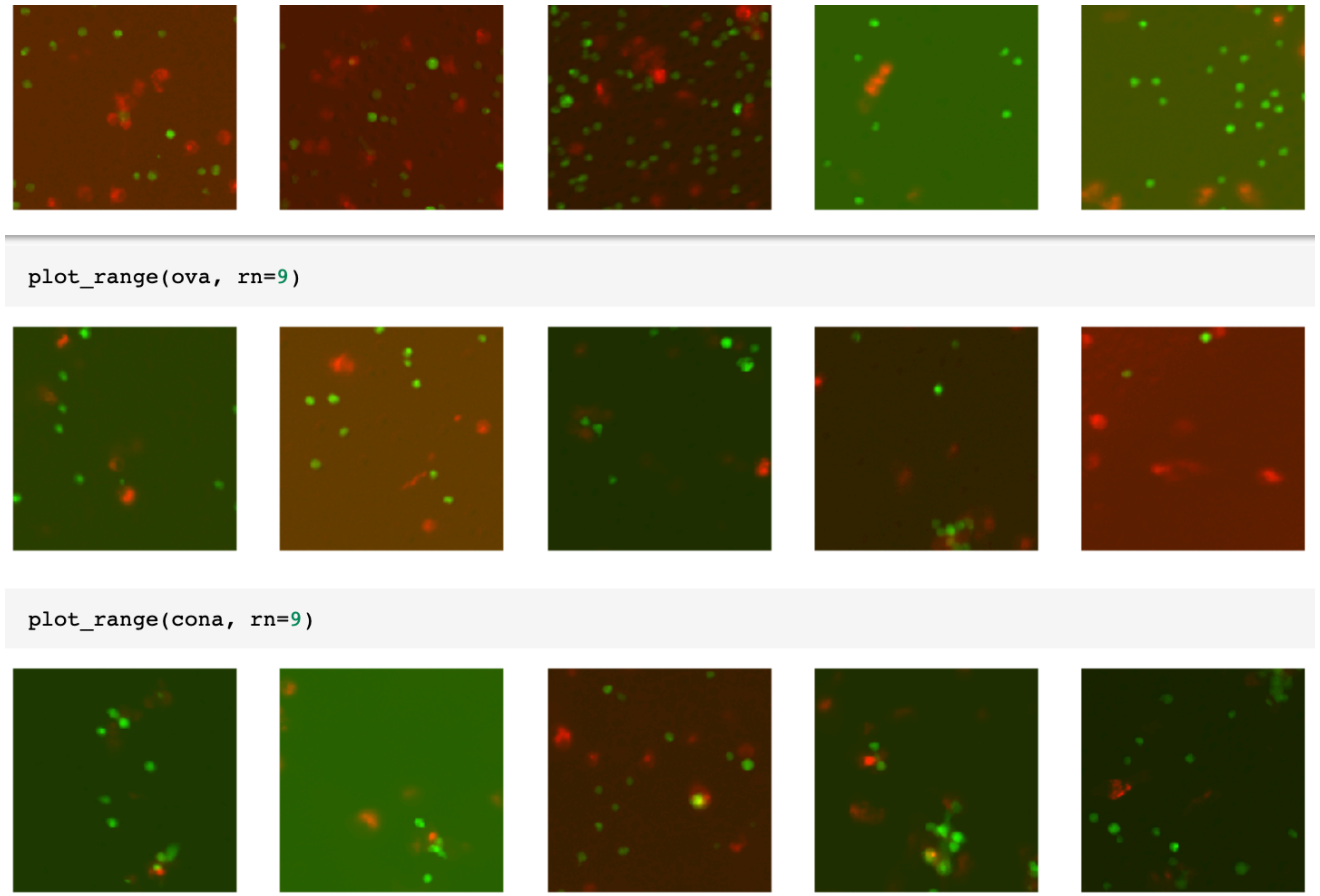


Investigating structural similarity index



Doesn’t seem to be a very promising technique as an interaction detection technique as they are all quite structurally similar (blobs of cells, lots of background)

Combining K-Means + autoencoder



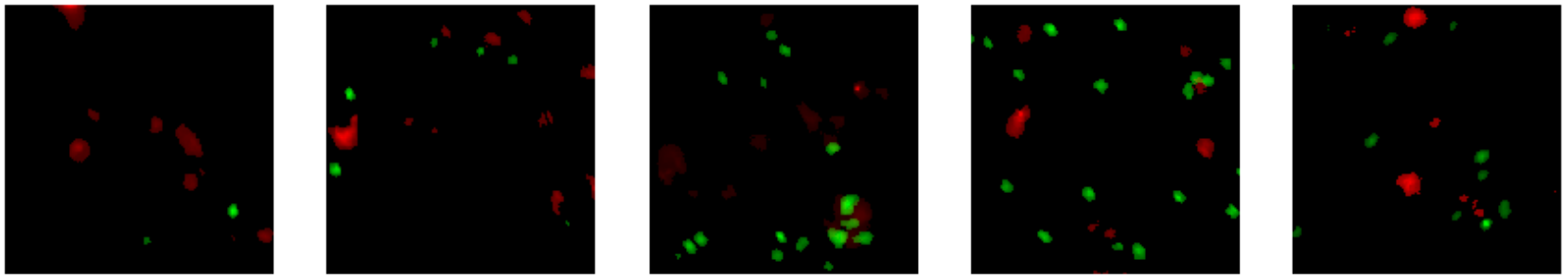
```
plot_range(ova, rn=9)
```



```
plot_range(cona, rn=9)
```


- Making a dataset with the images, but their background is masked out with the masks obtained through K-Means
- This is because as you can see, the images seem to be very dependent on the shade of the background

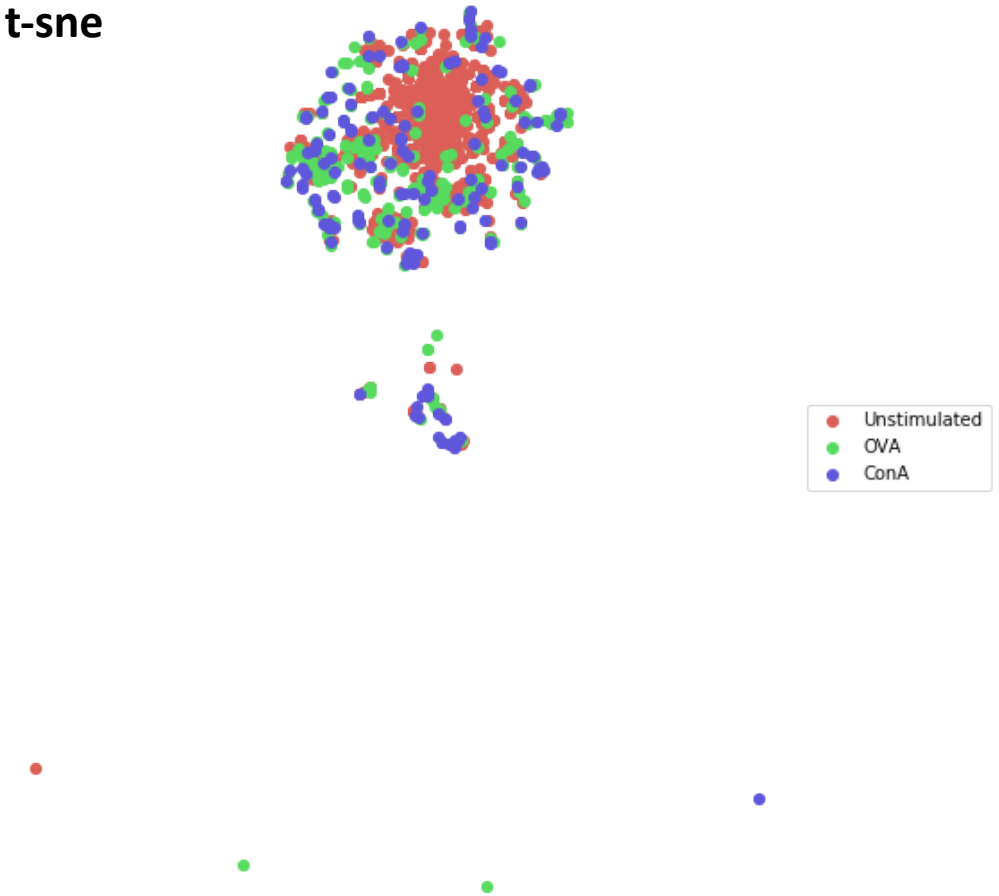
Combining K-Means + autoencoder



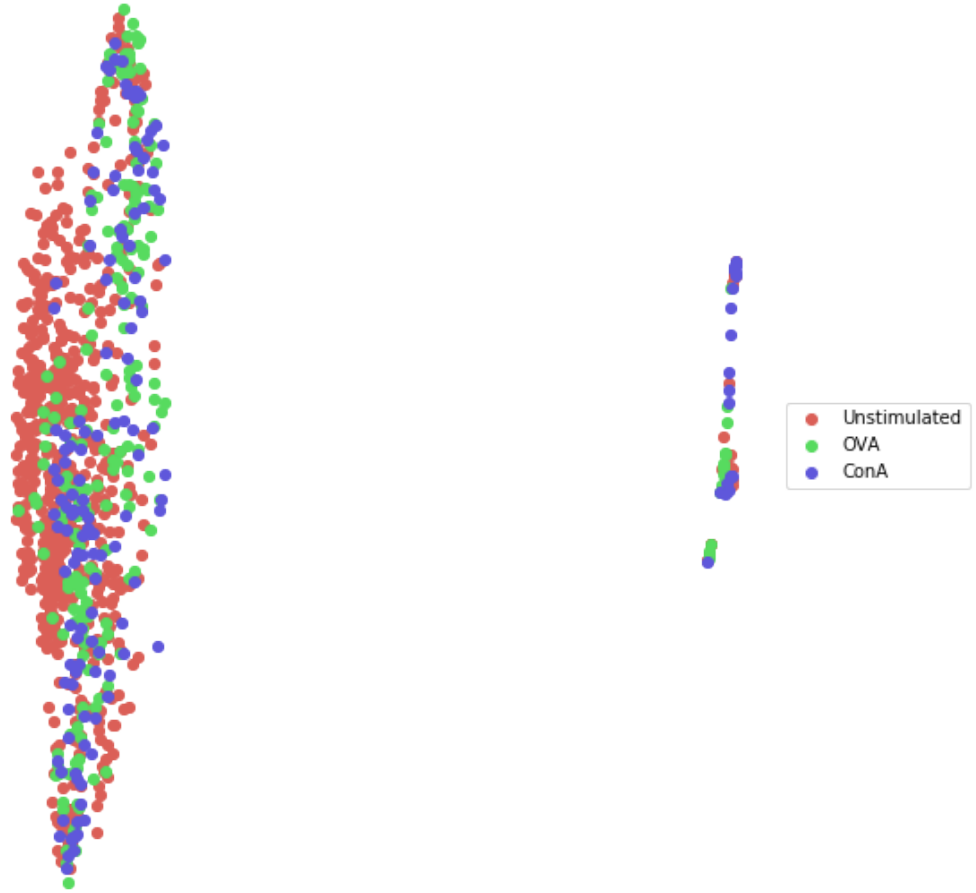
- As you can see the shades are much more consistent.
- And the clustering results are much more promising:
  - One main cluster but very tight “unstimulated” cluster within it
  - Some outliers which can come from empty images, images that suffer from the pre-processing, etc.

Combining K-Means + autoencoder

t-sne



UMAP



Next steps

- I want to re-investigate using a UNet like structure for detecting masks in unseen images and be able to differentiate between dendritic cells and t-cells.
  - It seems that this is Hannah mentioned at the beginning of the semester: how software struggles with dendritic cells, but not t-cells as they are blobs.
  - What do you think?
- Full plan is in status report