Ice Core Data Analysis

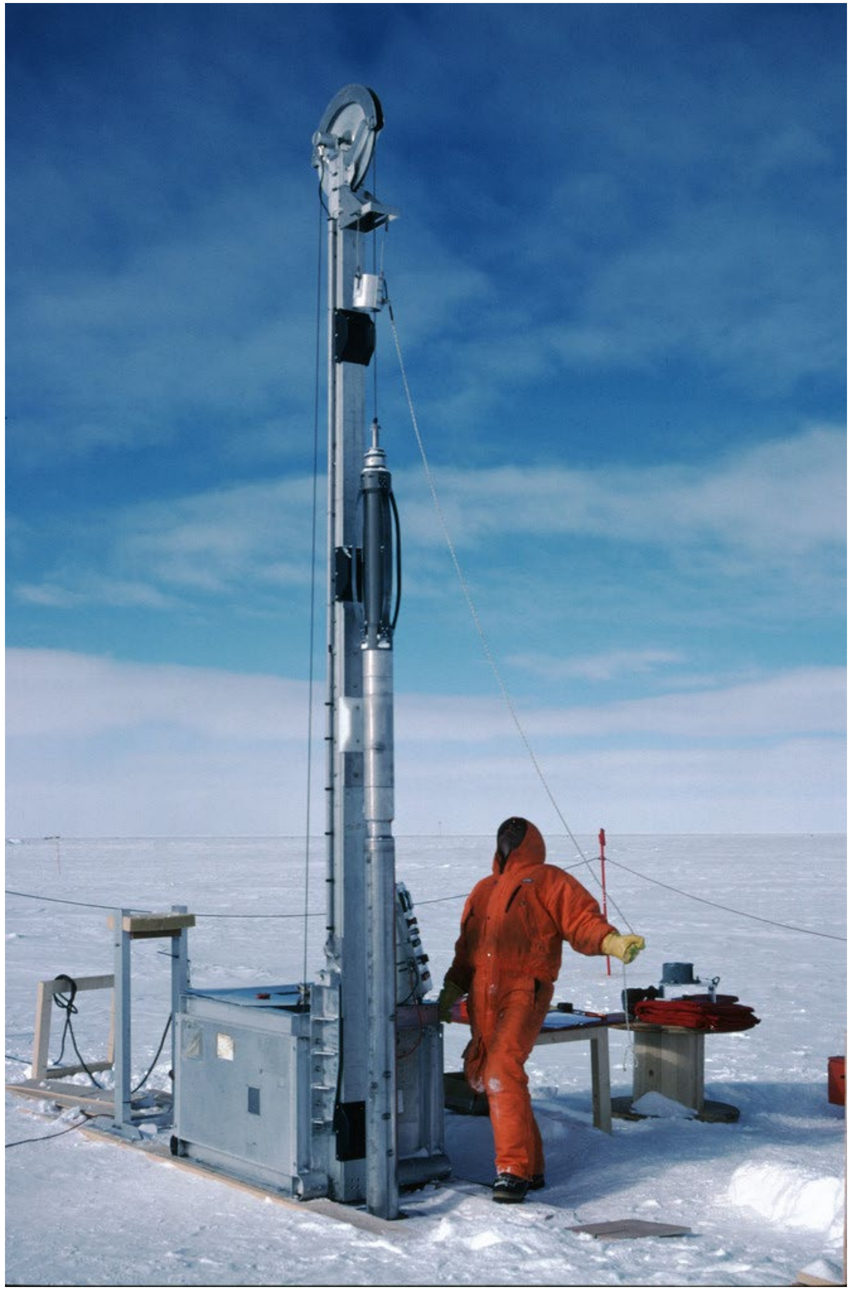
```
> get_participants()

{
    'Fabian Depenau Bjørnholt Jacobsen',
    'Povl Filip Sonne-Frederiksen',
    'Magnus Guldbæk Hansen',
    'Marcus Melhedegaard Thomsen'

}
```

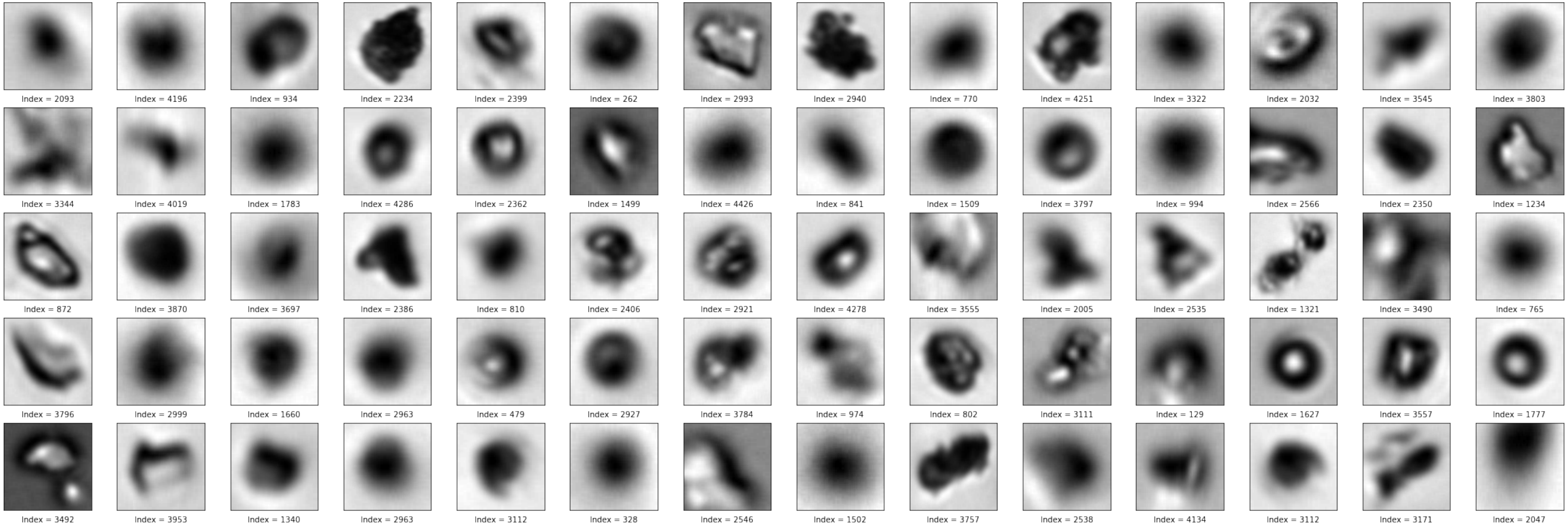
Overview

- Introduction
- Data
  - MNIST
  - Artificial dataset (labeled dataset from Nicolo)
  - Peruvian Ice Core Samples
- Methods
  - Classifier (Resnet18)
  - AutoEncoder (Resnet18 + Decoder)
  - Variational AutoEncoder (Resnet18 + Decoder)
- Analysis
  - Classifier
  - AutoEncoder
  - Variational AutoEncoder
- Findings
- Further Work



Introduction

The goal is to use dimensionality reduction to identify "interesting" objects within the Peruvian ice core dataset.



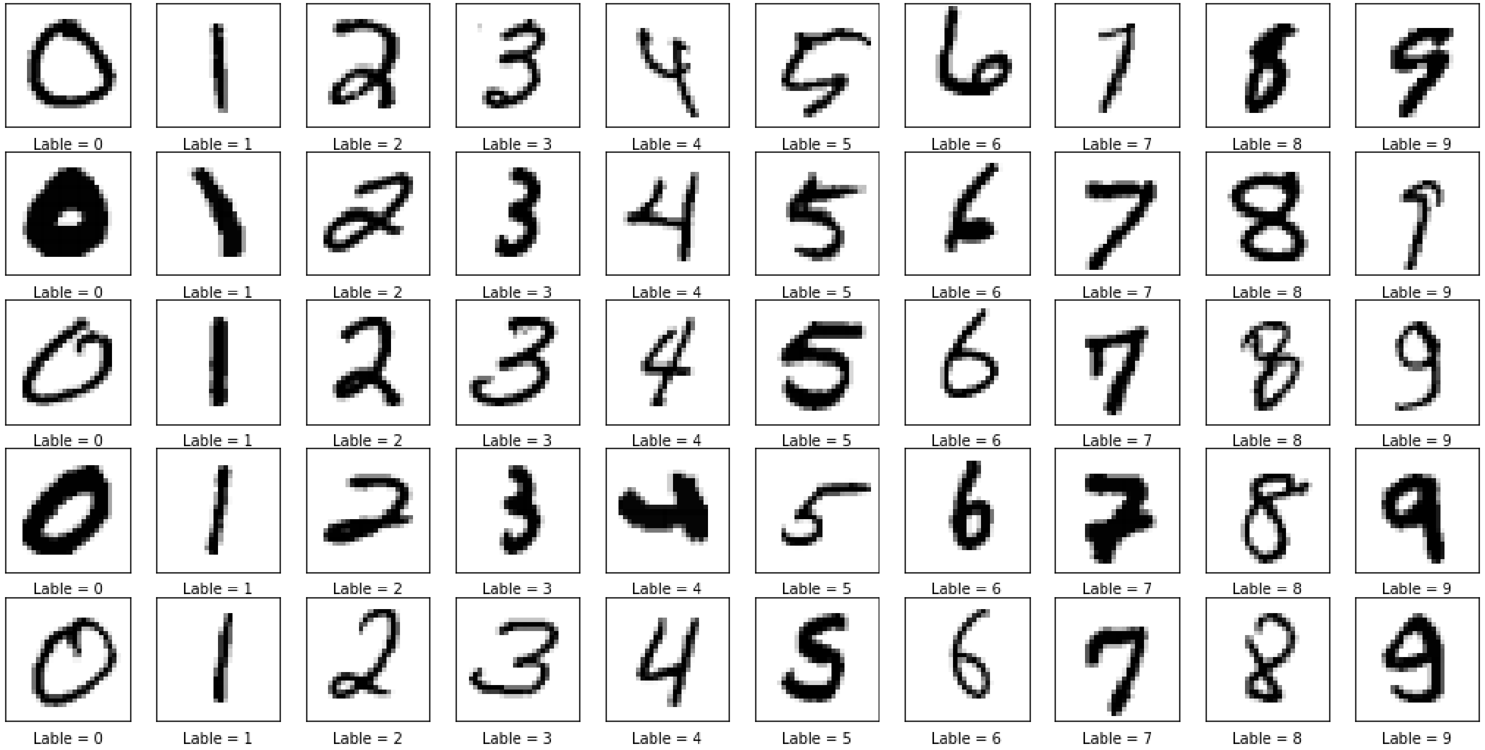
Data

A look at the datasets and their structures

MNIST

Handwritten digits

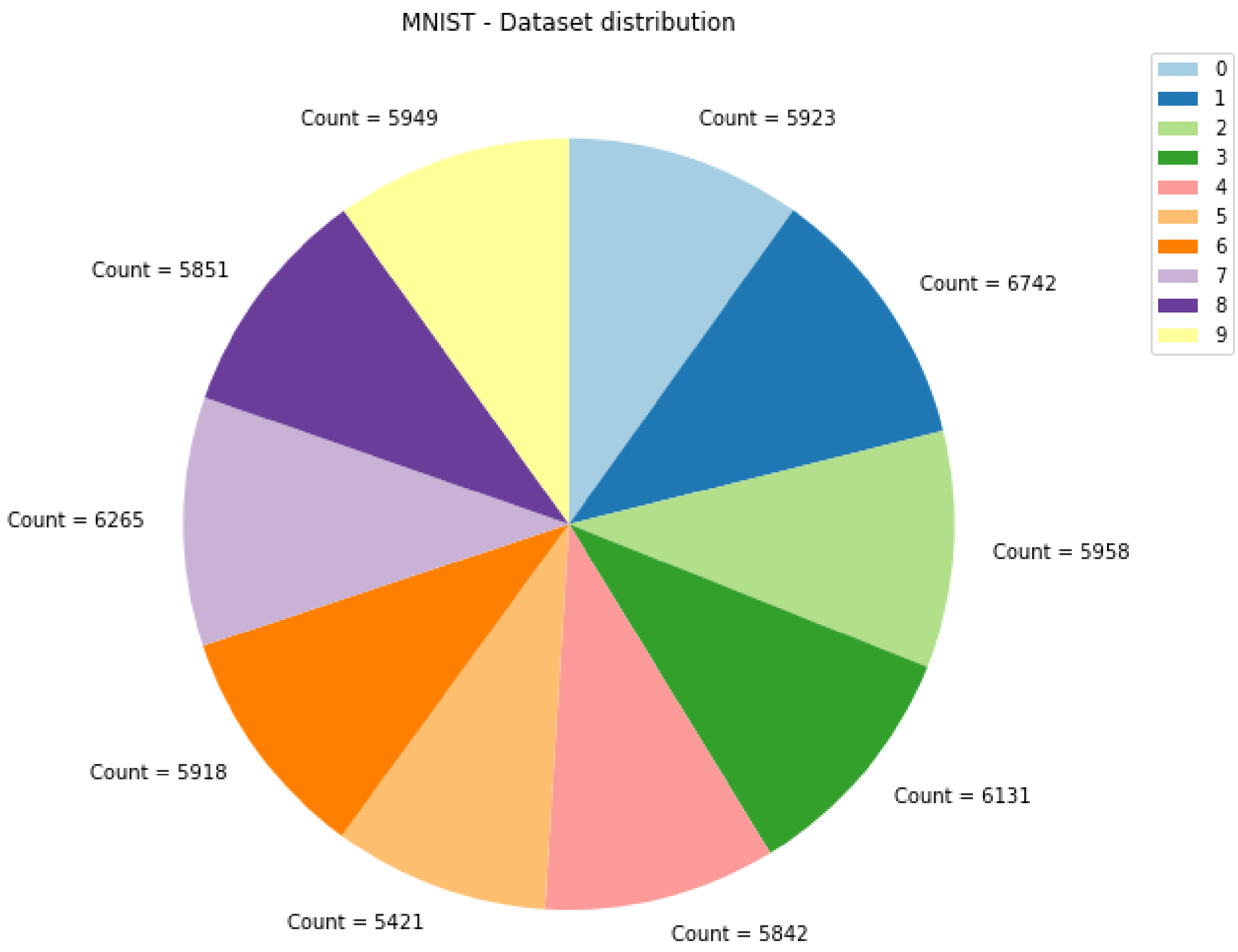
- 70.000 samples
- 28x28p size
- Balanced
- Preprocessed



MNIST

Handwritten digits

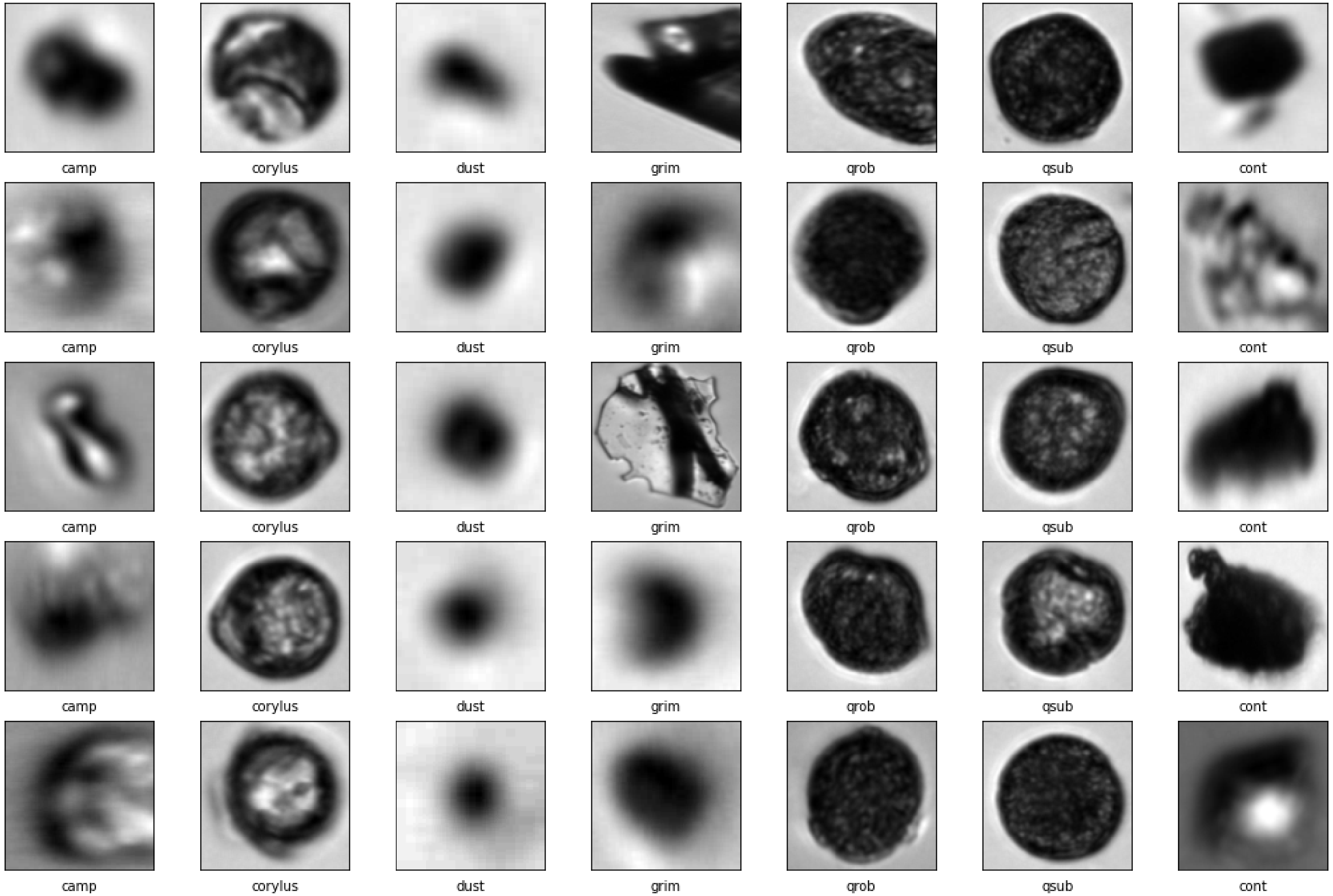
- 70.000 samples
- 28x28p size
- Balanced
- Preprocessed



Artificial

Image samples

- 145.242 samples
- Resized to 128x128p
- 34 scalars
- 7 Labels
- Unbalanced



Artificial

Image samples

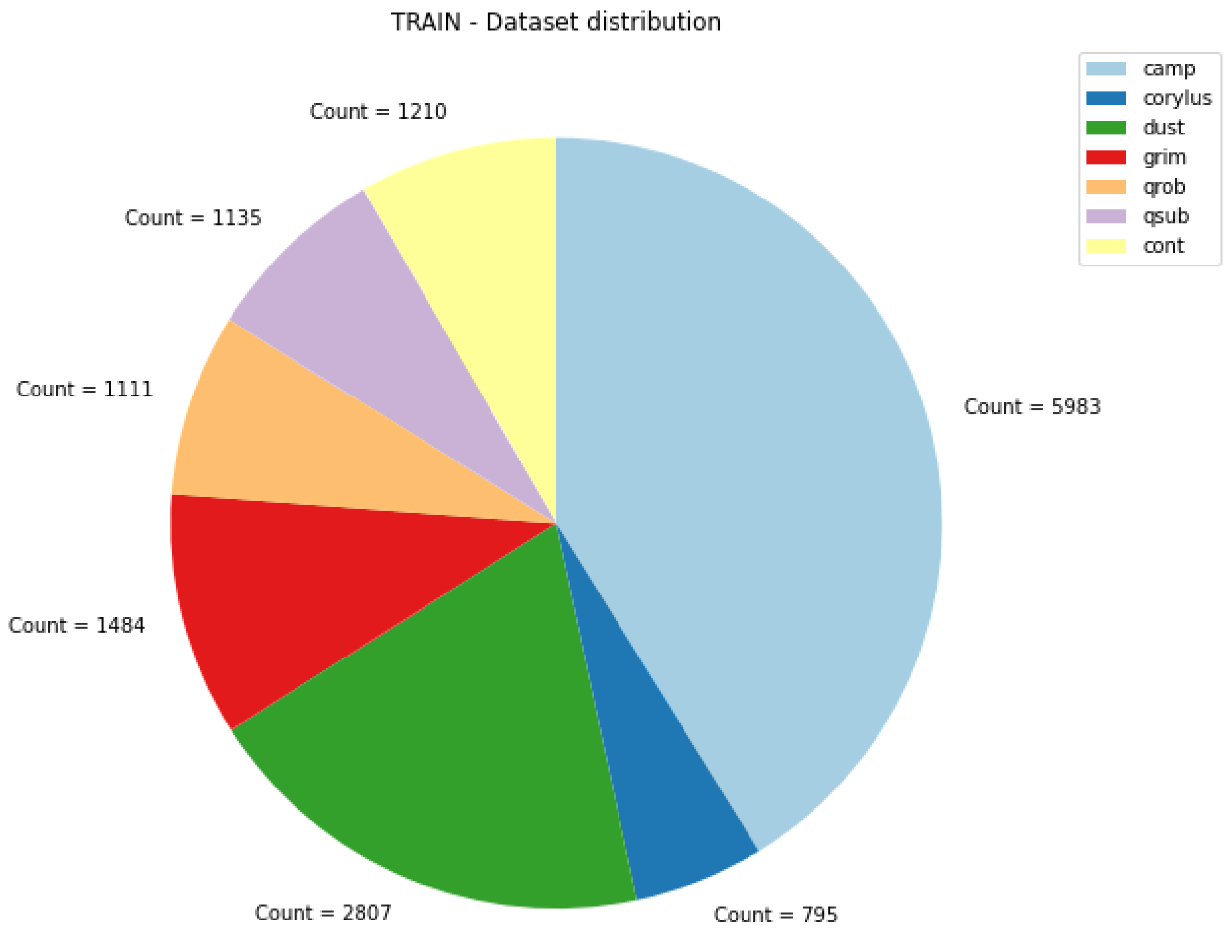
- 145.242 samples
- Resized to 128x128p
- 34 scalars
- 7 Labels
- Unbalanced

	Particle ID	Area (ABD)	Area (Filled)	Aspect Ratio	Biovolume (Cylinder)	...	Fiber Straightness	Filter Score	Geodesic Aspect Ratio	Geodesic Length	Geodesic Thickness
0	3733	79.16	79.97	0.88	446.21	...	0.69	0	0.32	17.89	5.64
1	5528	9.85	9.85	0.67	47.64	...	1.2	0	1.0	3.93	3.93
2	27	711.31	716.33	0.71	13121.26	...	1.05	0	0.6	36.07	21.52
3	2780	54.6	54.6	0.5	293.64	...	0.88	0	0.37	13.87	5.19
4	3510	41.53	43.93	0.77	209.93	...	0.69	0	0.34	13.33	4.48
5	4560	49.15	49.15	0.6	227.46	...	0.79	0	0.3	14.92	4.41
6	231	492.14	1453.44	0.5	30433.77	...	1.01	0	0.41	61.07	25.19
7	3730	20.15	20.15	0.8	102.9	...	0.85	0	0.56	7.43	4.2
8	359	34.46	36.44	0.42	154.61	...	0.89	0	0.29	13.16	3.87
9	1041	2.16	2.16	0.55	12.67	...	1.08	0	1.0	2.53	2.53
10	538	3010.61	3043.2	0.56	57890.21	...	0.67	0	0.17	135.95	23.28
11	985	953.0	953.0	0.24	11772.3	...	0.99	0	0.21	69.68	14.67
12	3630	6.57	6.57	0.59	32.51	...	1.26	0	1.0	3.46	3.46
13	1024	24.32	25.07	0.53	95.4	...	0.79	0	0.29	11.43	3.26
14	4421	34.97	36.12	0.54	178.42	...	0.92	0	0.4	11.23	4.5
15	24910	11286.34	12293.88	0.29	217302.8	...	0.5	0	0.04	568.22	22.07
16	9695	1356.39	1357.79	0.94	22830.78	...	0.63	0	0.28	71.32	20.19
17	611	125.32	125.32	0.29	695.85	...	1.06	0	0.23	25.77	5.86
18	114	466.39	508.6	0.89	9016.04	...	1.24	0	1.0	22.56	22.56
19	7737	1339.99	1474.94	0.46	16981.02	...	0.67	0	0.12	112.62	13.86
20	76	10.36	10.36	0.71	55.07	...	1.12	0	1.0	4.12	4.12
21	2304	27.5	27.5	0.44	70.27	...	0.6	0	0.12	18.07	2.22
22	82	494.23	501.66	0.87	9878.65	...	1.22	0	1.0	23.26	23.26
23	932	3841.71	3854.44	0.8	100845.08	...	0.71	0	0.26	124.11	32.16
24	1002	5.06	5.06	0.53	26.5	...	1.21	0	1.0	3.23	3.23
25	436	846.26	846.26	0.32	11550.66	...	1.05	0	0.29	56.47	16.14
26	1718	791.41	791.41	0.95	15907.62	...	1.23	0	1.0	27.26	27.26
27	4090	19.84	19.84	0.81	113.07	...	1.13	0	1.0	5.24	5.24
28	2251	21.55	21.55	0.69	124.98	...	1.24	0	1.0	5.42	5.42
29	2558	10.19	10.19	0.55	53.09	...	1.3	0	1.0	4.07	4.07
30	557	0.4	0.4	0.39	1.33	...	0.95	0	1.0	1.19	1.19
31	5305	815.77	815.77	0.76	13897.48	...	0.87	0	0.46	43.73	20.12
32	3892	24.01	24.01	0.24	27.0	...	0.77	0	0.03	37.65	0.96
33	386	1887.91	1907.26	0.63	14429.87	...	0.34	0	0.04	218.66	9.17
34	17161	401.0	402.07	0.51	4071.36	...	0.86	0	0.3	38.6	11.59
35	967	21.24	21.24	0.8	115.7	...	1.14	0	1.0	5.28	5.28
36	2869	1076.36	1083.77	0.86	21755.35	...	0.89	0	0.5	48.35	23.94
37	227	35.1	35.1	0.55	169.29	...	0.83	0	0.39	11.28	4.37
38	49	465.67	489.97	0.9	6801.59	...	0.82	0	0.48	33.64	16.04
39	927	1139.05	1143.45	0.81	15048.28	...	0.67	0	0.2	77.54	15.72
40	1382	919.66	942.76	0.82	15861.03	...	0.79	0	0.39	50.65	19.97
41	18998	80.1	80.1	0.6	388.58	...	0.72	0	0.24	20.6	4.9
42	4874	21.67	21.67	0.63	118.23	...	0.97	0	0.63	7.28	4.55
43	5382	958.78	958.78	0.89	20170.92	...	0.92	0	0.61	41.14	24.98
44	199	476.19	504.41	0.86	4743.64	...	0.56	0	0.21	50.98	10.88
45	1024	39.91	41.27	0.57	203.68	...	0.82	0	0.37	12.34	4.58
46	4216	47.71	48.29	0.2	135.1	...	0.8	0	0.11	24.38	2.66
47	3207	7.0	7.0	0.84	30.93	...	1.04	0	1.0	3.4	3.4
48	3247	602.16	604.12	0.6	4836.81	...	0.57	0	0.13	70.58	9.34
49	1371	394.89	394.89	0.55	5590.32	...	1.19	0	0.6	27.16	16.19

Artificial

Image samples

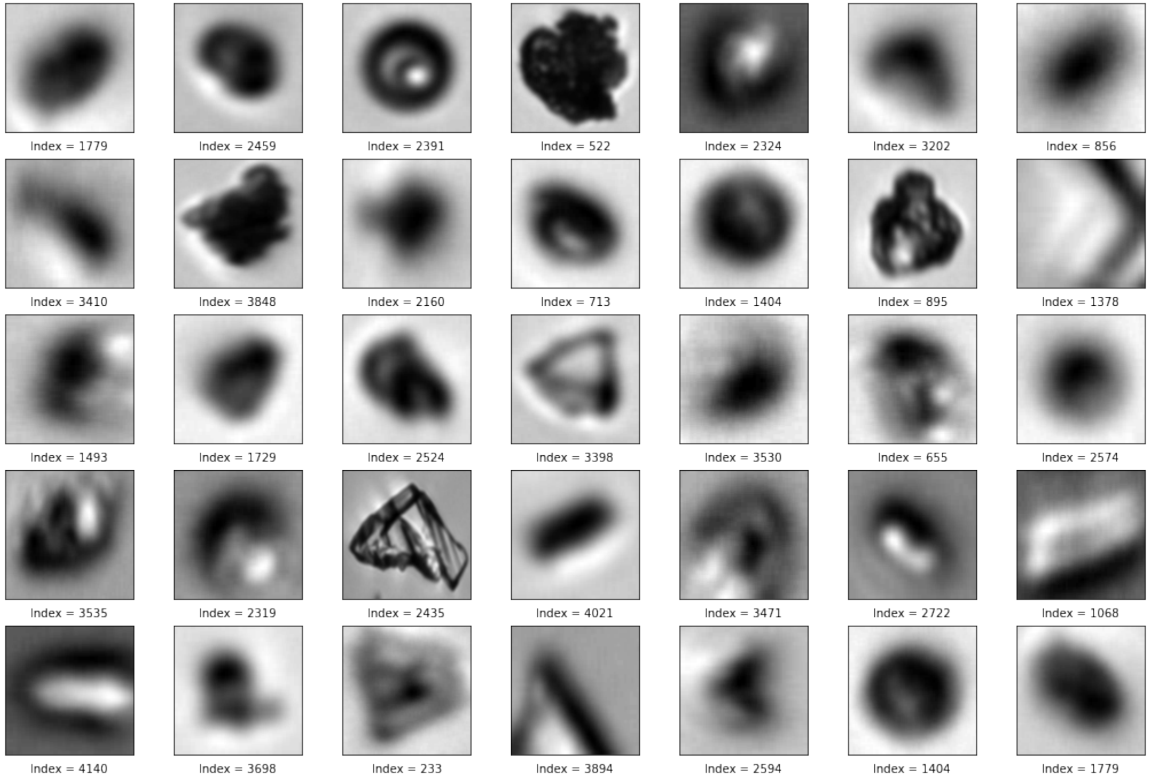
- 145.242 samples
- Resized to 128x128p size
- 34 scalars
- 7 Labels
- Unbalanced



Peruvian Ice Core Samples

Image samples

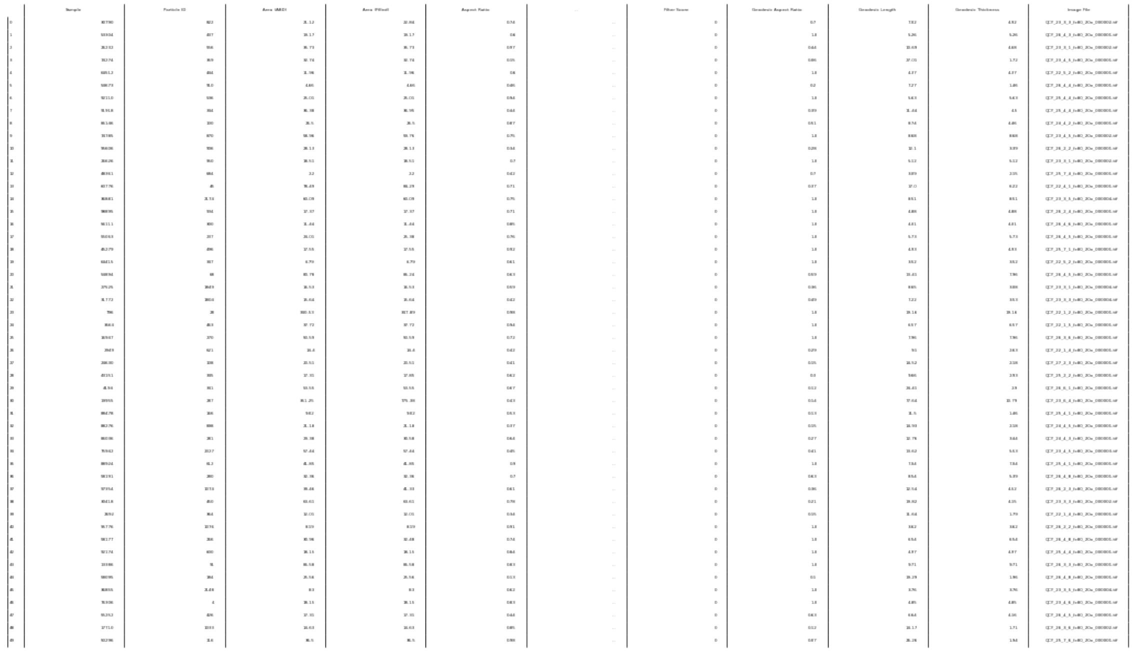
- 102.763 samples
- Resized to 128x128p
- 34 scalars
- Probably extremely unbalanced with a lot being dust



Peruvian Ice Core Samples

Image samples

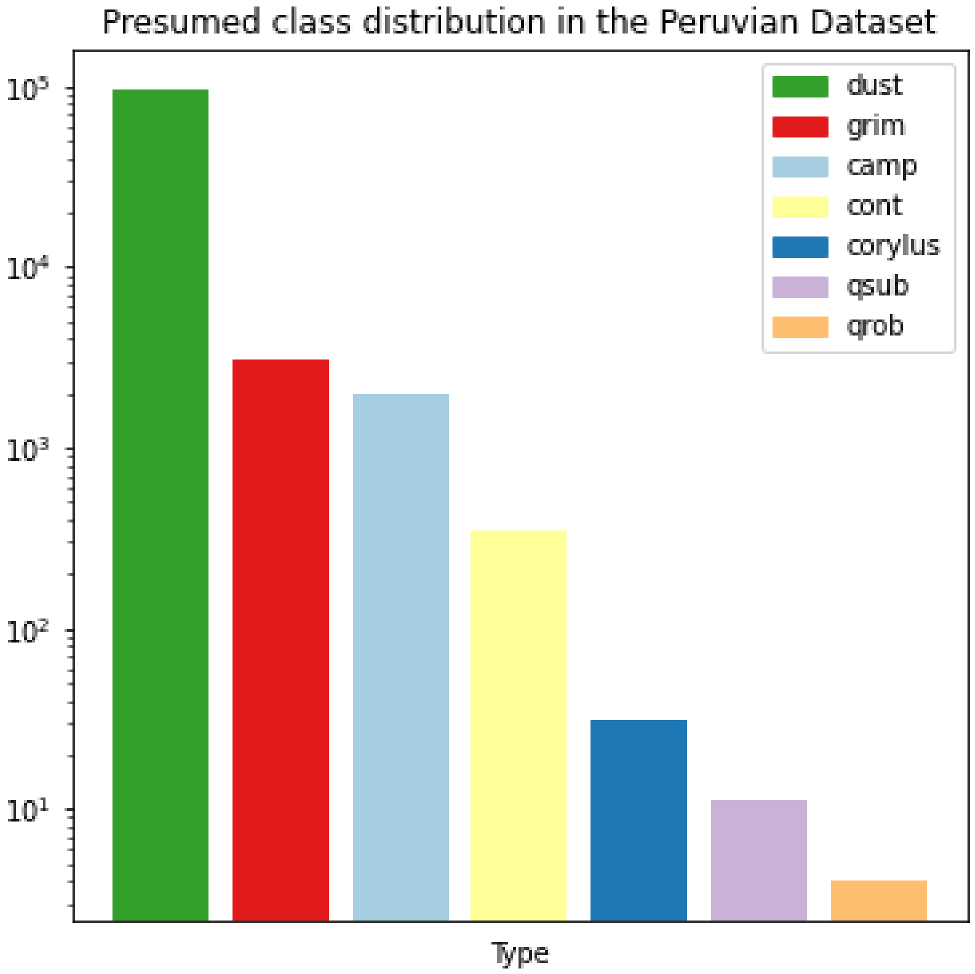
- 102.763 samples
- Resized to 128x128p size
- 34 scalars
- Probably extremely unbalanced with a lot being dust



Peruvian Ice Core Samples

Image samples

- 102.763 samples
- Resized to 128x128p size
- 34 scalars
- Probably extremely unbalanced with a lot being dust



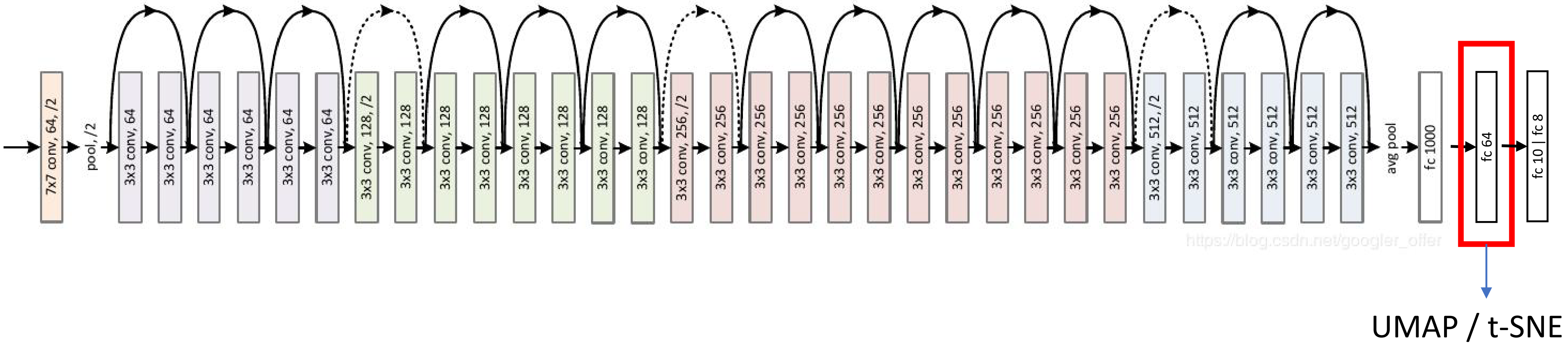
We remove all that the classifier characterizes as dust from the Peruvian set and look at the rest.
We include only images above 15x15 pixels in size

Methods

How did we engage with the data

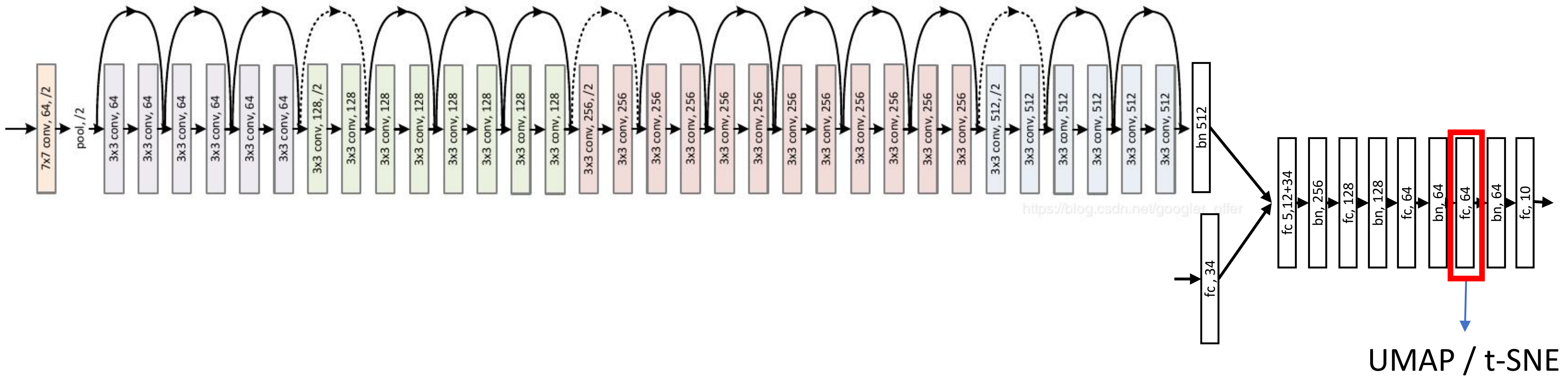
Classifier – Resnet18 on MNIST

- See if classification can be used to find unknow categories by mapping the n-1 layer in the network



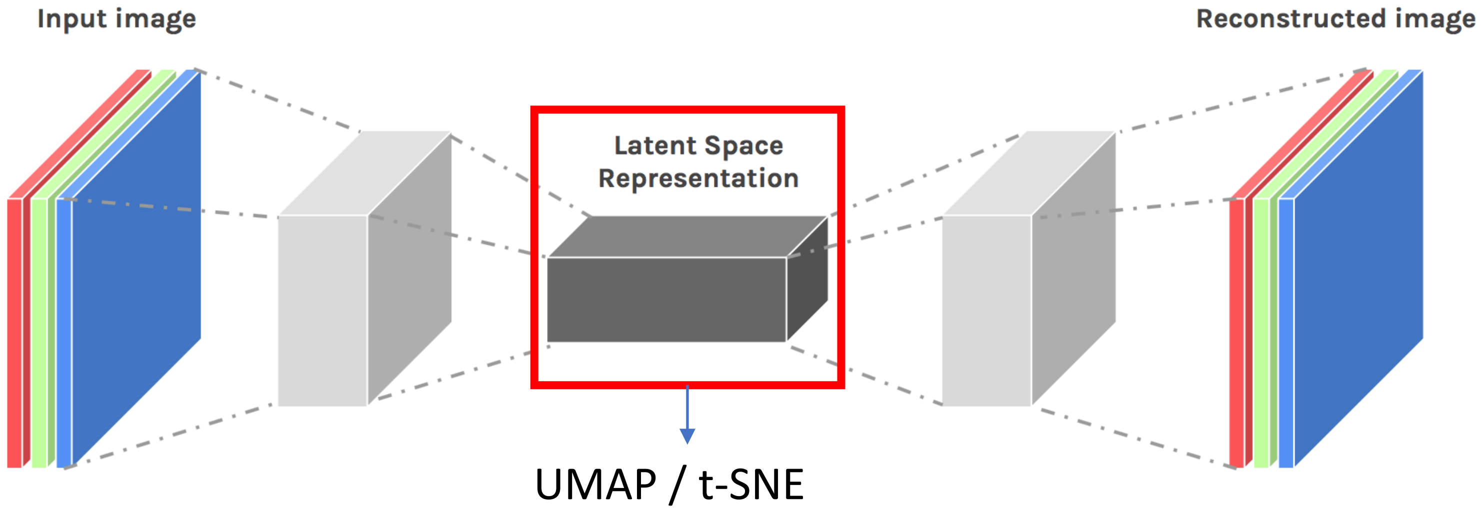
Classifier – Resnet18 on Artificial and Peruvian Datasets

- See if classification can be used to find unknow categories by mapping the n-1 layer in the network



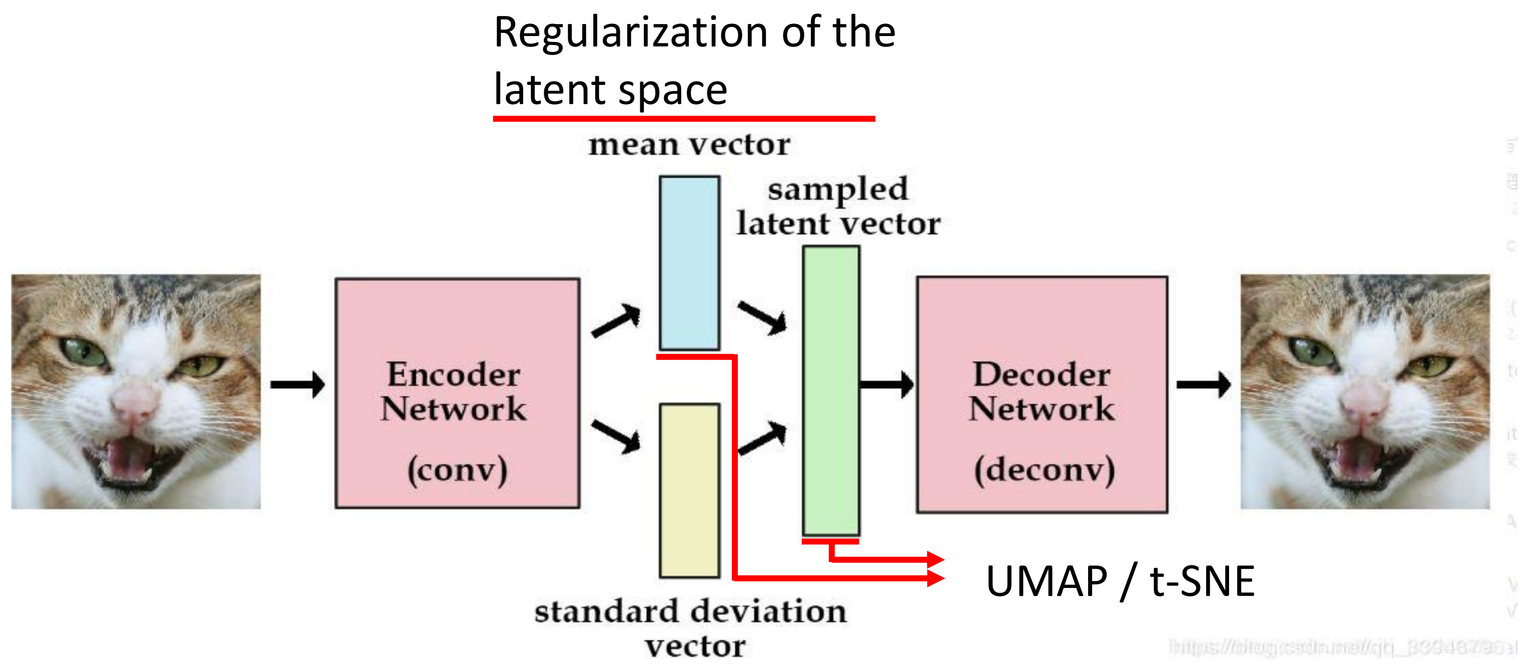
AutoEncoder

- See if the bias can be reduced by encoding a latent space

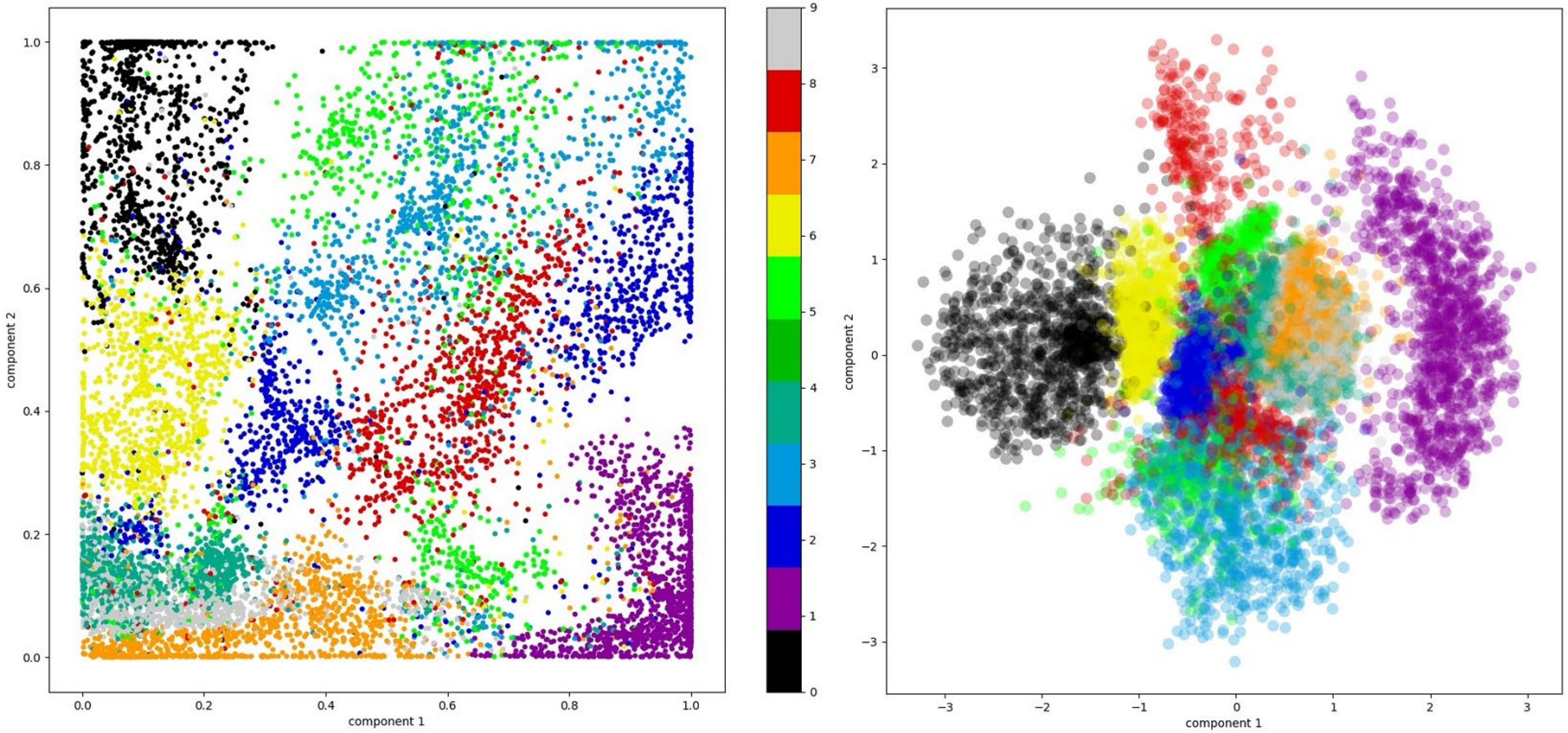


Variational AutoEncoder

- Regularizing the latent space to reduce overfitting



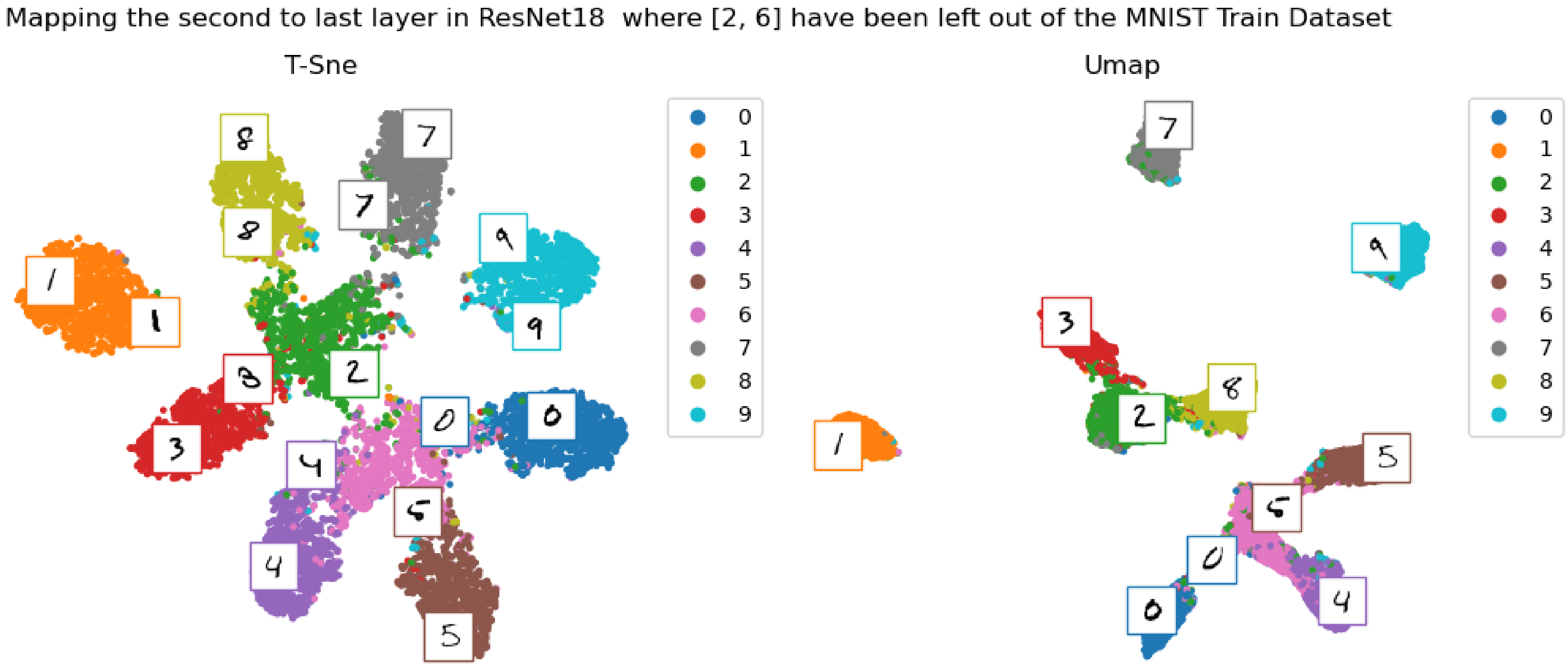
Example of latent space for AE vs VAE



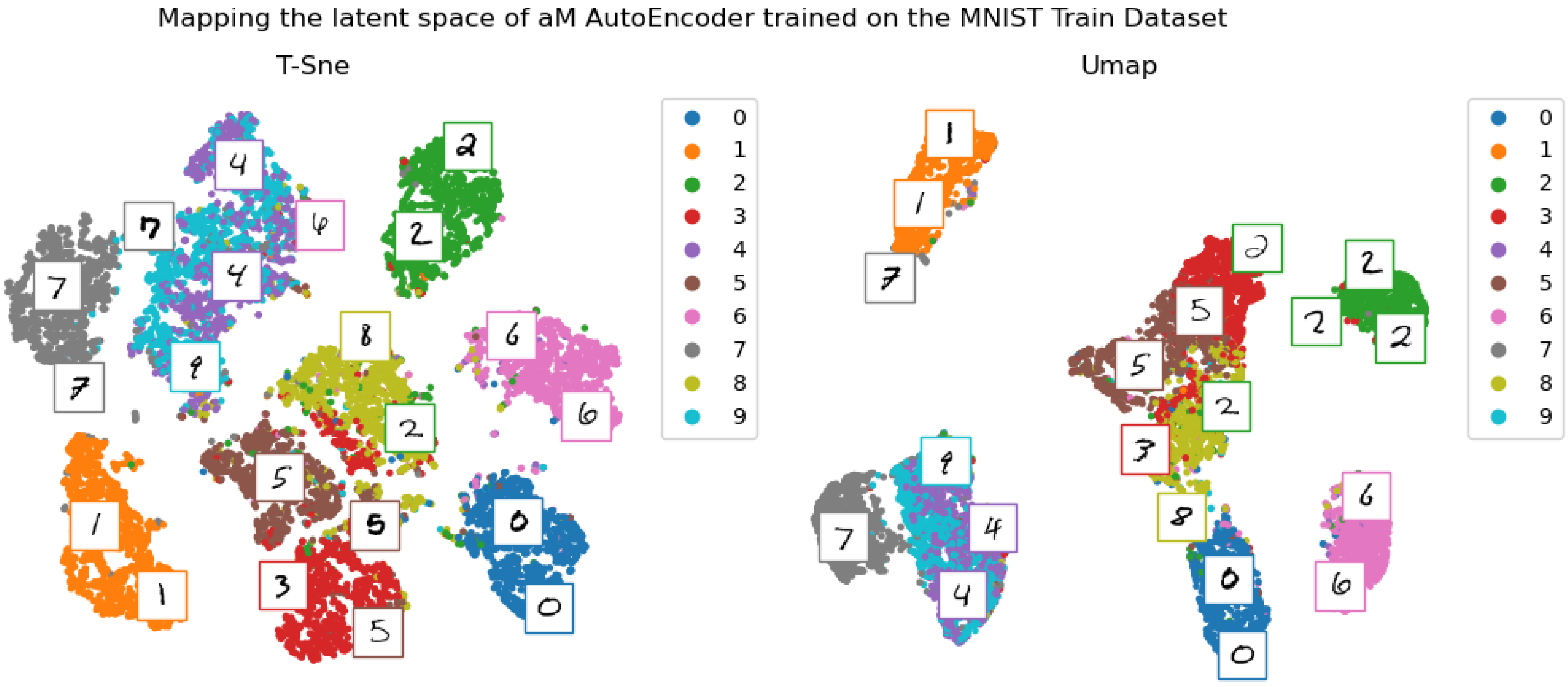
Analysis

Our results

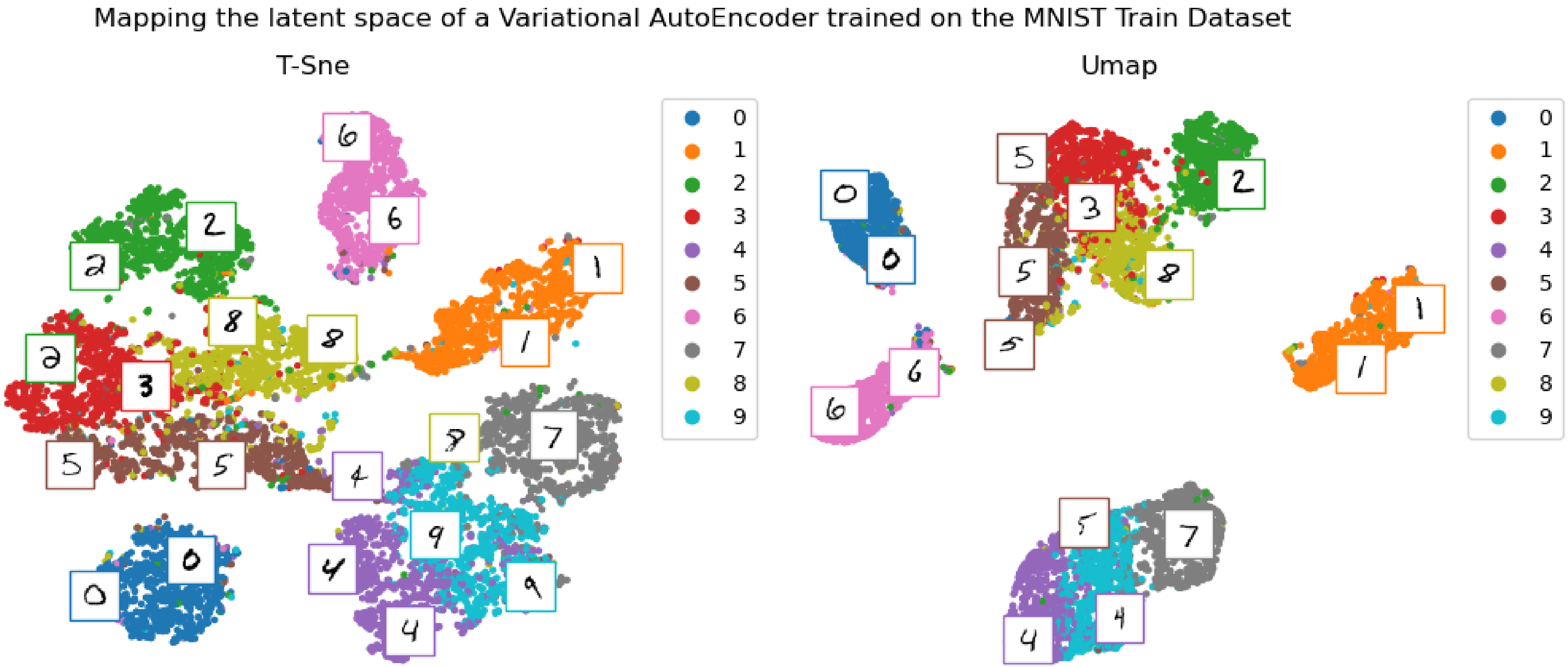
Classifier – MNIST



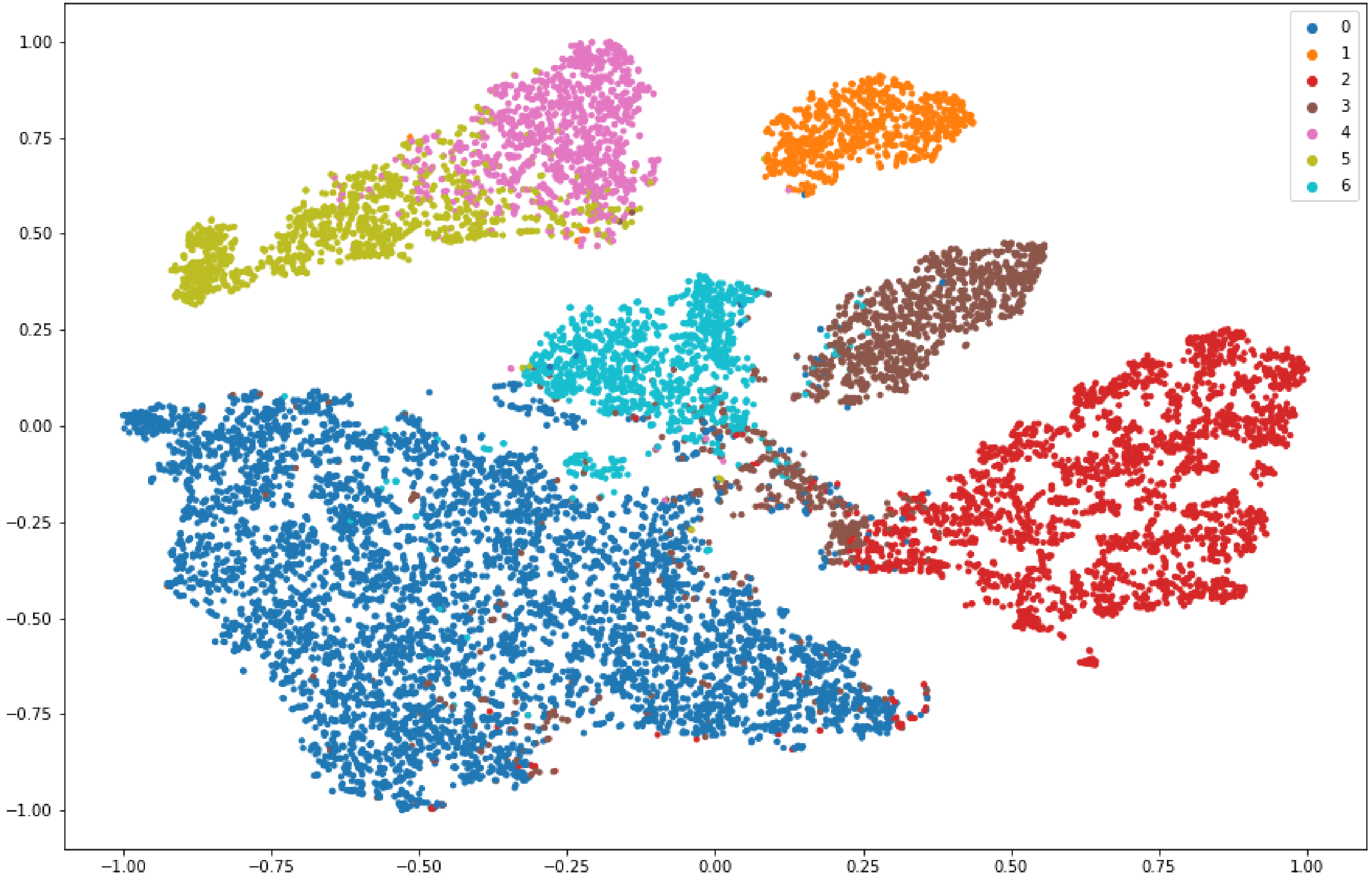
AutoEncoder – MNIST



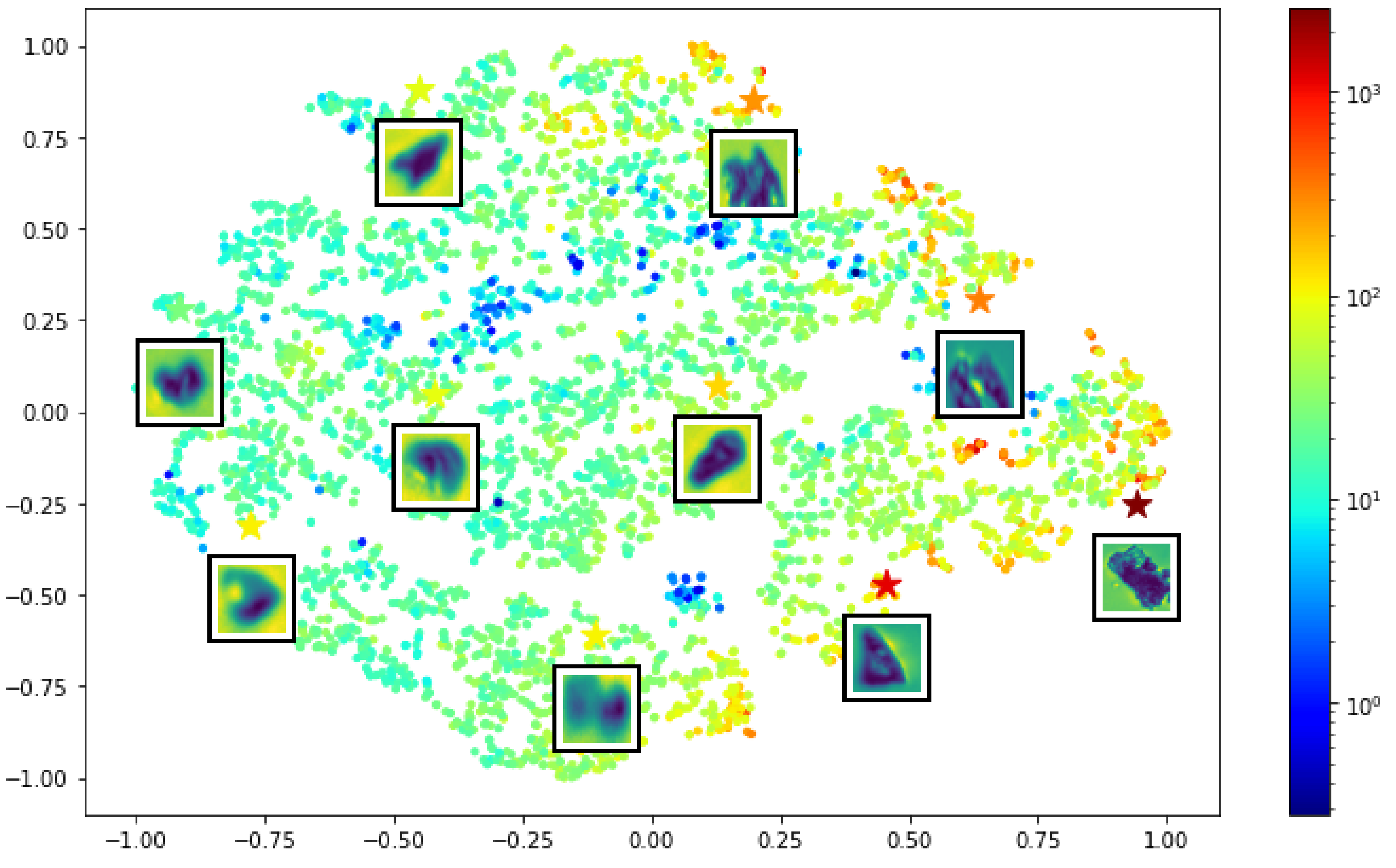
Variational AutoEncoder – MNIST



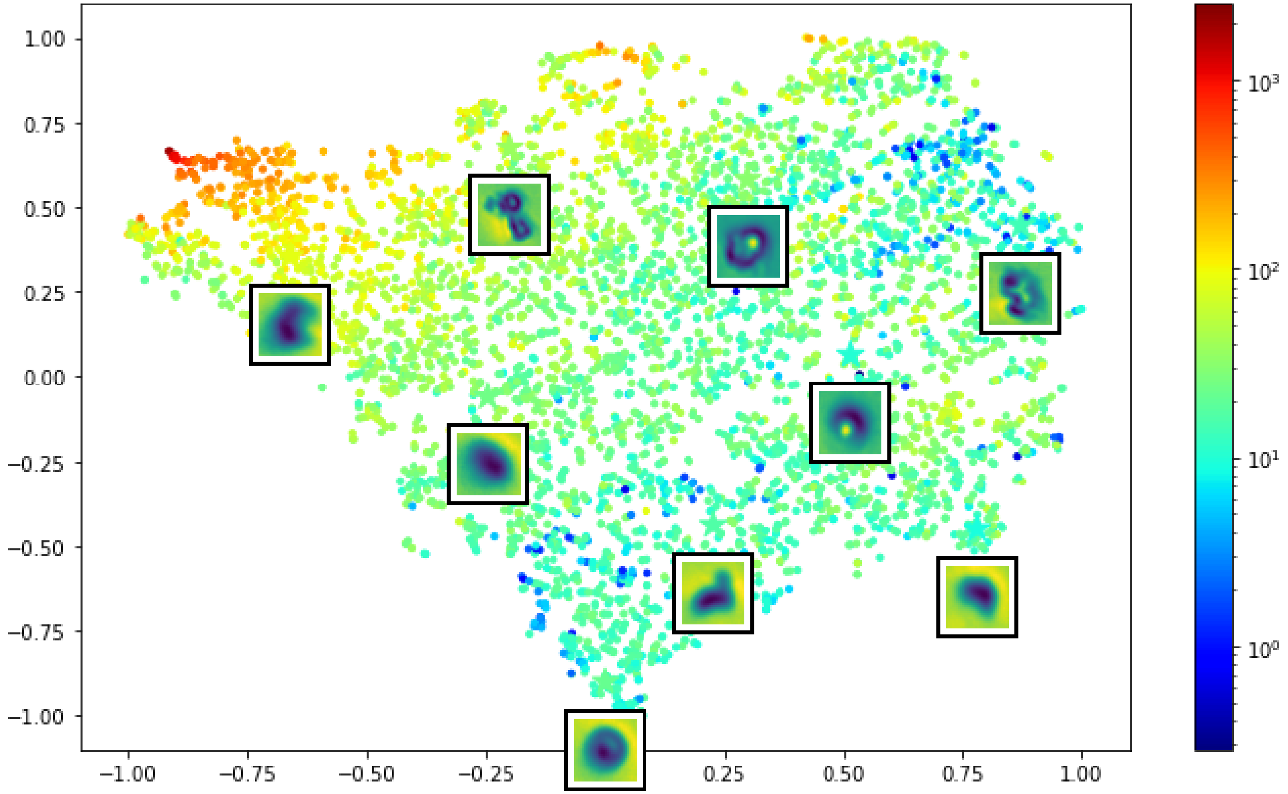
CNN-Classifier on Artificial Dataset



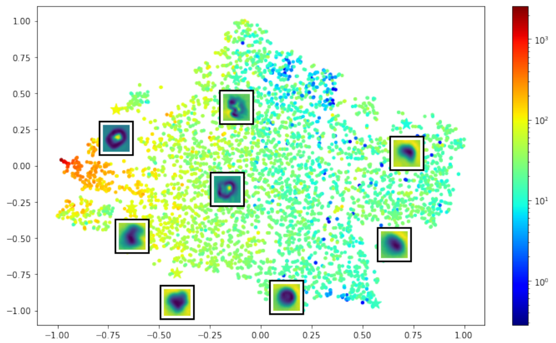
CNN-Classifier on Peruvian Dataset



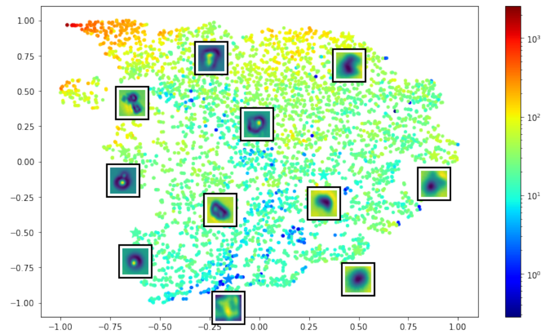
AutoEncoders on Peruvian Data



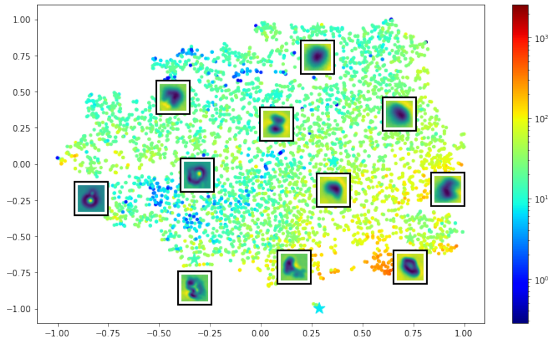
AutoEncoder with mean correction



AutoEncoder plain



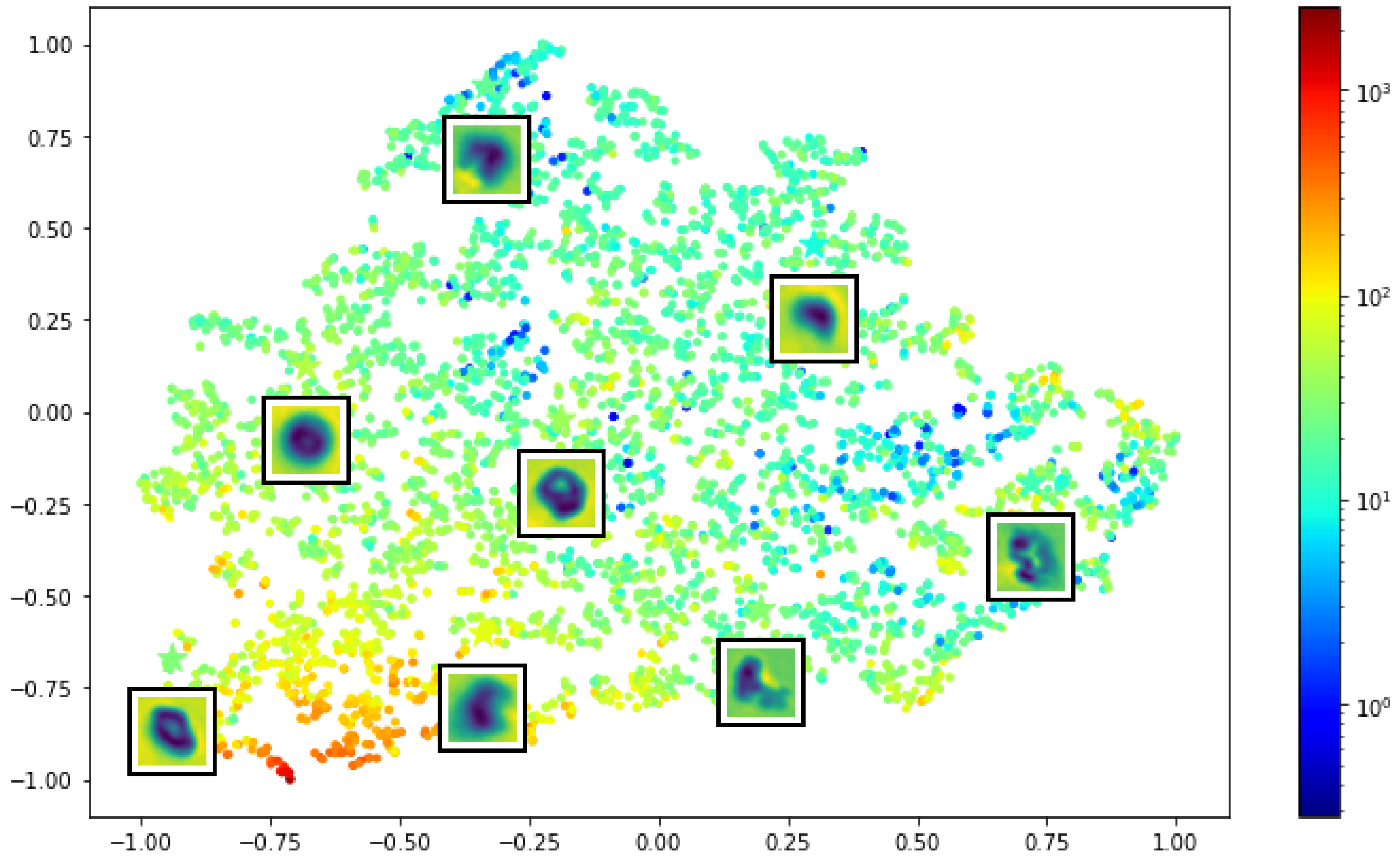
AutoEncoder with mean correction and no Dust



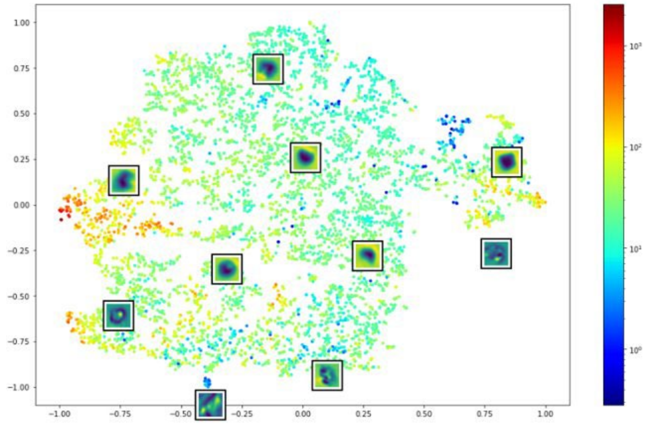
AutoEncoder plain latent space 128

Decoder: [VAE-ResNet18-PyTorch/model.py at master · julianstastny/VAE-ResNet18-PyTorch (github.com)]

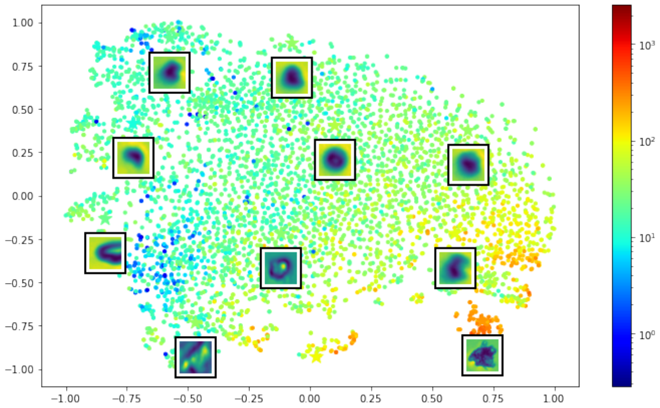
Variational AutoEncoders on Peruvian Dataset



Variational AutoEncoder with mean correction



Planar flow model trained without artificial data



Planar flow trained with artificial data

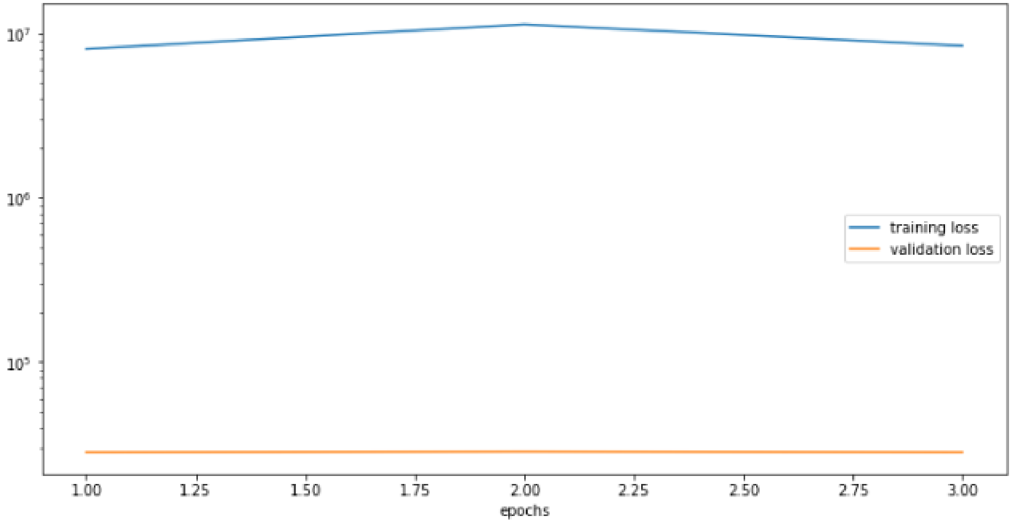
planar-flow-pytorch/vae-pf.py at master · abdulfatir/planar-flow-pytorch (github.com)

Difficulties during training

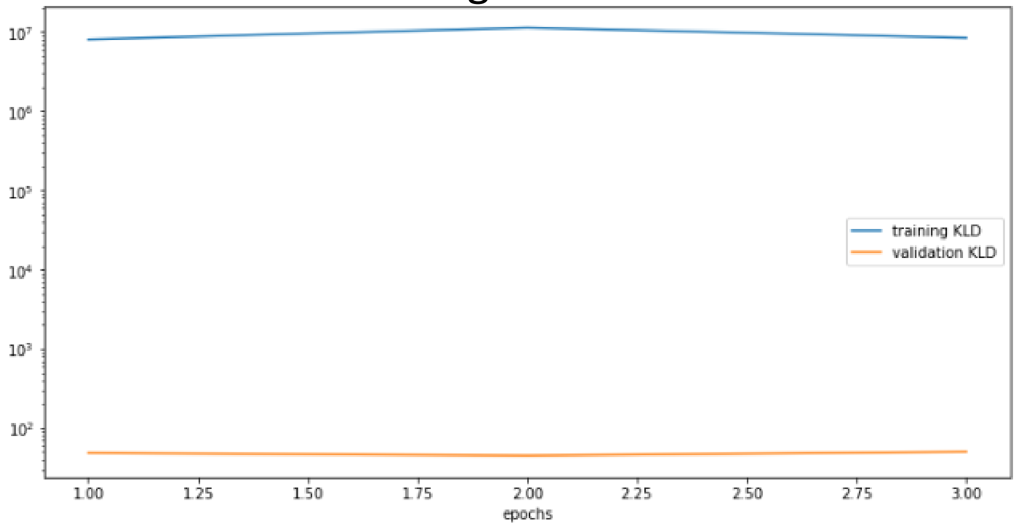
– Overfitting on initial parameters?

- KL divergence for some events in the Peru set explodes before training has even started and never seem to drop. Most likely overfitting encoder from classifier on the artificial set. And yes! Problem solved by reducing epochs on pretrained encoder.

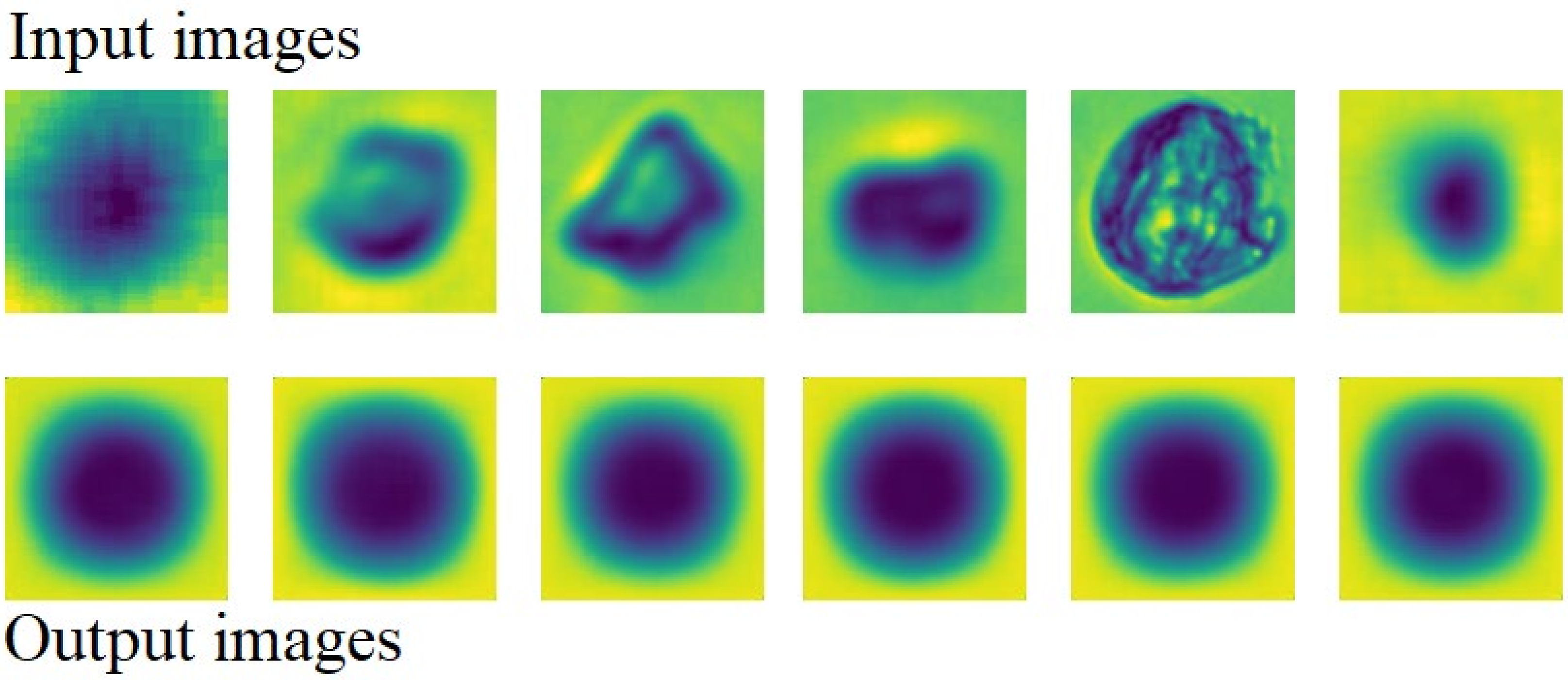
Total loss



Kullback Liebler divergence



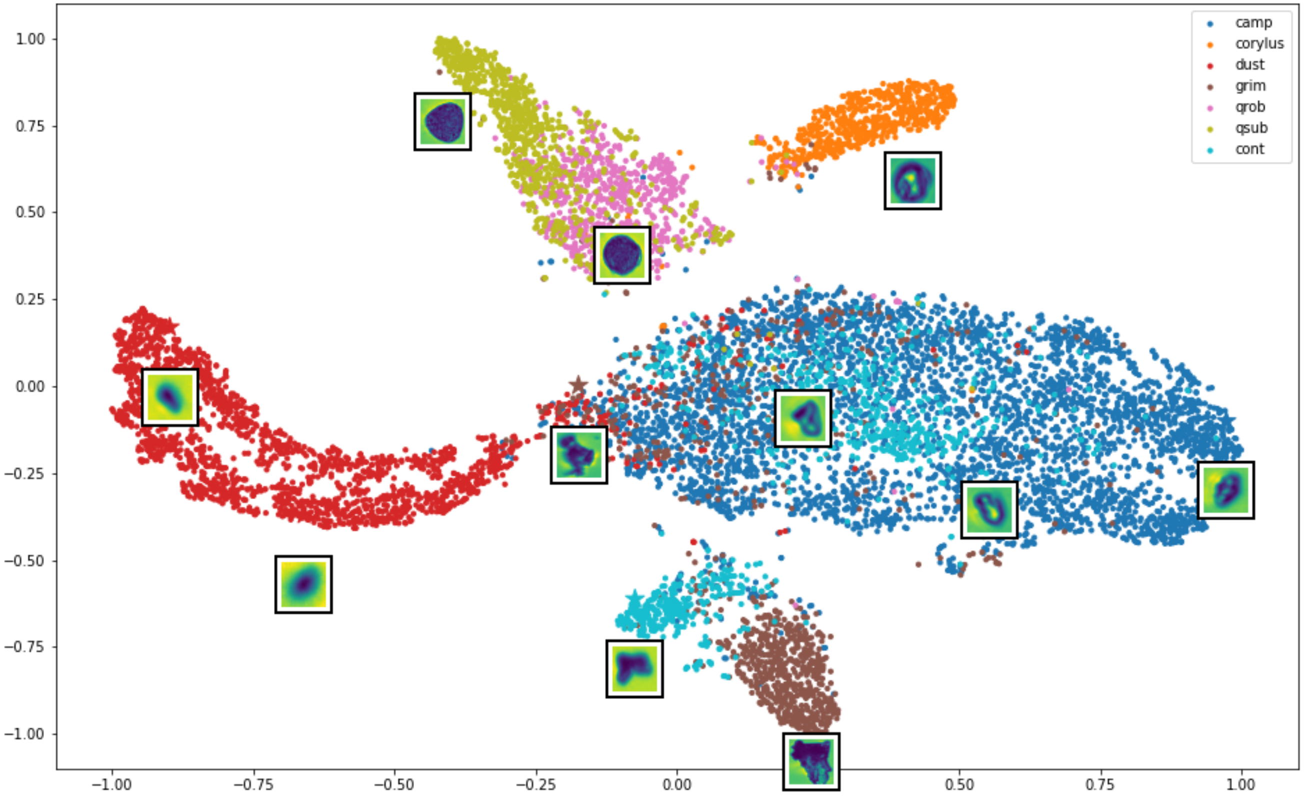
Difficulties during training – mode collapse?



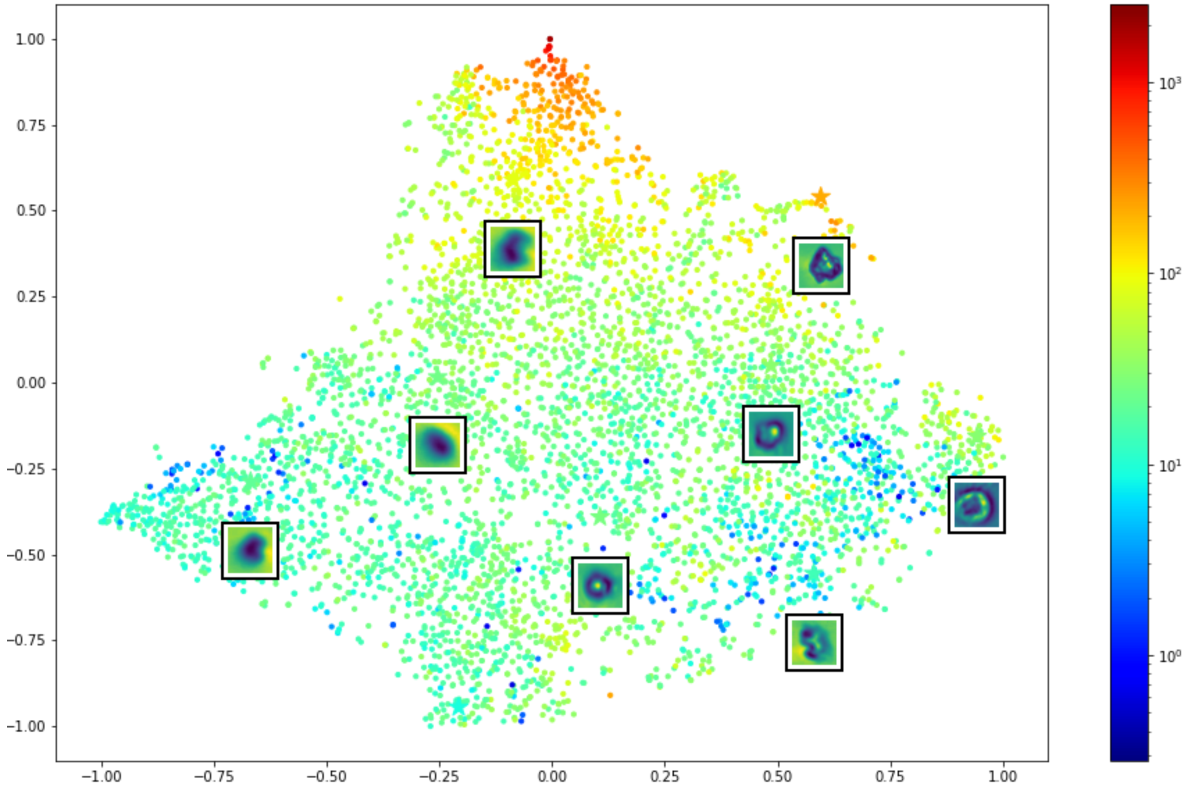
KLD was about 2 vs about 40 in "good" trainings indicating information depleted latent space.

Difficulties with overfitting

Artificial dataset

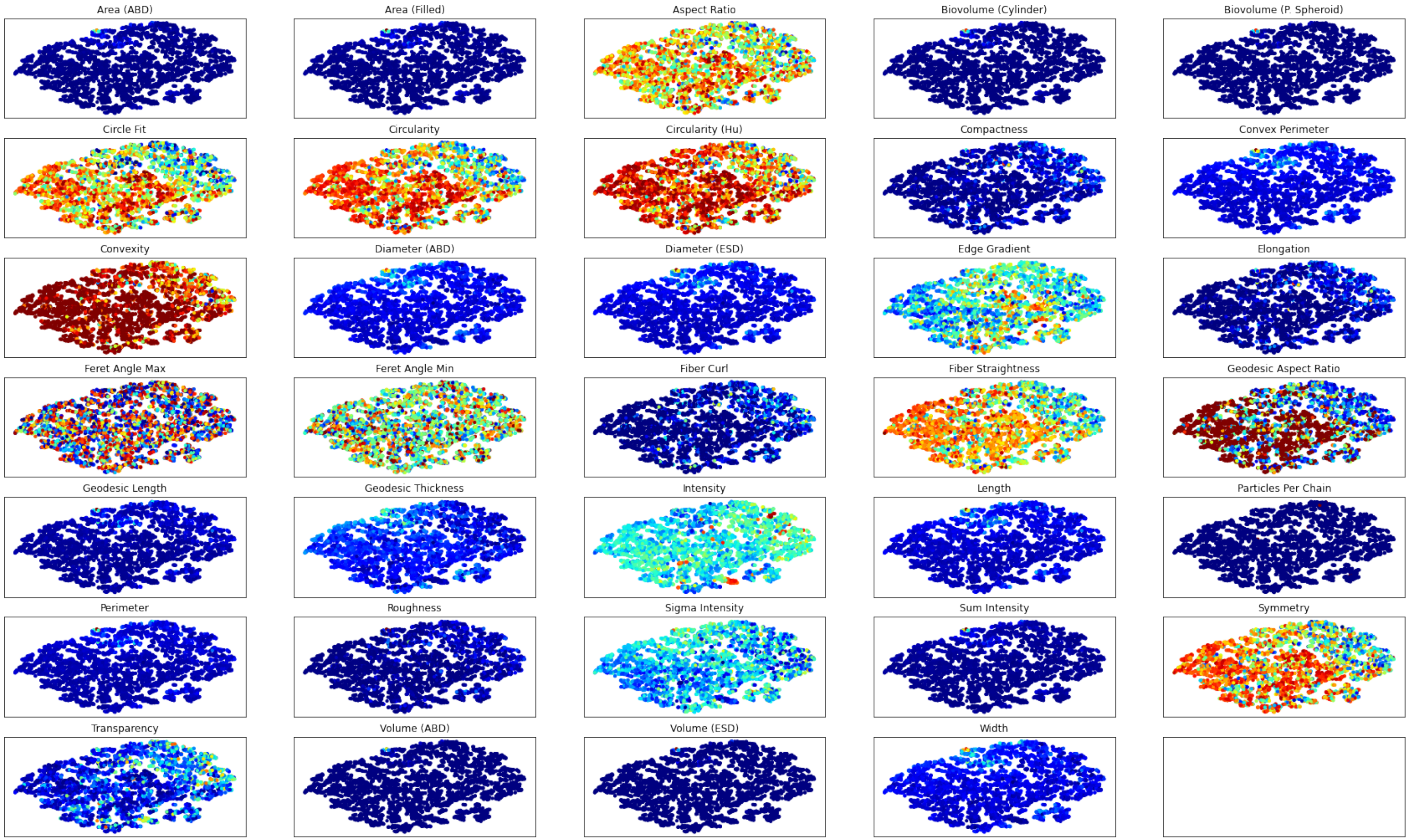


Peruvian set

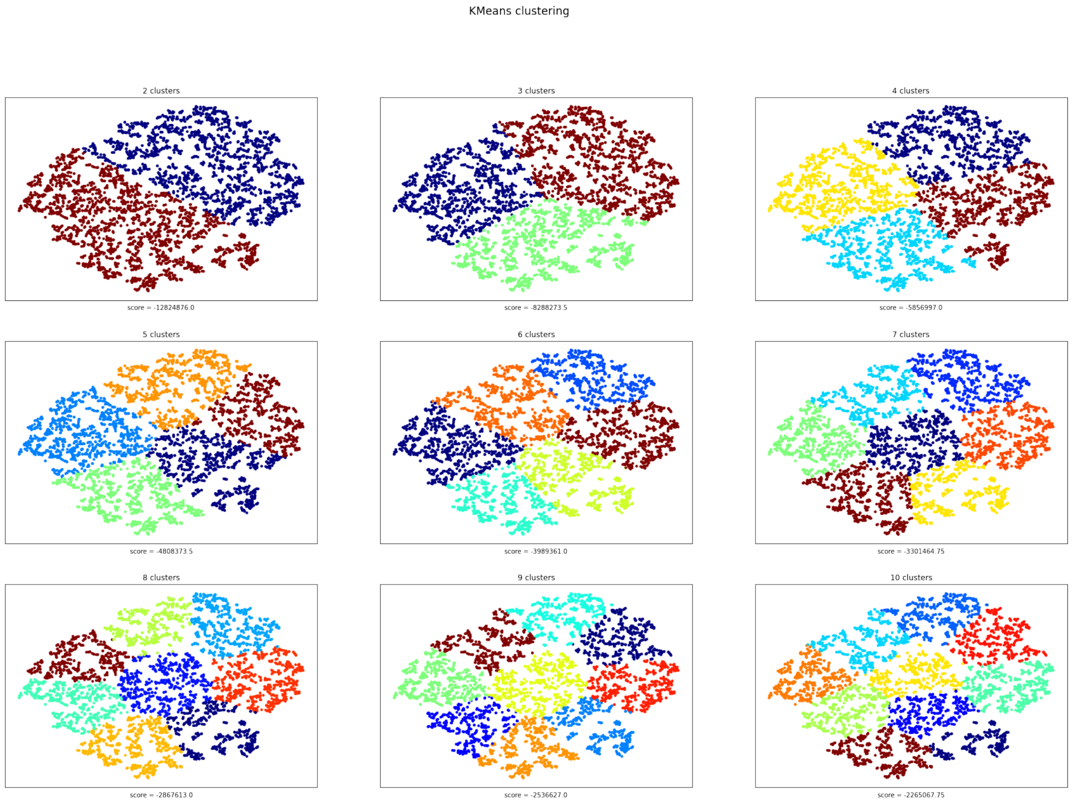


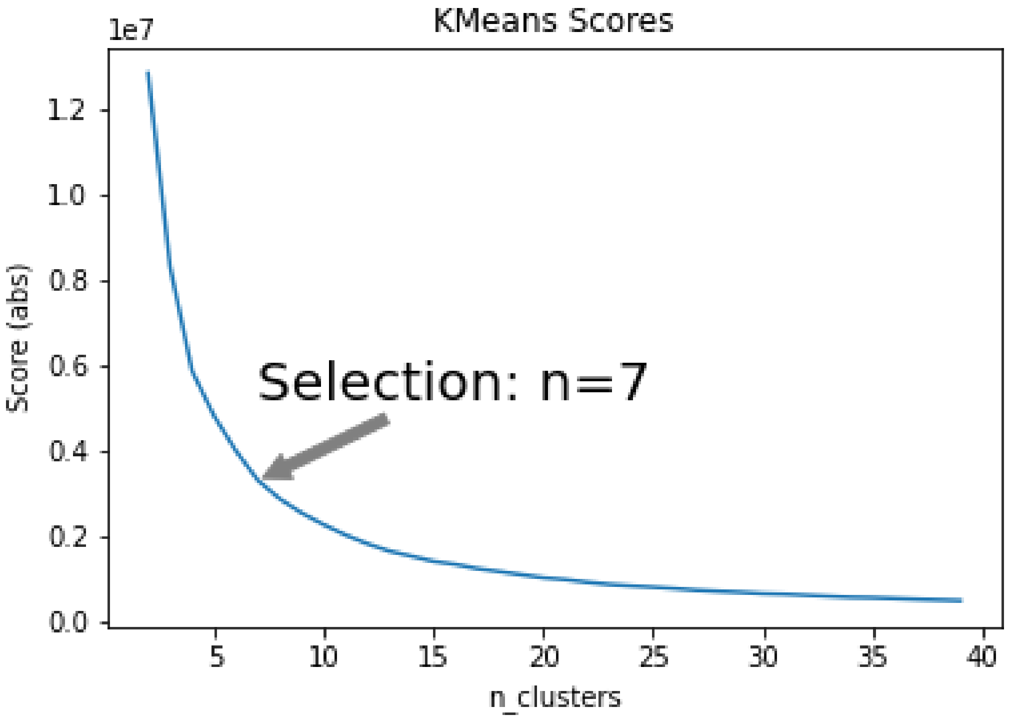
Fitted too much on labeled data?

Plotting scalars

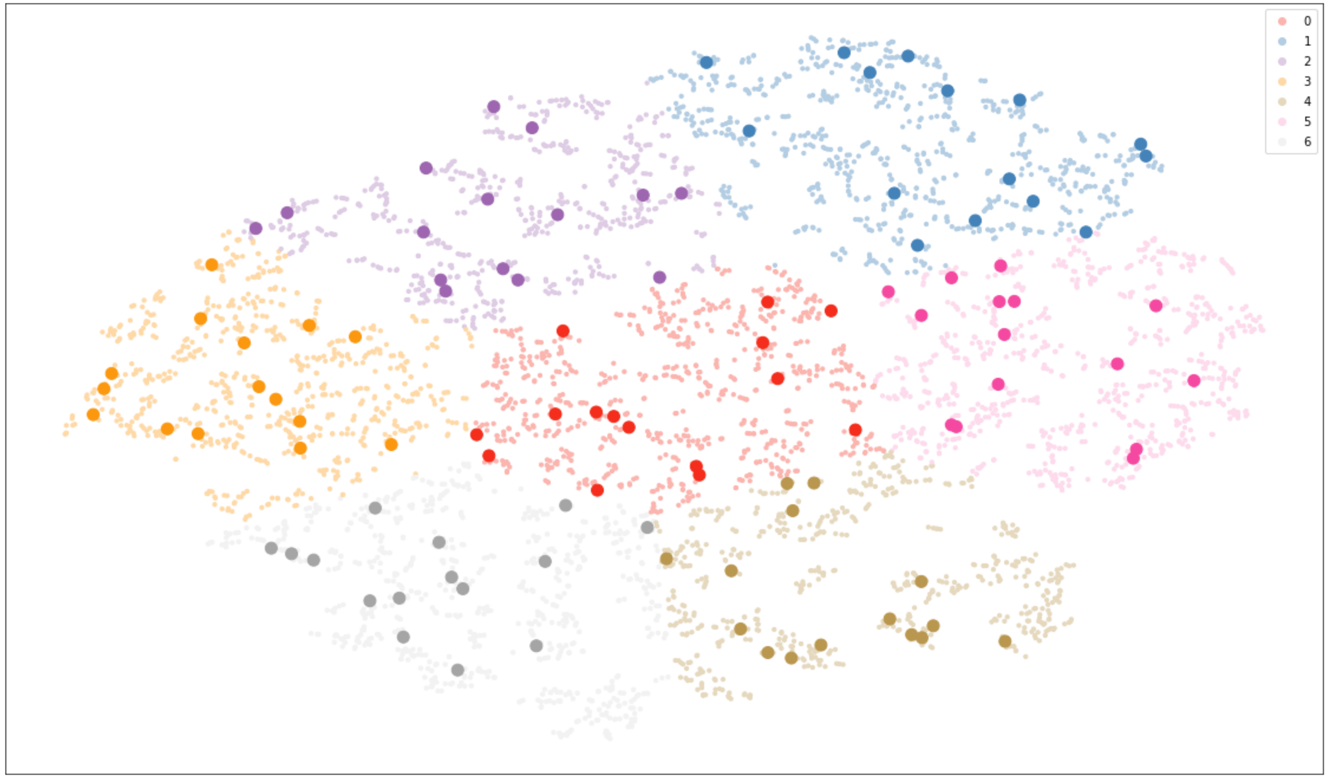


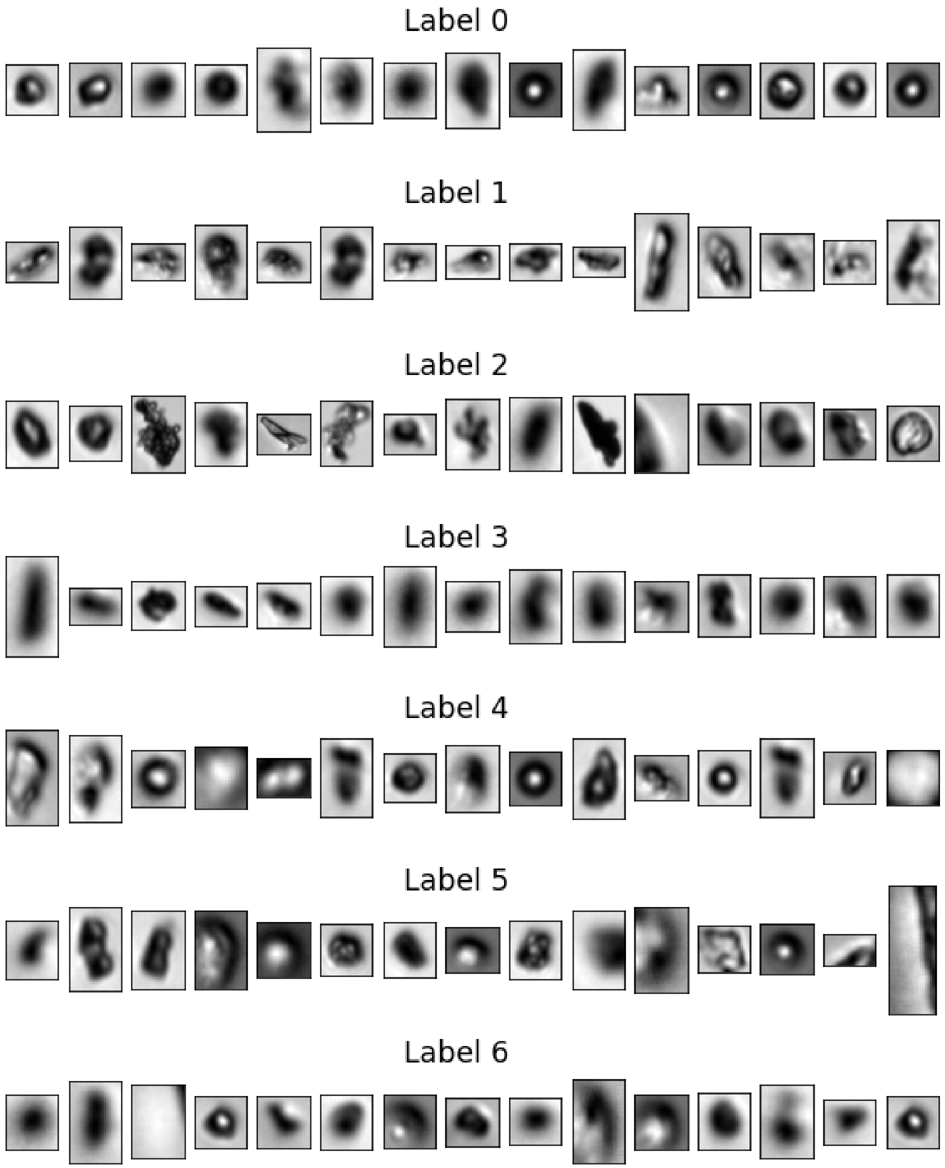
KMeans Clustering



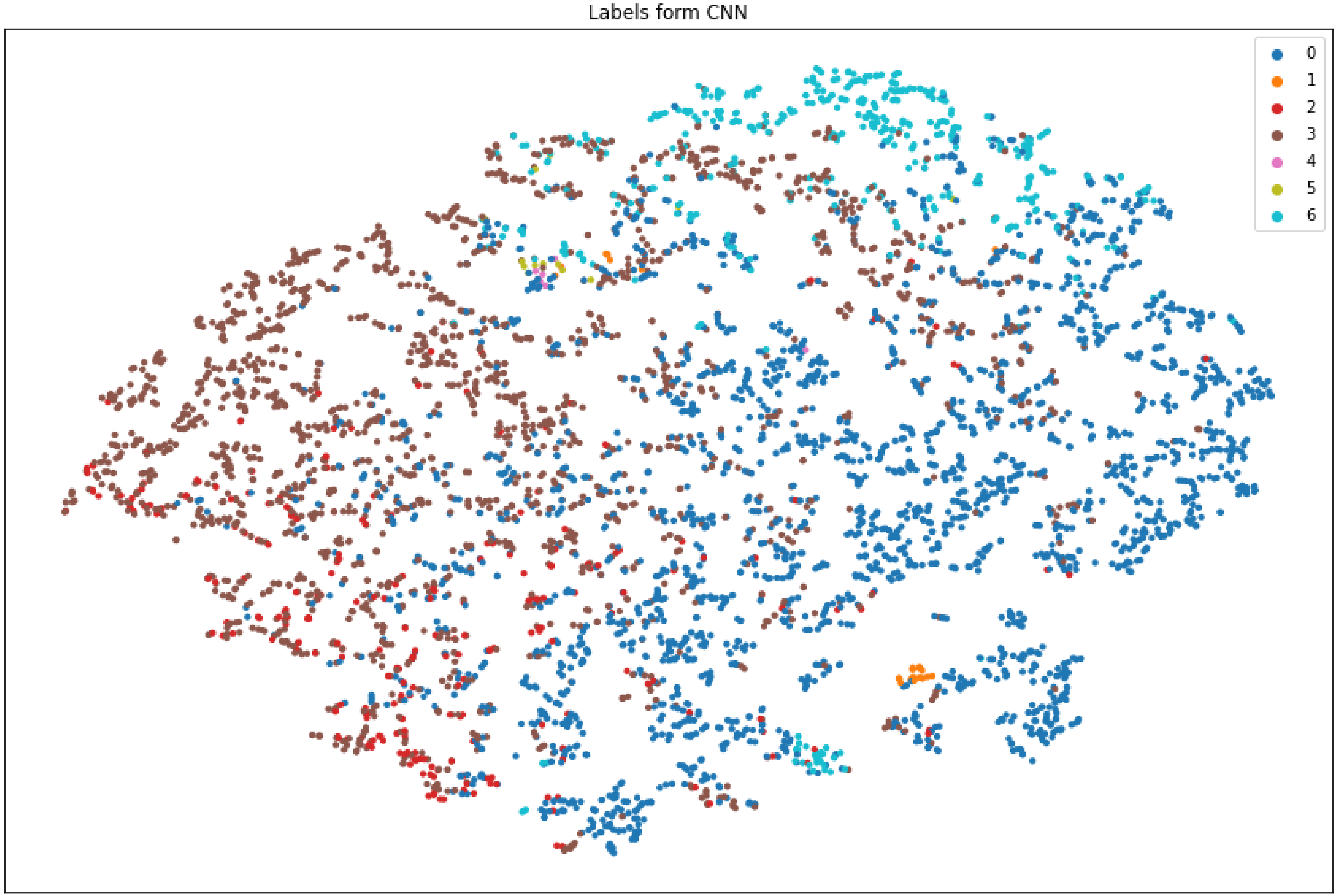


KMeans for n=7



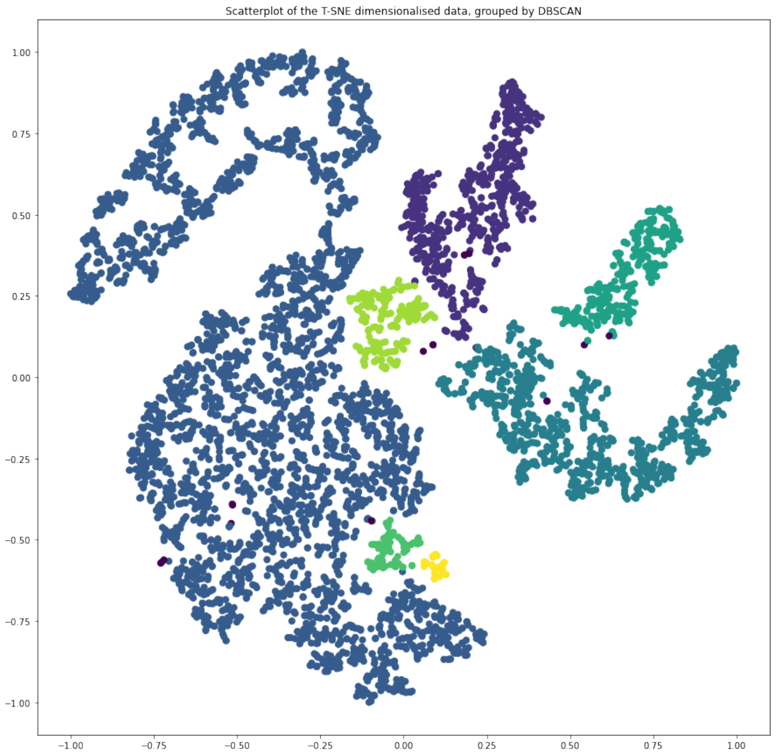


Overlaying labels from CNN-Classifier

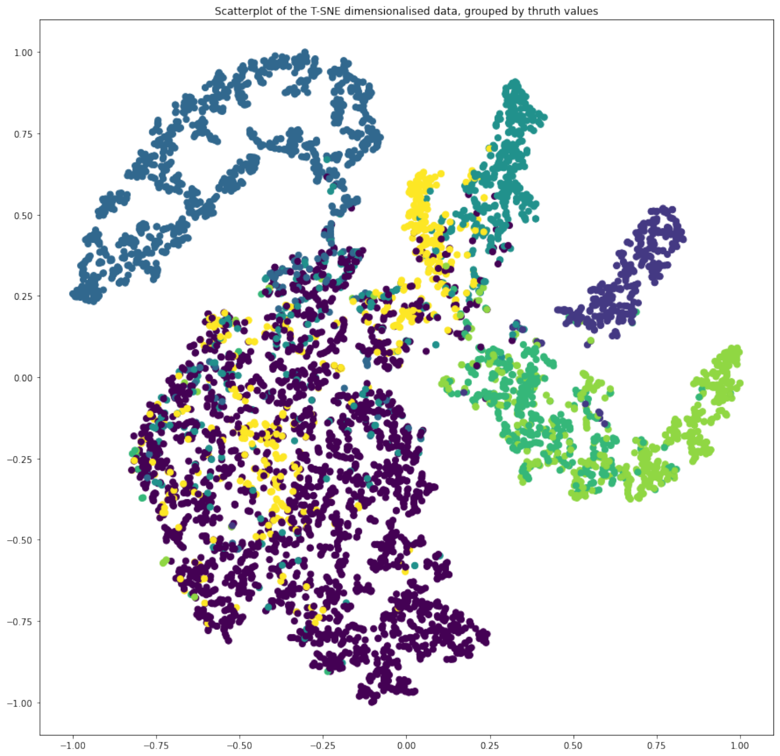


DBSCAN

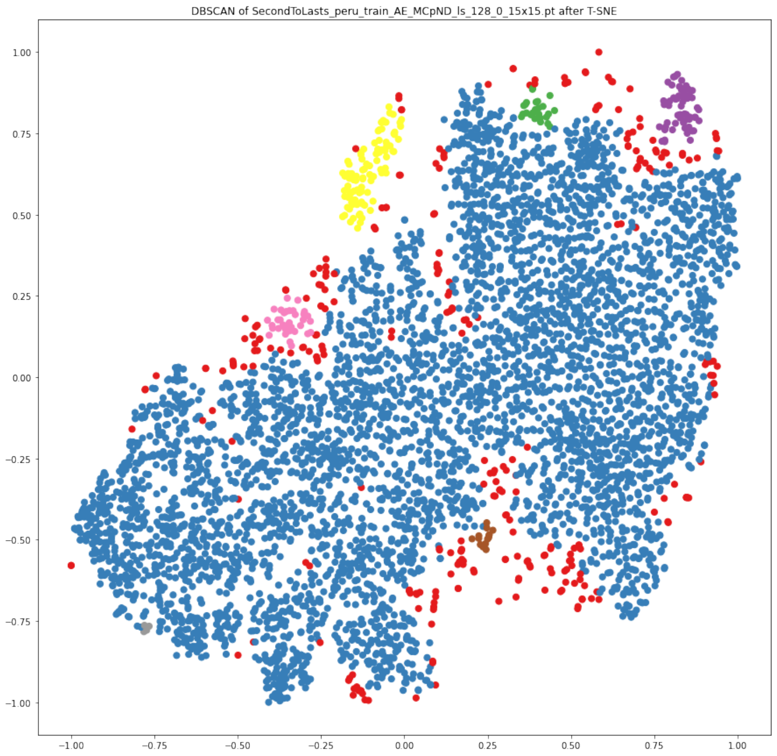
- DBSCAN on the AE/VAE tensors allows for partial clustering



Artificial Dataset (Prediction)

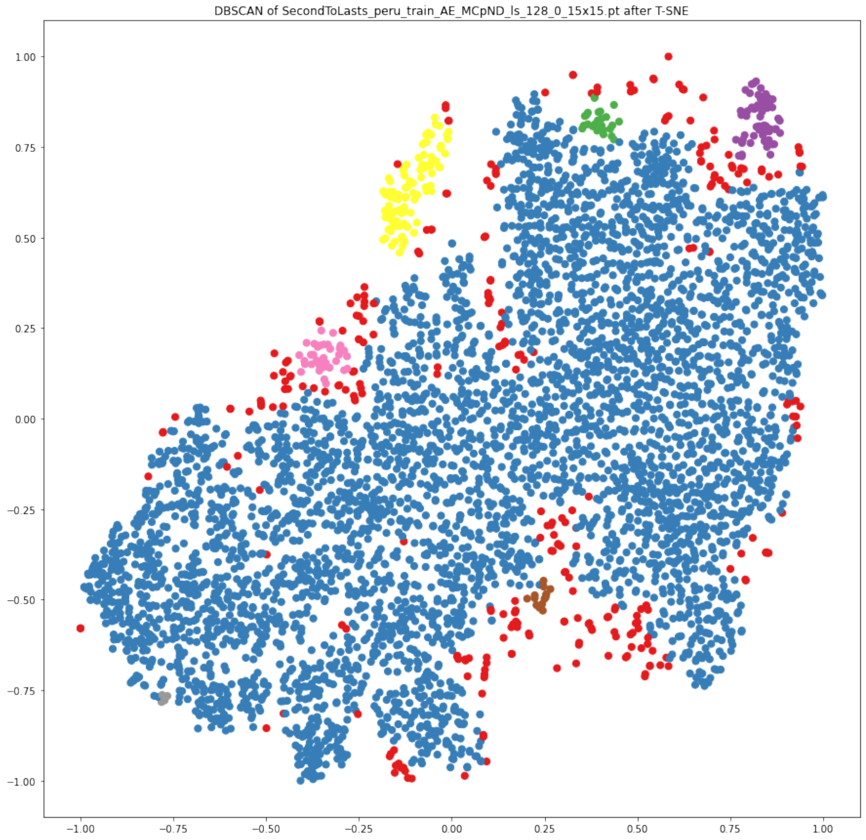


Artificial Dataset (Truth)

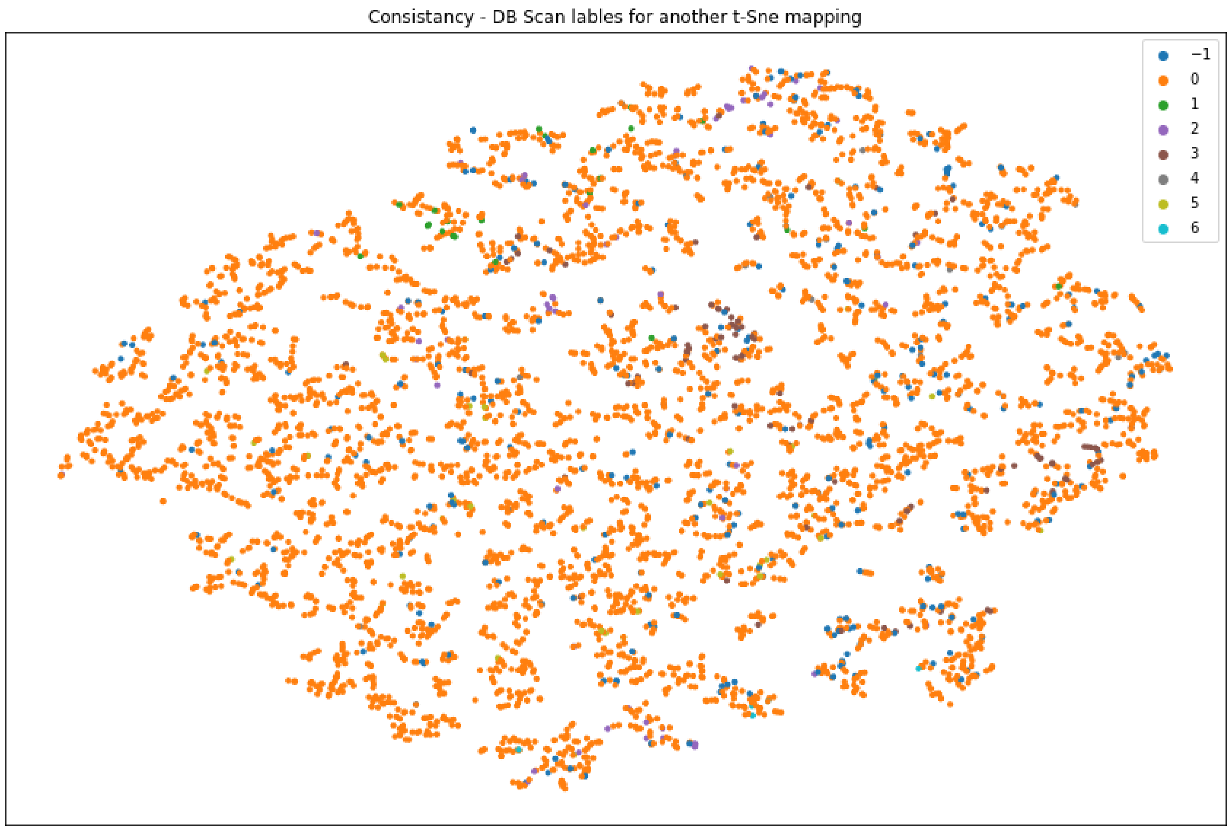


Peruvian Dataset

DBSCAN

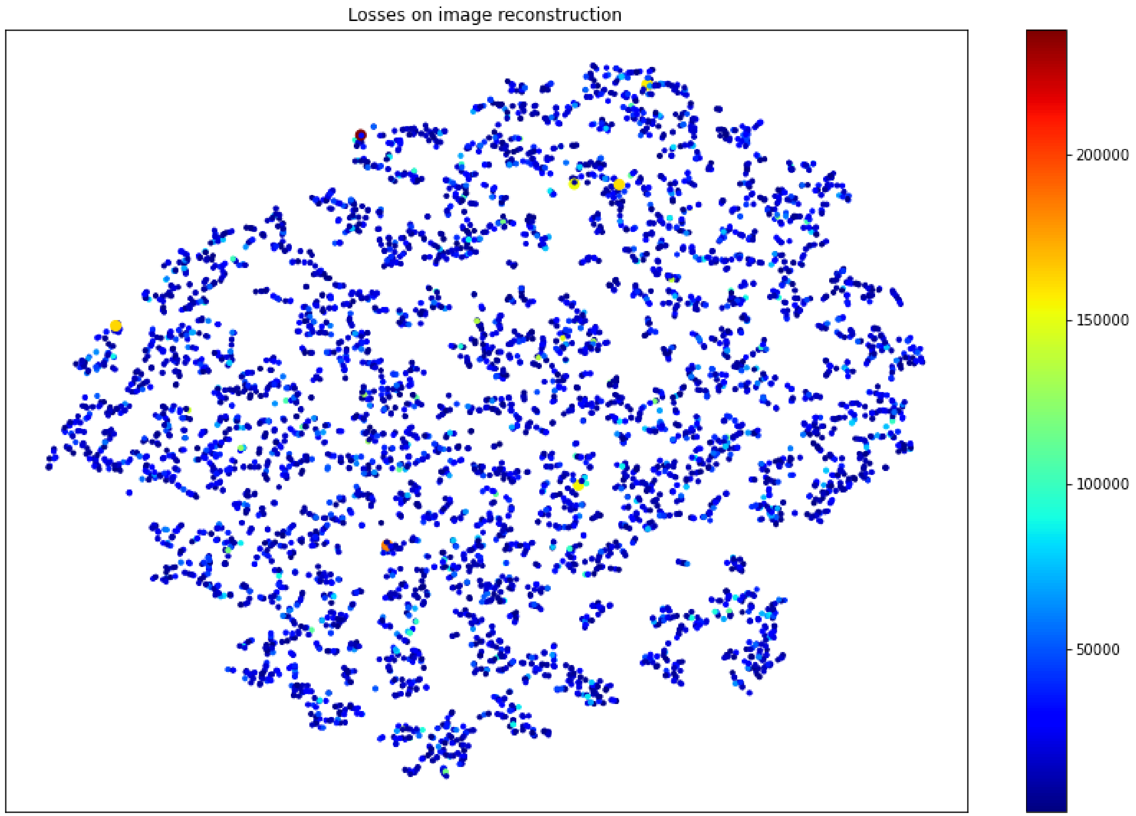


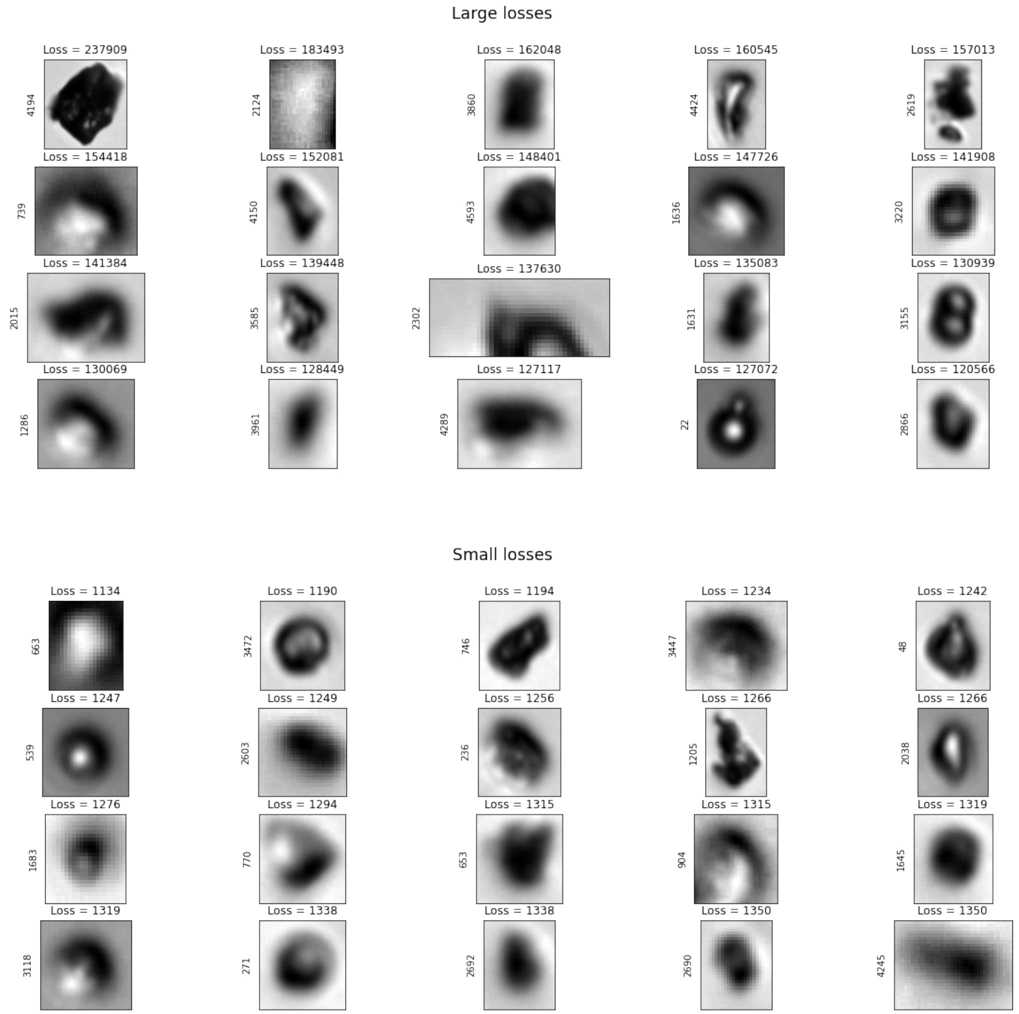
DBSCAN on the AE/VAE tensors allows for partial clustering



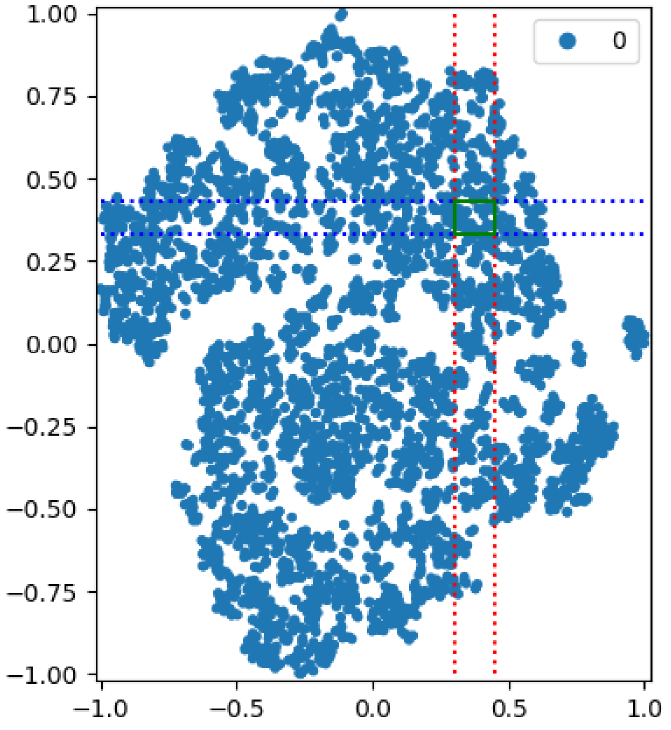
Consistency between mappings

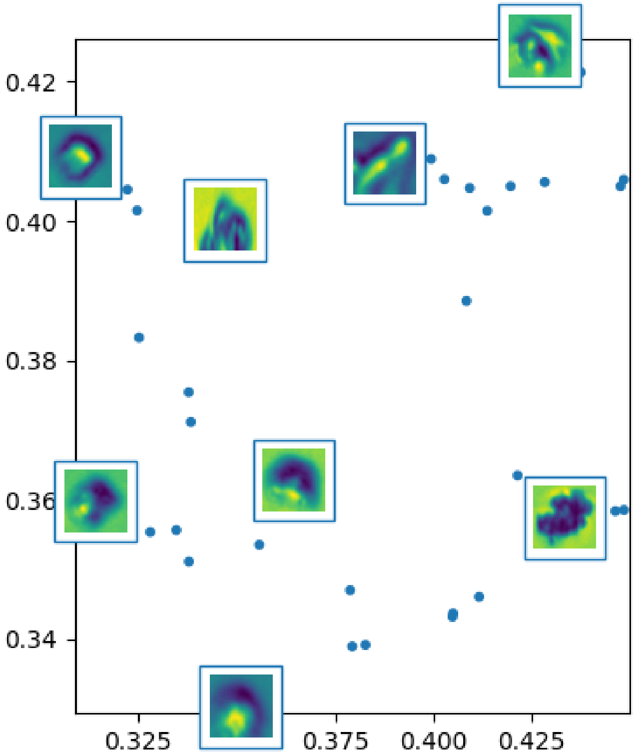
Plotting the loss

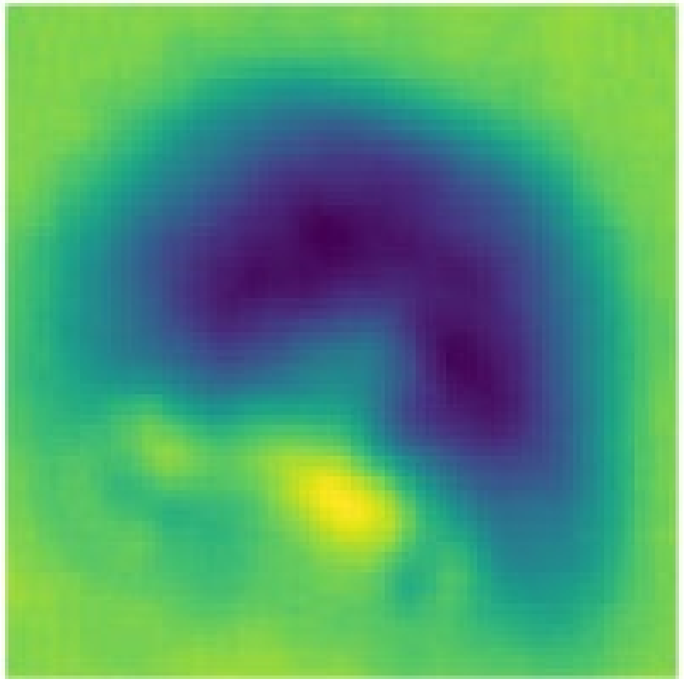




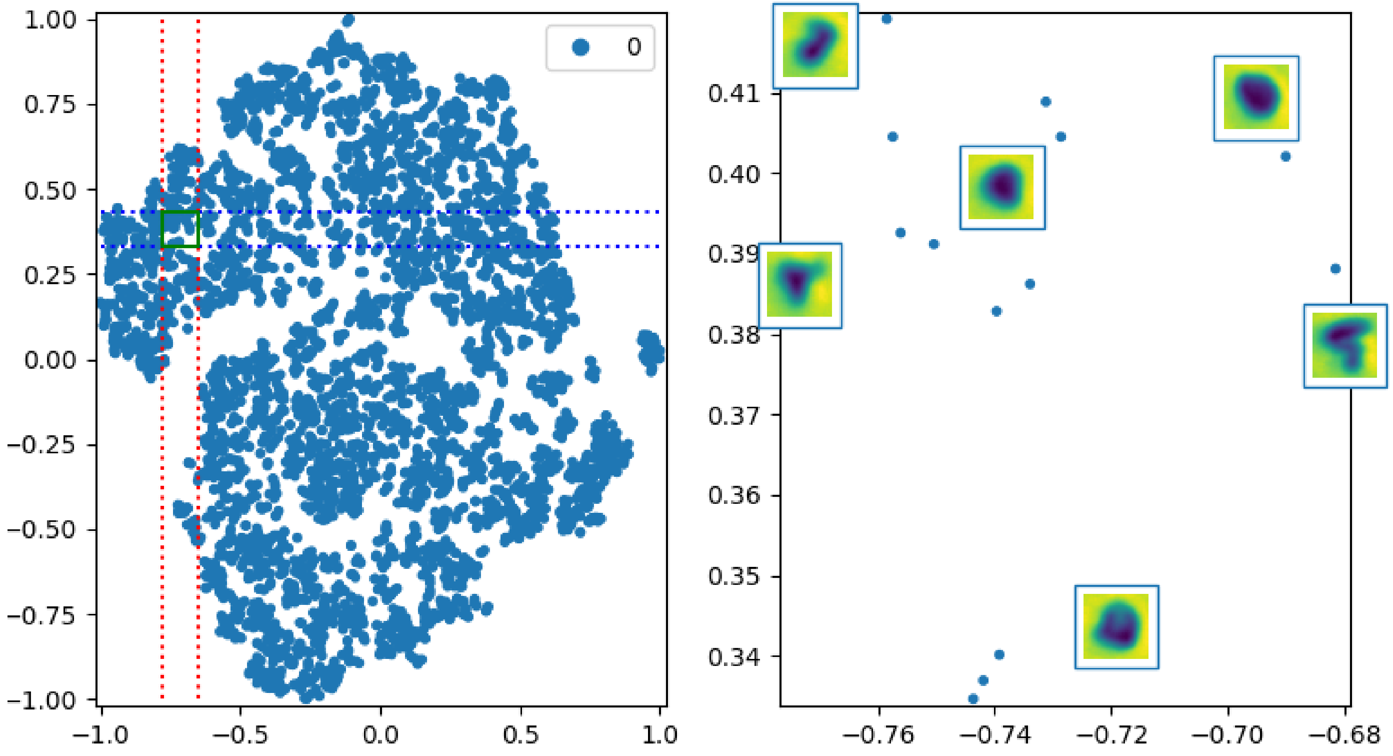
CNN classifier - Area 1

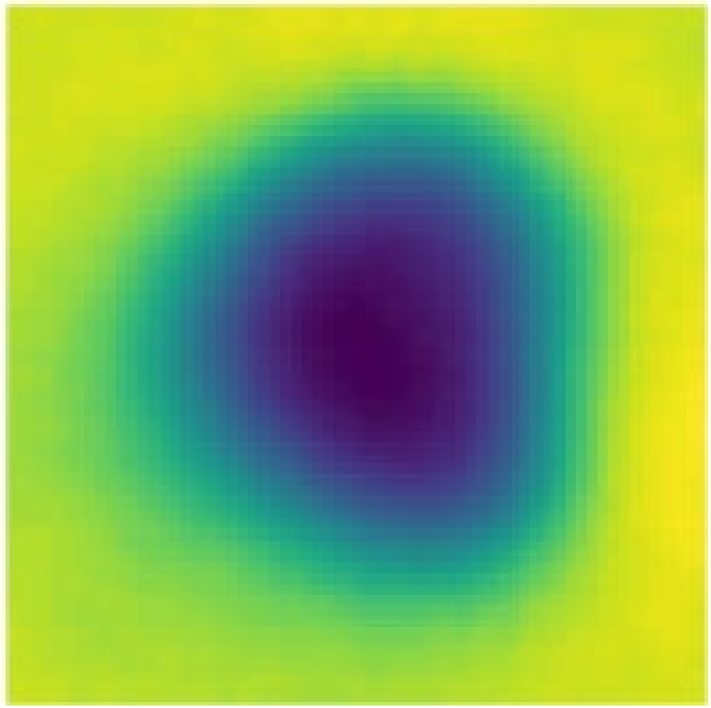




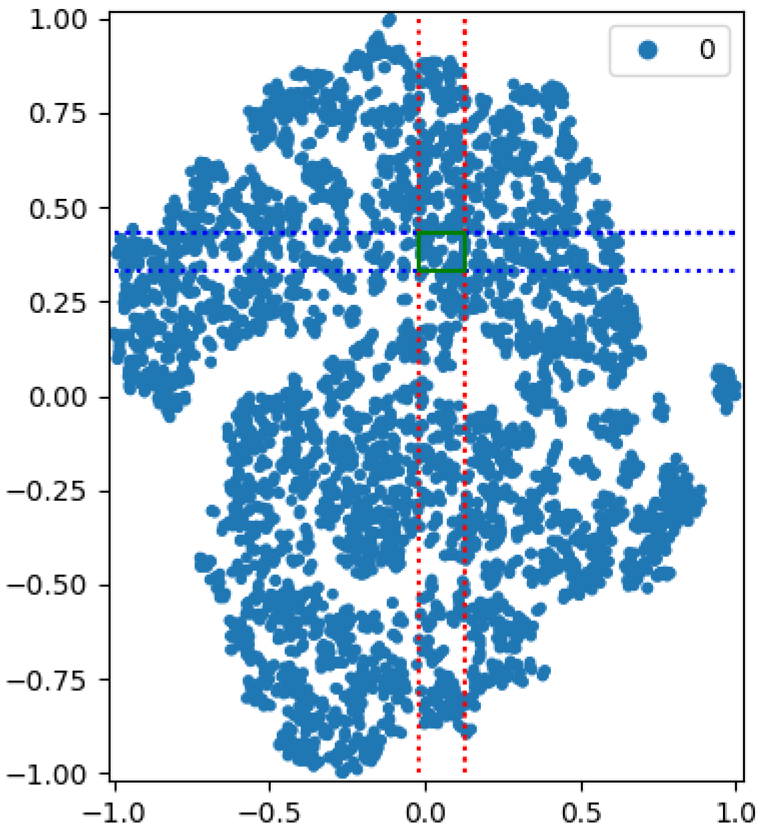


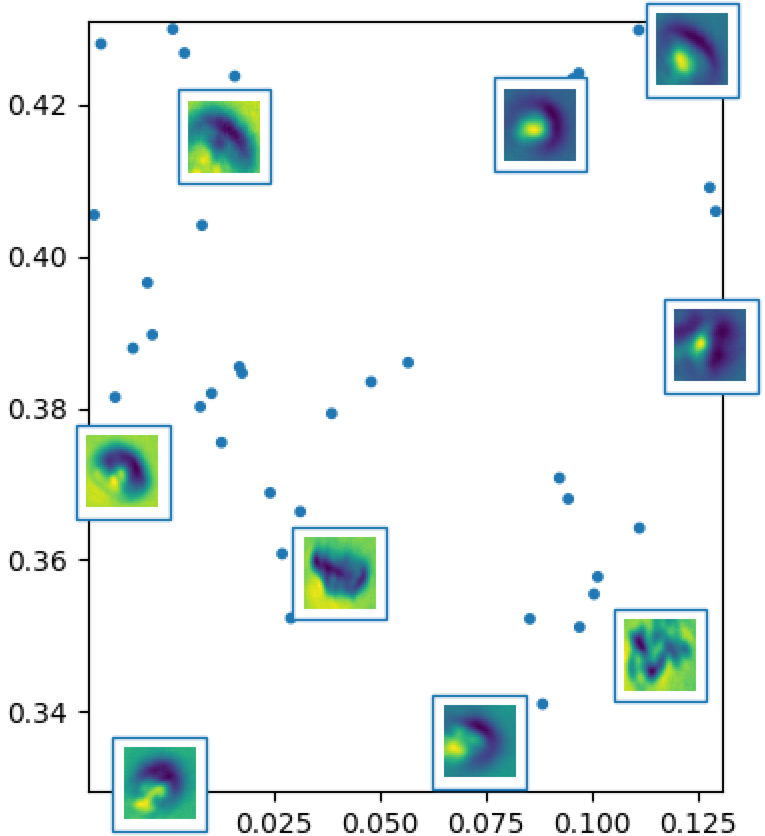
CNN classifier - Area 2



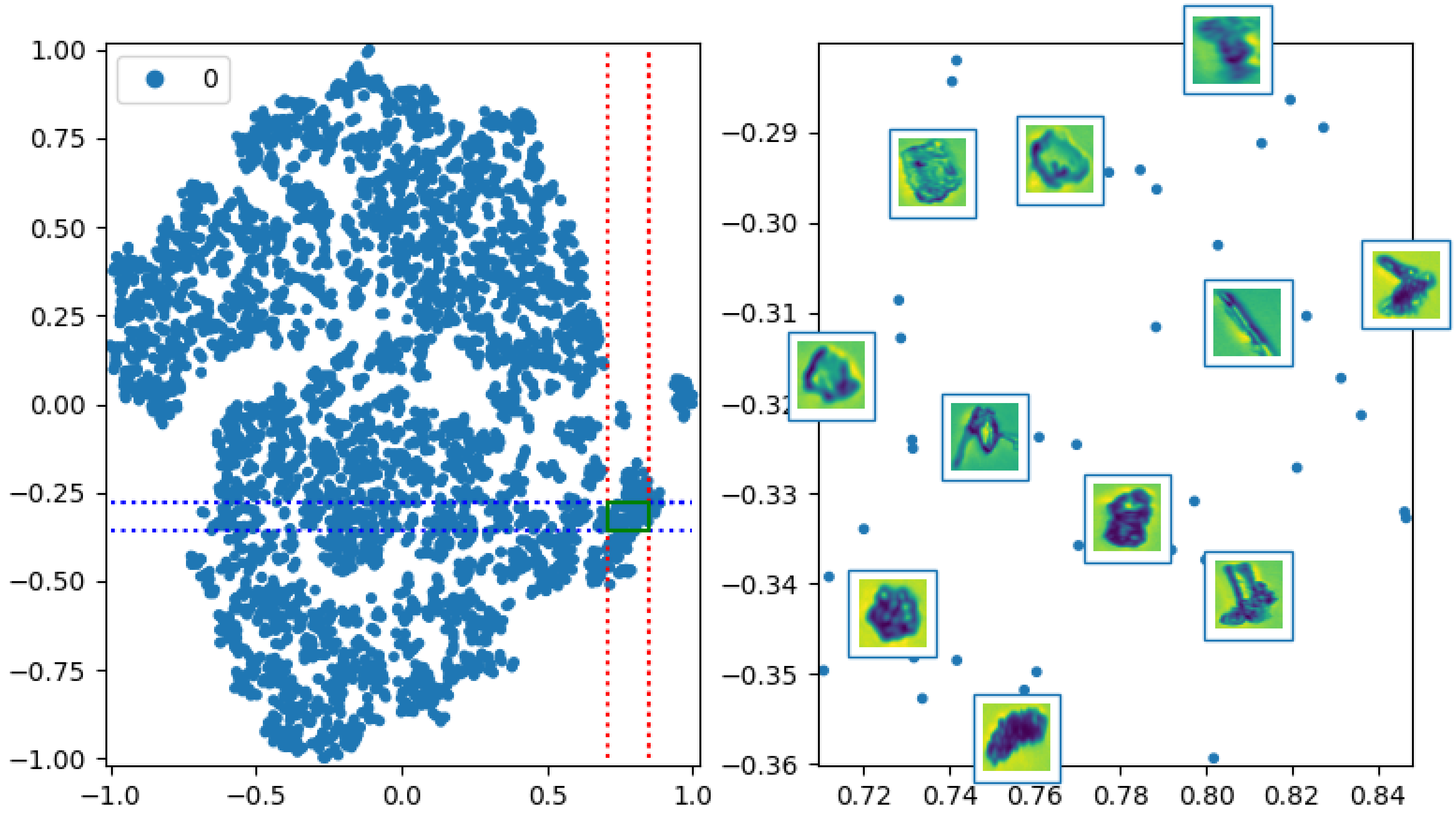


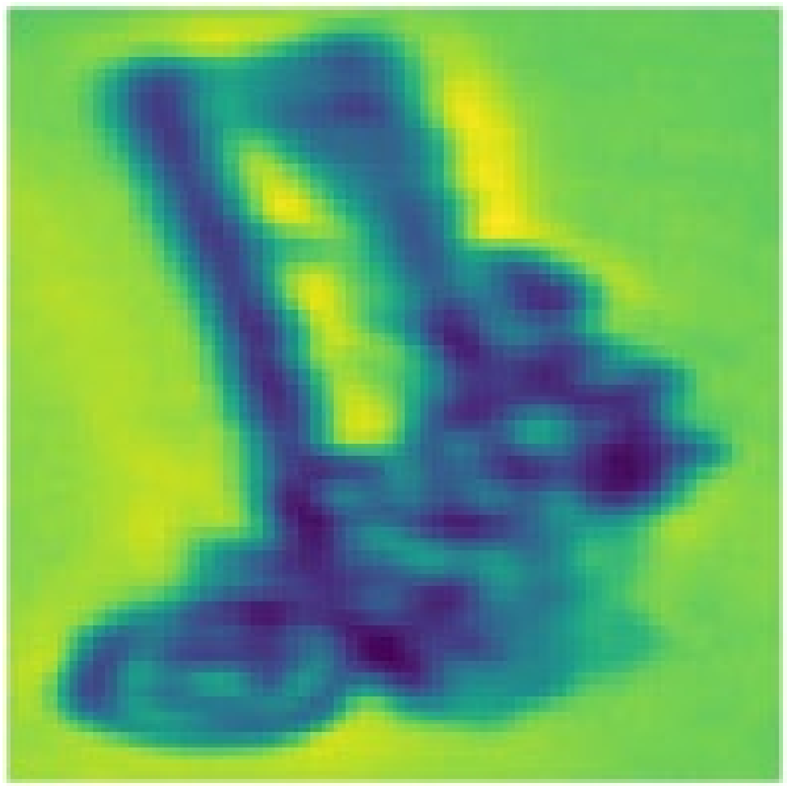
Between area 1 and 2



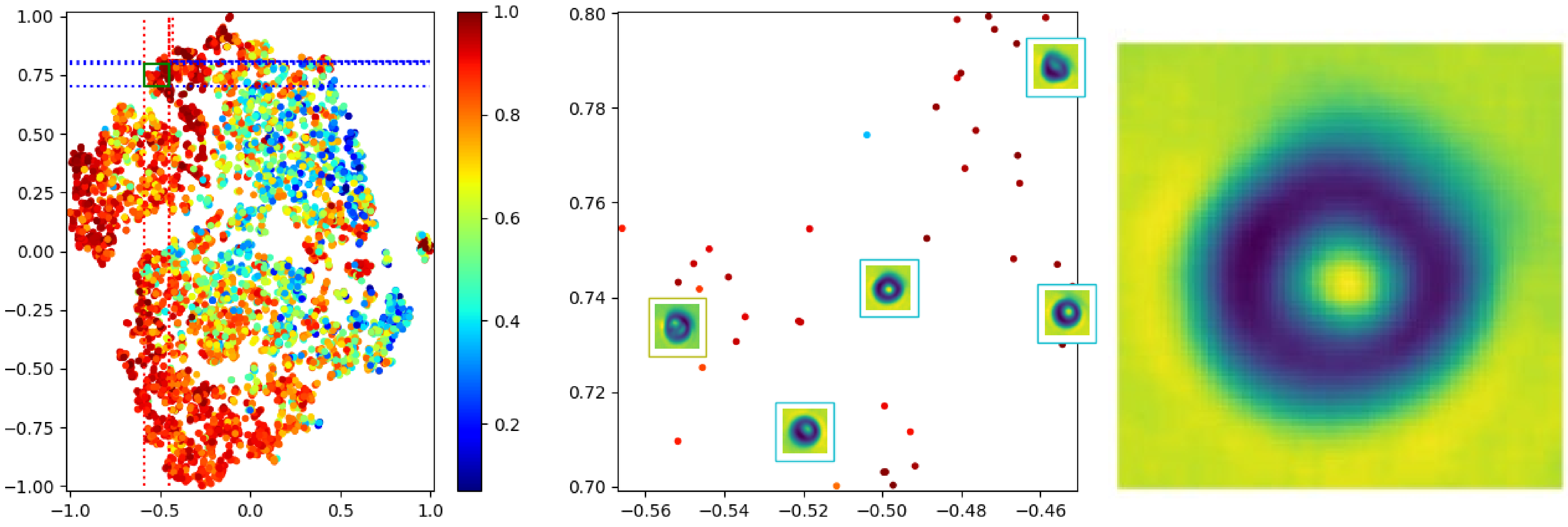


CNN classifier - Area 3

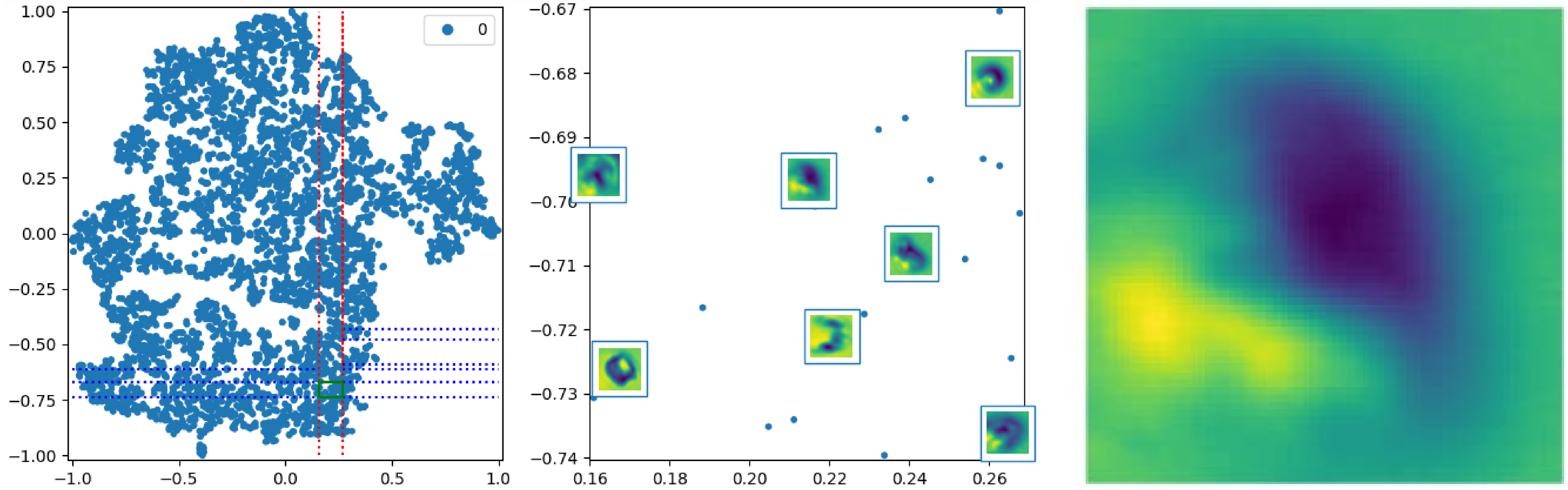




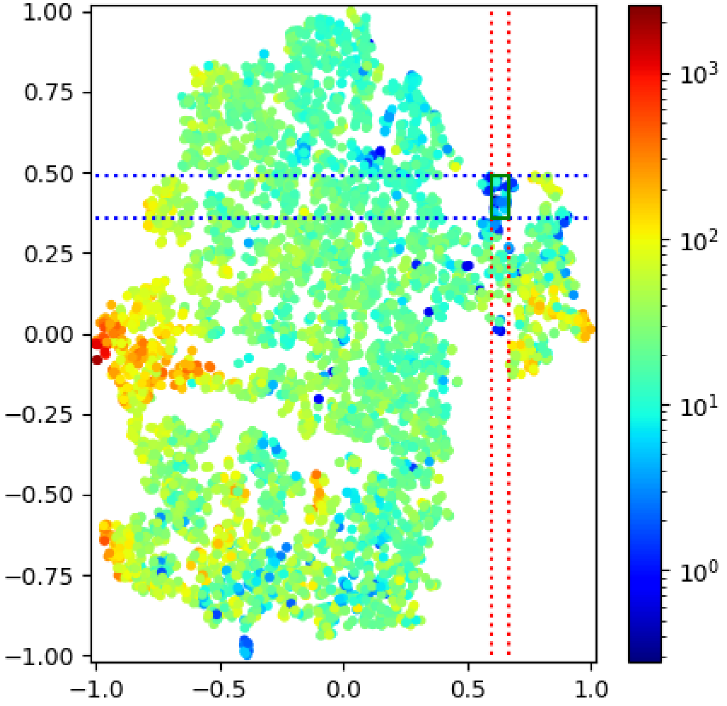
CNN classifier - Area 4

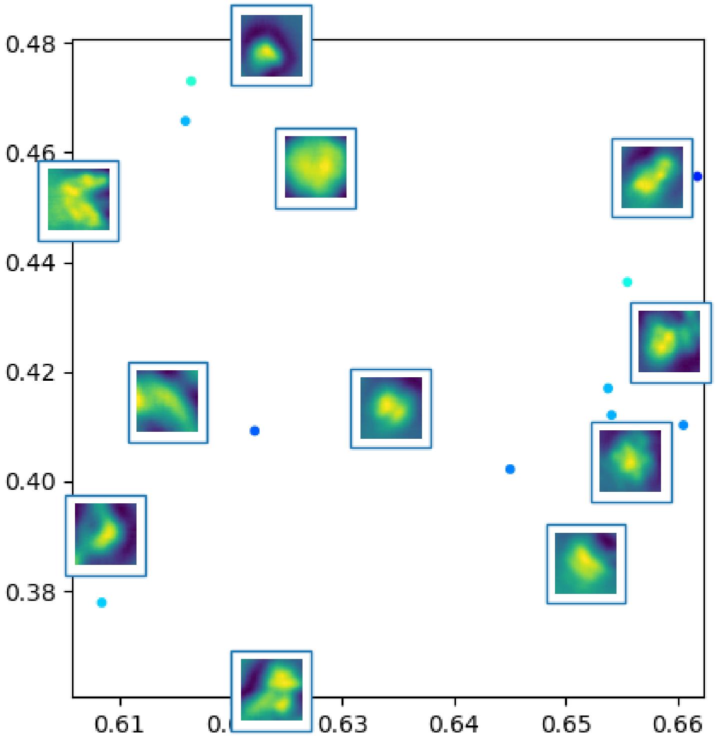


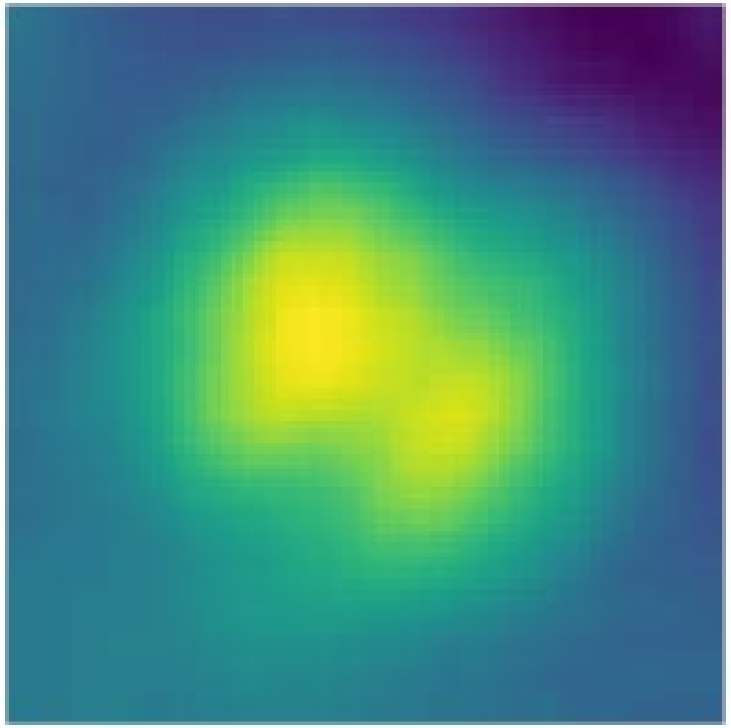
VAE, planar flow Area 1



VAE, planar flow area 2





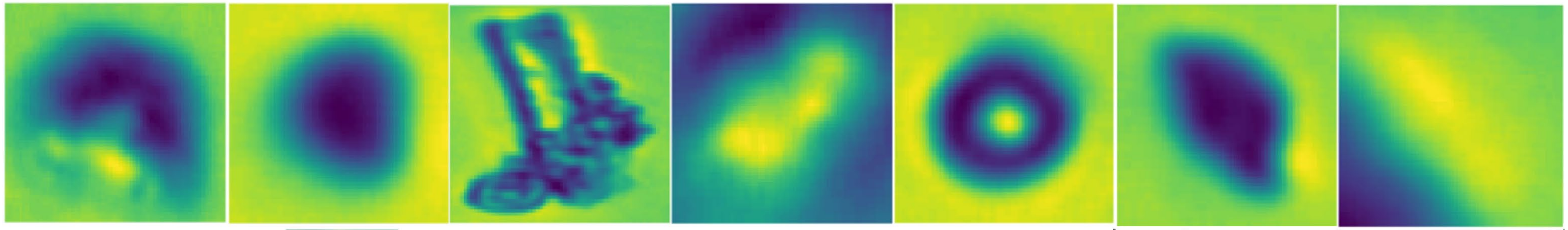


Findings

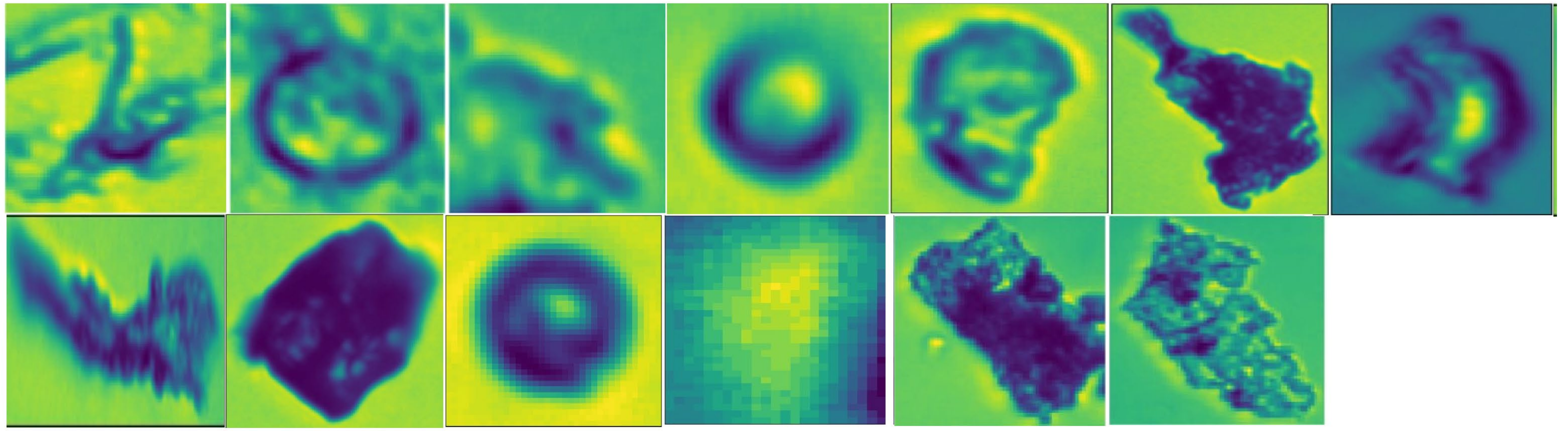
Our results

20 Images

Main categories

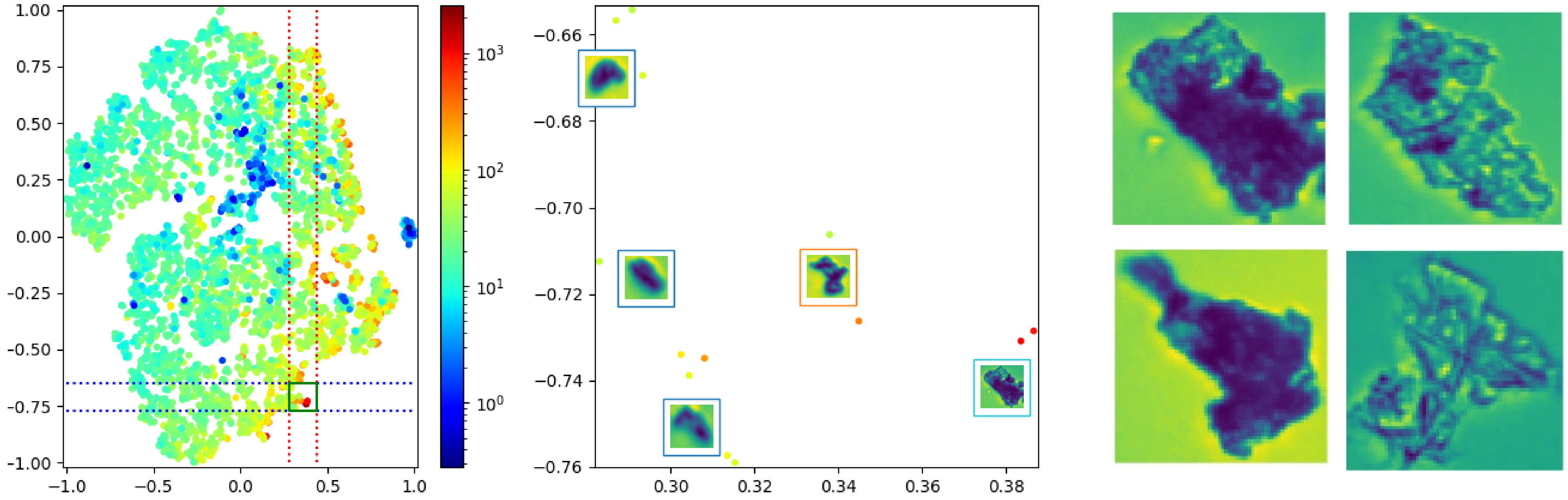


Outliers



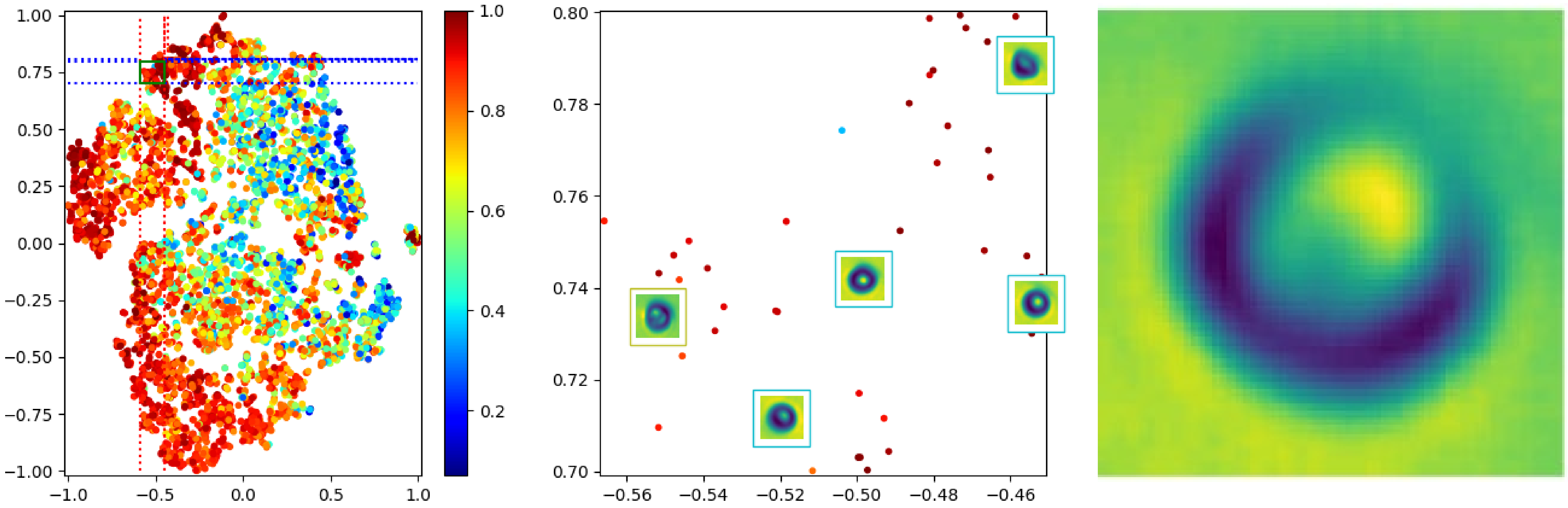
Special 1

Separate area of high-resolution images similar area #5 (Outlier)



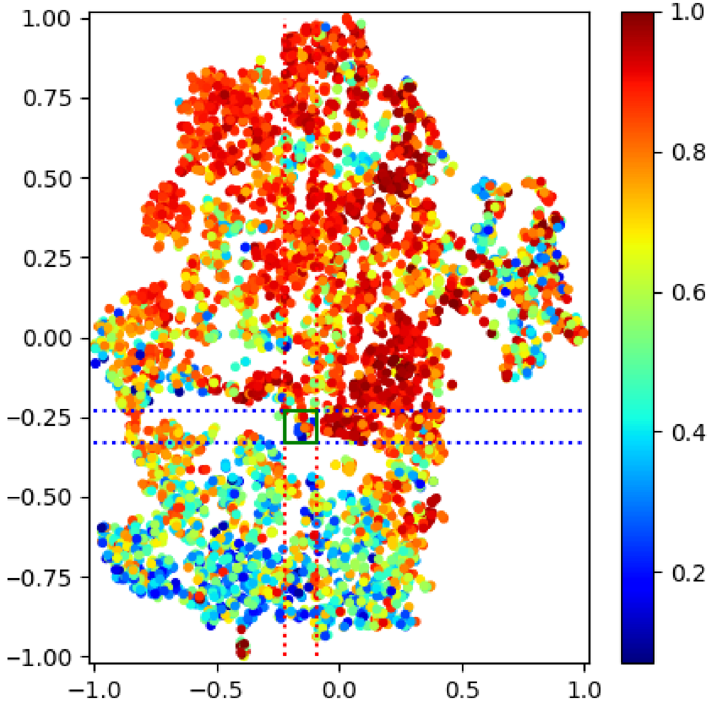
Special 2

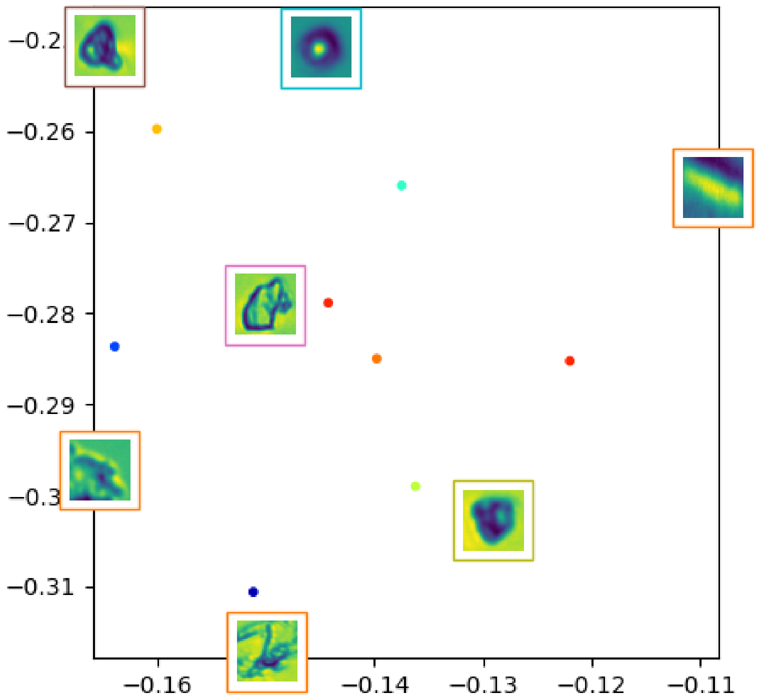
Low circularity image surrounded by ones with high circularity. (Outlier)

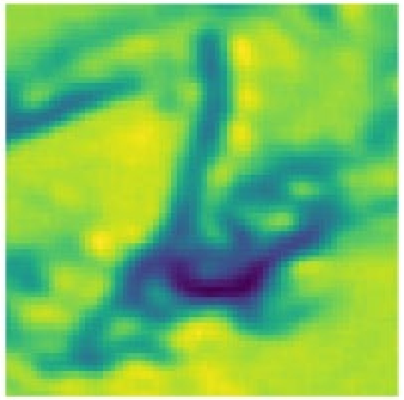


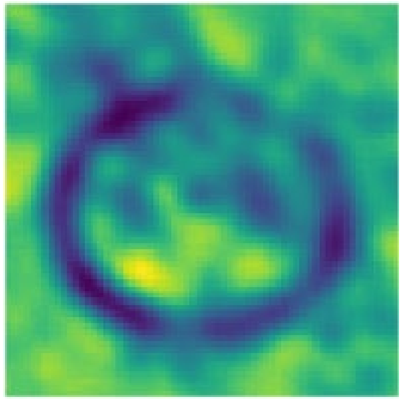
Special 3

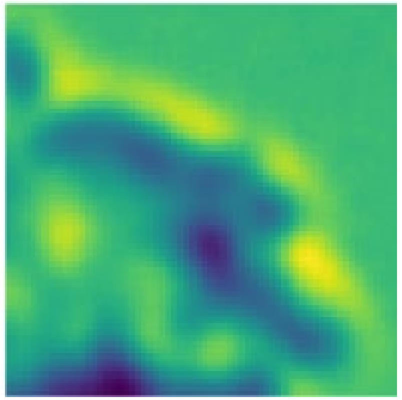
Low circularity images surrounded by ones with high circularity. (Outlier)
Unusually deform











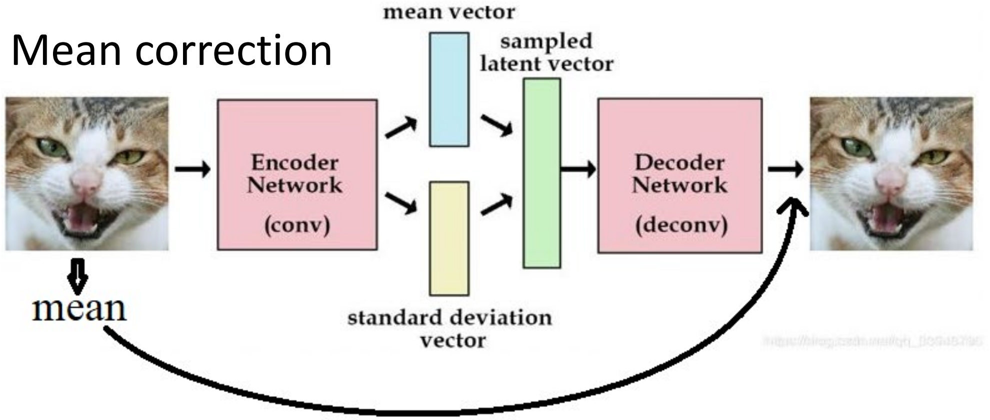
Further work

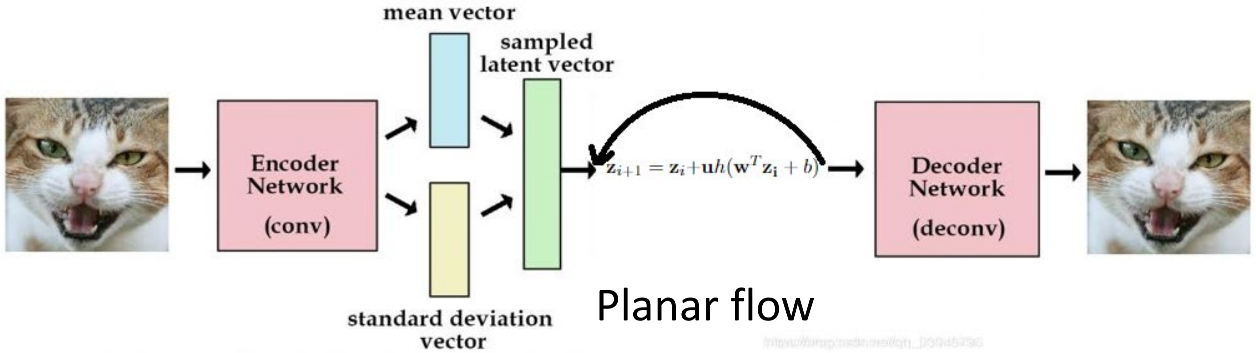
- Try implementing auto encoders with added scalar output.
- Use permutation importance on scalar parameters in classifier to reduce the number of parameters.
- Work further with the dataset to see if more balanced data for training can be created.

Conclusion

- Using MNIST we were able to reconstruct all classes on partially trained classifiers and from latent extracted form AutoEncoders and VAEs
- Using resnet18 we were able to build models that can make representations of Peruvian ice core images.
- Encountered and overcame problems while training VAEs.
- Fitting less to the artificial set seemed to help the VAE classes.
- Tried clustering and different (V)AE architectures.
- Were able to use representations and dimensionality reduction to identify main classes and outlier images.

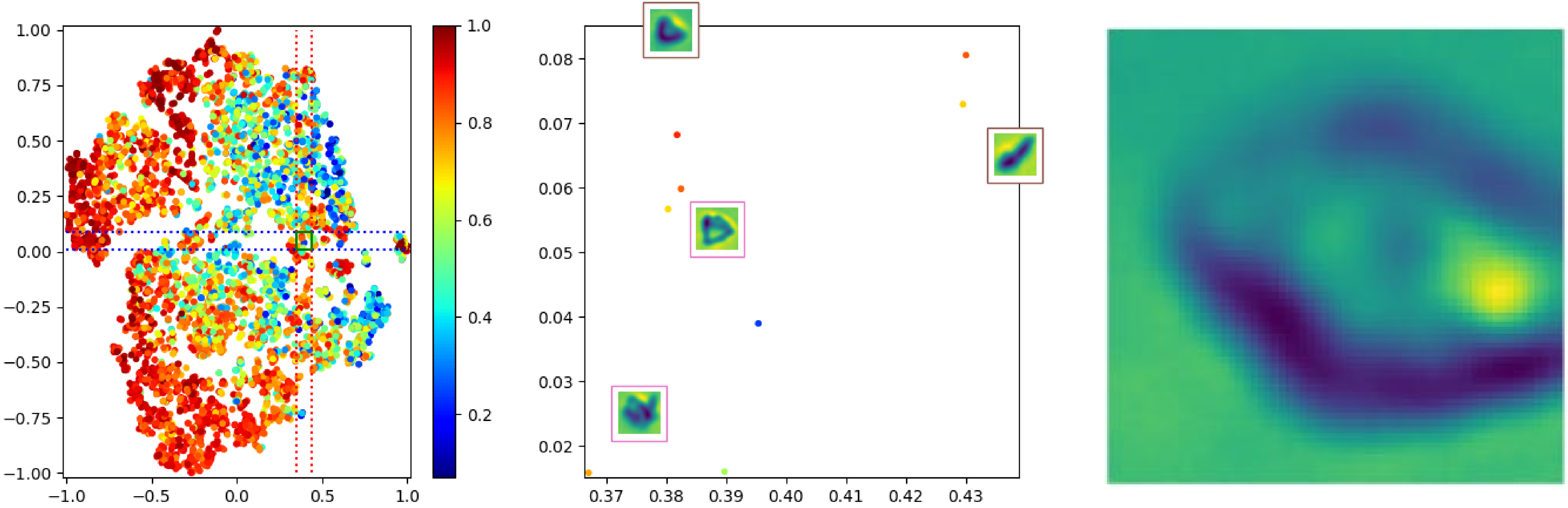
Mean correction





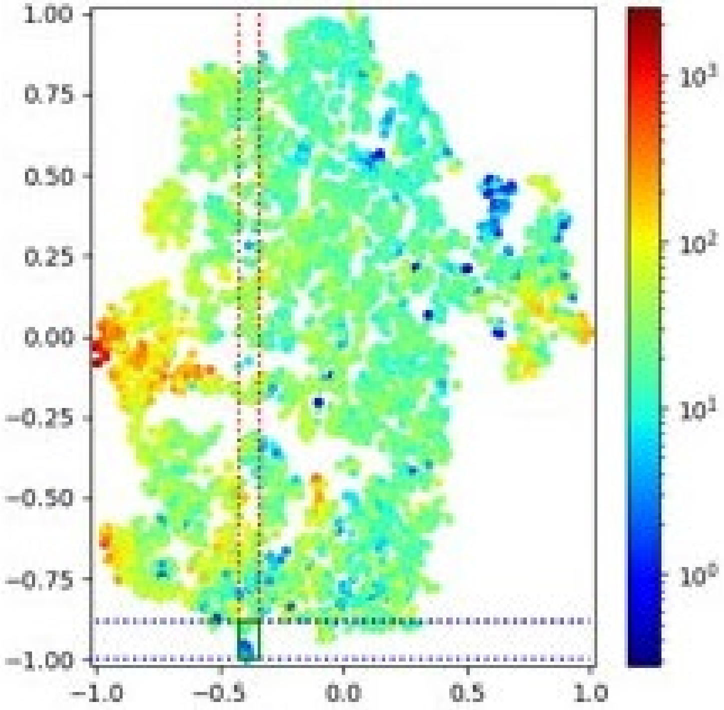
Special 4

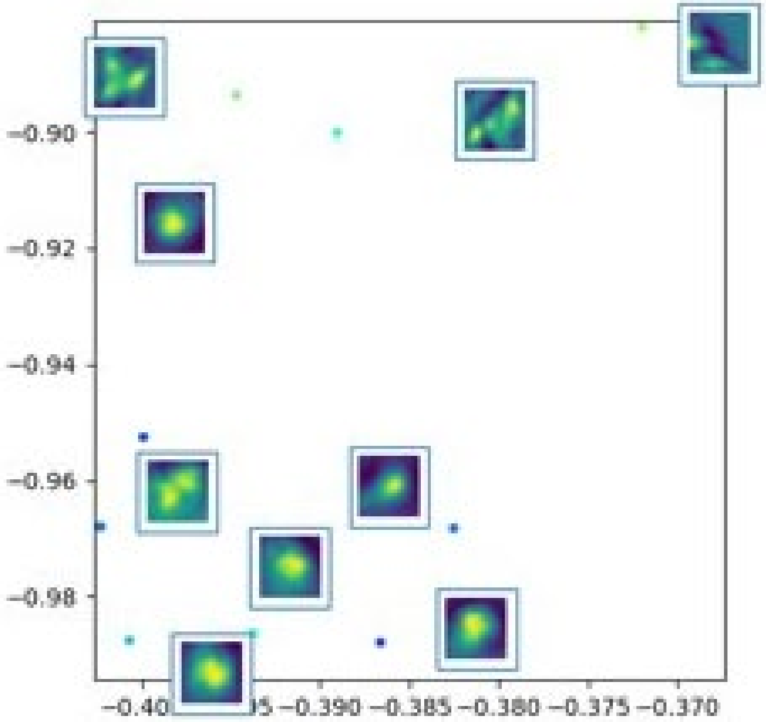
Low circularity image surrounded by ones with high circularity. (Outlier)

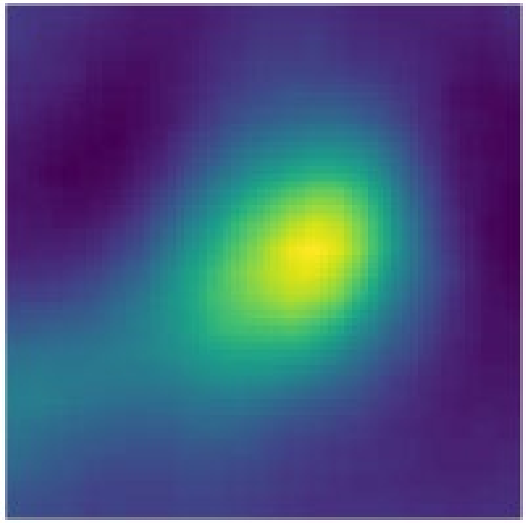


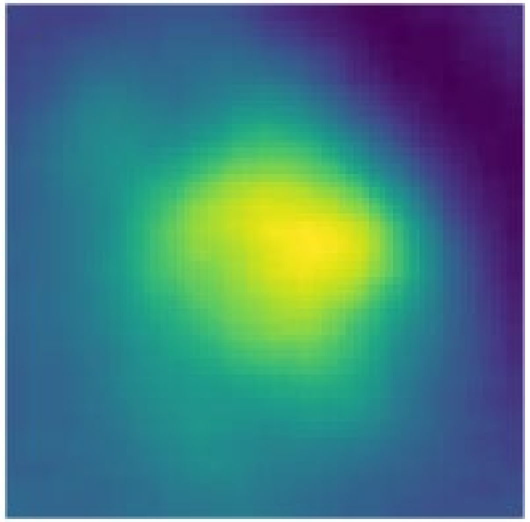
Special 5

Local area of small image "light blobs"



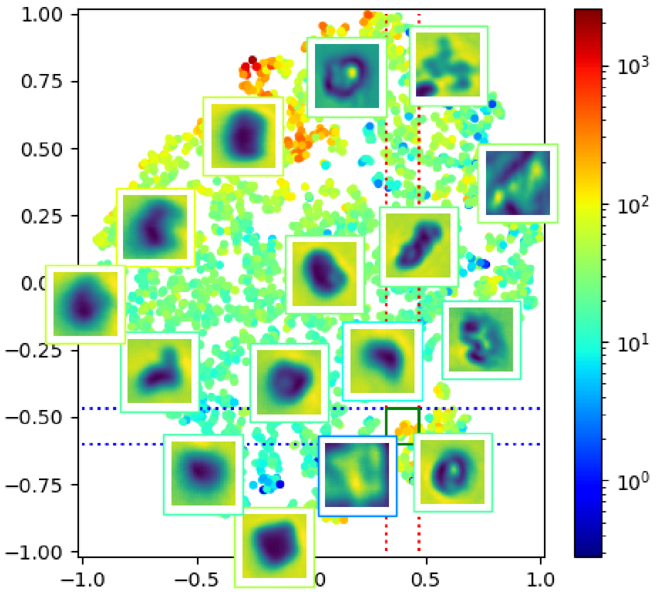


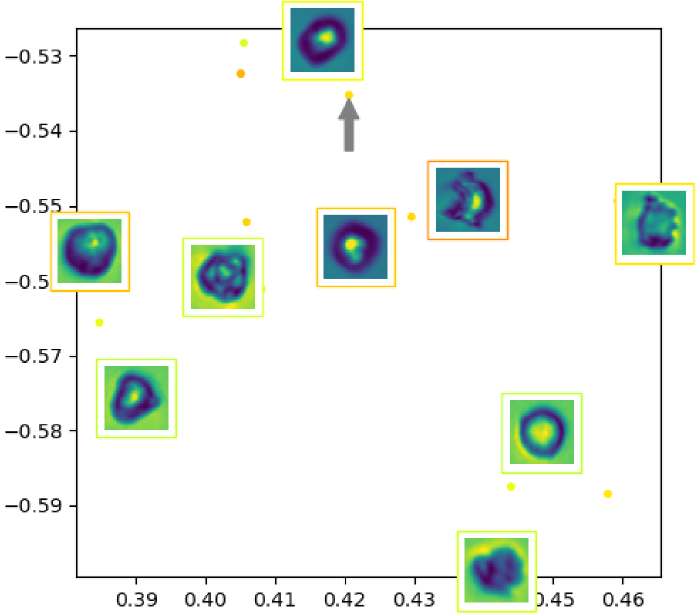


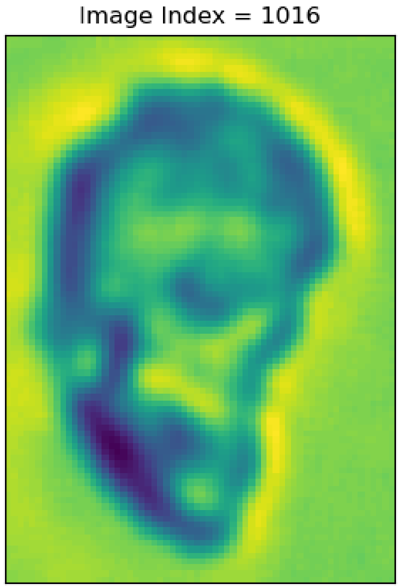


QCY_25_7_5_401

Why?



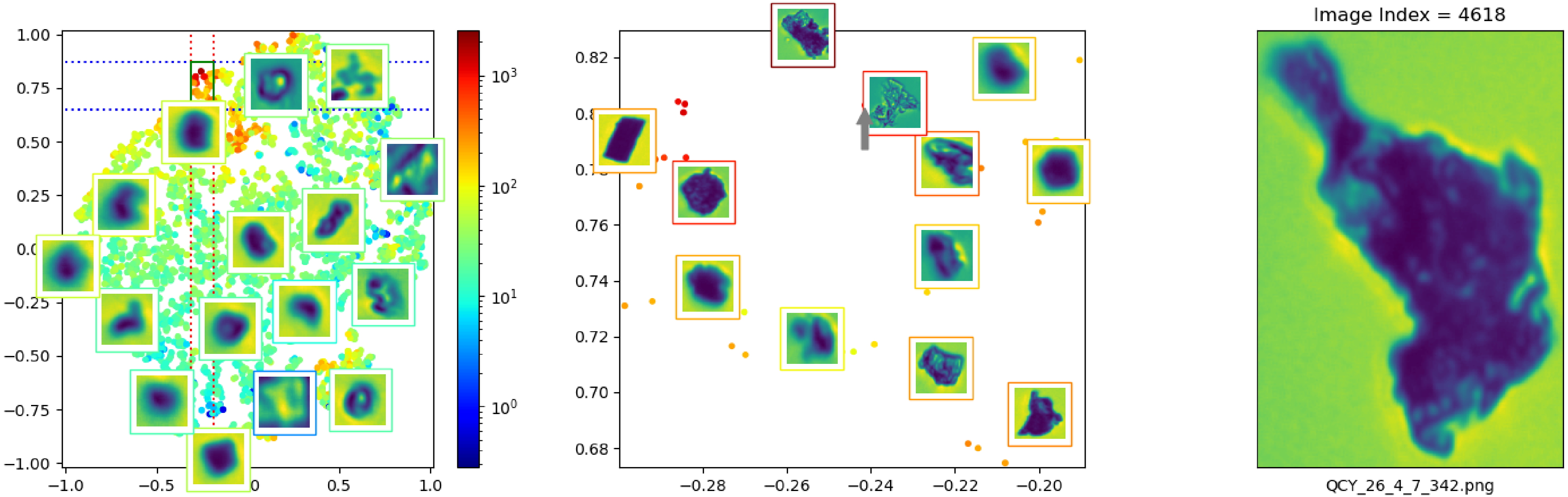




QCY_25_7_5_401.png

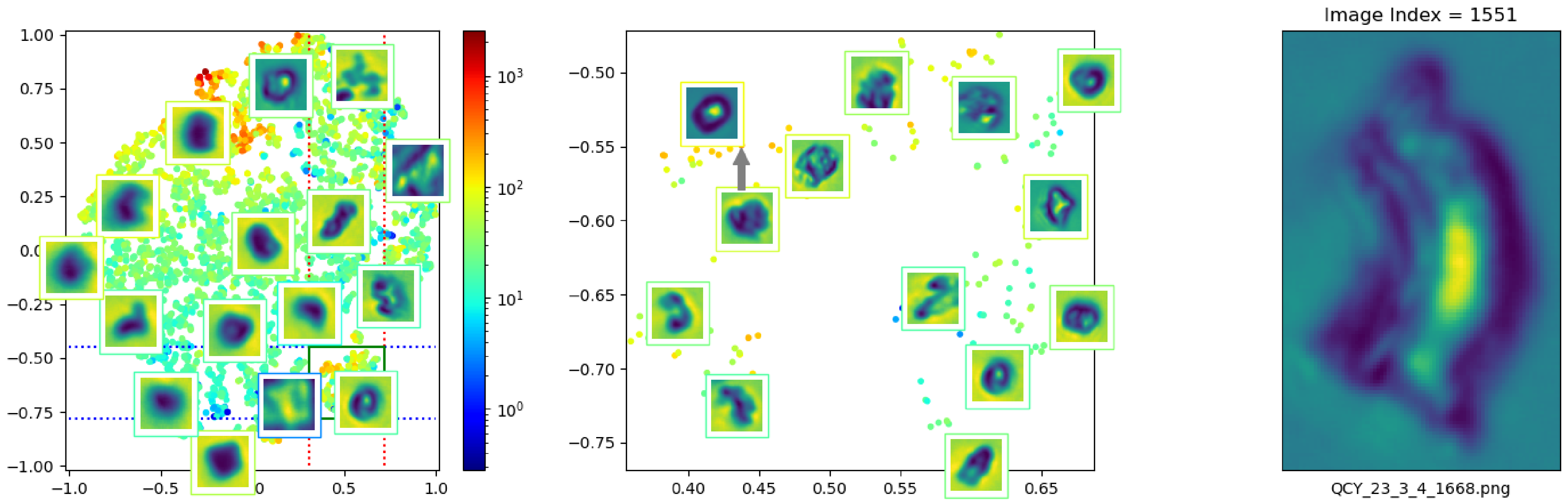
QCY_26_4_7_342

Why?



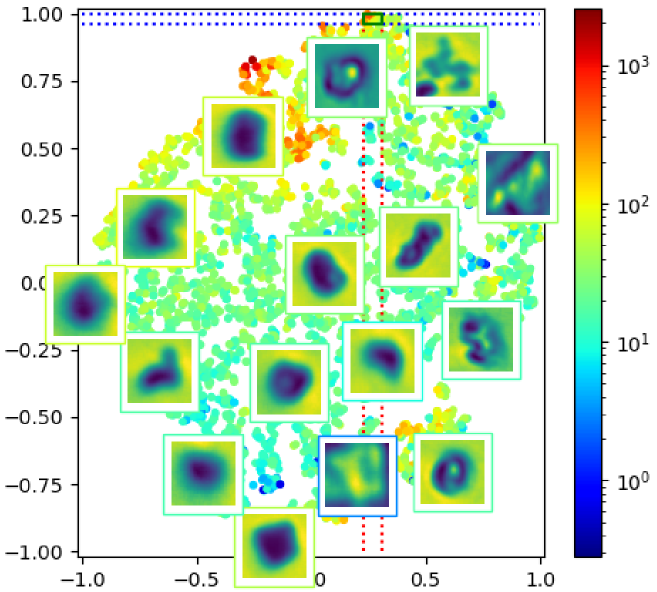
QCY_32_3_4_1668

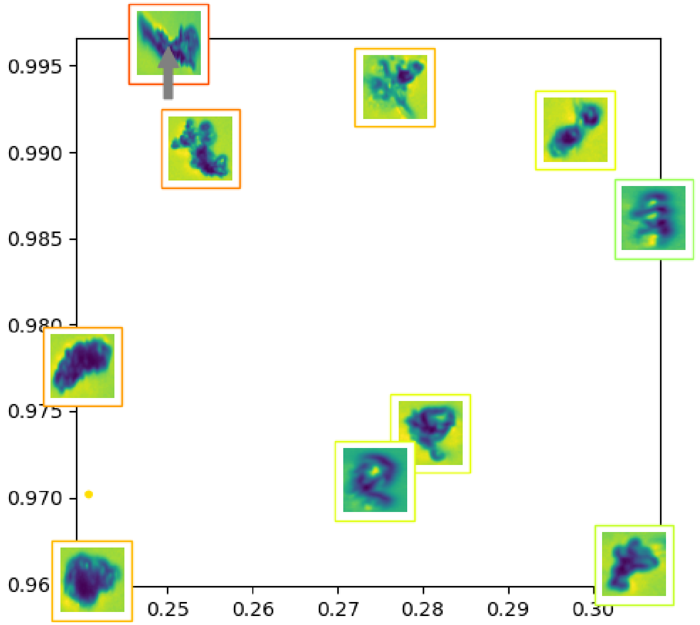
This looks to be a separate cluster and the selected images is an example of it

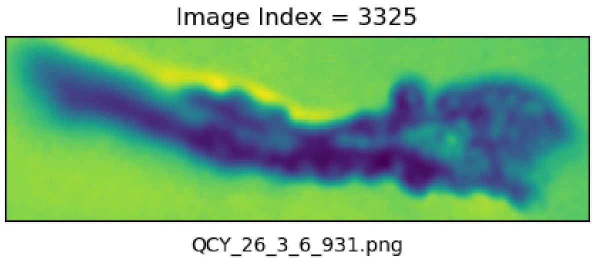


OCY_26_3_6_931

Visually distinct large object. (15th largest object)

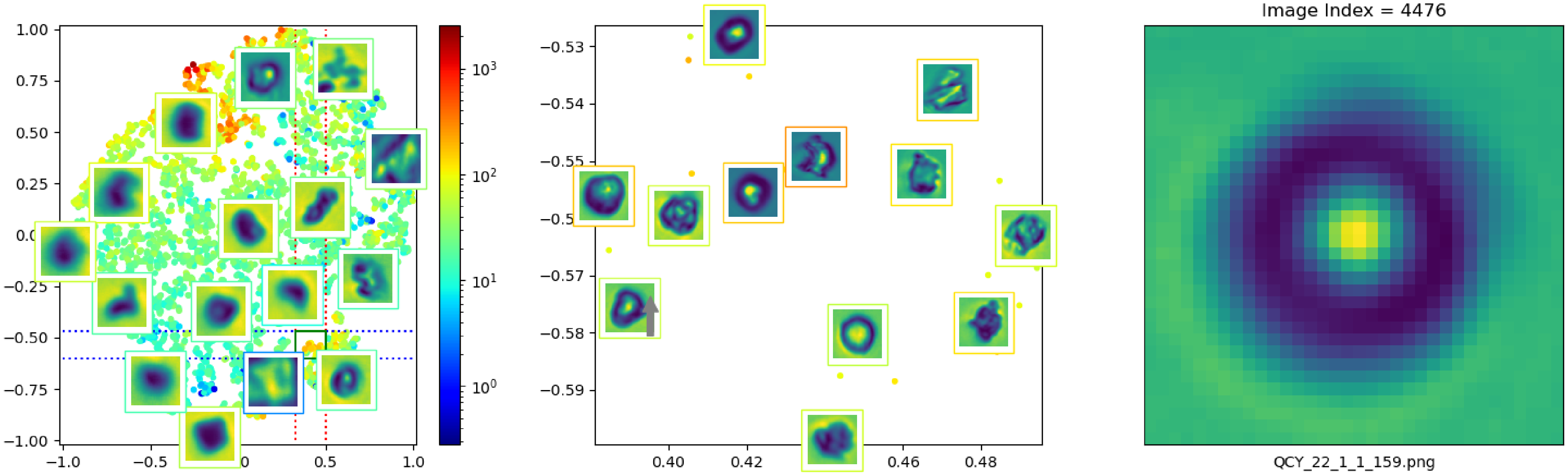






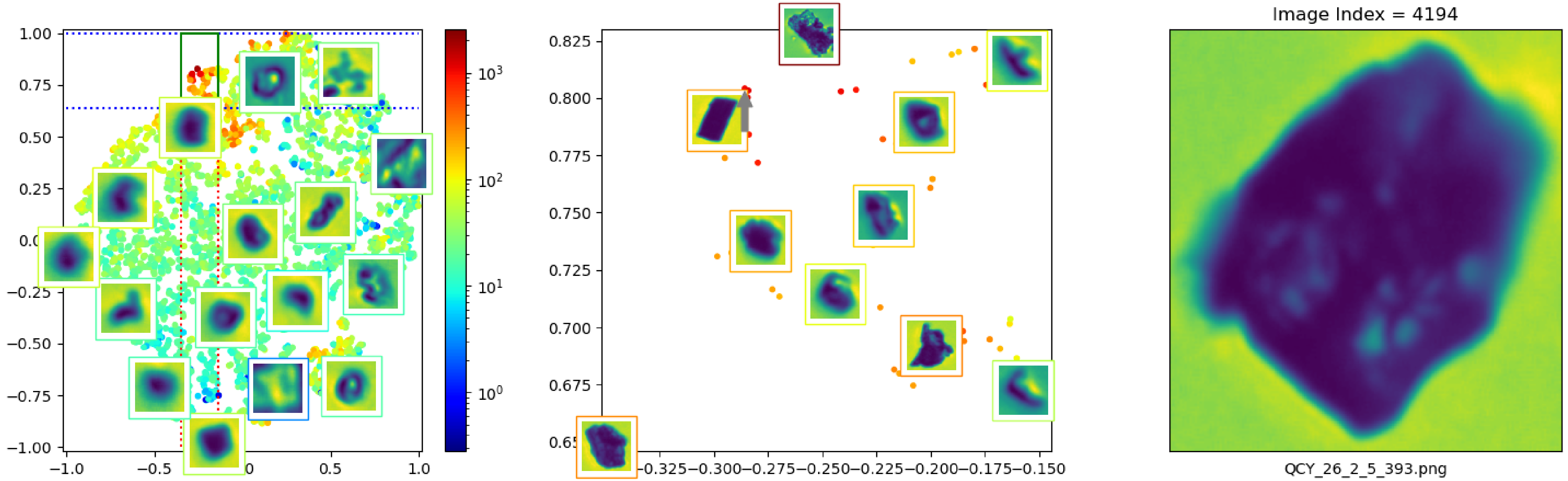
QYC_22_1_1_159

Small image (Outlier)



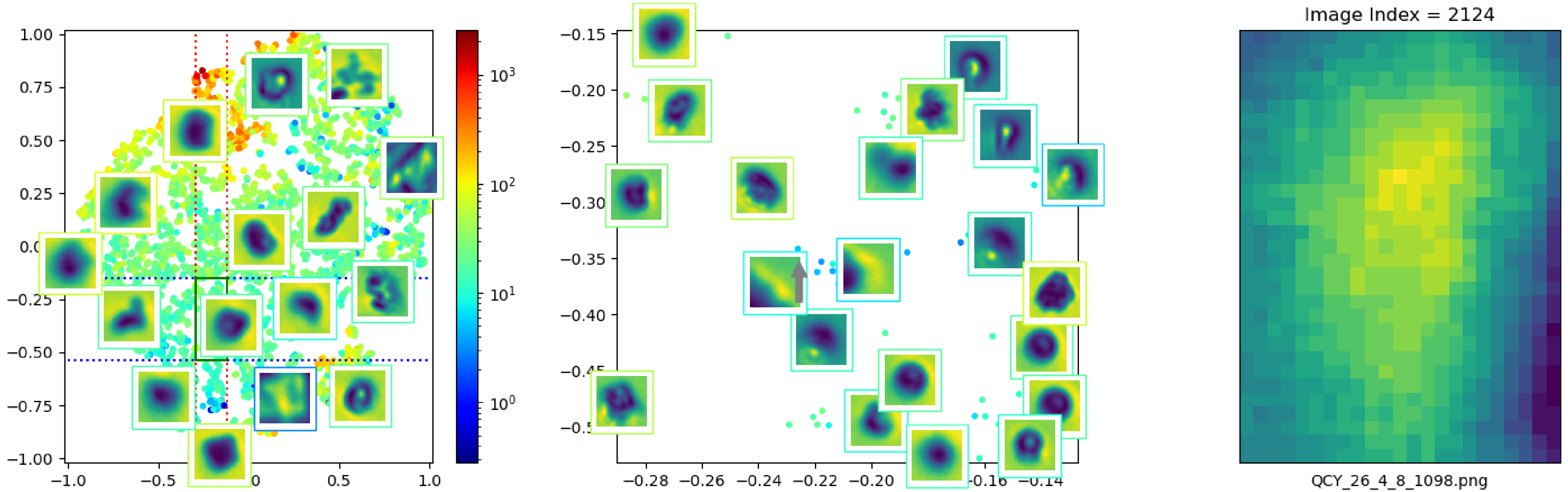
QCY_26_2_5_393

Highest loss for on VAE



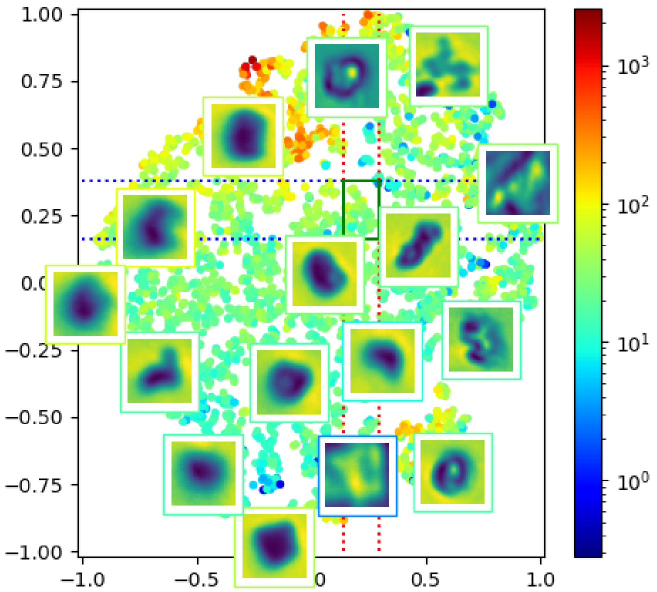
QCY_26_4_8_1098

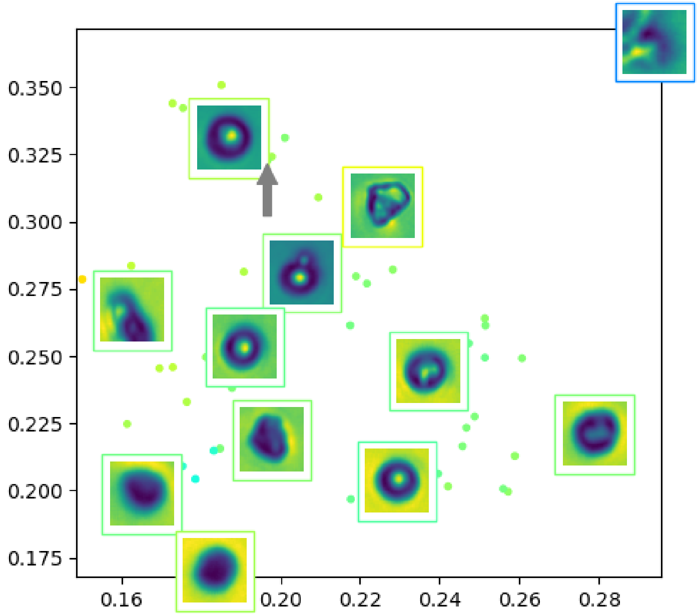
Second highest loss on VAE

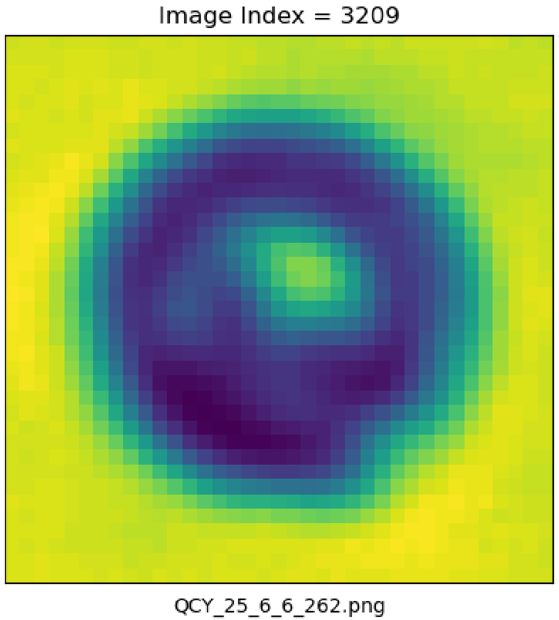


QCY_25_6_6_262

Previously clustered differently



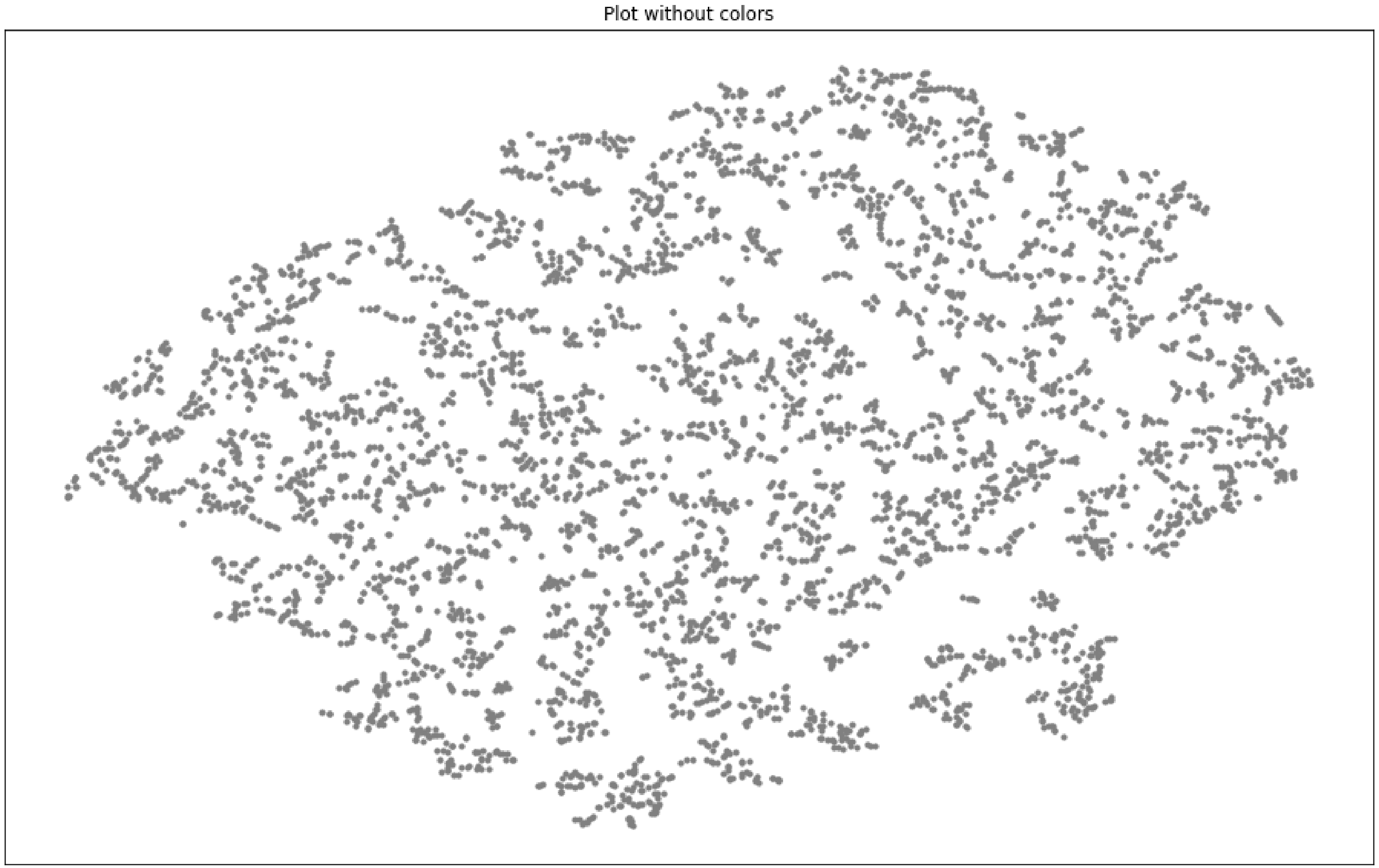




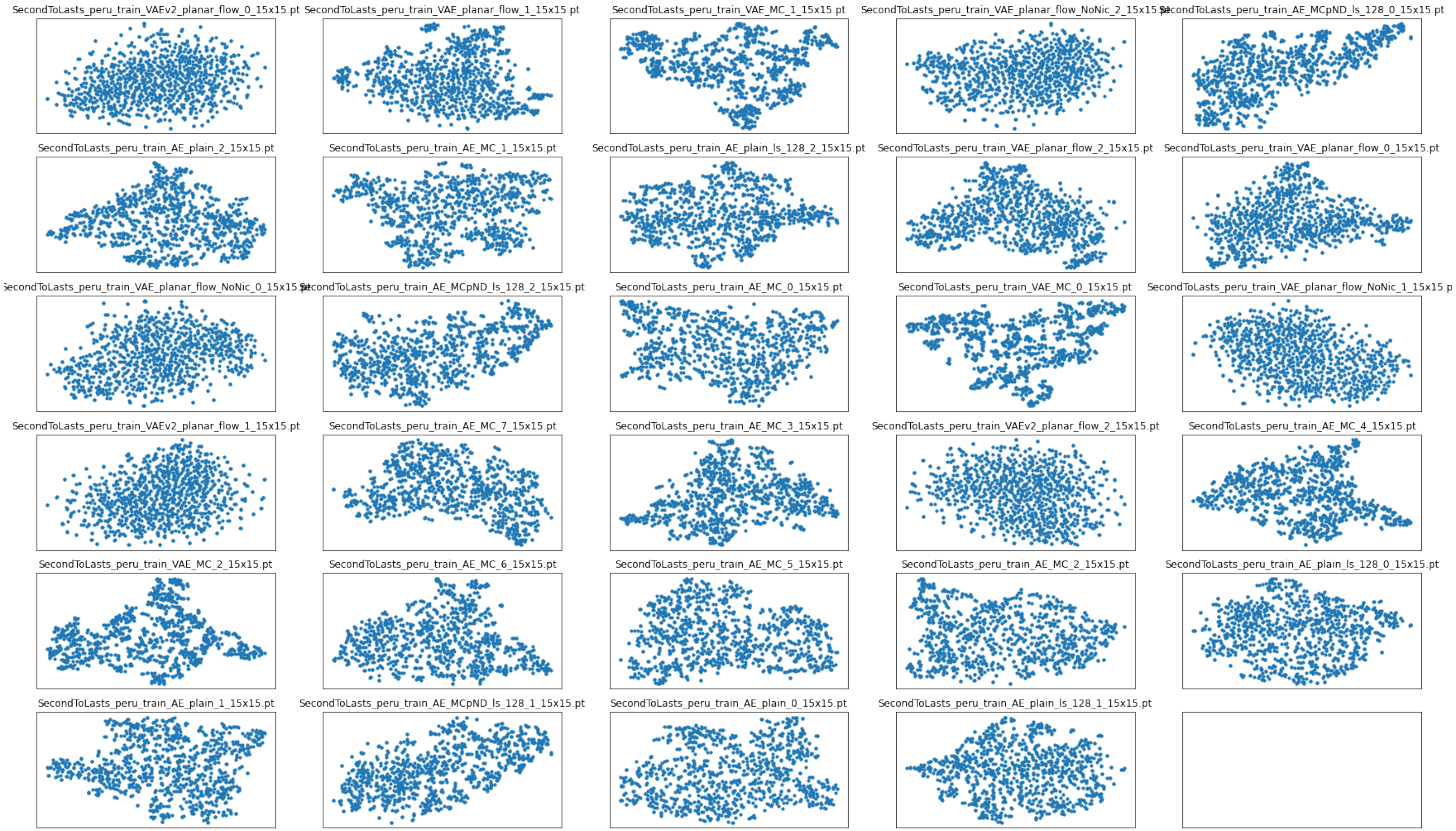
Appendix

Are you looking for more?

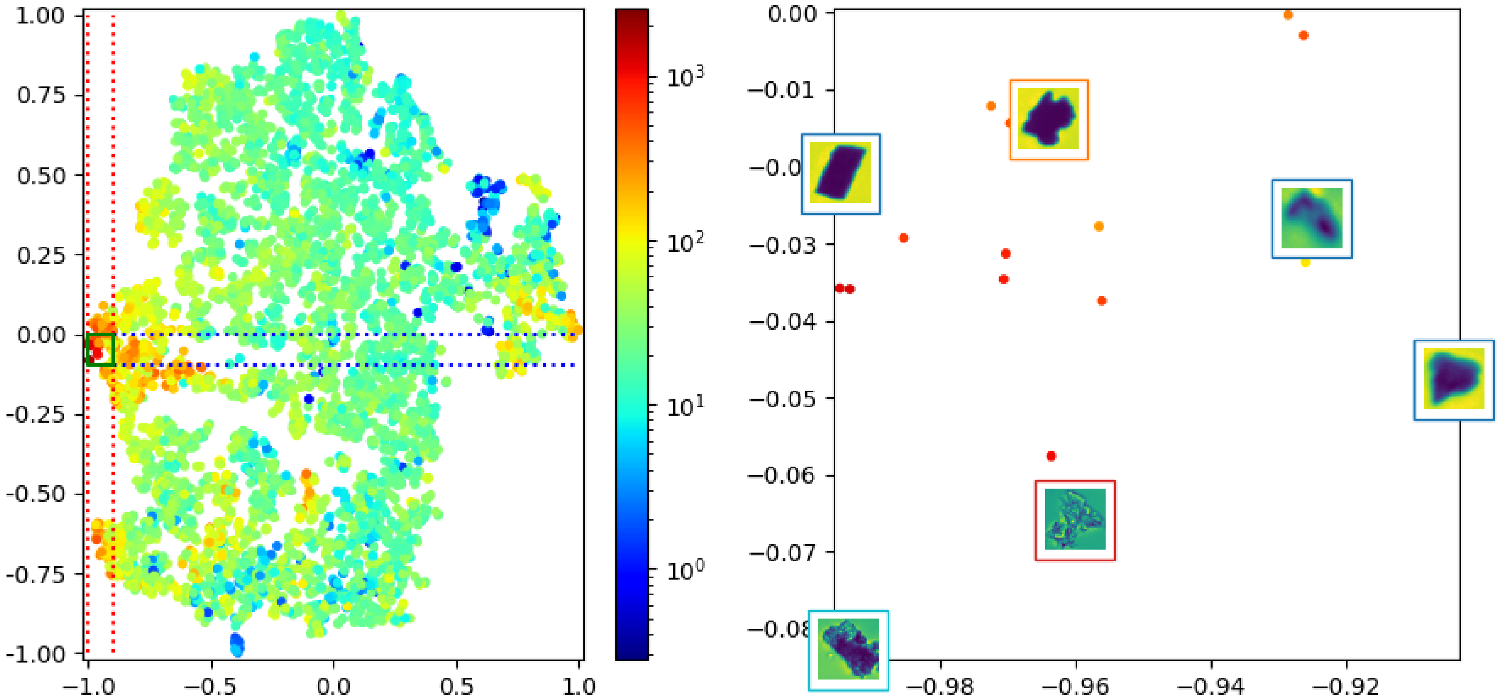
Plain plot

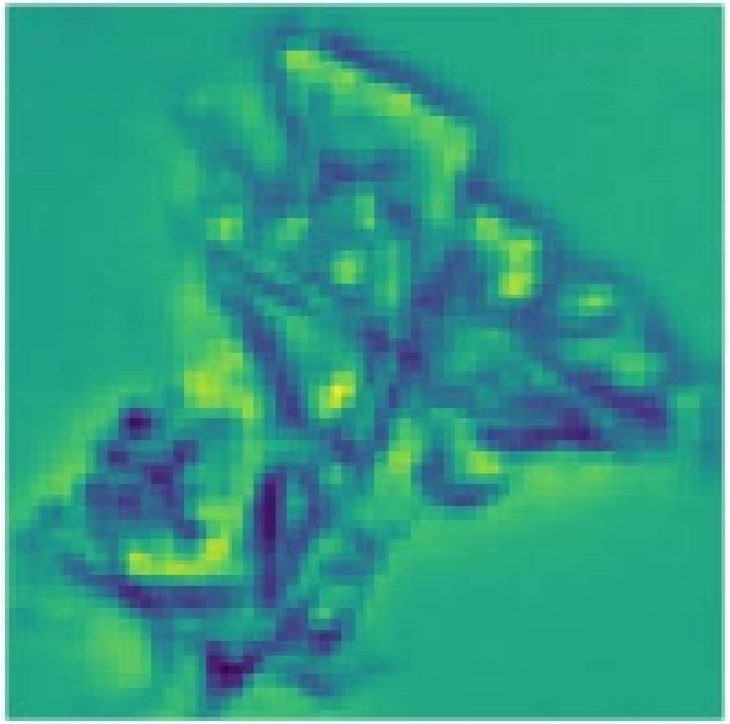


t-Sne for different (V)AEs latent spaces

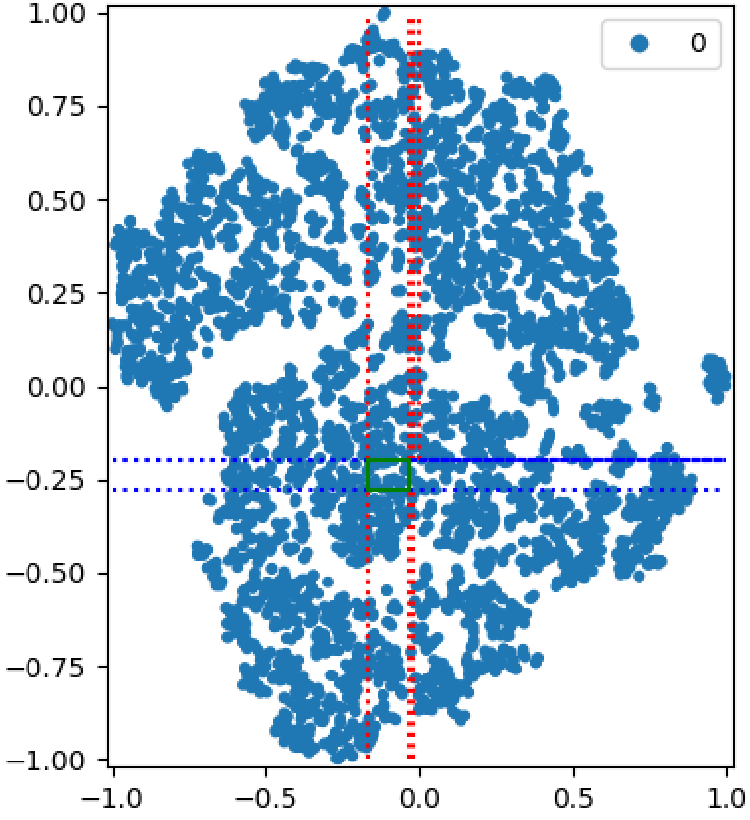


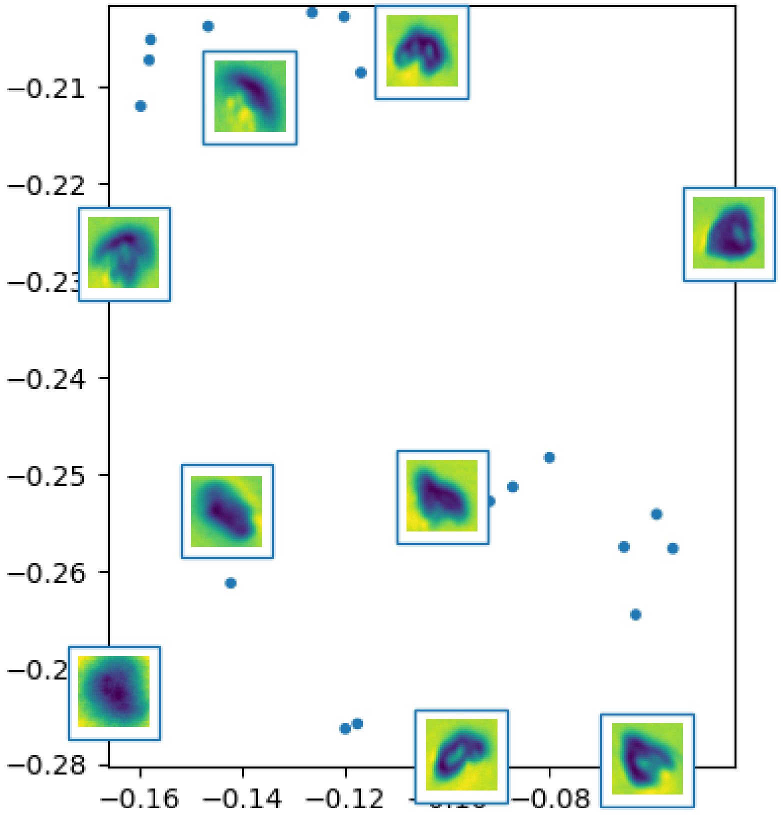
Refound largest images

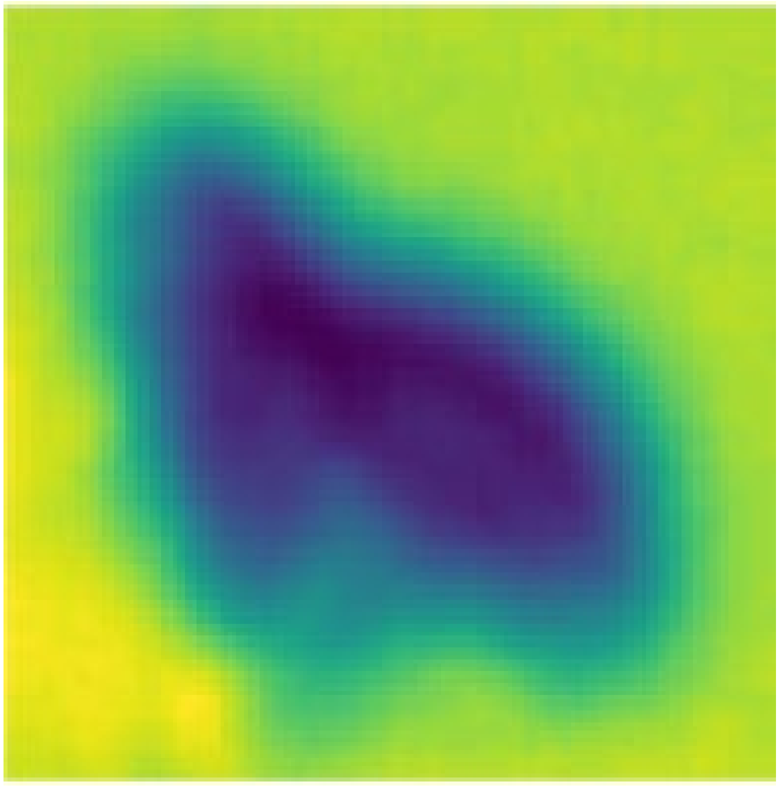




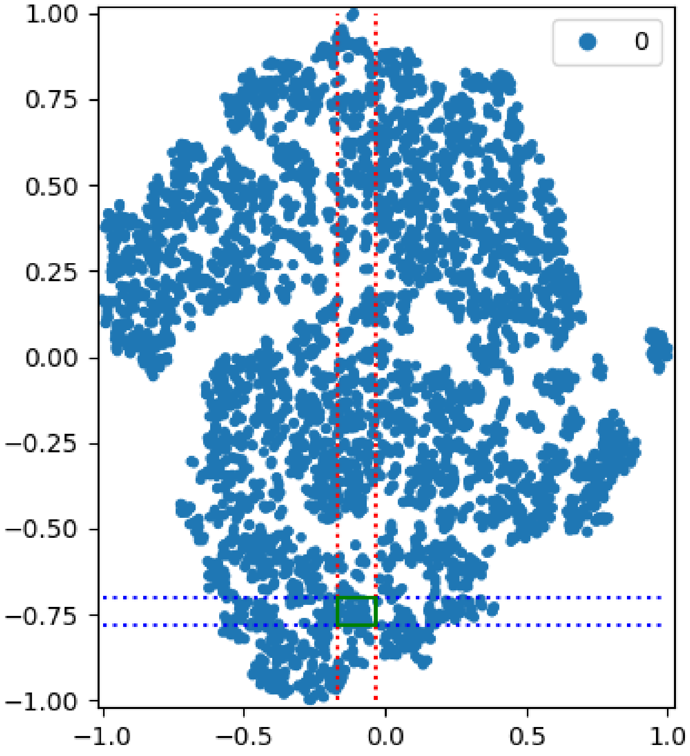
CNN classifier - Area 3

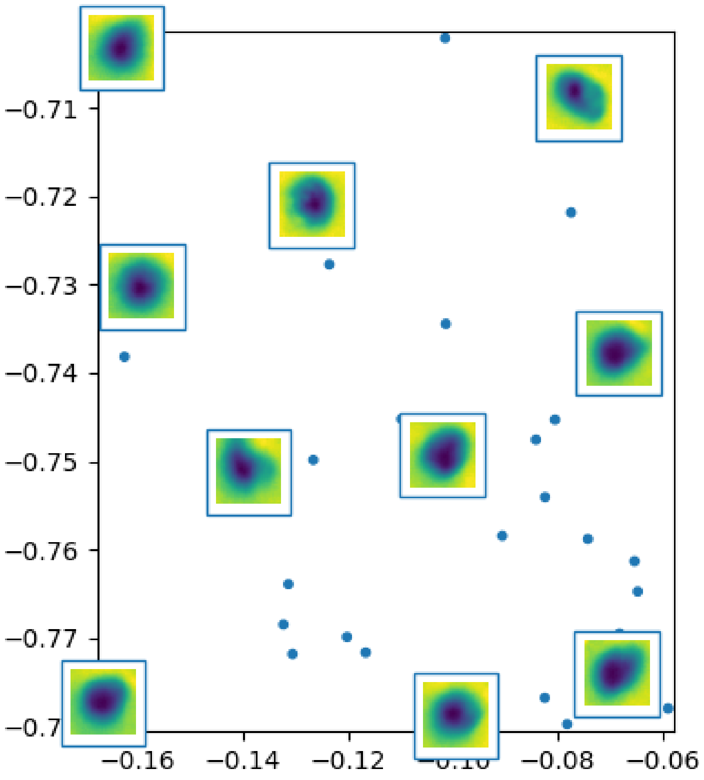


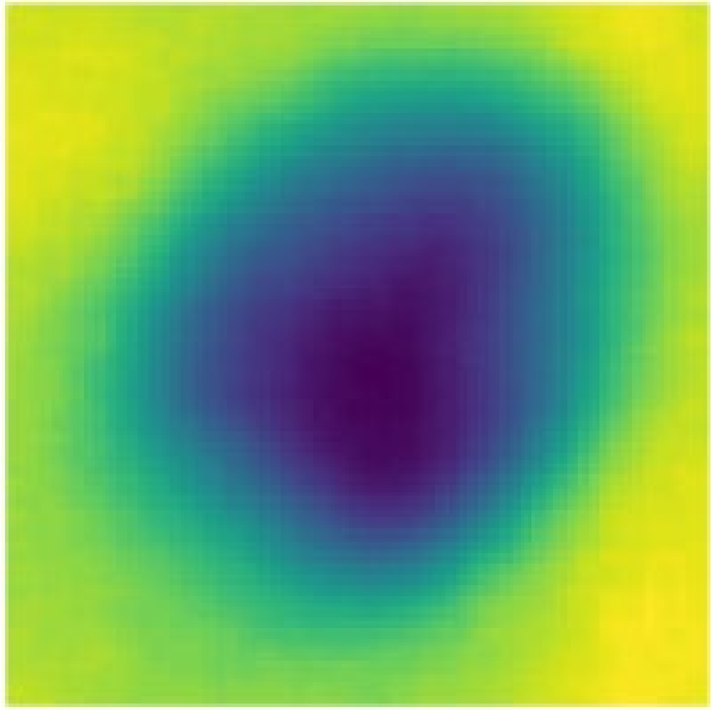




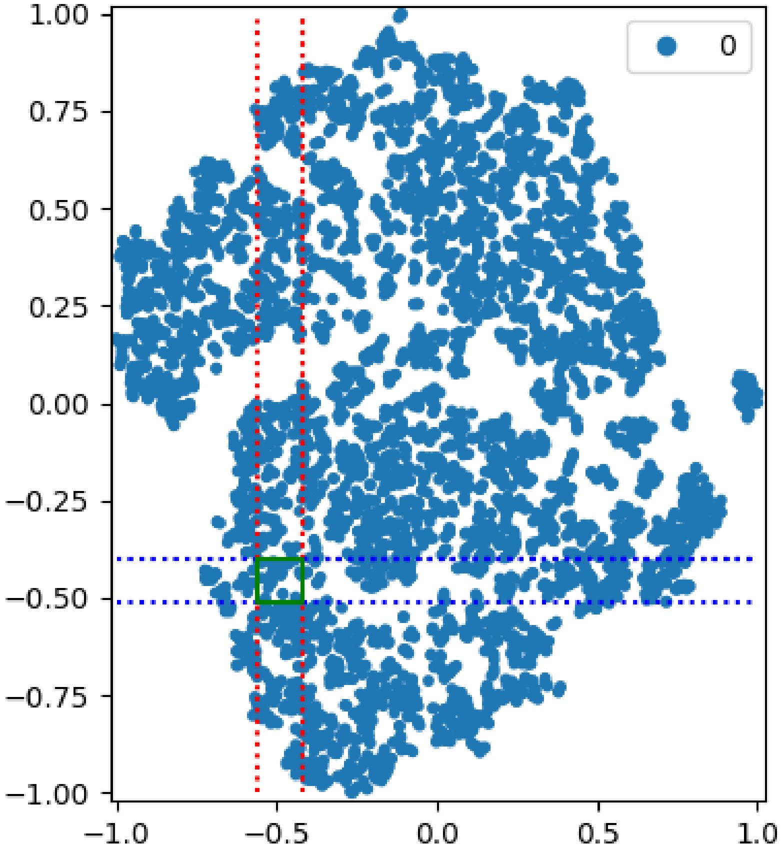
CNN classifier - Area 4

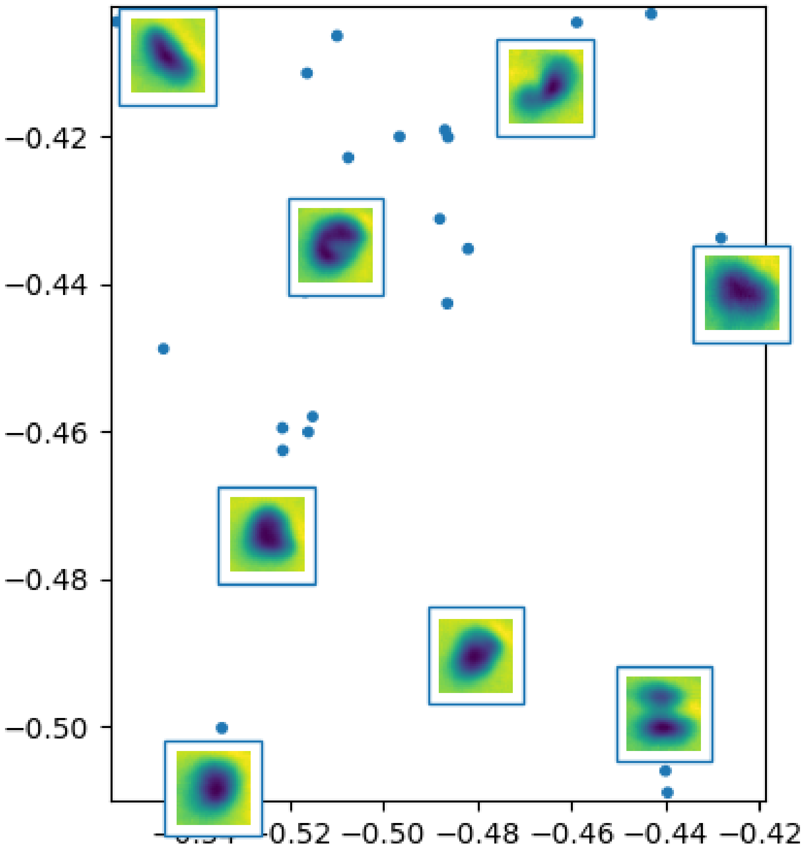




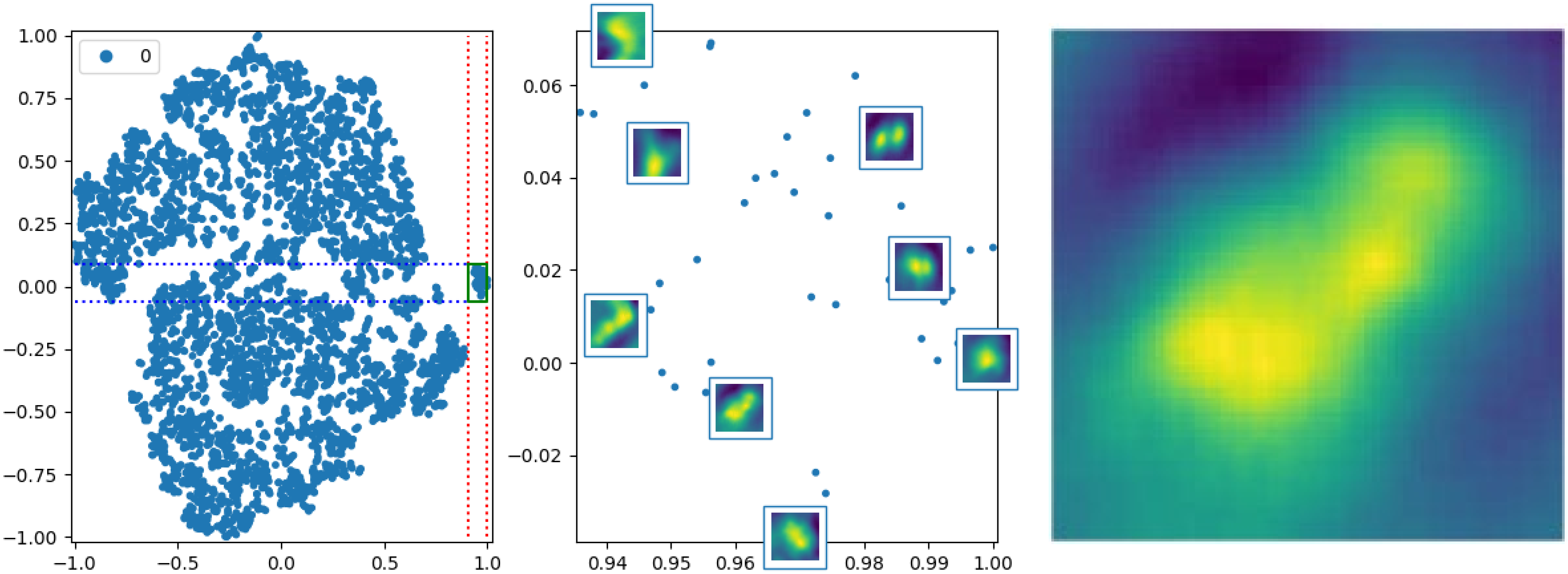


Between area 3 and 4



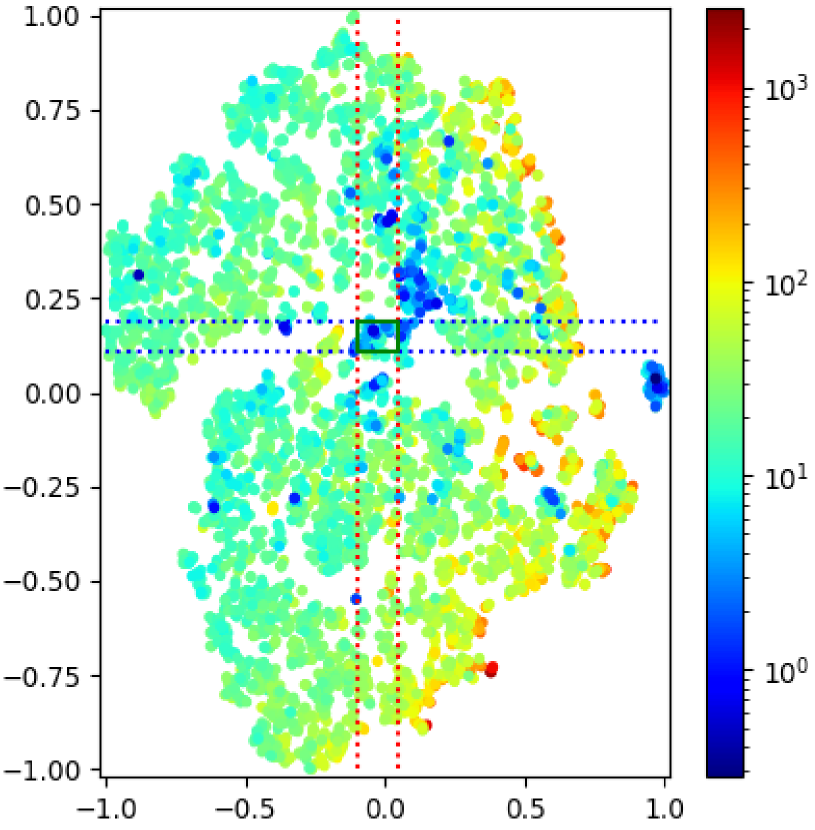


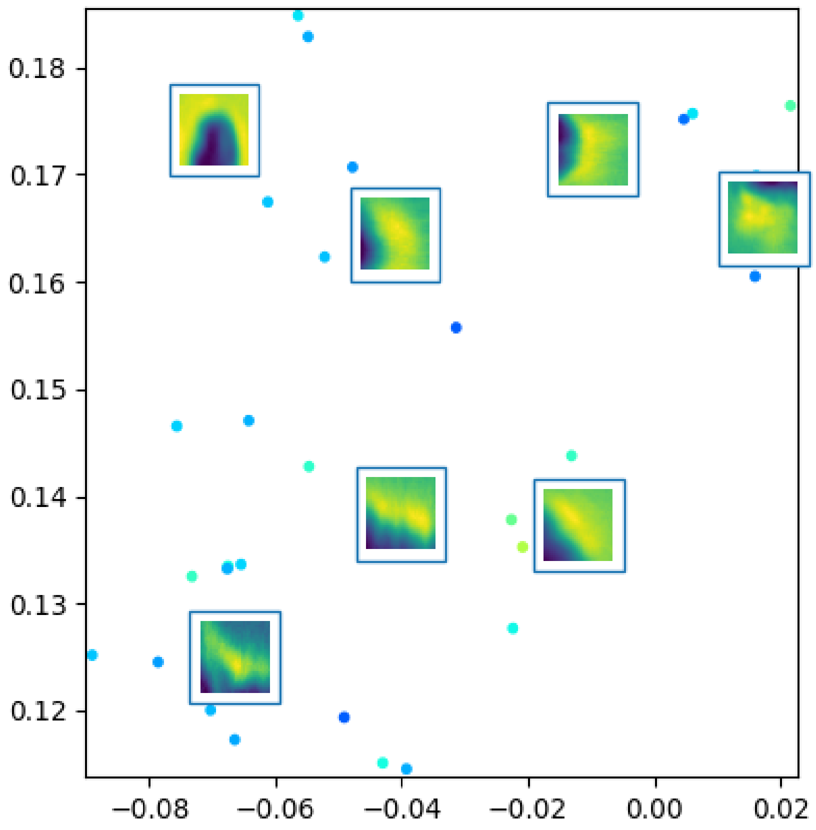
CNN classifier - Area 6

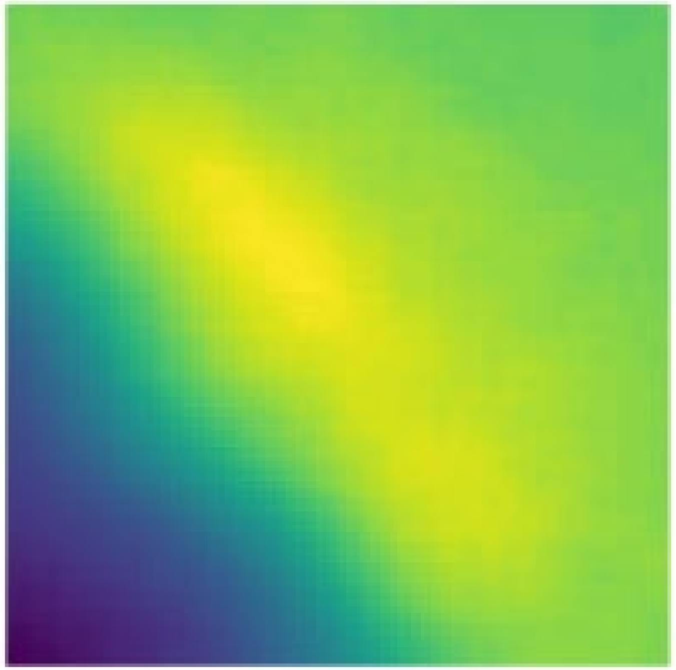


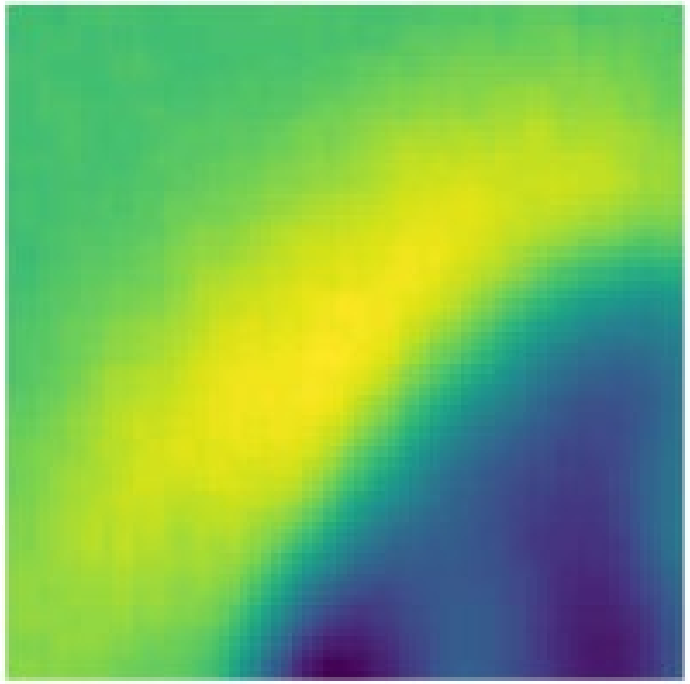
Area 7

Low area

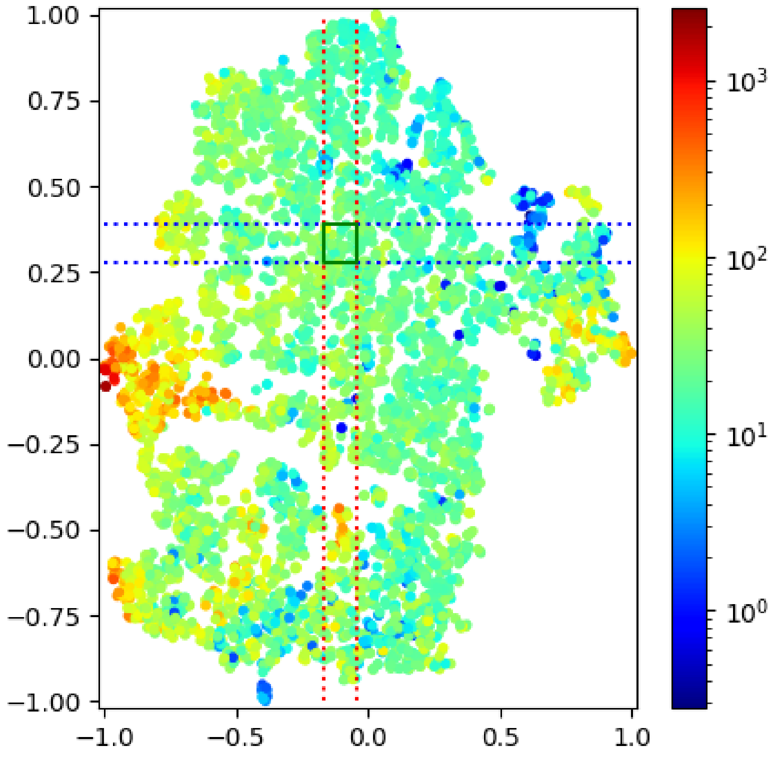


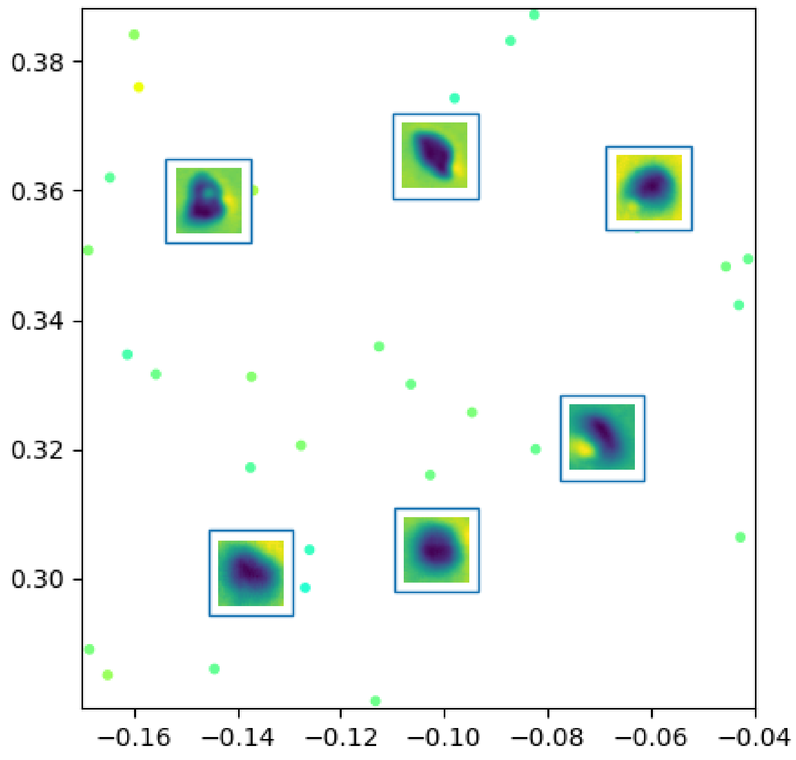


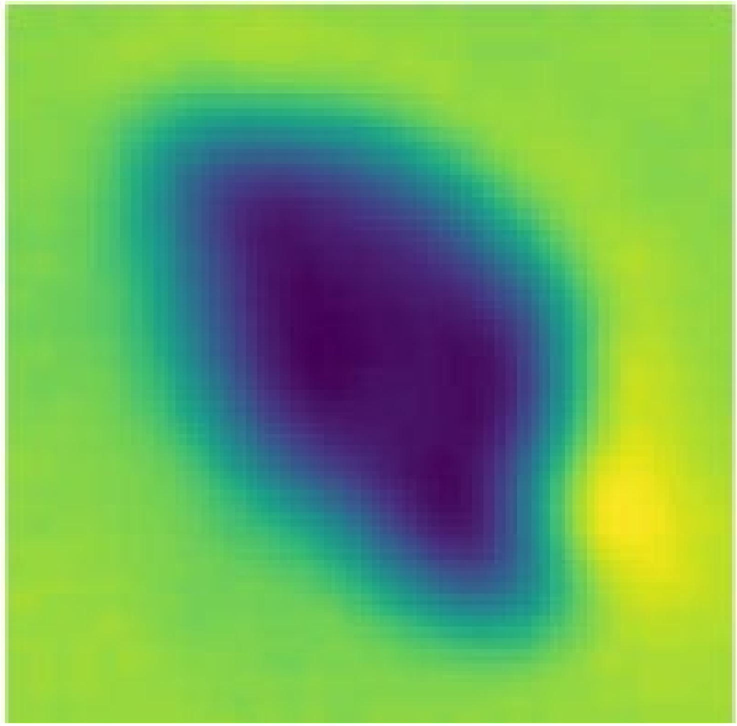




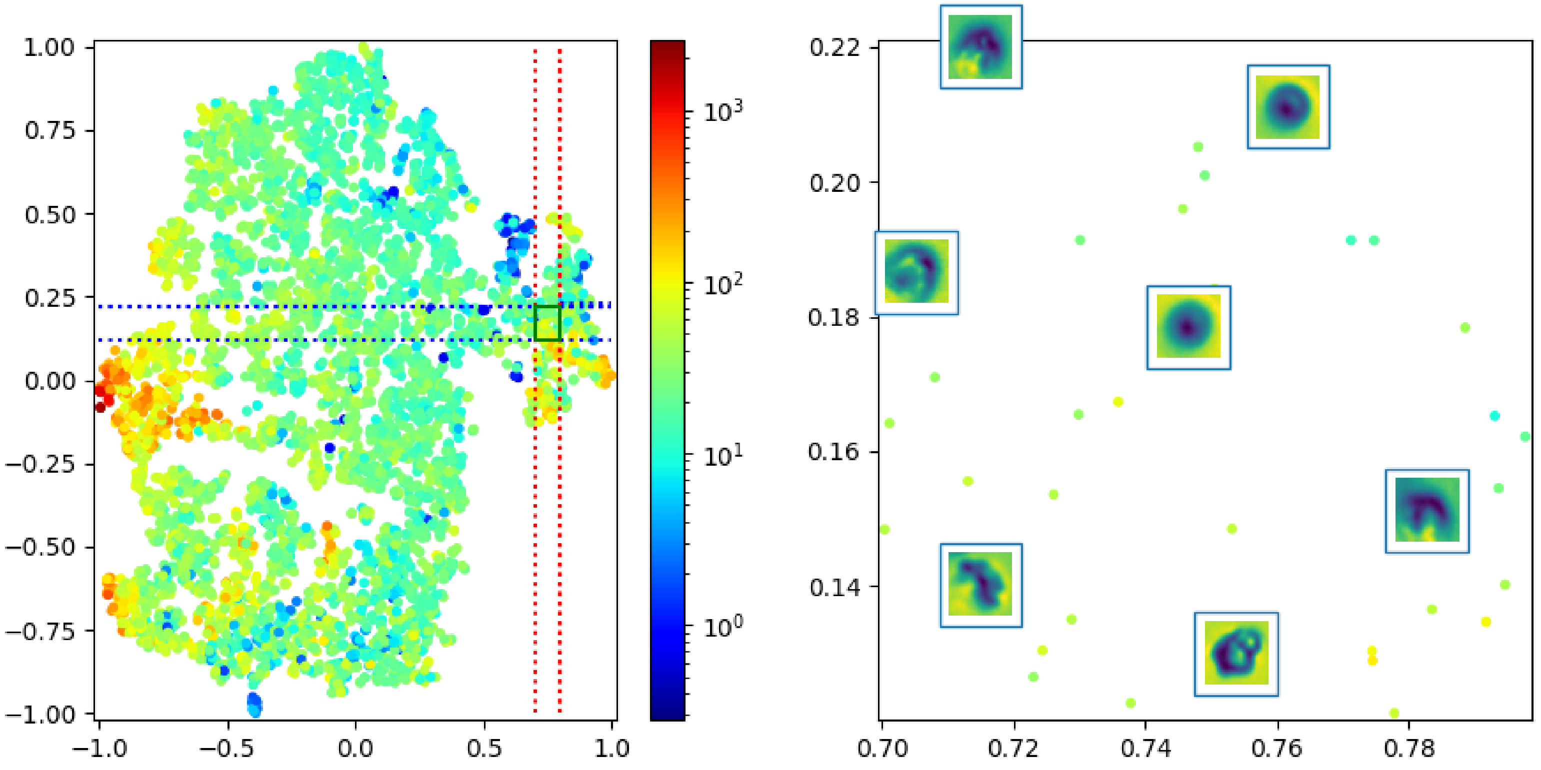
VAE, planar flow Area 1

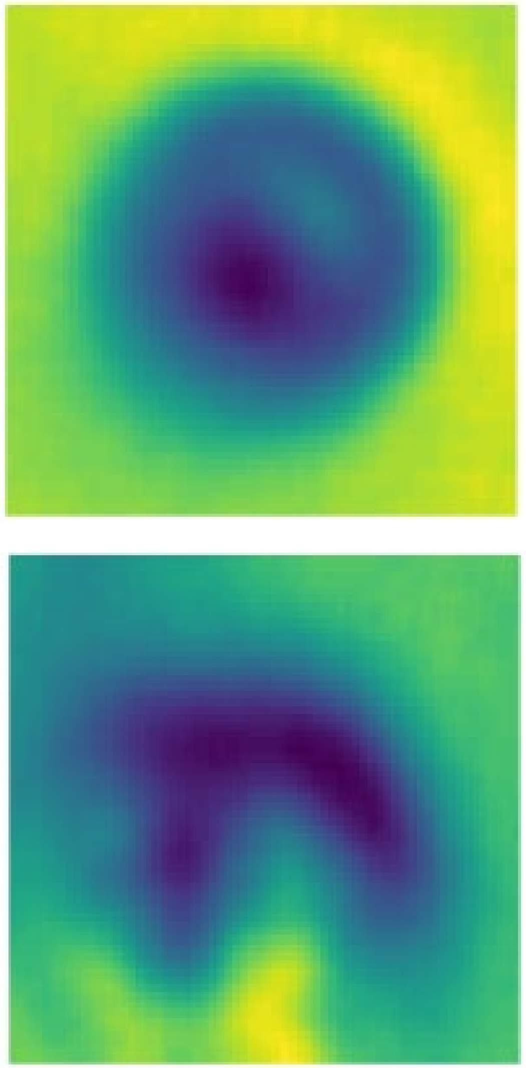




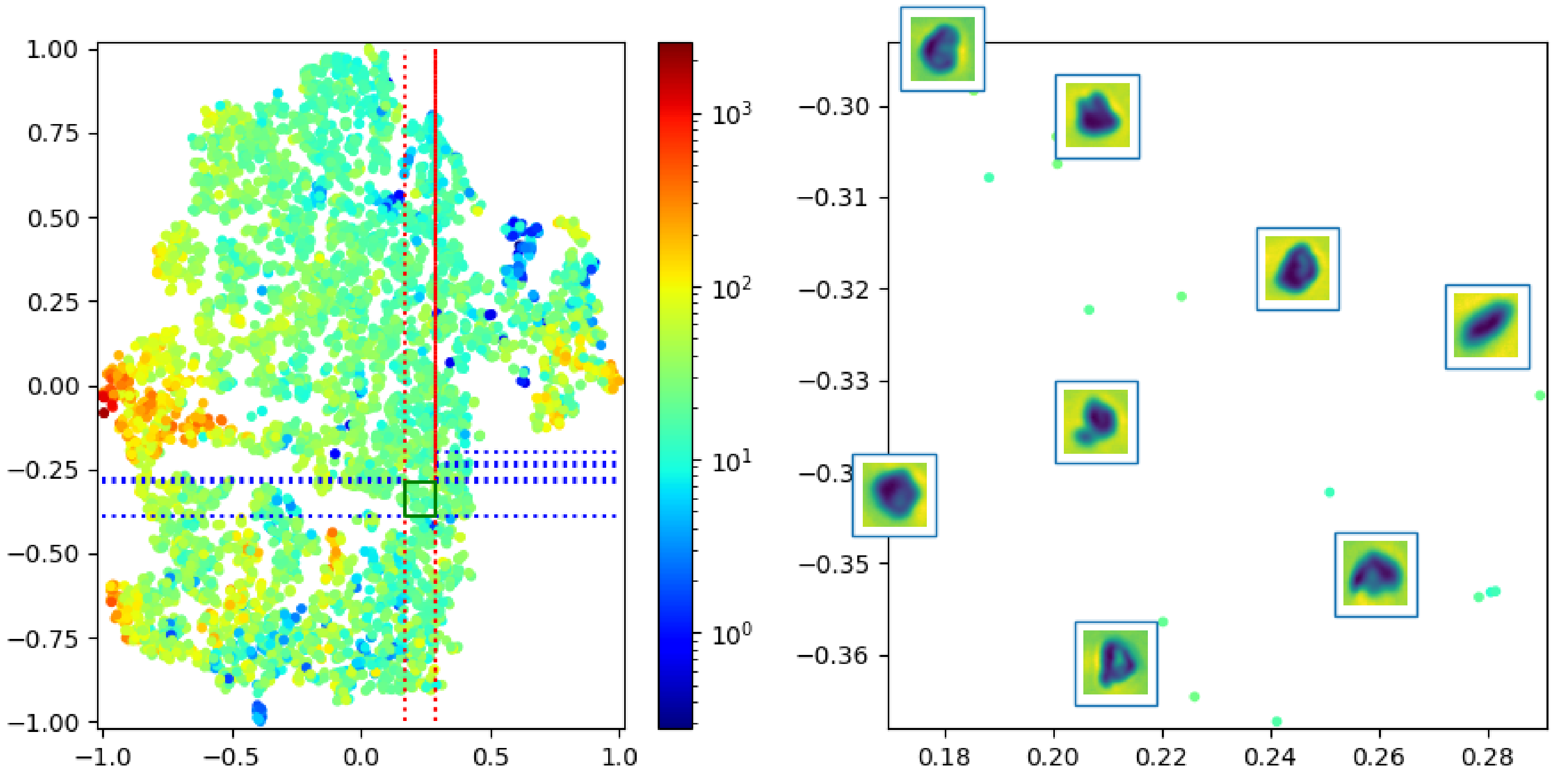


VAE, planar flow Area 2

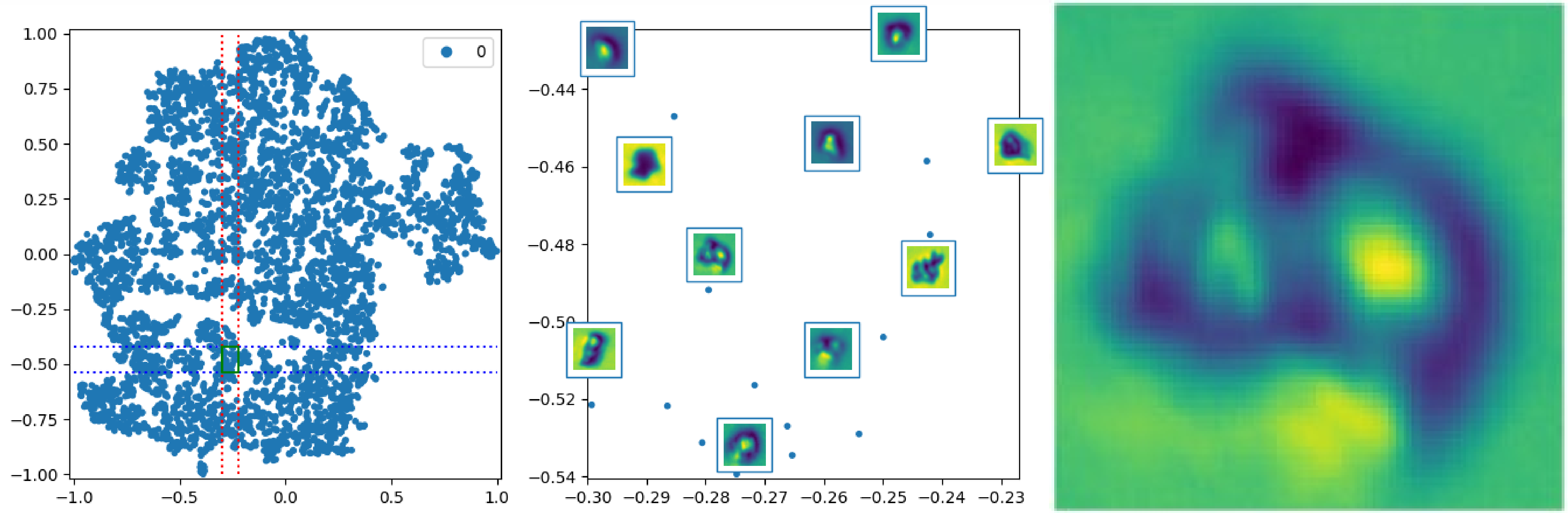




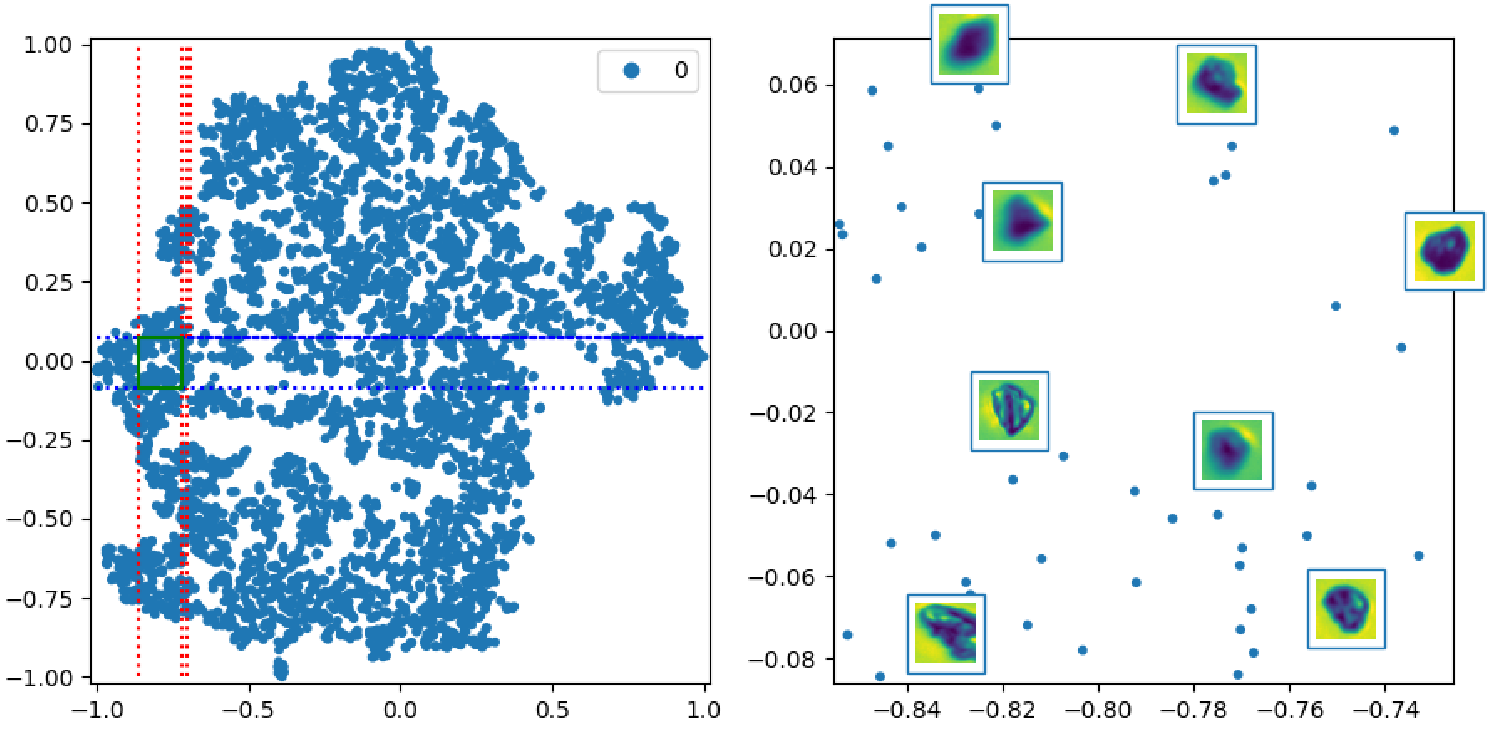
Between area 1 and 3

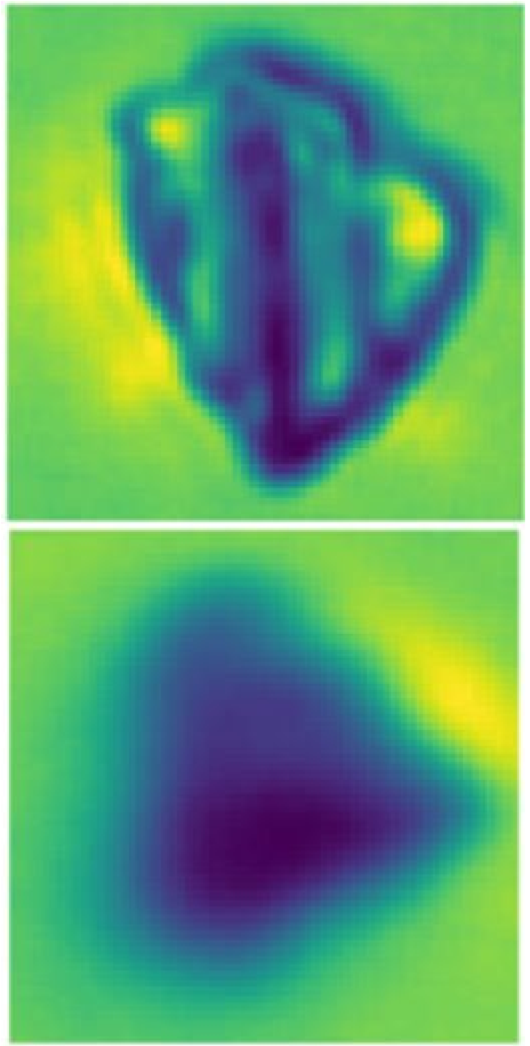


VAE, planar flow Area 4

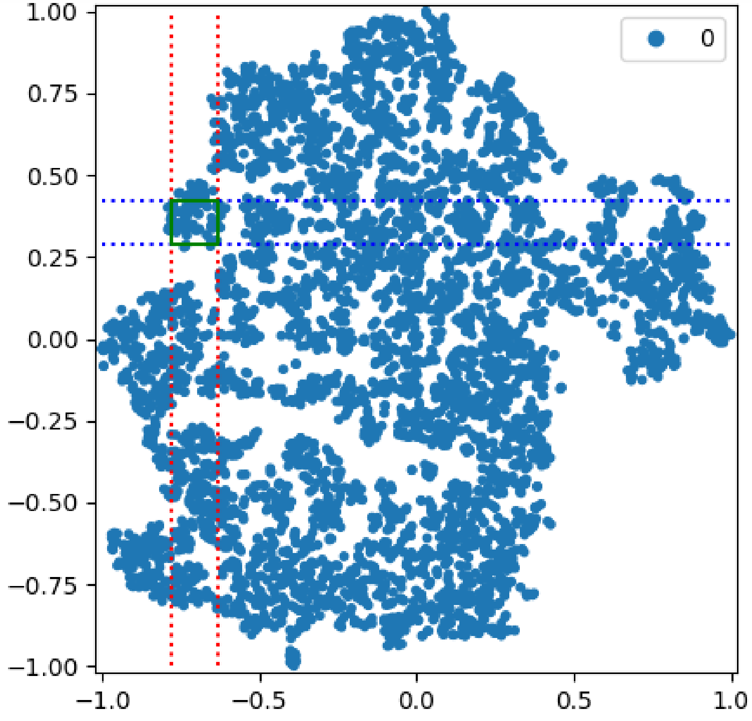


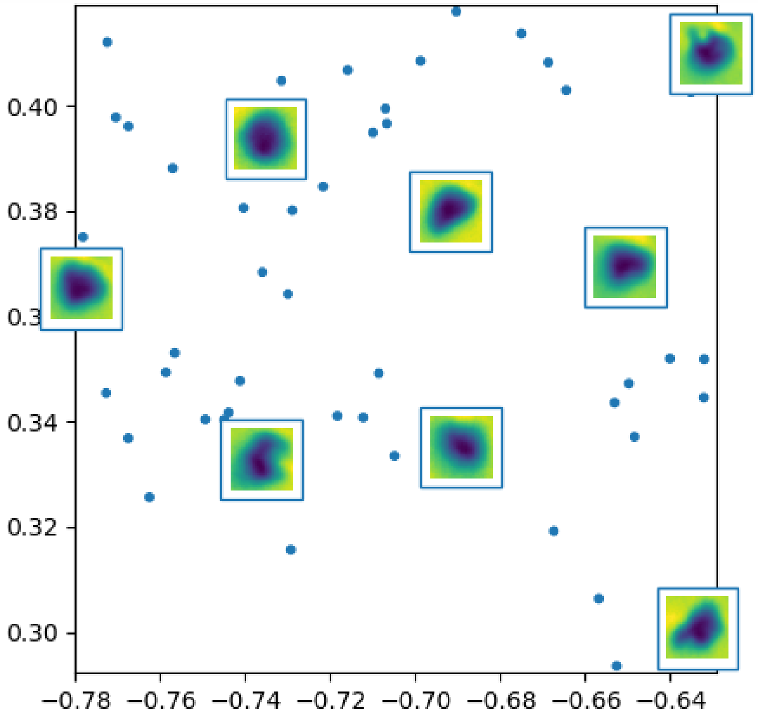
VAE, planar flow Area 5

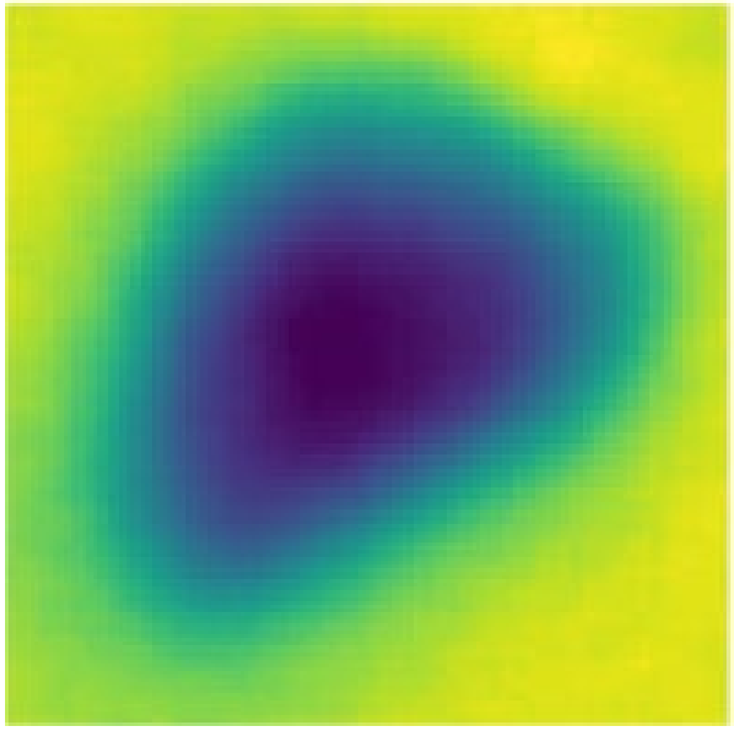




VAE, planar flow Area 6







Peruvian Dataset with Labels obtained through the CNN-Classifier