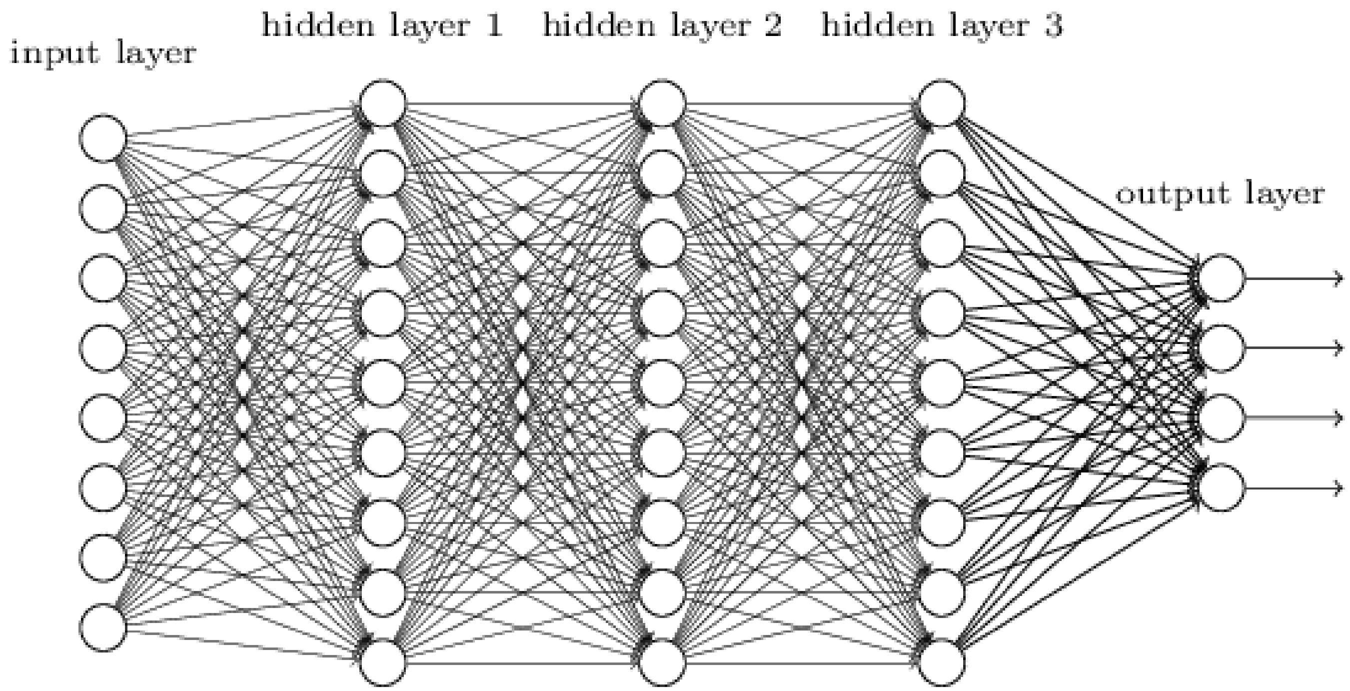
THE UNIVERSITY of TENNESSEE KNOXVILLE

Lecture 28: Deep Convolutional Neural Networks

Sergei V. Kalinin

Putting Neurons Together

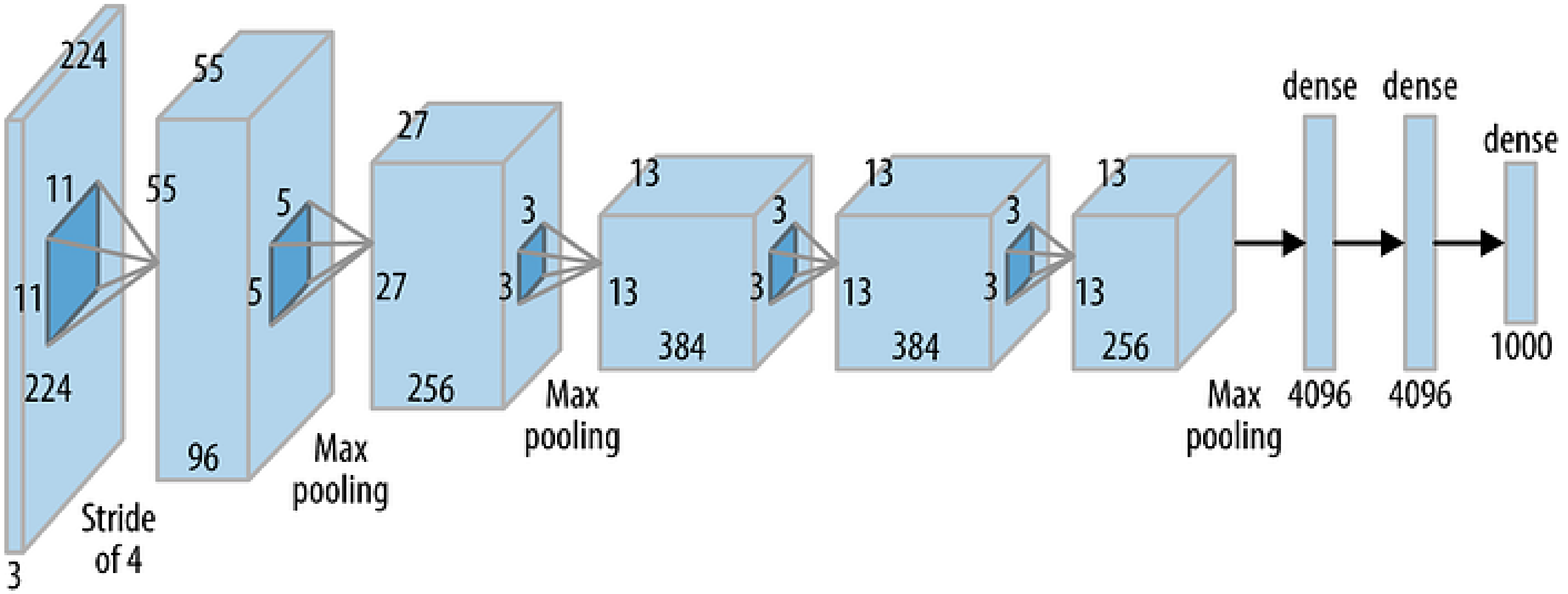


- Linear neurons and activation functions
- Input dimensionality
- Output dimensionality
- Special attention: activation function of the last layer

- Loss function – how different is what we get from what we want (supervised ML)
- Backpropagation – how we adapt the neuron settings given the loss
- Metrics – how we monitor the training/performance

Deep Convolutional Neural Networks

Structure of AlexNet



https://towardsdatascience.com/the-w3h-of-alexnet-vggnet-resnet-and-inception-7baaaecccc96

DCNNs vs. MLP

1. Parameter Efficiency

- Fewer Parameters: CNNs require significantly fewer parameters than FCNs. They use shared weights and convolutional filters, reducing the total number of trainable parameters. This makes CNNs more efficient and less memory-intensive.
- Reduces Overfitting: With fewer parameters, CNNs are less prone to overfitting, especially with image data.

2. Exploitation of Spatial Structure

- Local Connectivity: CNNs exploit the spatial structure of the data by applying convolutional filters that capture local features (like edges, textures) in early layers and more complex features (like patterns or object parts) in deeper layers.
- Preservation of Spatial Relationships: Unlike FCNs that lose spatial relationships by flattening the input, CNNs maintain the spatial hierarchy and relationships between different parts of the input.

3. Translation Invariance

- Robust to Translation: Due to pooling layers and the nature of convolution operations, CNNs are inherently more robust to the translation of input data. This means that if an object shifts in an image, a CNN can still detect it effectively.

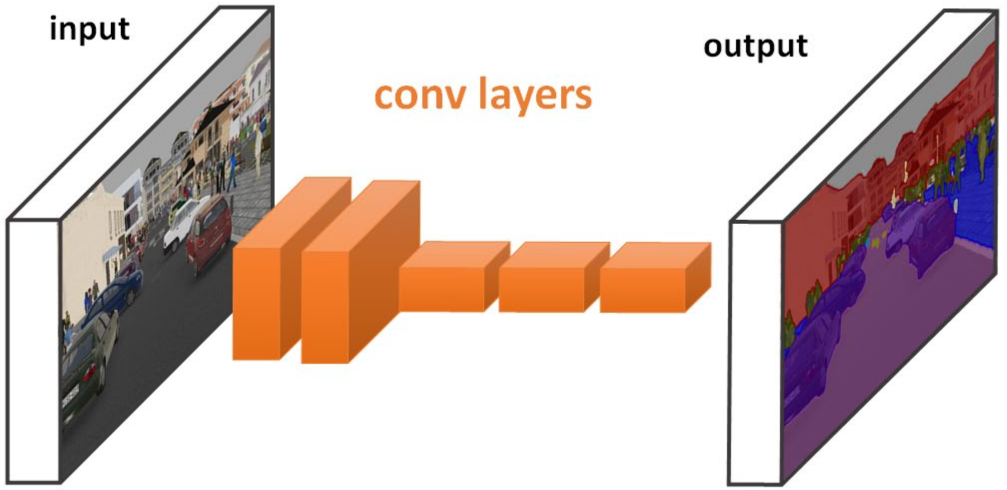
Hyperparameter tuning

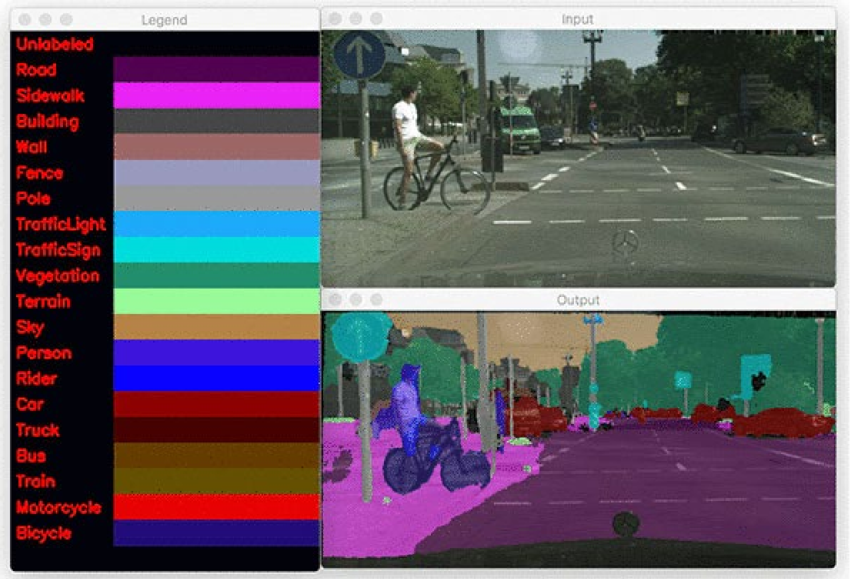
- Hyperparameters are the configuration settings used to structure the neural network and guide the learning process.

- Hyperparameters are not learned from the data but set prior to the training process.
  - Number of Layers and Neurons: Too few may lead to underfitting, while too many can cause overfitting and increased computational cost.
  - Learning Rate: If it's too high, the model may overshoot minima; if too low, training may be slow or get stuck in local minima.
  - Batch size: Influences the accuracy of the gradient estimation and can affect both the speed of convergence and the stability of the learning process.

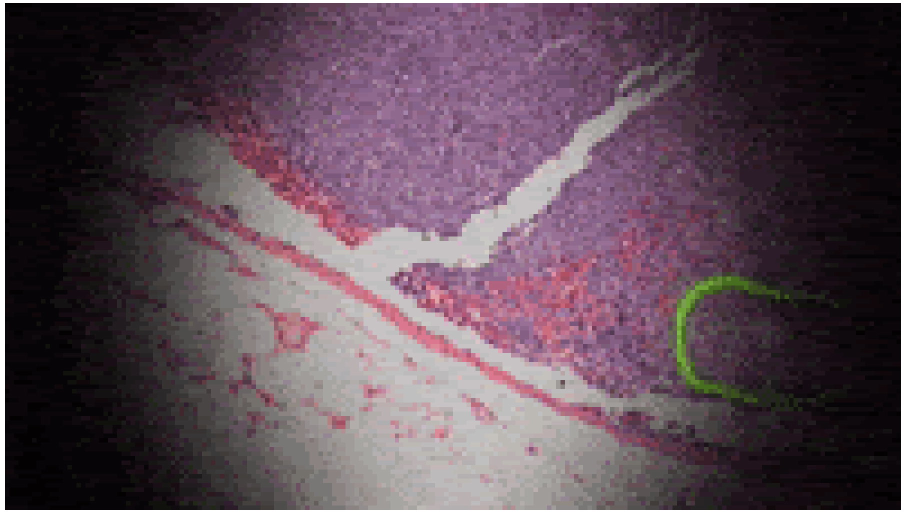
- How do we do it?
  - Grid Search
  - Random Search
  - Bayesian Optimization
  - Leveraging Experience: One can use their knowledge from previous projects to choose a good starting point for hyperparameter settings. This intuition, can significantly reduce the search space and lead to faster convergence on optimal or near-optimal configurations.

Semantic Segmentation





Semantic segmentation of street views *[1], [2]*



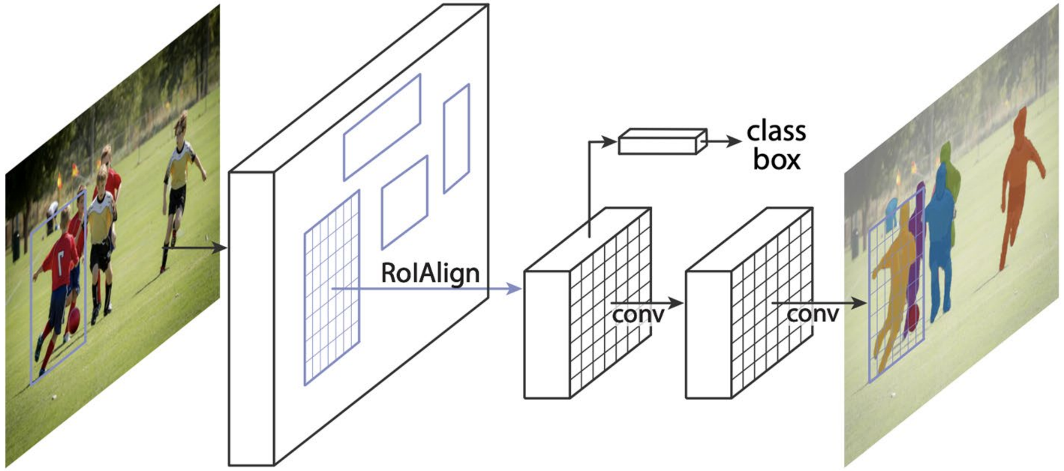
Augmented reality microscope *[3]*

[1] G. Heinrich, “Image Segmentation Using DIGITS 5,” NVIDIA Developer Blog, Nov. 10, 2016. https://developer.nvidia.com/blog/image-segmentation-using-digits-5/ (accessed Feb. 25, 2021).

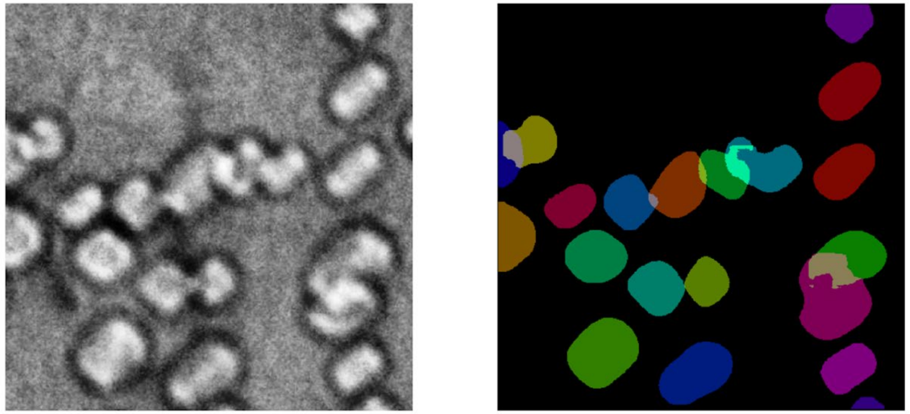
[2] M. Cordts et al., “The Cityscapes Dataset for Semantic Urban Scene Understanding,” in 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, Jun. 2016, pp. 3213–3223, doi: 10.1109/CVPR.2016.350.

[3] P.-H. C. Chen et al., “Microscope 2.0: An Augmented Reality Microscope with Real-time Artificial Intelligence Integration,” arXiv:1812.00825 [cs], Dec. 2018, doi: 10.1038/s41591-019-0539-7.

Instance Segmentation



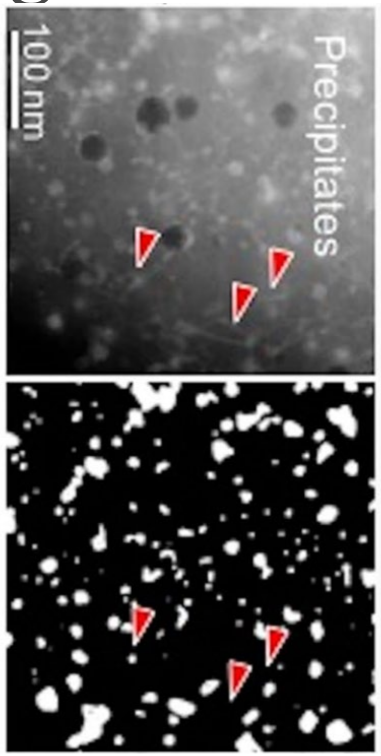
Mask R-CNN



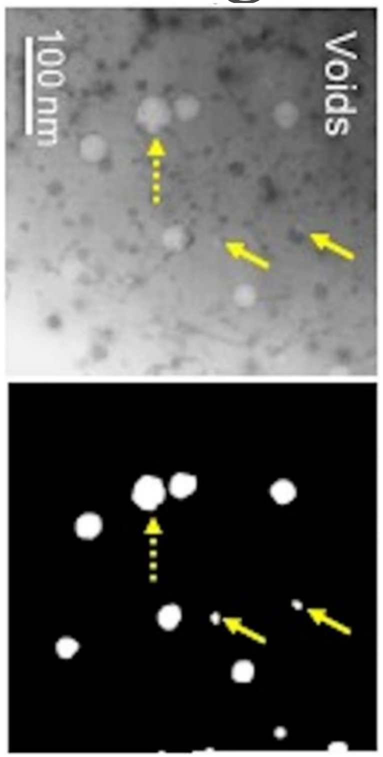
$$Loss = L_{class} + L_{box} + L_{mask}$$

Microstructure Analysis

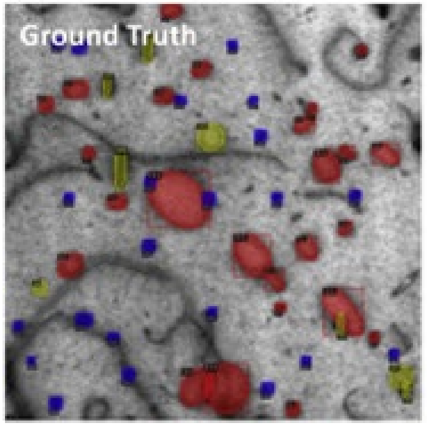
- Segmentation: associating each pixel in an image with a class



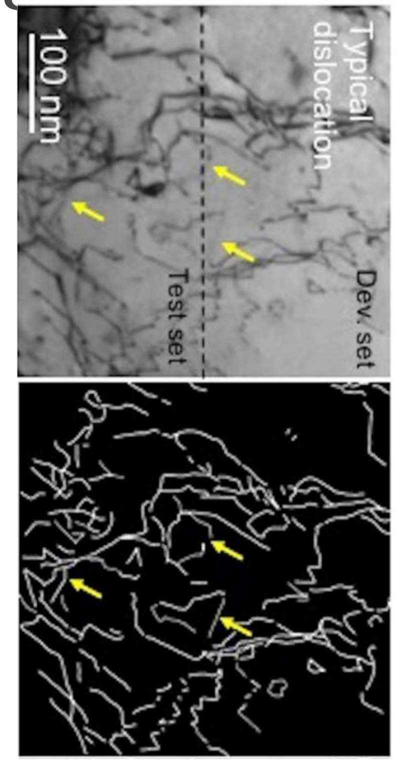
Cavities (bubbles & voids)



Precipitates



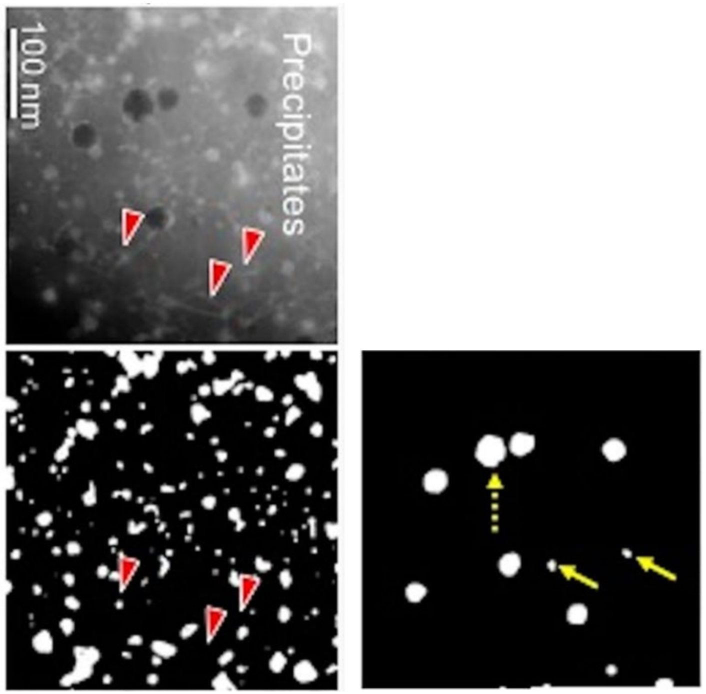
Dislocation loops



Dislocation lines

Adapted from Roberts et al. 2019 10.1038/s41598-019-49105-0, Jacobs et al. 2022 10.1016/j.xcrp.2022.100876

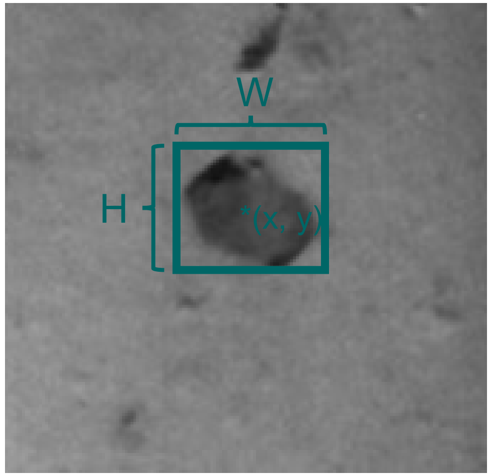
Need Labeled Data!



Precipitates

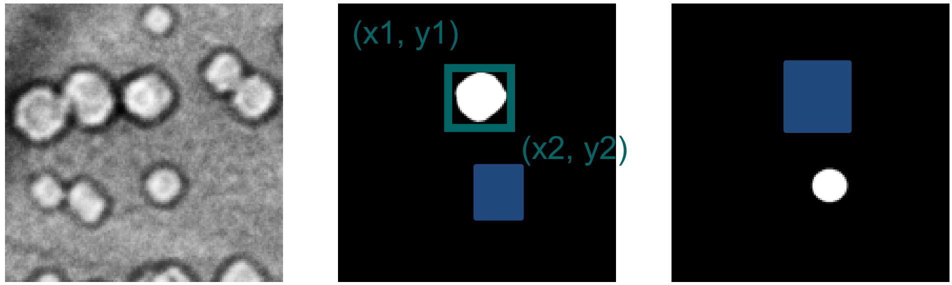
Voids

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



[Class x y W H]

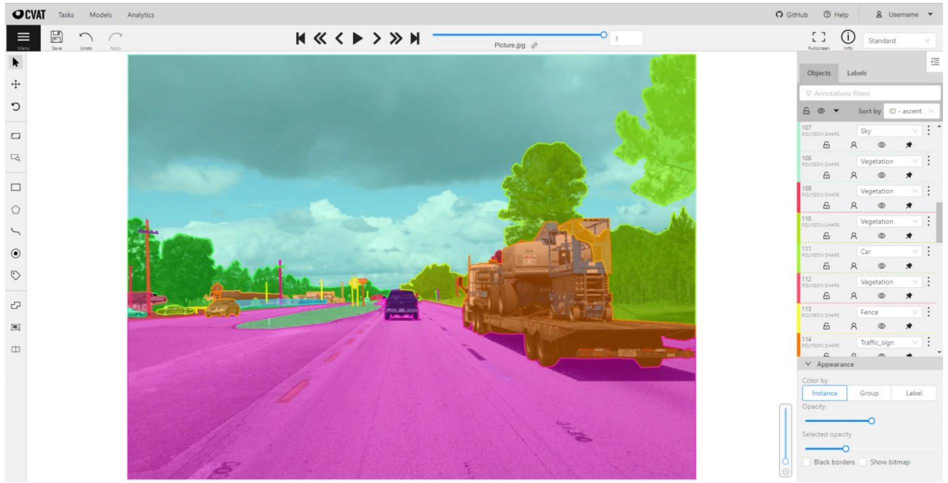
Object identification: bounding box (YOLO)



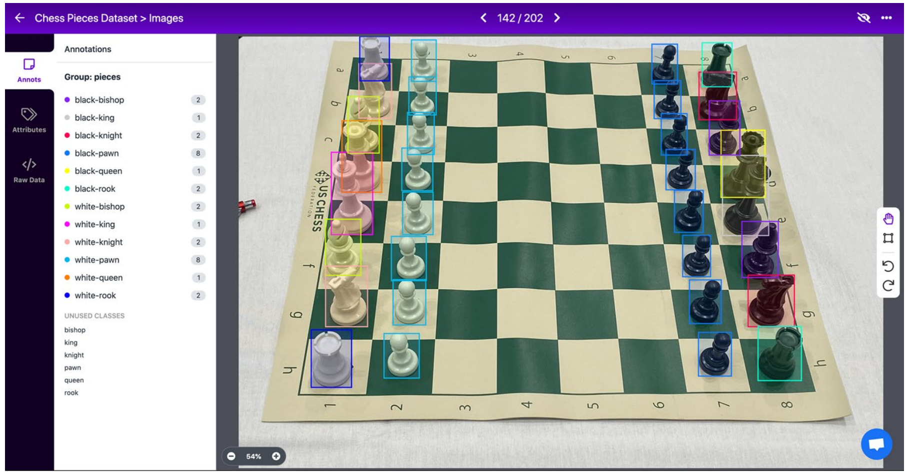
Dictionary{
‘boxes’: [x1 y1 x2 y2]
‘labels’: class
‘masks’: feature mask
}

Instance segmentation: label encoding (Mask R-CNN)

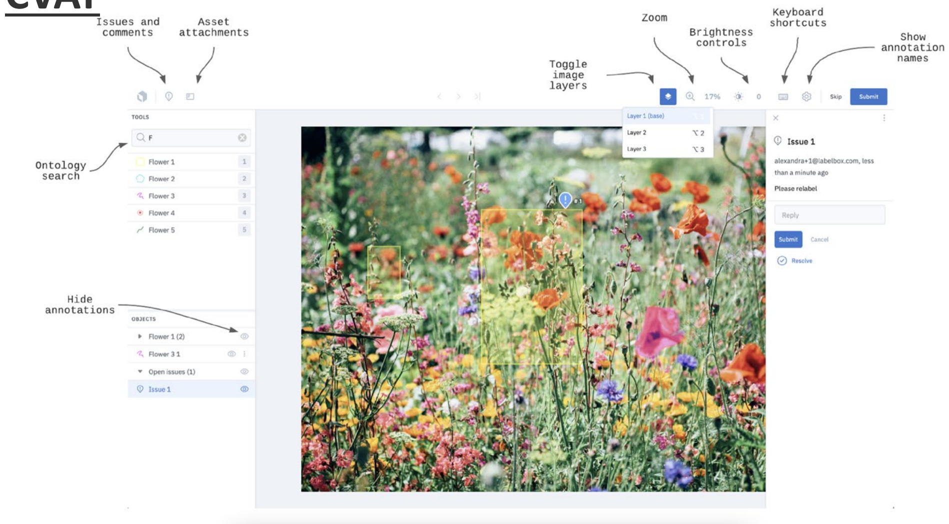
Web-based GUI tools



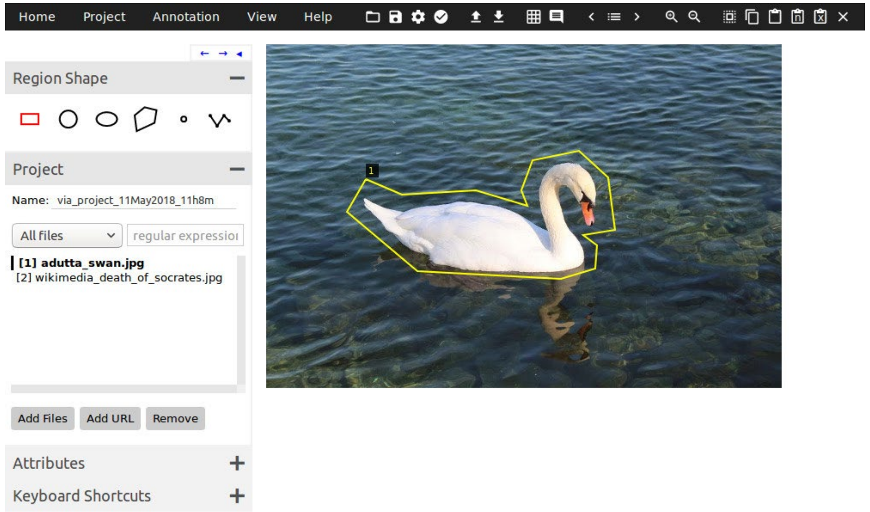
CVAT



Roboflow



Labelbox



VIA

Computer Vision Annotation Tool (CVAT)

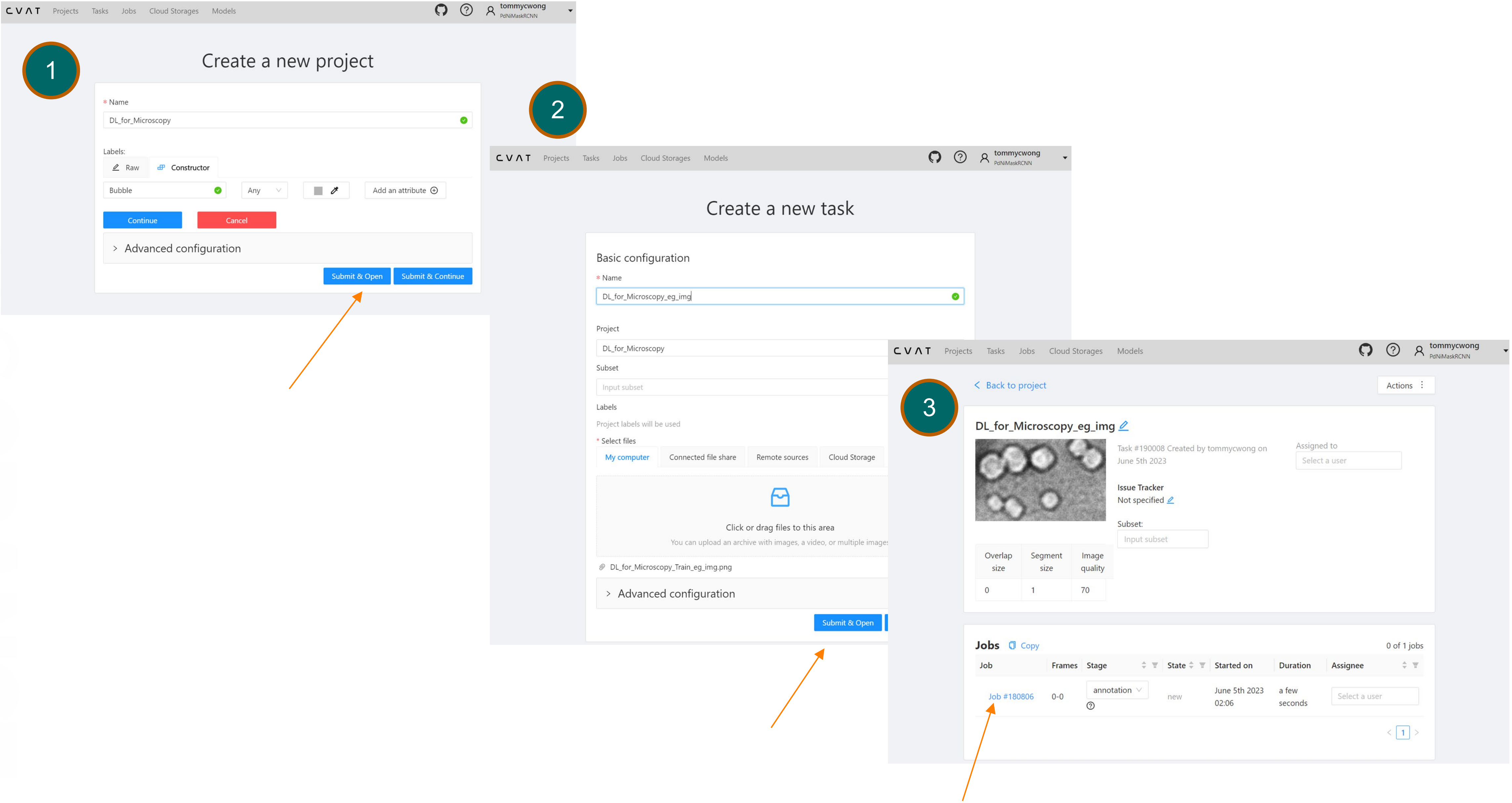
Important: before labeling, ensure all image data have the same dimensions e.g. 1024x1024

Create project → Create tasks → Labeling jobs → Export labels → Parse labels using Python

cvat.ai

Documentation:
github.com/TaSeeMba/cvat/blob/master/cvat/apps/documentation/user_guide.md

Creating Project and Labeling Task



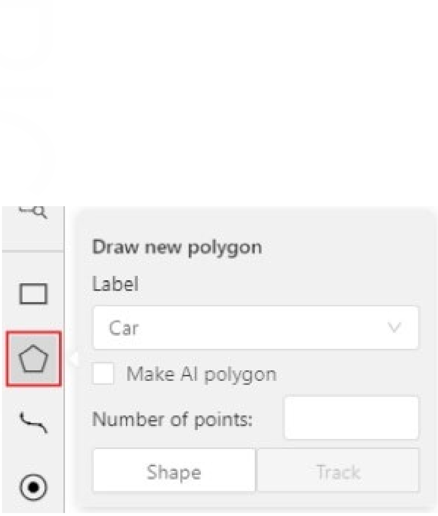
Labeling Using Polygon and Ellipse Tools

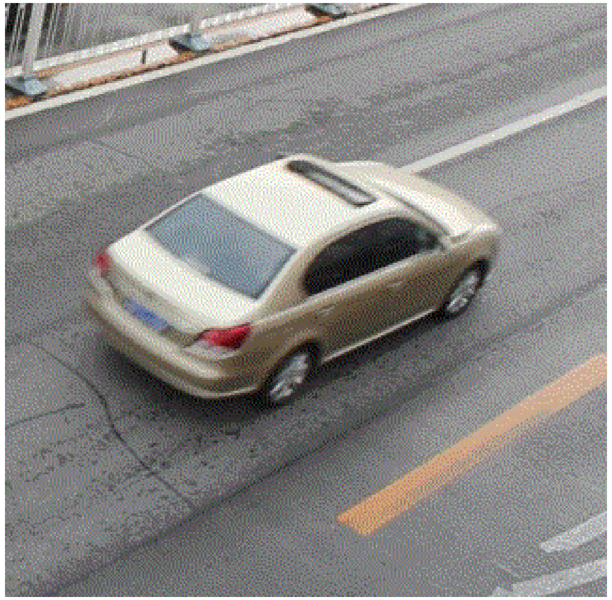
Remember to click **Save**

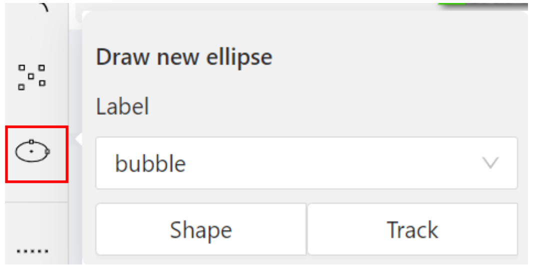
Polygon tool (preferred)

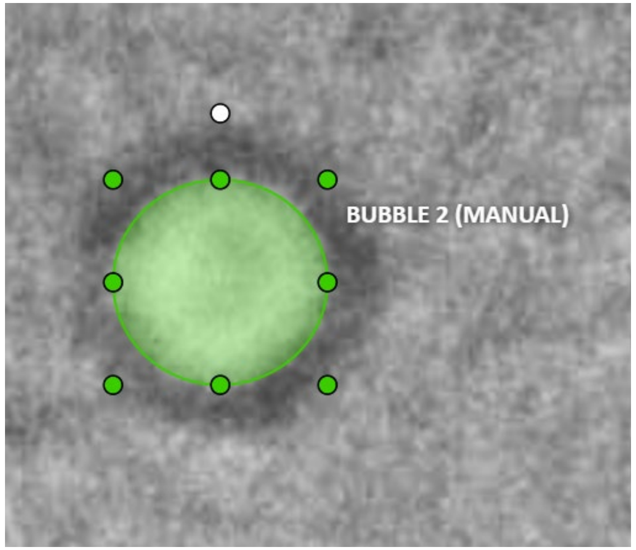
⑩ Hold **Shift** to draw

Ellipse tool



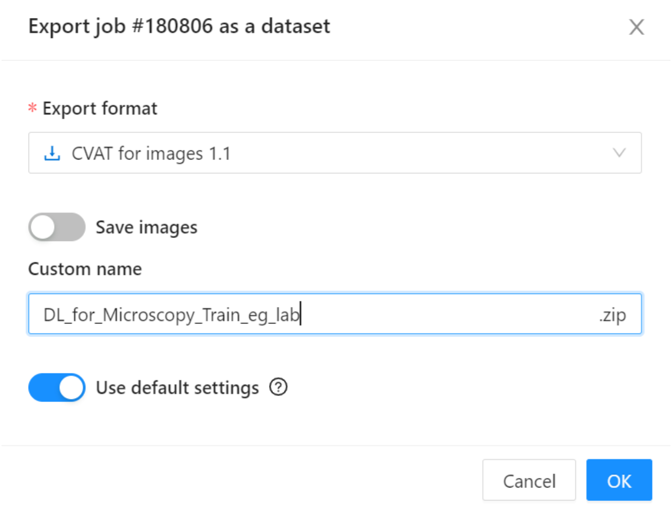






Exporting and Parsing

Export as .xml



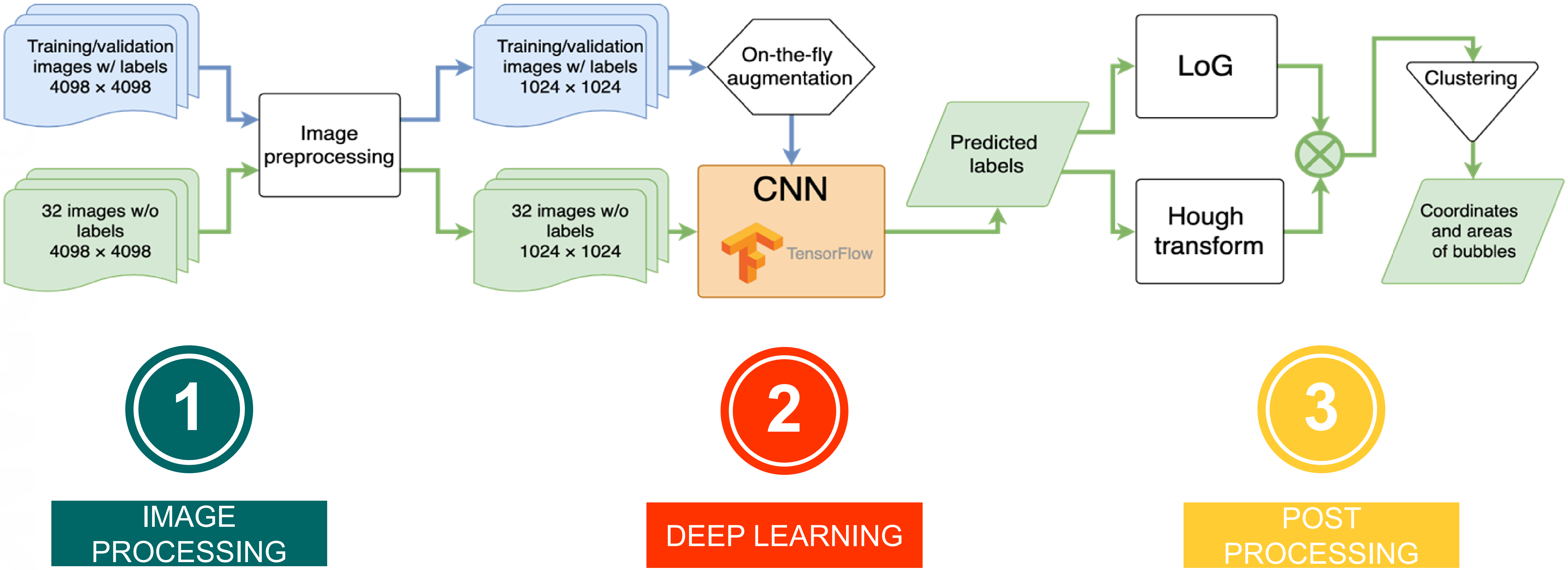
Parsing labels using Python

```
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)
```

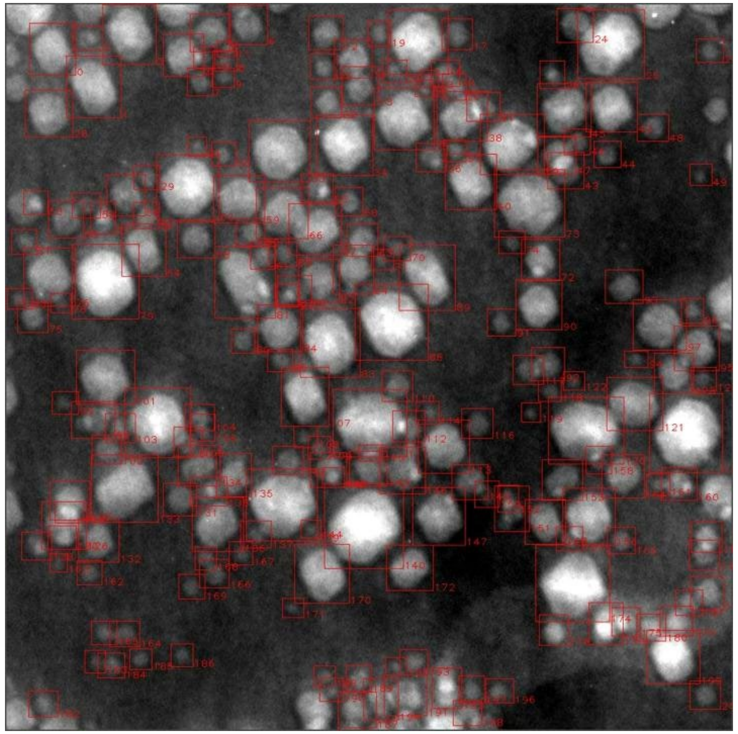
```xml
<image id="0" name="DL_for_Microscopy_Train_eg_img.png" width="512" height="512">
  <ellipse label="Bubble" source="manual" occluded="0" cx="291.95" cy="334.99" rx="32.35" ry="30.94" z_order="0">
  </ellipse>
  <polygon label="Bubble" source="manual" occluded="0" points="282.22,131.08;289.47,135.54;295.61,142.23;300.07,1
  </polygon>
```

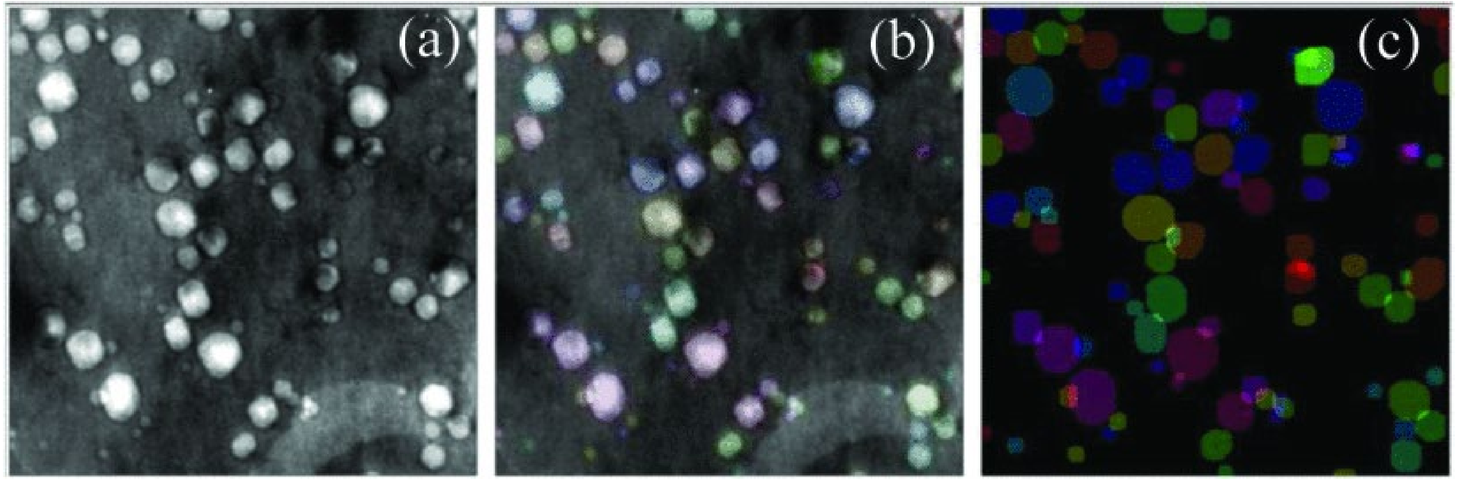
- Features typically should have a convex mask
  - Concave masks are likely occluded convex masks
- Don’t leave holes between multiple overlapping masks
- Keep in mind output files: different parsing scripts needed for .xml, .json, etc.

Training and Classification Workflows

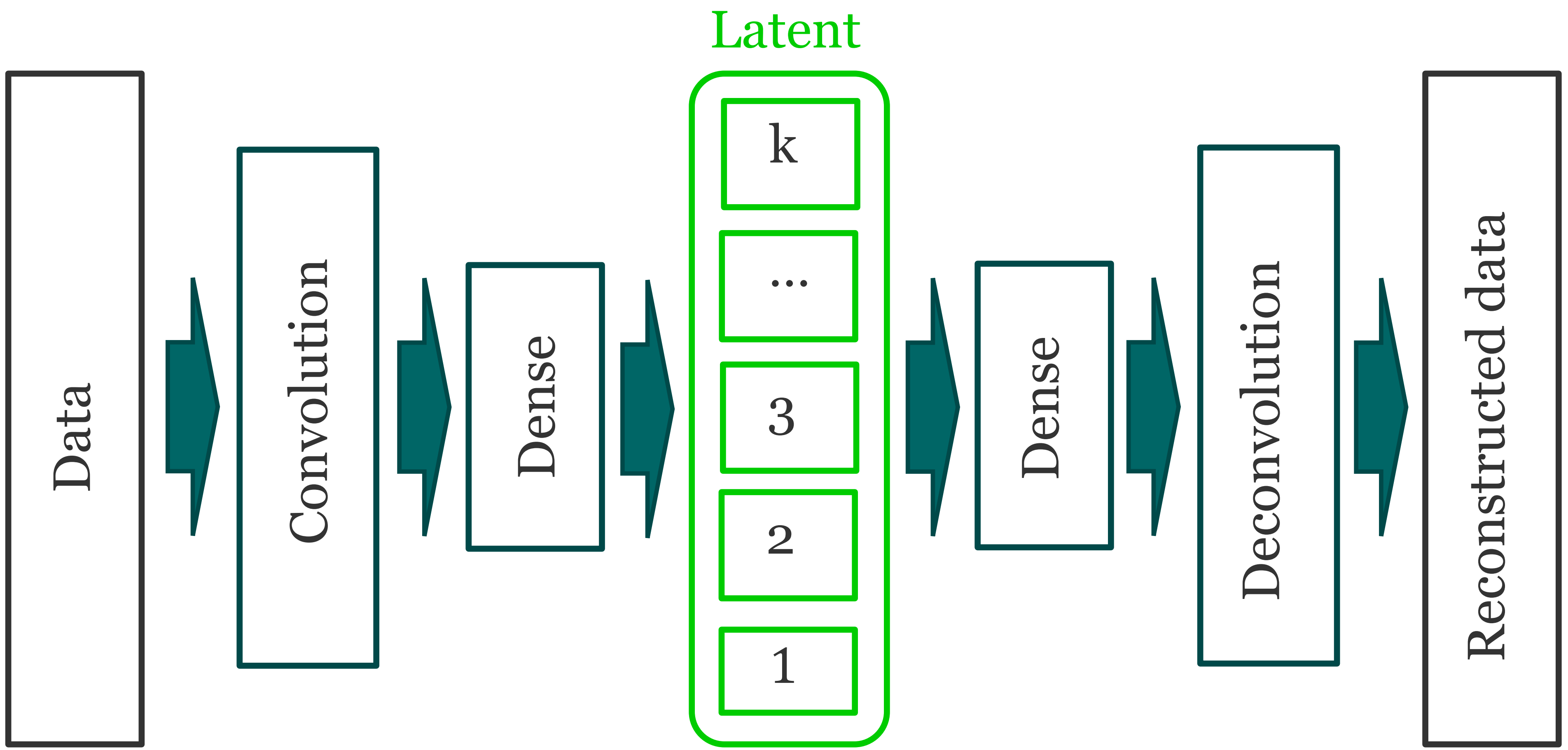


Example: He-atom Bubbles





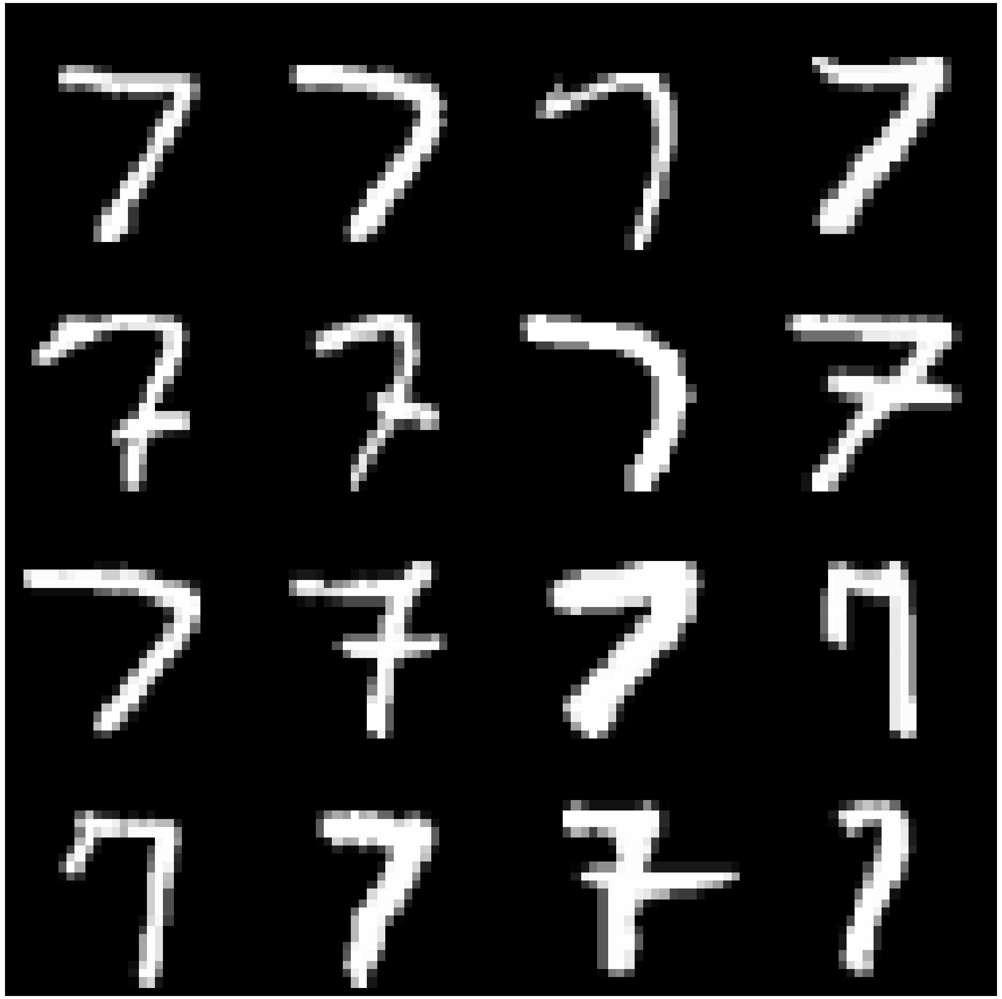
Autoencoders



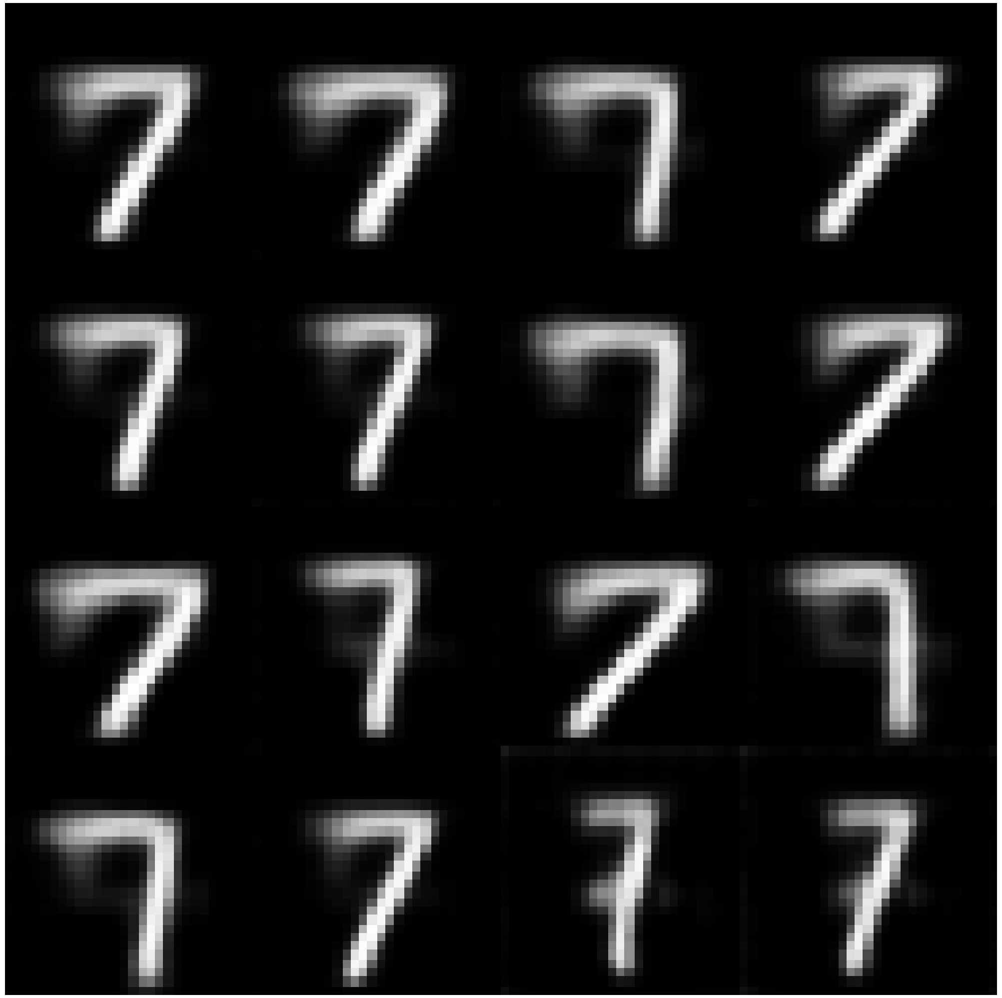
Loss: reconstruction loss

The AE reconstructs data

Input data



Decoded data



Why are AE important?

Geoffrey Hinton

FOLLOW

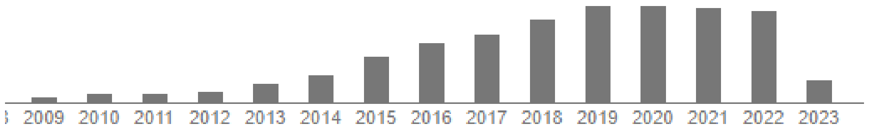
Emeritus Prof. Comp Sci, U.Toronto & Engineering Fellow, Google
Verified email at cs.toronto.edu - Homepage

machine learning psychology artificial intelligence cognitive science computer science

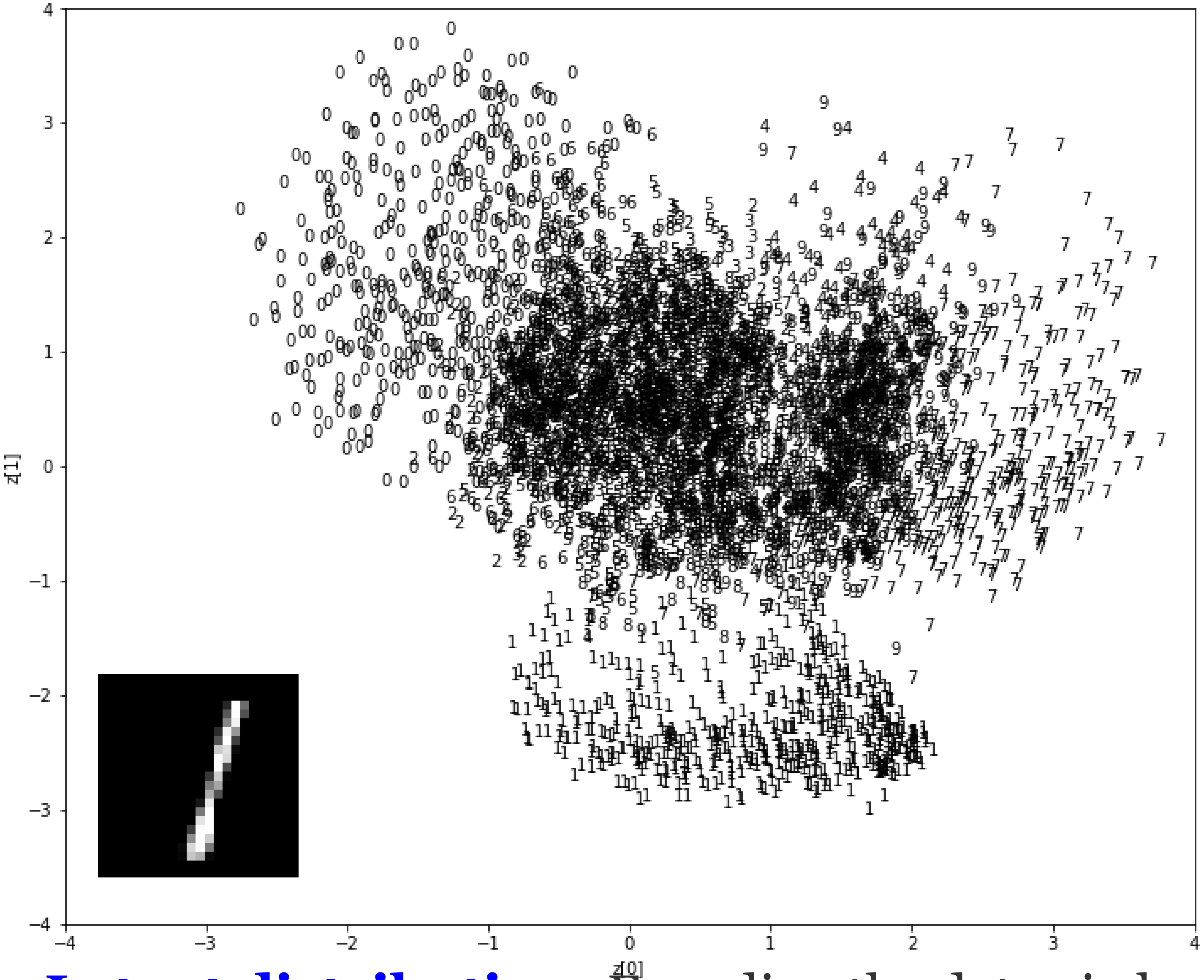
TITLE	CITED BY	YEAR
Imagenet classification with deep convolutional neural networks A Krizhevsky, I Sutskever, GE Hinton Communications of the ACM 60 (6), 84-90	130318	2017
Deep learning Y LeCun, Y Bengio, G Hinton Nature 521 (7553), 436-44	62790	2015
Dropout: a simple way to prevent neural networks from overfitting N Srivastava, G Hinton, A Krizhevsky, I Sutskever, R Salakhutdinov The journal of machine learning research 15 (1), 1929-1958	42078	2014
Visualizing data using t-SNE L van der Maaten, G Hinton Journal of Machine Learning Research 9 (Nov), 2579-2605	35035	2008
Learning representations by back-propagating errors DE Rumelhart, GE Hinton, RJ Williams Nature 323 (6088), 533-536	32239	1986
Learning internal representations by error-propagation DE Rumelhart, GE Hinton, RJ Williams Parallel Distributed Processing: Explorations in the Microstructure of ...	30711	1986
Schemata and sequential thought processes in PDP models. D Rumelhart, P Smolensky, J McClelland, G Hinton Parallel distributed processing: Explorations in the microstructure of ...	28073 *	1986
Learning multiple layers of features from tiny images A Krizhevsky, G Hinton	21876	2009
Rectified linear units improve restricted boltzmann machines V Nair, GE Hinton Proceedings of the 27th international conference on machine learning (ICML ...	21050	2010
Reducing the dimensionality of data with neural networks GE Hinton, RR Salakhutdinov Science 313 (5786), 504-507	19930	2006

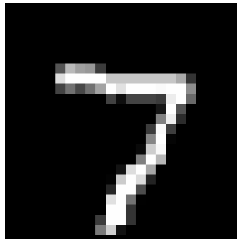
Reducing the dimensionality of data with neural networks

Authors	Geoffrey E Hinton, Ruslan R Salakhutdinov
Publication date	2006/7/28
Journal	Science
Volume	313
Issue	5786
Pages	504-507
Publisher	American Association for the Advancement of Science
Description	High-dimensional data can be converted to low-dimensional codes by training a multilayer neural network with a small central layer to reconstruct high-dimensional input vectors. Gradient descent can be used for fine-tuning the weights in such “autoencoder” networks, but this works well only if the initial weights are close to a good solution. We describe an effective way of initializing the weights that allows deep autoencoder networks to learn low-dimensional codes that work much better than principal components analysis as a tool to reduce the dimensionality of data.
Total citations	Cited by 19930



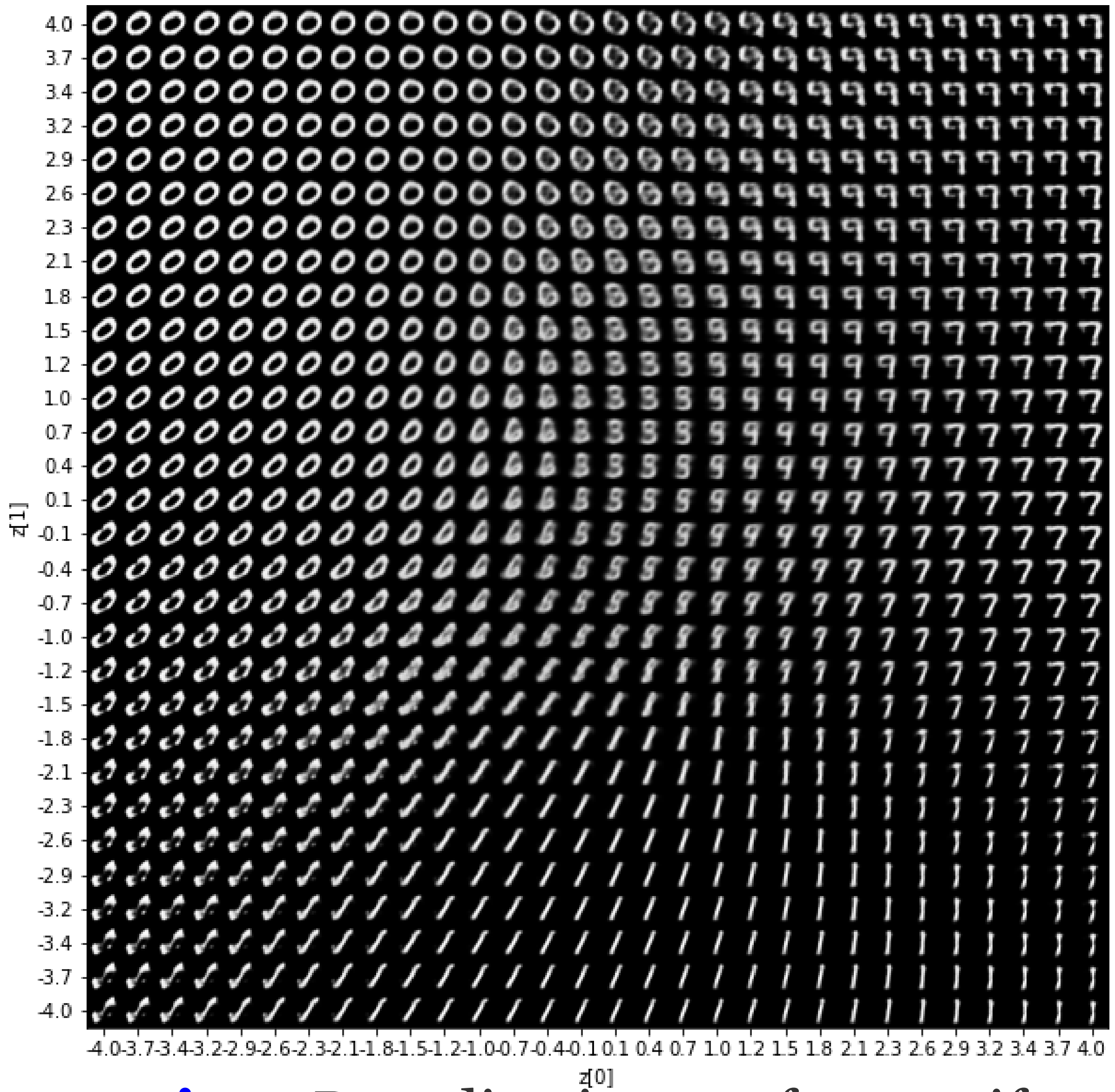
Encoding: Image → Latent Space





Latent distribution: Encoding the data via low dimensional vector

Decoding: Latent Space → Image



Latent representation: Decoding images from uniform grid in latent space

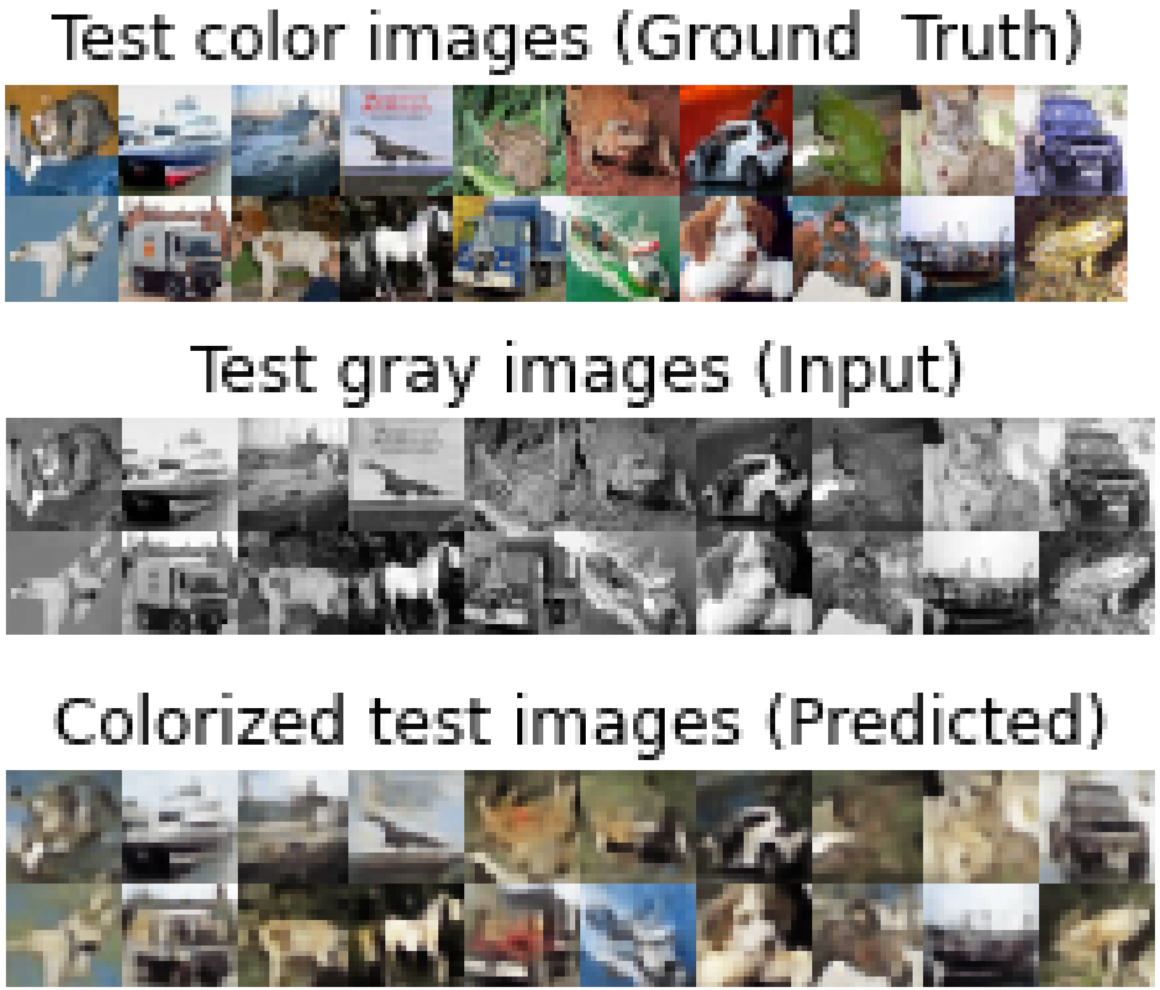
Image Reconstruction

Image Reconstruction

Test color images (Ground Truth)

Colorized test images (Predicted)

Image Reconstruction



- **Training:** pairs of the grayscale and color images
- **Application:** new grayscale images (from the same distribution)
- **Concern:** has to be from the same distribution

Image Denoising

Ground truth

Noisy input

Reconstruction

- **Training:** pairs of the high-noise and low-noise images
- **Application:** new high noise images (from the same distribution)
- **Concern:** has to be from the same distribution

Variational Autoencoders

Diederik P. Kingma

Other names

Research Scientist, Google Brain
Verified email at google.com - Homepage

Machine Learning   Deep Learning   Neural Networks   Generative Models   Variational Inference

FOLLOW

TITLE	CITED BY	YEAR
Adam: A Method for Stochastic Optimization DP Kingma, J Ba Proceedings of the 3rd International Conference on Learning Representations ...	141306	2014
Auto-Encoding Variational Bayes DP Kingma, M Welling arXiv preprint arXiv:1312.6114	26540	2013
Semi-Supervised Learning with Deep Generative Models DP Kingma, S Mohamed, DJ Rezende, M Welling Advances in Neural Information Processing Systems, 3581-3589	2946	2014

- Variational Autoencoder (VAE): uses “reparameterization trick” to sample from the latent space
- Can be used for same tasks as AE
- Have a much better-behaved latent space: disentanglement of the representations

VAE Training

Latent manifold → Image space

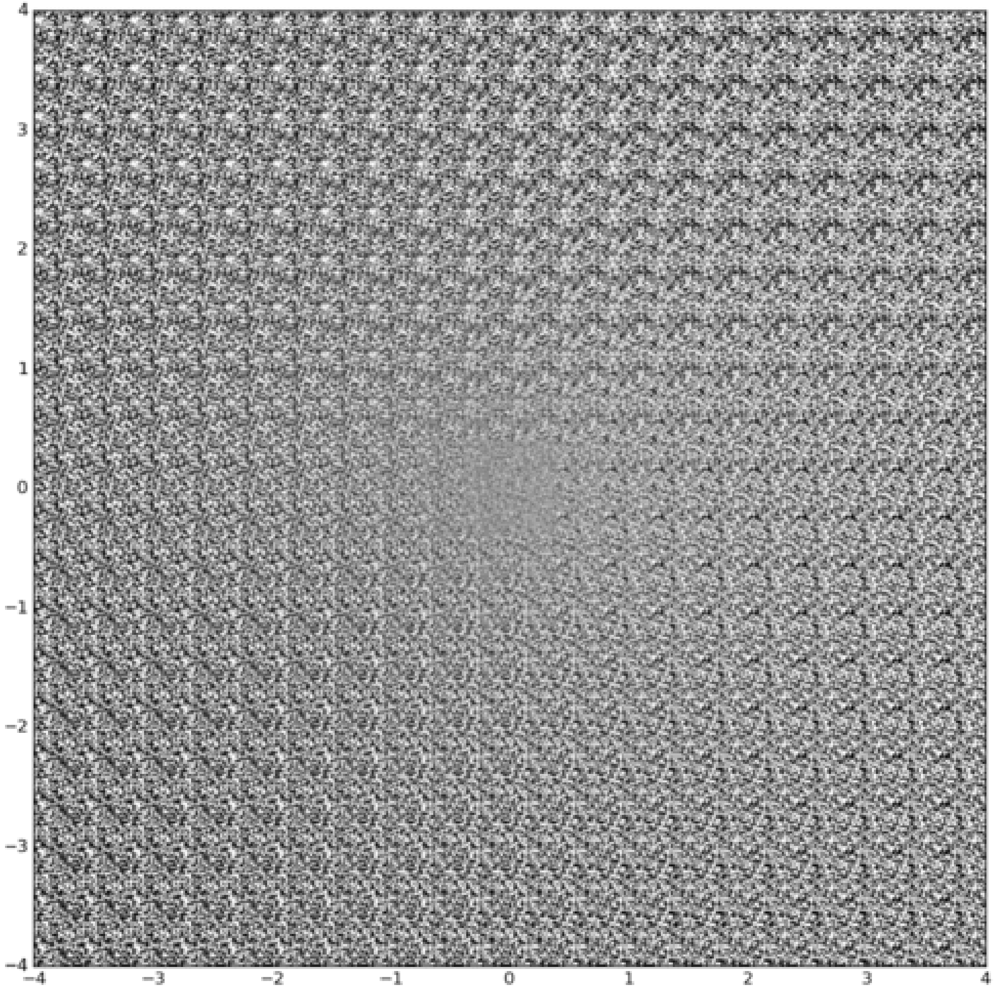
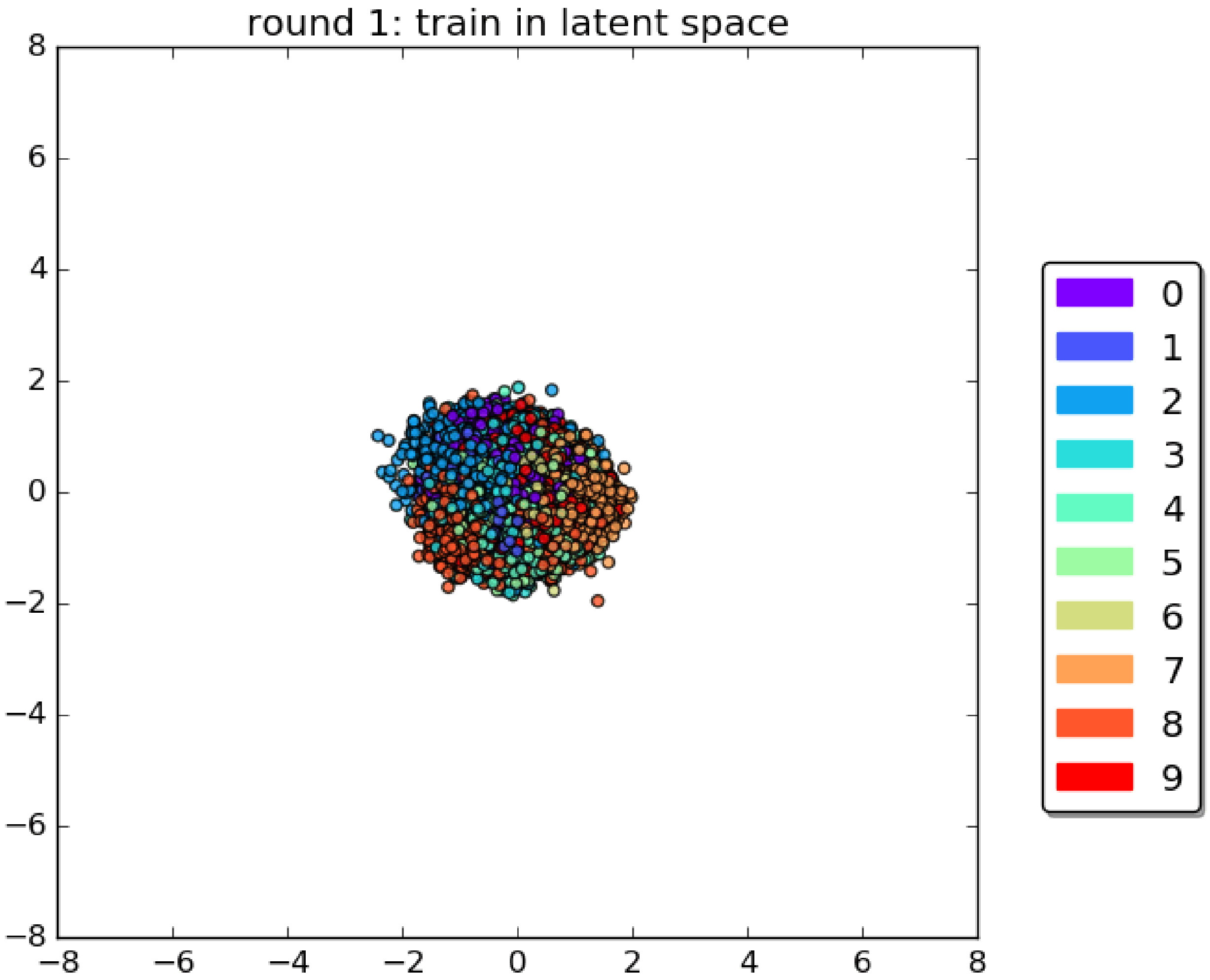
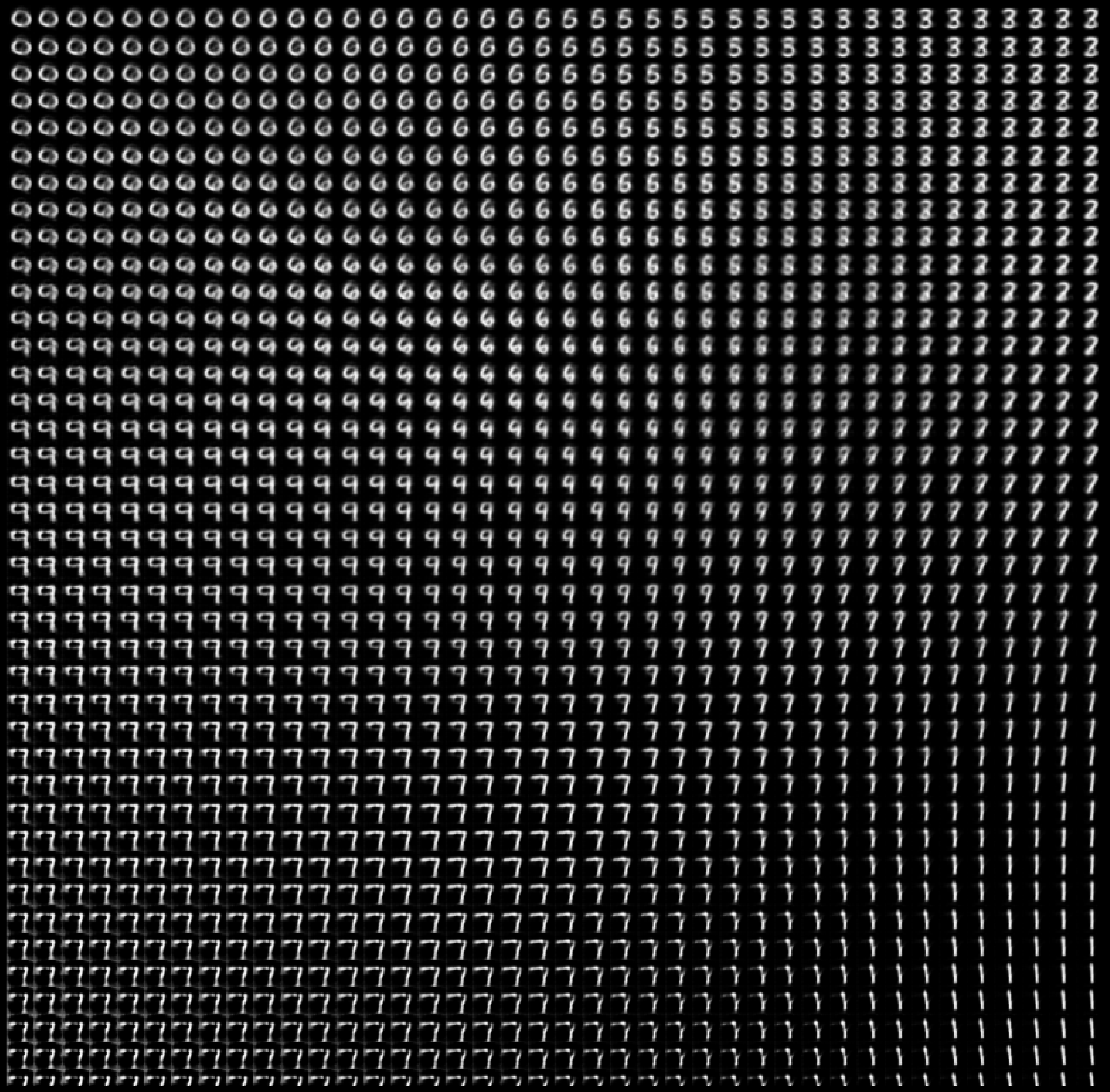


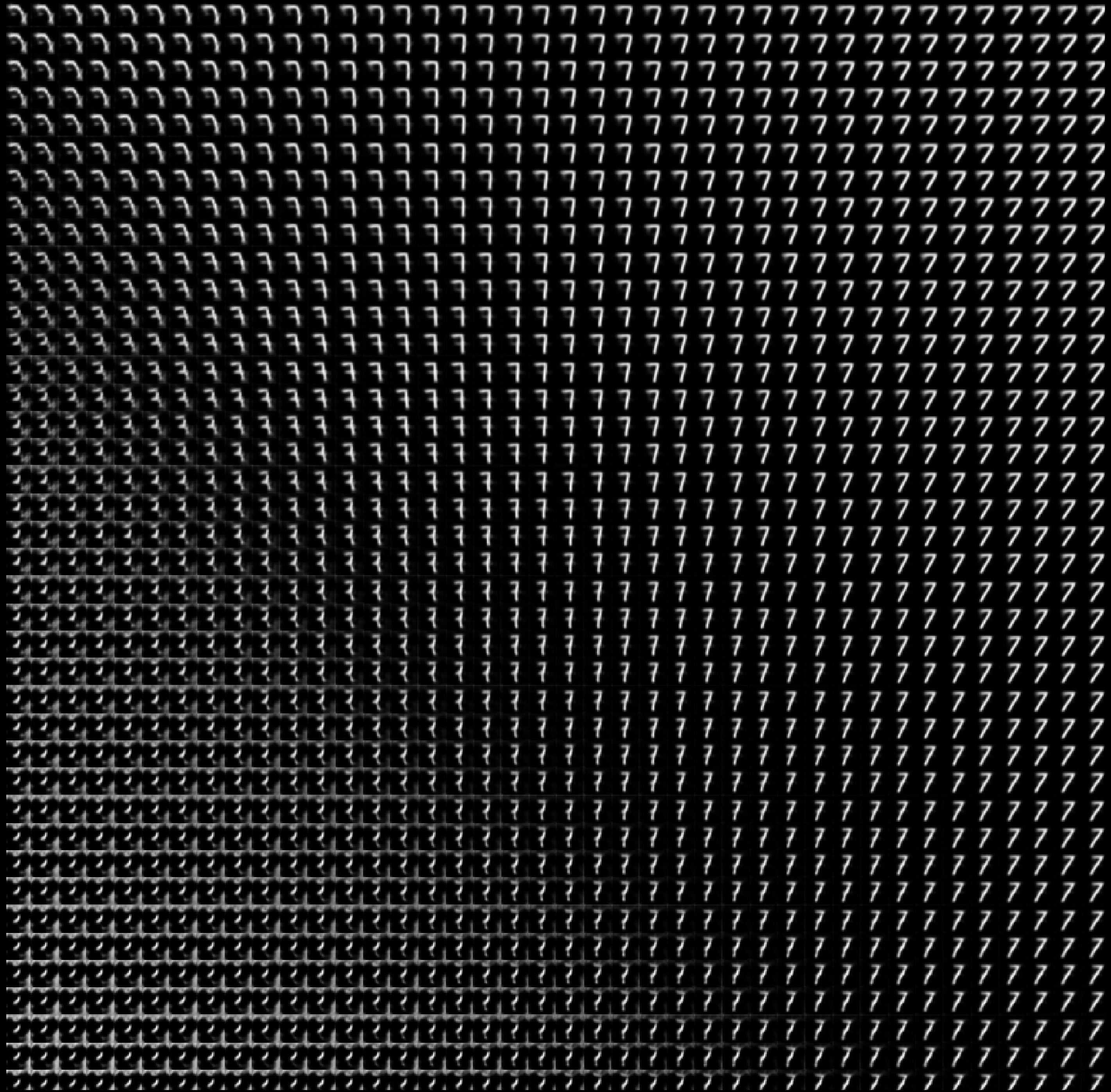
Image space → Latent space



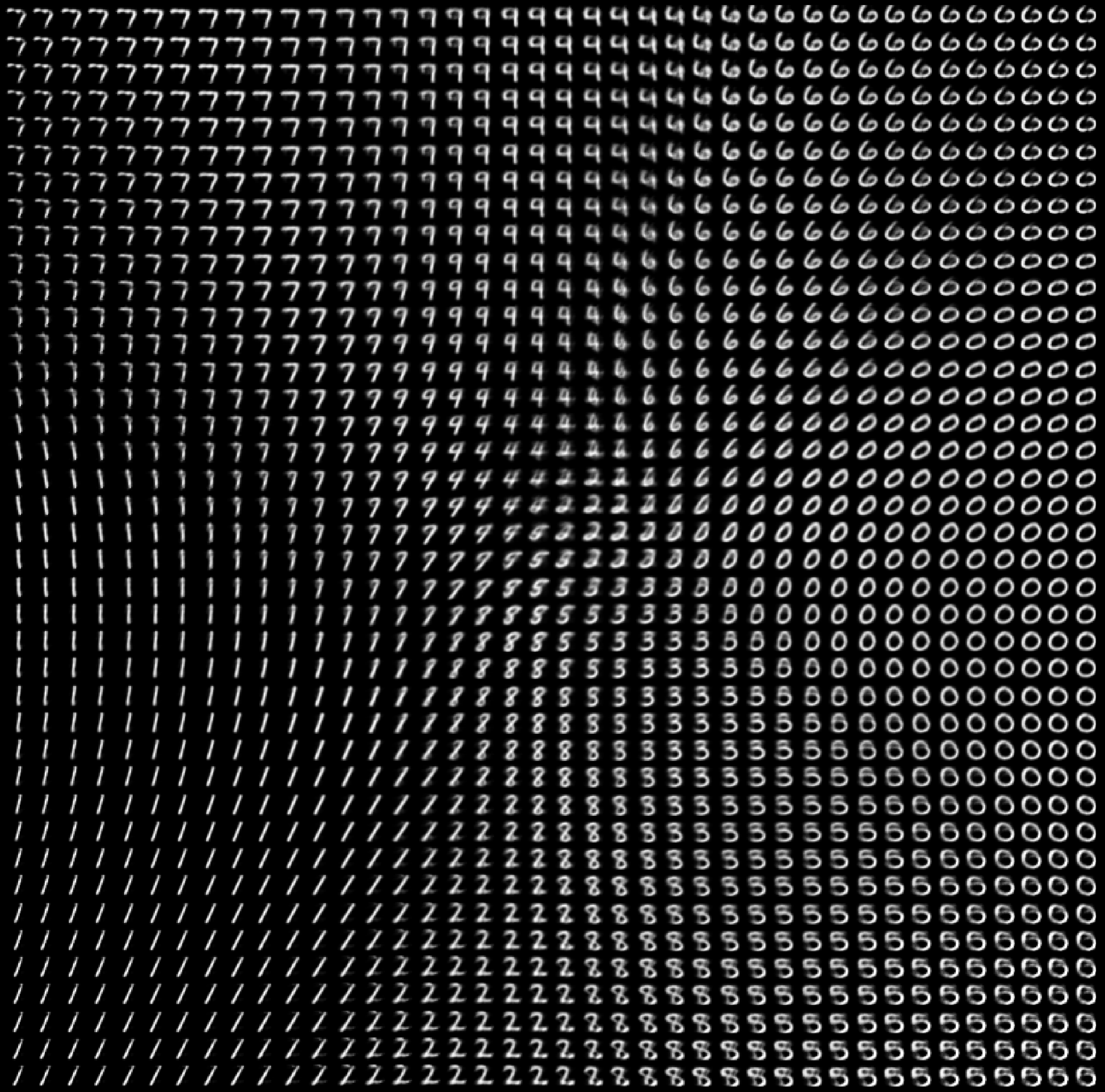
Autoencoder latent representation



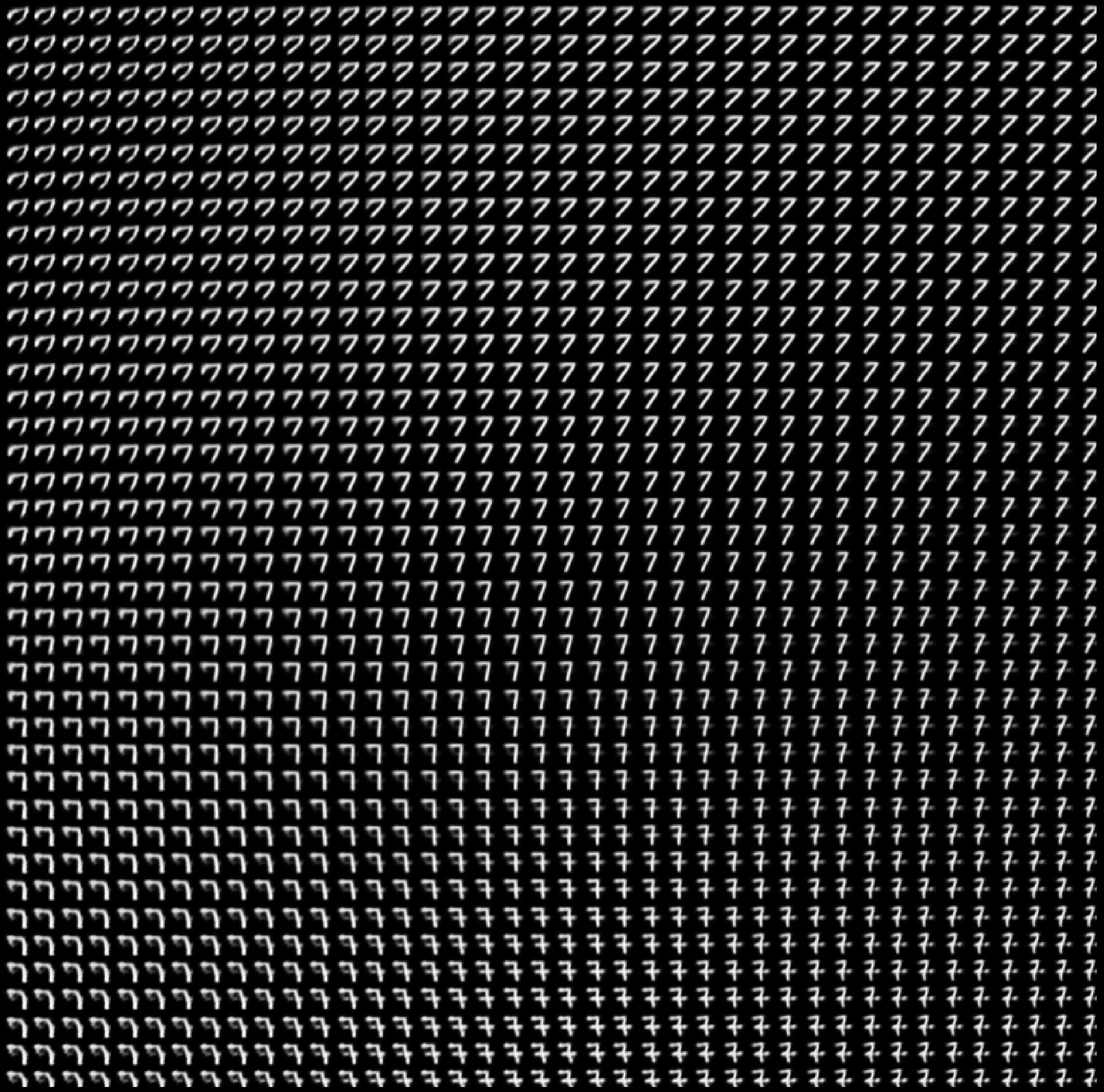
Autoencoder latent representation (digit 7)



VAE latent representation



VAE latent representation (digit 7)



VAE latent representation (digit 8)