Lecture 3

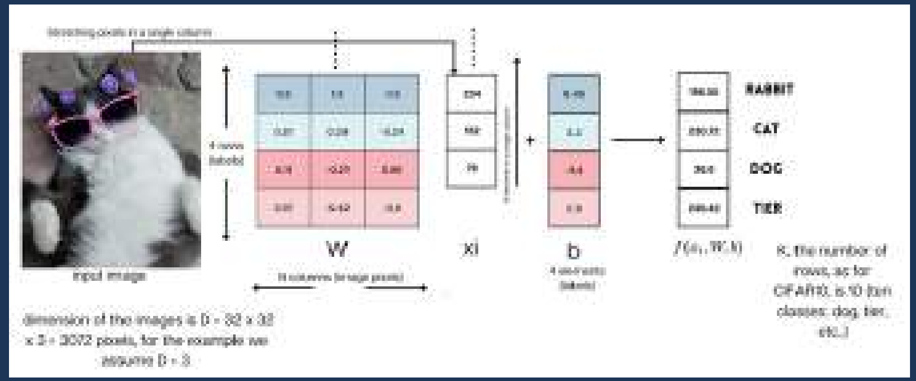
NEURAL NETWORKS AND THEIR APPLICATIONS

Claudia Acquistapace
Istitute for Geophysics and Meteorology
University of Cologne
email: **cacquist@uni-koeln.de**

UNIVERSITY
OF COLOGNE

Recap from last week

1. The problem of **image classification** or… assigning a label to an image with a computer

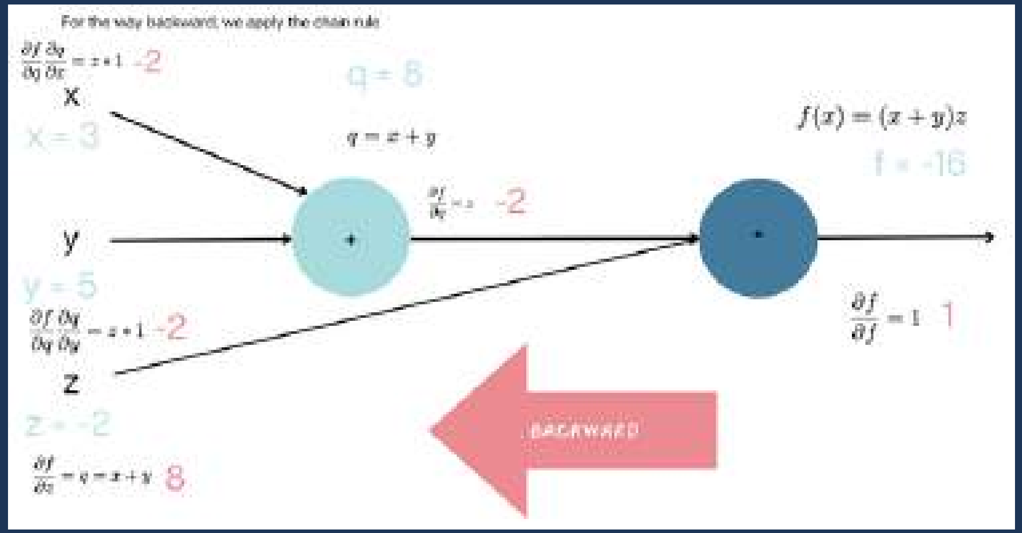


dimension of the images is D = 32 x 32 x 3 = 3072 pixels, for the example we assume D = 3

K, the number of rows, as for CIFAR10, is 10 (ten classes: dog, tier, etc.)

2. **The simplest data driven approach**: a linear classifier and a loss function

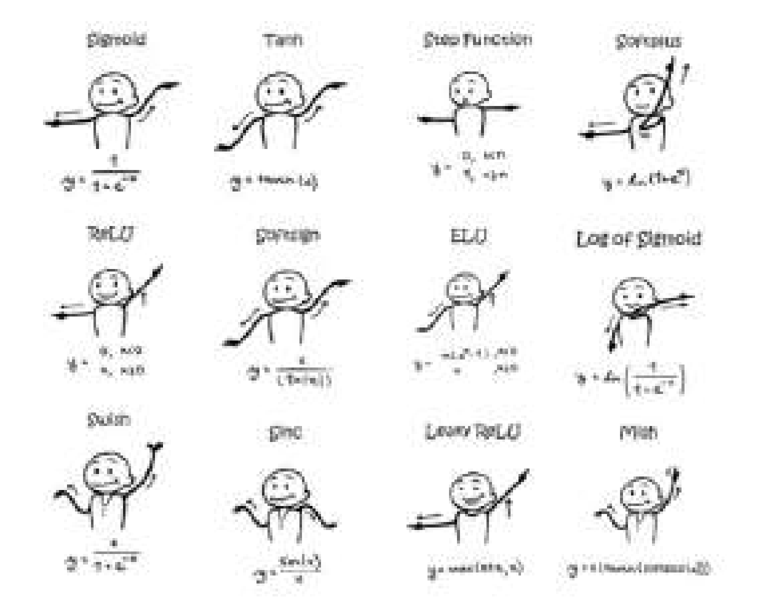
3. **Learning process via optimization** (and the various processes behind it)



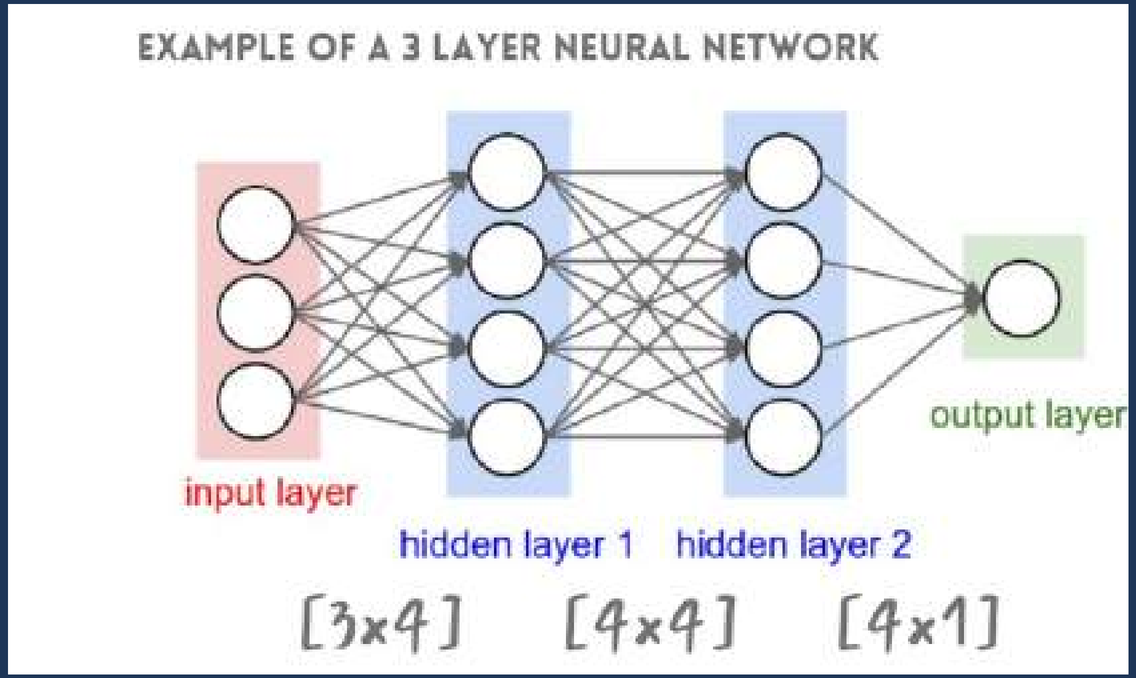
For the way backward, we apply the chain rule

$\frac{\partial f}{\partial q}\frac{\partial q}{\partial x} = z \cdot 1$ -2
x
x = 3
q = 8
$q = x + y$
$f(x) = (x + y)z$
f = -16
$\frac{\partial f}{\partial q} = z$ -2
y
y = 5
$\frac{\partial f}{\partial q}\frac{\partial q}{\partial y} = z \cdot 1$ -2
z
z = -2
$\frac{\partial f}{\partial z} = q = x + y$ 8
$\frac{\partial f}{\partial f} = 1$ 1
BACKWARD

4. **Activation** functions and multilayer perceptron

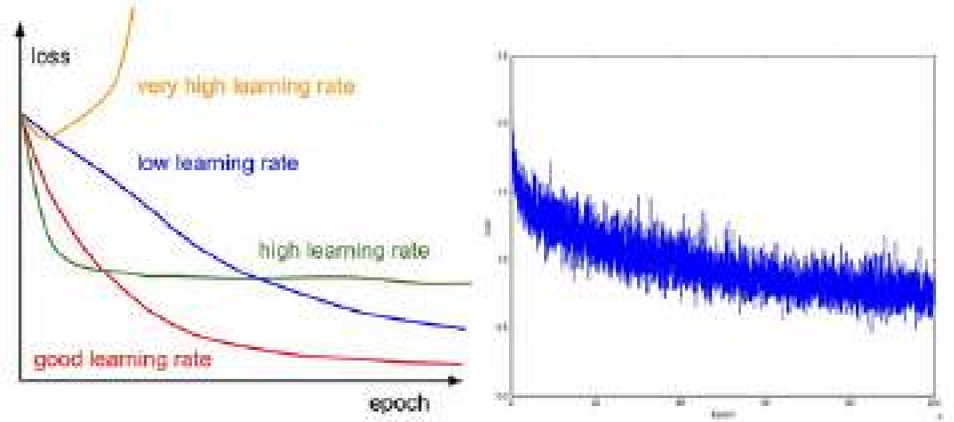


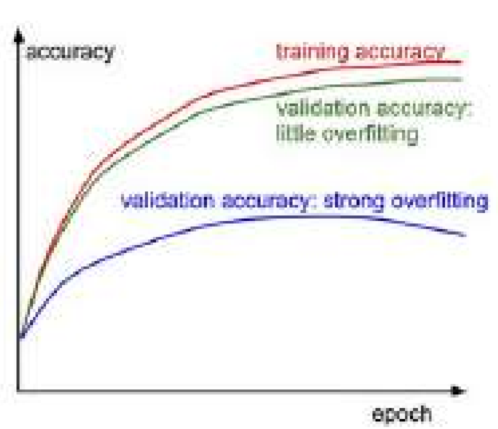
5. **Artificial neural networks**



EXAMPLE OF A 3 LAYER NEURAL NETWORK

6. **Monitoring** the learning process





Topics for today

1. Convolutional neural networks (CNN)
2. Visualizing CNNs and examples
3. Generative models: auto encoders
4. Applications of CNNs: classifying clouds
5. Ways of learning
6. Cloud classification approaches: an overview

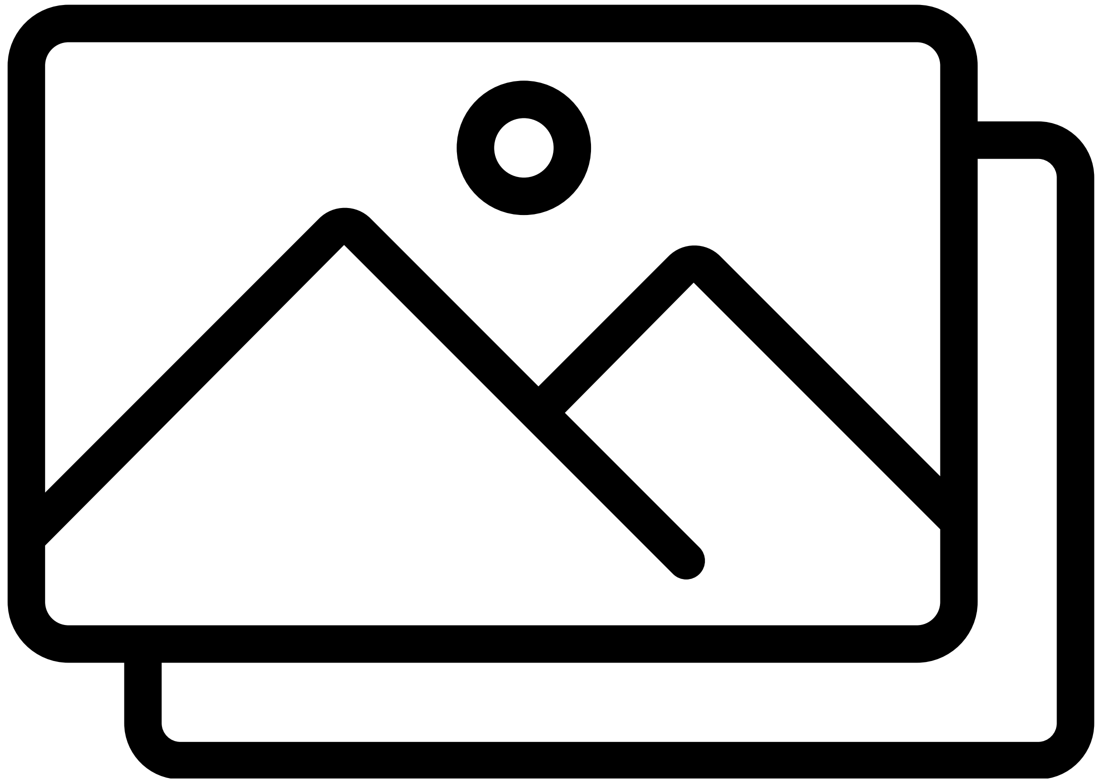
1

Convolutional neural networks (CNN)

Neural network and images

How do neural network scale with images?

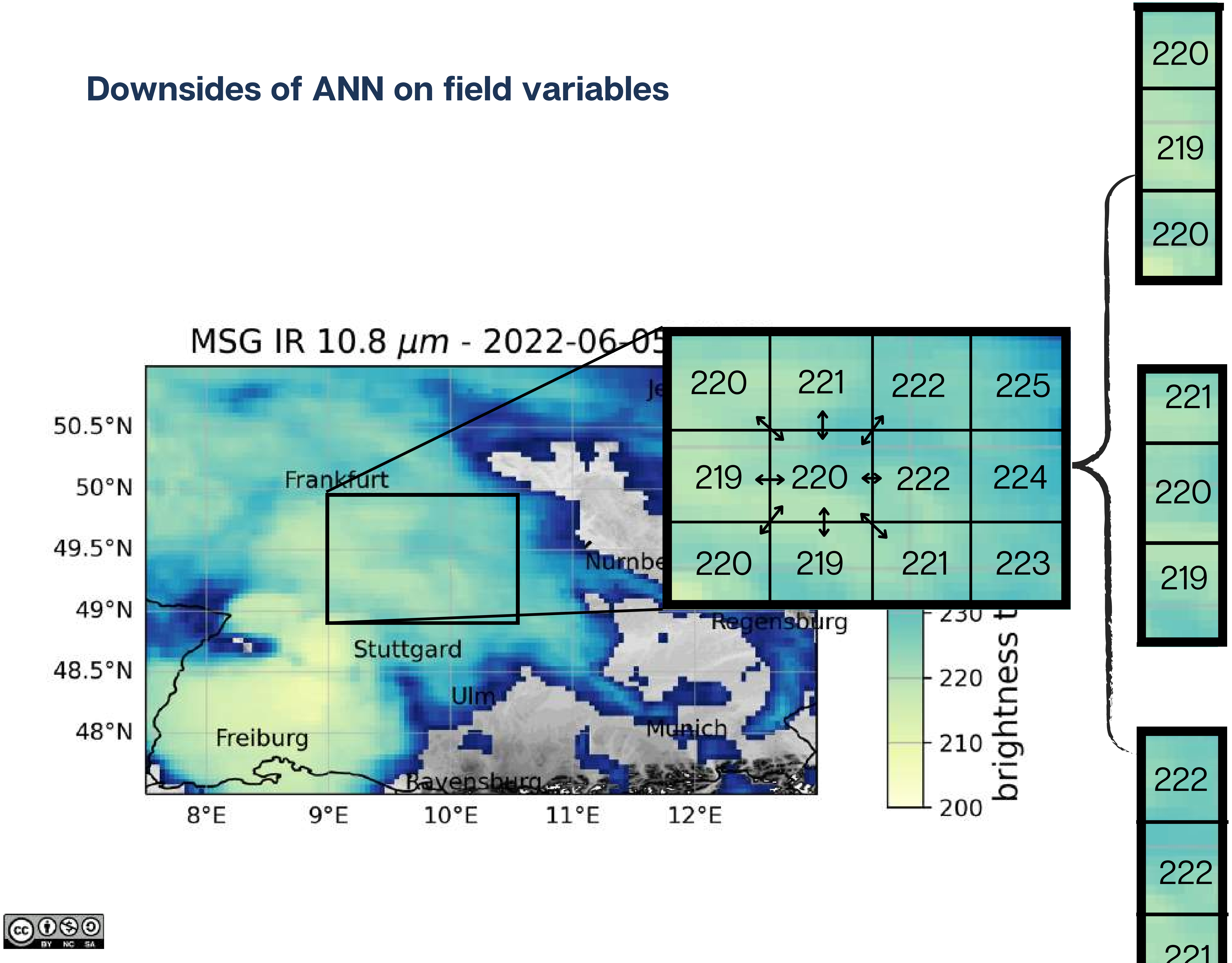
1 fully connected layer would have 10x2073 weights for the small image of 2073 elements



typical image sizes: 200 x 200 x 3 = 120000 elements.

Huge number of weights to manage for the network, only in the first hidden layer, and weights will add up quickly if you add more layers.

Downsides of ANN on field variables



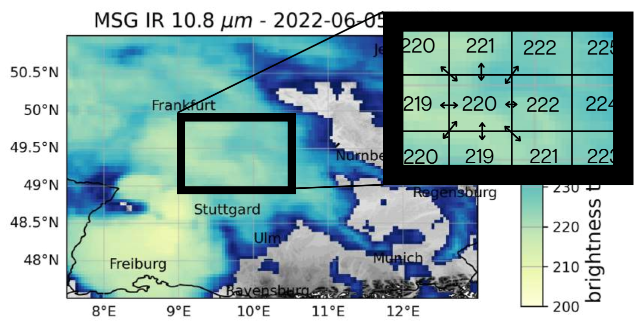
220	221	222	225
219 ↔	220 ↔	222	224
220	219	221	223

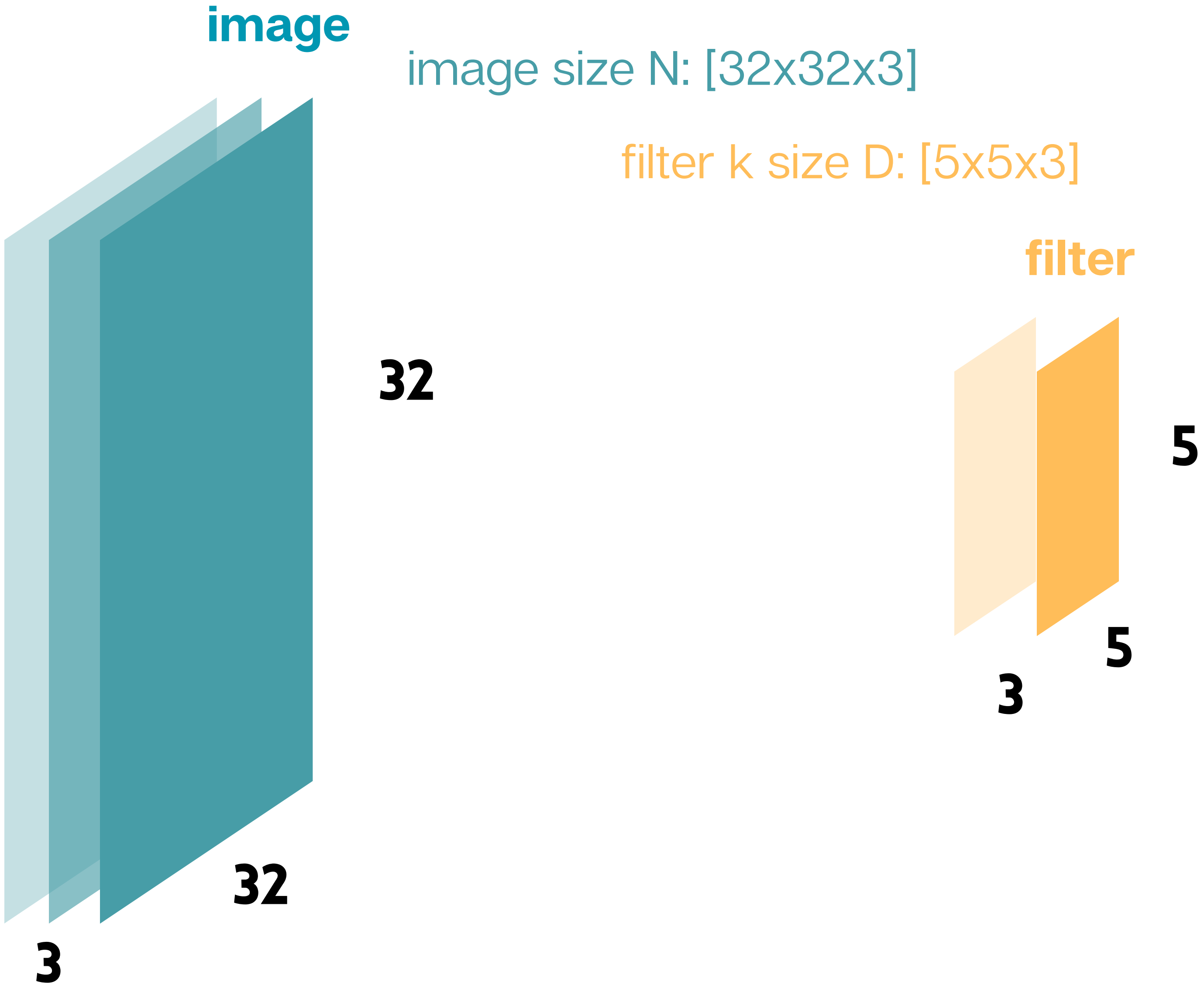
contextual and spatial **relationships** of **nearby pixels** in the data **ARE LOST**

Convolutional neural networks

What changes in convolutional neural network compared to the regular ones?

CNNs do preserve the spatial structure, i.e. they assume that their input are images, i.e. 3D objects





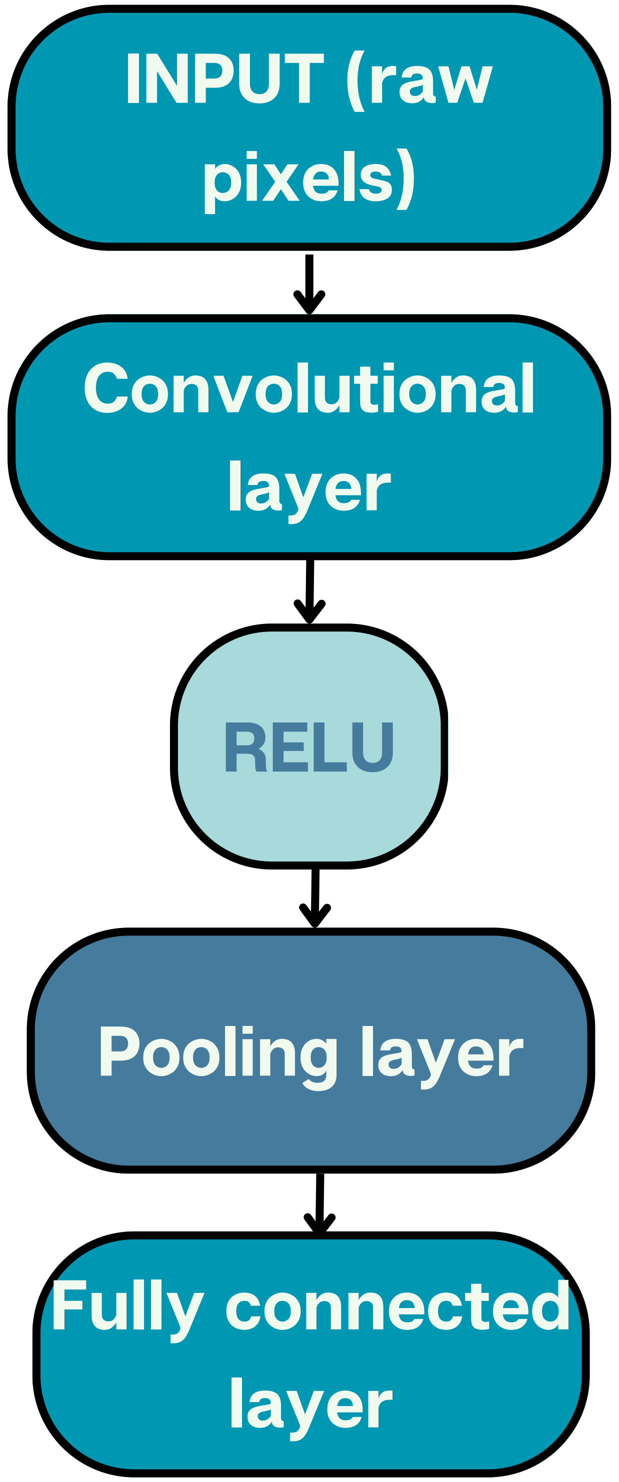
how?

by convolving the filter with the image:
calculating the dot product of the filter and the area of the image under the filter and then sliding the filter to a new position

Convolutional neural networks

These architectures are composed by stacking different types of layers in order:

- Convolutional layer
- Pooling layer
- Fully-connected layer



Convolutional layer

building block of CNN. It consists of k-learnable **filters** (called “**kernels**”), each of them with a width and a height. The filters are convolved across the height and the width of the input 3D volume.

Pooling layer

downsamples the input volume it receives, making the representations smaller and more manageable in terms of size.

Fully connected layer

derives the class scores of the output categories

Convolution for each of the k filters

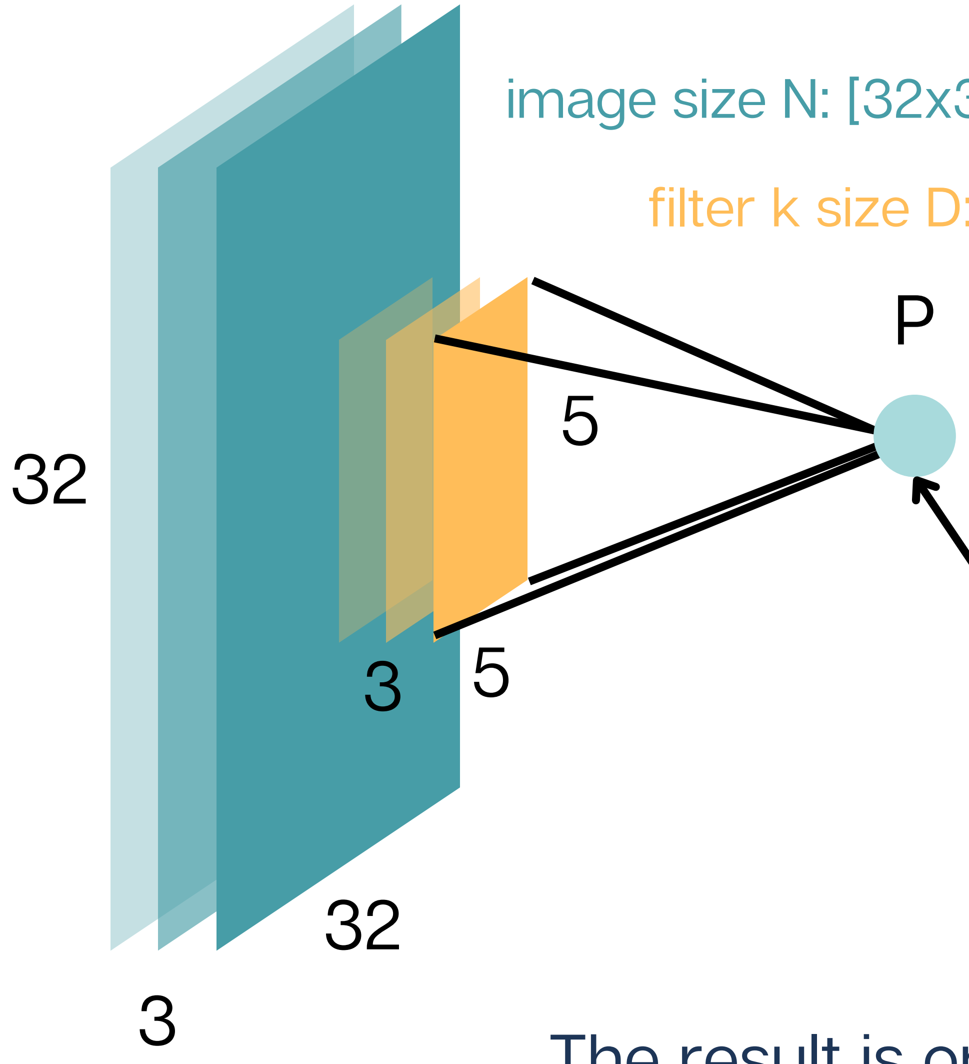


image size N: [32x32x3]

filter k size D: [5x5x3]

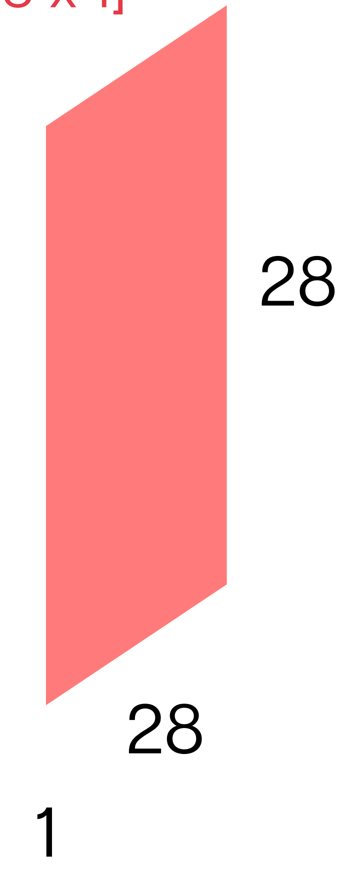
Convolve (slide) over all spatial locations of the image

The result is one number P that is the dot product between the filter and a small chunk of the image. Calling x = {x1.....xD} the portion of input image covered by the filter array and f={f1...fD} the filter, the number P is:

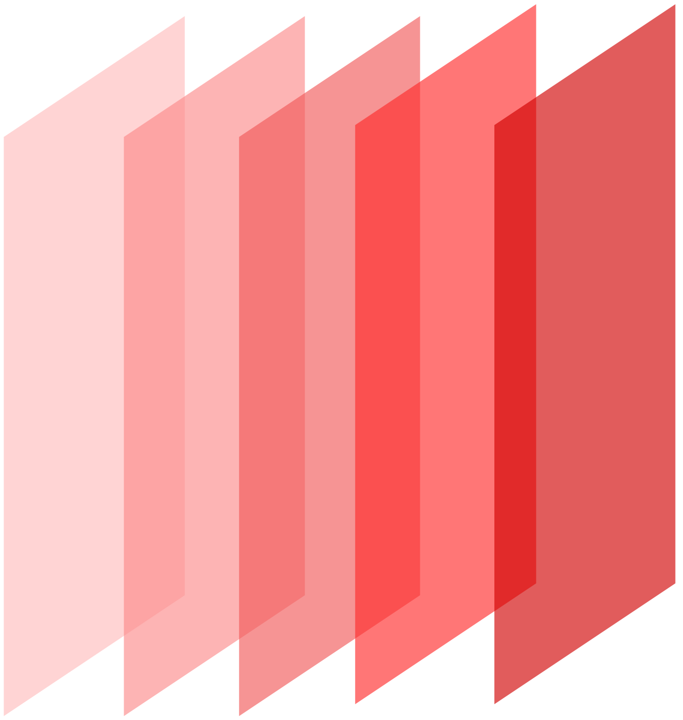
P = x1*f1+x2*f2+x3*f3+... +xD*fD

After sliding on all positions to cover the image, **we obtain one value for each position that will compose the activation map**

activation map: [28 x 28 x 1]

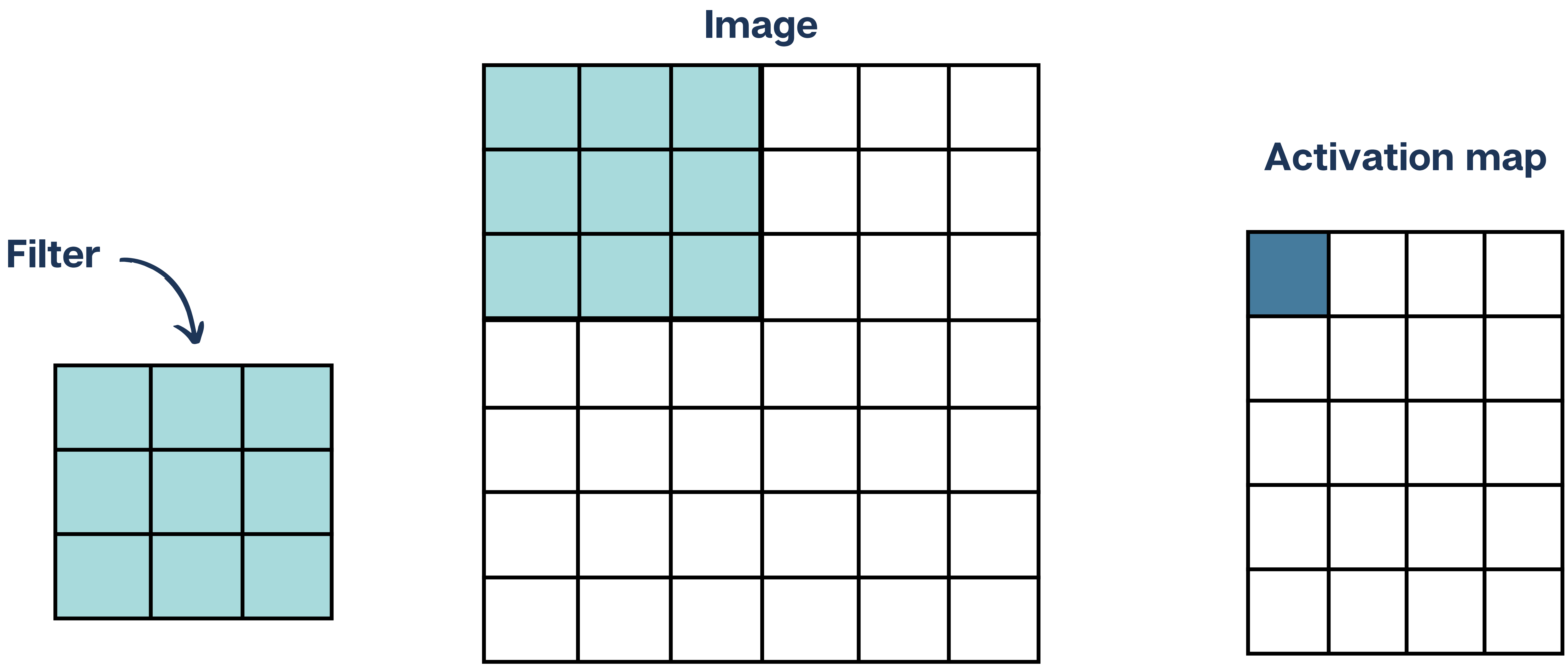


if k = 6 and we have 6 filters of size 5x5, we get 6 activation maps separate



We can stack them up to get a **new image of size 28 x 28 x 6**

Seeing on the 2d plane of the image

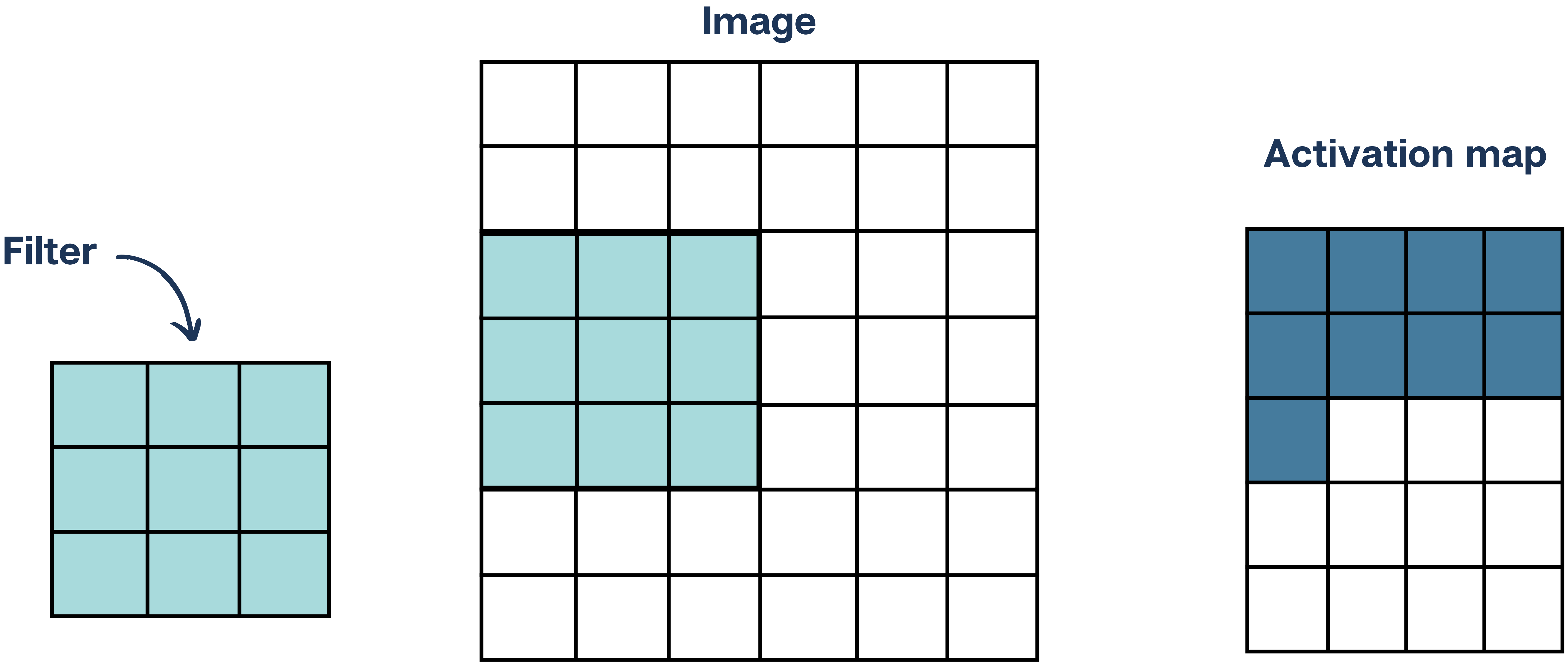


Image

Activation map

Filter

Seeing on the 2d plane of the image

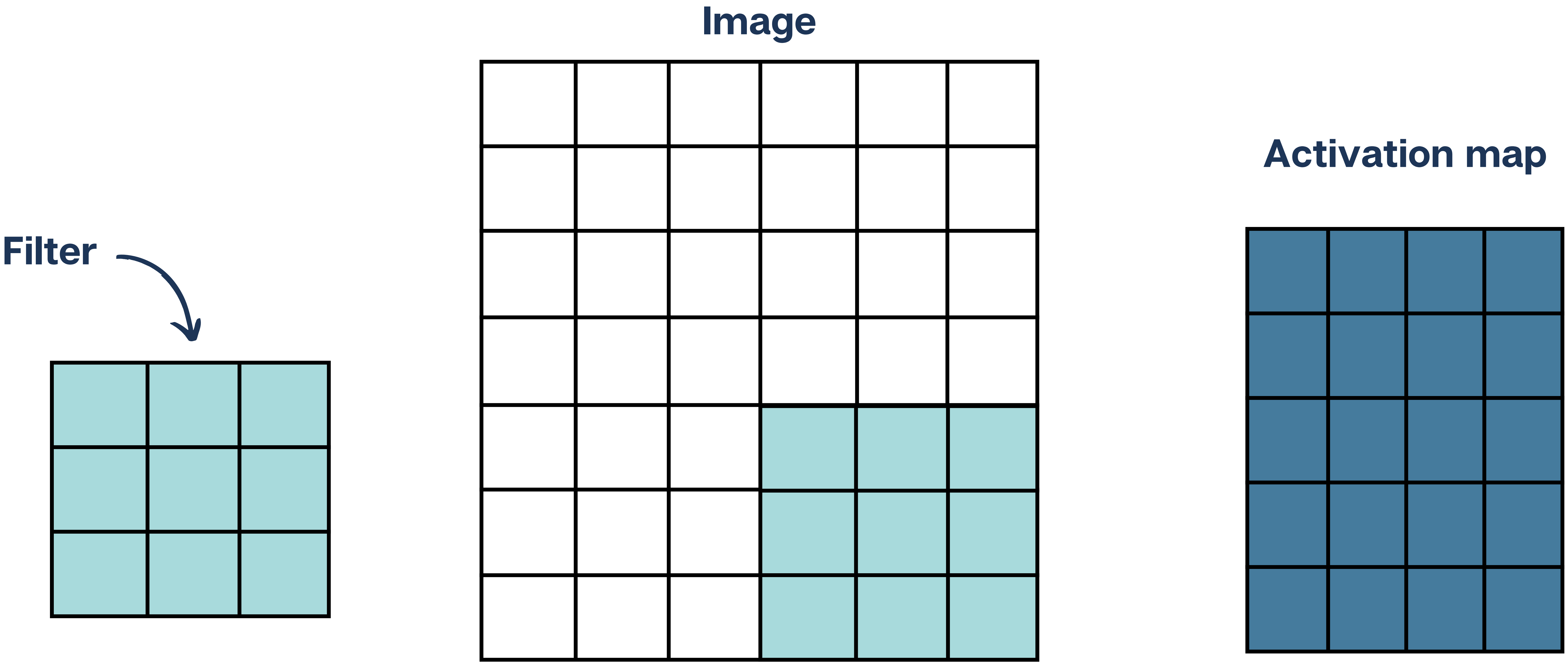


Image

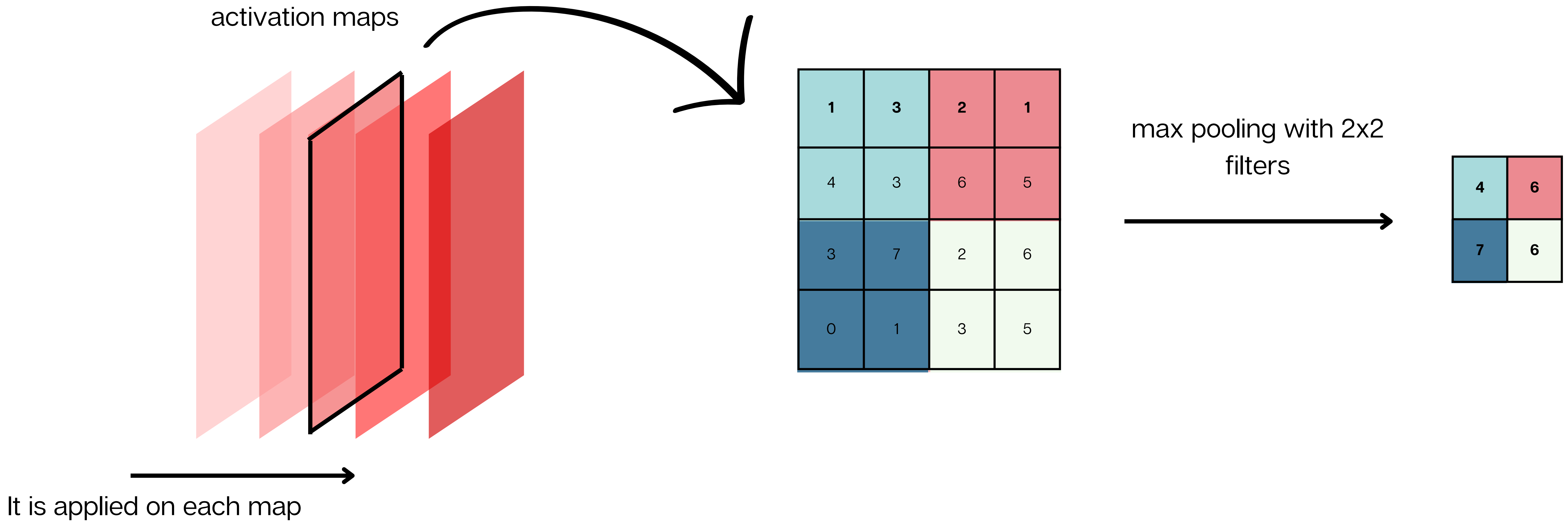
Activation map

Filter

Seeing on the 2d plane of the image



Pooling layer



activation maps

It is applied on each map

1	3	2	1
4	3	6	5
3	7	2	6
0	1	3	5

max pooling with 2x2 filters

4	6
7	6

Pooling works in the same way as CNN but instead of taking the dot product of the elements of the filter with the area covered of the image, it selects the maximum.

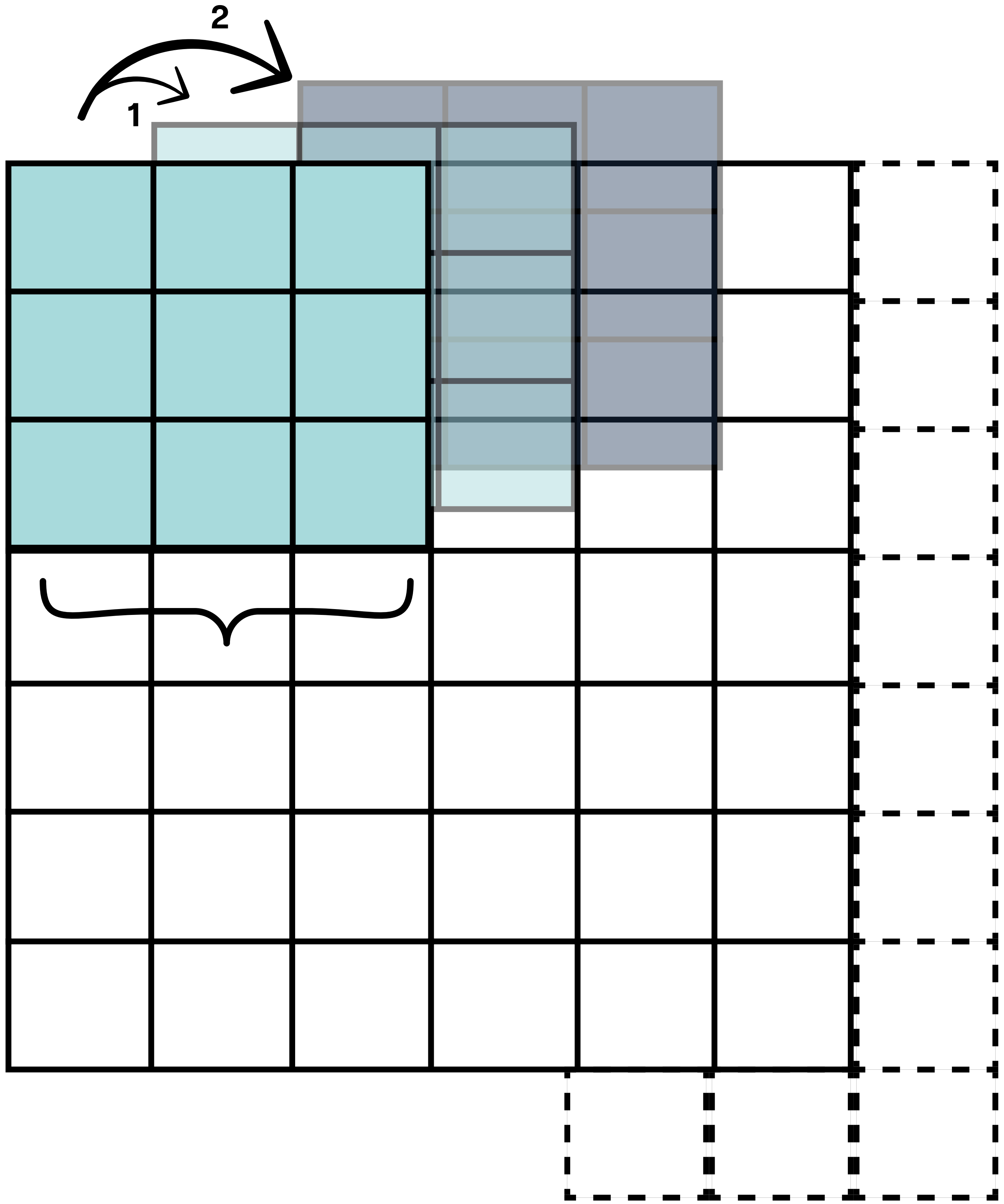
Other forms of pooling are also possible instead of max, for example, one can take the average. For some applications, like recognition, higher values are preferred to be transferred, so max pooling is better in such cases

The pooling layer **downsamples the input volume** it receives and operates on each activation map independently so that the downsampling is applied uniformly to all of them.

Nomenclature of CNNs and Pooling layers

FILTER SIZE

it is the spatial dimension of the sliding window over the input. This is a crucial parameter in image classification tasks: large kernel sizes extract less information and lead to a faster reduction of the dimensions in the layer, but they are better suited to extract features that are larger. Small filter sizes can extract larger amount of information containing highly local features from the input.



STRIDE

it is the number that indicates of how many pixels the kernel should be shifted over at a time.

PADDING

it is necessary if the filter size extends beyond the activation map. Most common approach is zero-padding because it maintains the same size of the input. Applying a padding of 1 means to add 1 pixels on each border of the input image

For more on CNN and animations explaining all the concepts presented here, check the following link

2

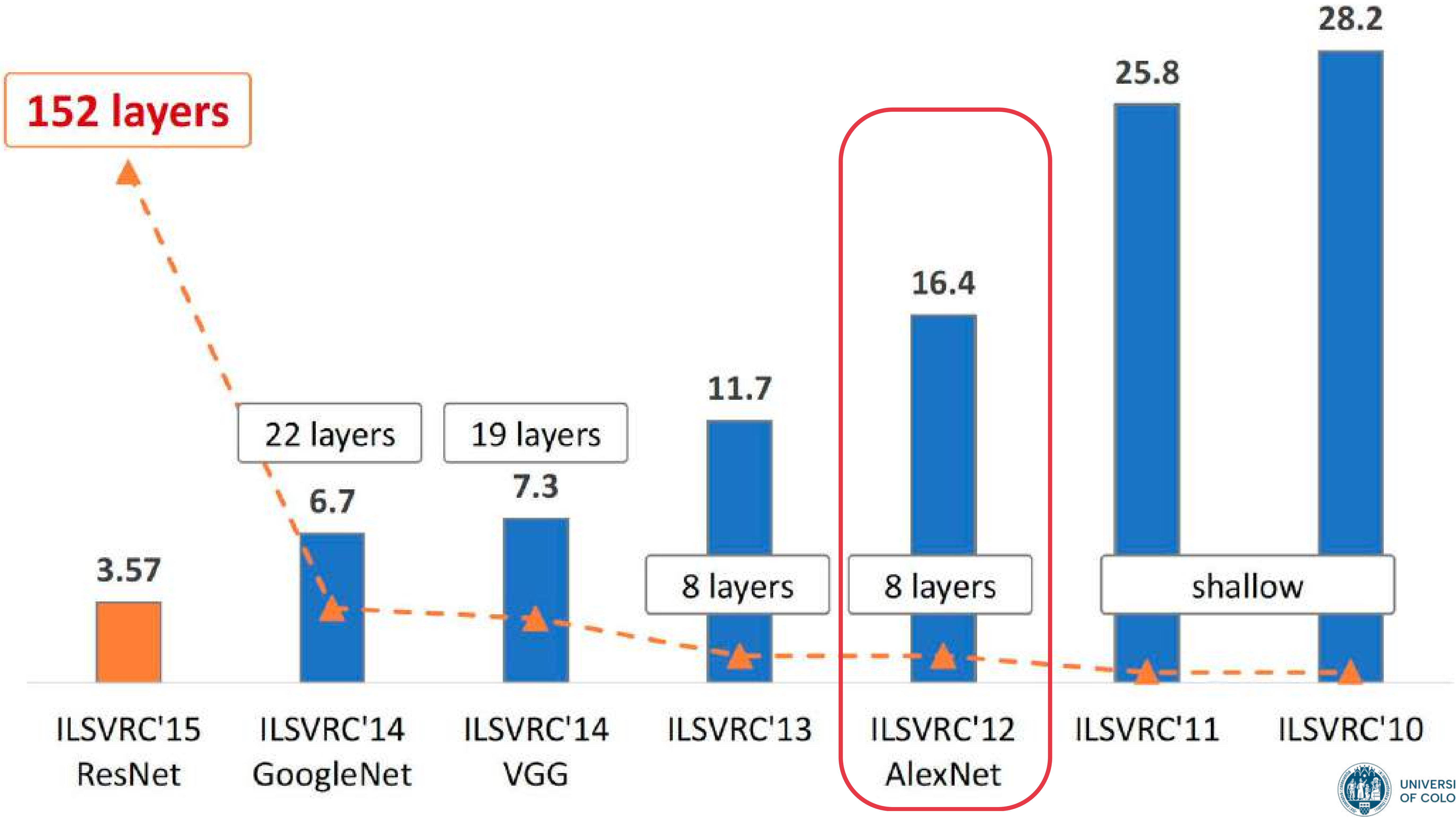
Visualizing CNN and examples

CNN examples

The ImageNet Large Scale Visual Recognition Challenge (ILSVRC) evaluates algorithms for object detection and image classification at large scale.

Why?

- to **compare progress in detection** across a wider variety of objects
- to **measure the progress of computer vision** for large scale image indexing for retrieval and annotation.



*The **evolution of the winning entries on the ImageNet Large Scale Visual Recognition Challenge from 2010 to 2015**. **Displayed is the error in the classifications. Since 2012, CNNs have outperformed hand-crafted descriptors and shallow networks by a large margin**. Image re-printed with permission from K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Jun 2016, pp. 770–778. (K. Nguyen, C. Fookes, A. Ross and S. Sridharan, "Iris Recognition With Off-the-Shelf CNN Features: A Deep Learning Perspective," in IEEE Access, vol. 6, pp. 18848-18855, 2018, doi: 10.1109/ACCESS.2017.2784352._*

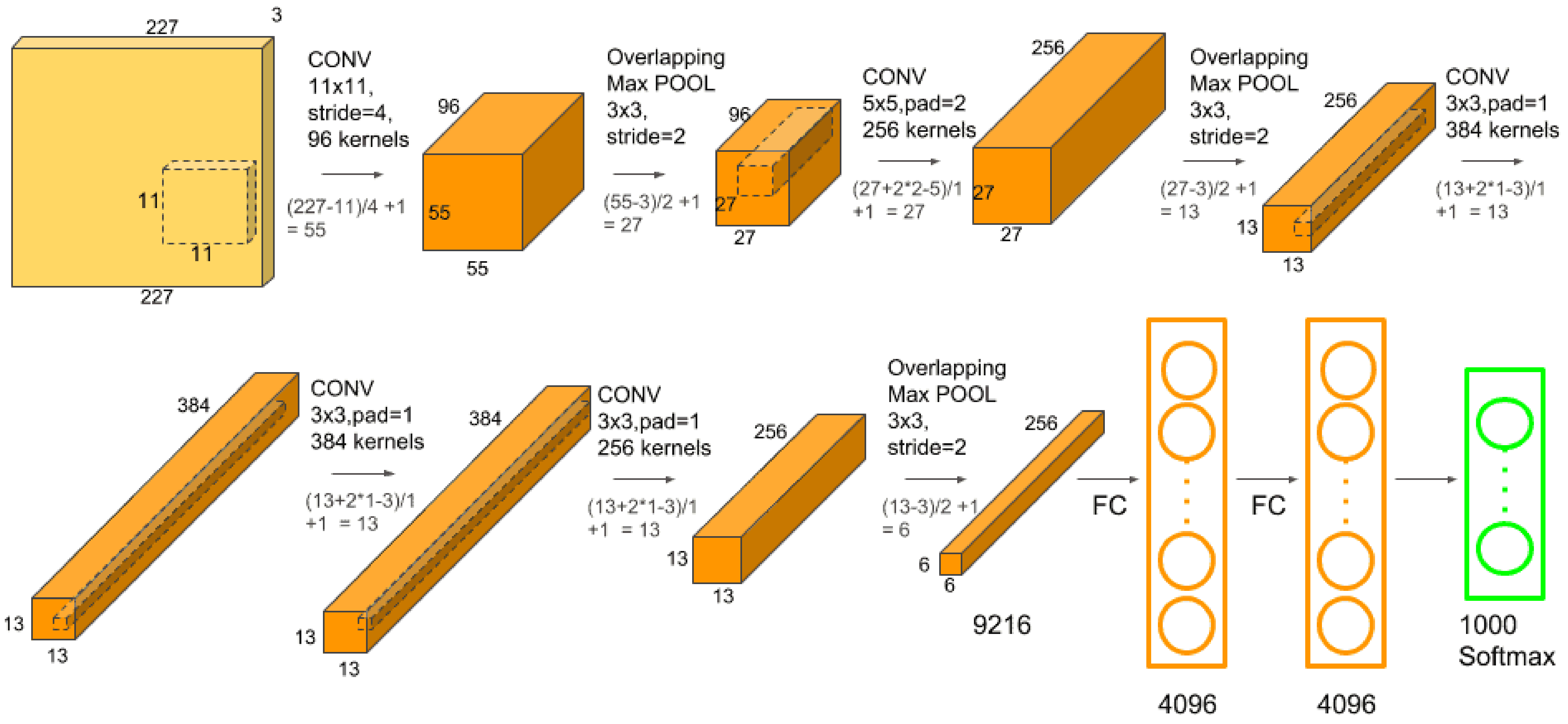
AlexNet

AlexNet solves the problem of image classification with subset of ImageNet dataset with roughly 1.2 million training images and 1000 classes:

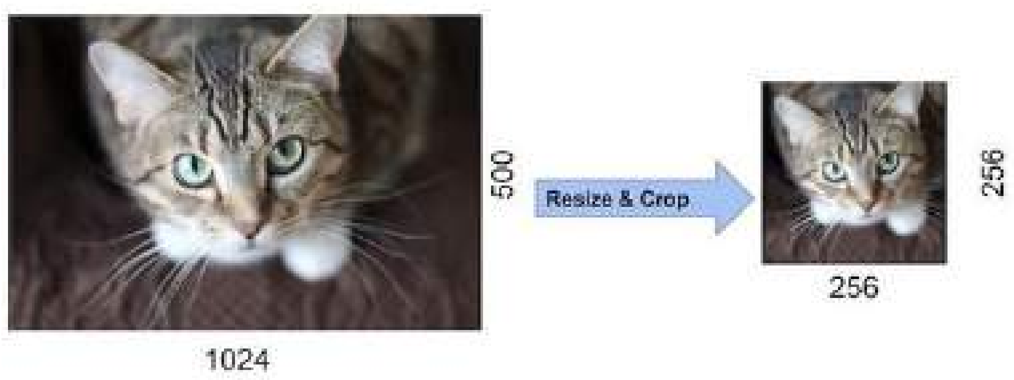
- 50,000 validation images,
- 150,000 testing images.

The output is a vector of 1000 numbers.

Full (simplified) AlexNet architecture:
[227x227x3] INPUT
[55x55x96] CONV1: 96 **11x11 filters** at stride 4, pad 0
[27x27x96] MAX POOL1: **3x3 filters** at stride 2
[27x27x96] NORM1: Normalization layer
[27x27x256] CONV2: 256 **5x5 filters** at stride 1, pad 2
[13x13x256] MAX POOL2: **3x3 filters** at stride 2
[13x13x256] NORM2: Normalization layer
[13x13x384] CONV3: 384 3x3 filters at stride 1, pad 1
[13x13x384] CONV4: 384 3x3 filters at stride 1, pad 1
[13x13x256] CONV5: 256 3x3 filters at stride 1, pad 1
[6x6x256] MAX POOL3: 3x3 filters at stride 2
[4096] FC6: 4096 neurons
[4096] FC7: 4096 neurons
[1000] FC8: 1000 neurons (class scores)



- Input: 227x227x3 images. If the input image is not 256*256, image is rescaled such that shorter size is of length 256, and cropped out the central 256*256 patch from the resulting image.



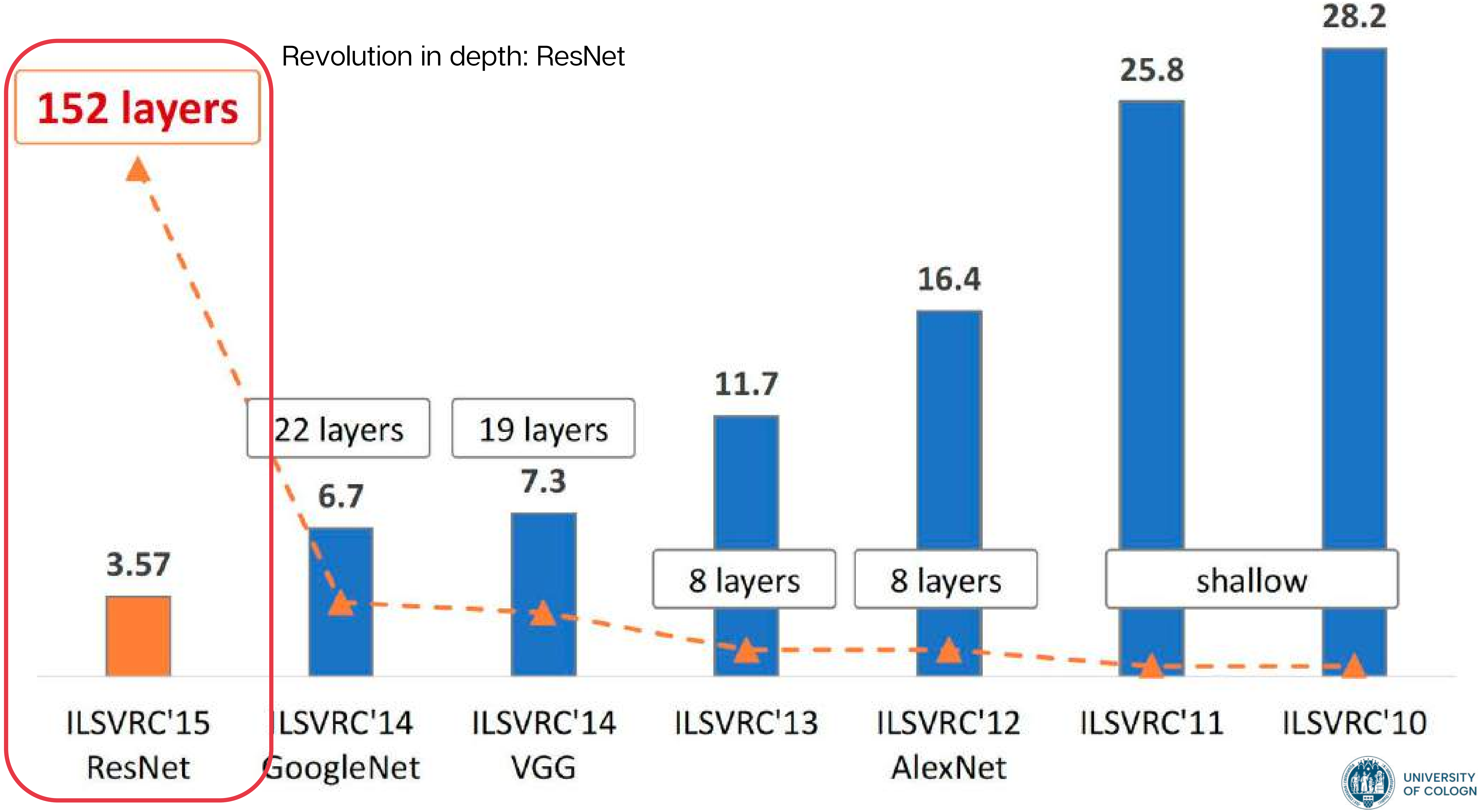
CNN examples

The ImageNet Large Scale Visual Recognition Challenge (ILSVRC) evaluates algorithms for object detection and image classification at large scale.

Why?

- to **compare progress in detection** across a wider variety of objects
- to **measure the progress of computer vision** for large scale image indexing for retrieval and annotation.

*The **evolution of the winning entries on the ImageNet Large Scale Visual Recognition Challenge from 2010 to 2015**. **Displayed is the error in the classifications. Since 2012, CNNs have outperformed hand-crafted descriptors and shallow networks by a large margin**. Image re-printed with permission from K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Jun 2016, pp. 770–778. (K. Nguyen, C. Fookes, A. Ross and S. Sridharan, "Iris Recognition With Off-the-Shelf CNN Features: A Deep Learning Perspective," in IEEE Access, vol. 6, pp. 18848-18855, 2018, doi: 10.1109/ACCESS.2017.2784352._*



Revolution in depth: ResNet

ResNet

Residual network are a type of neural networks that substantially simplify the training of deeper networks.

What happens when we continue stacking layers?

Two main problems:

1) Vanishing and exploding gradients will increase

2) It appears the so-called **degradation problem**, i.e. **deeper networks perform worse both in training and in test error,** but this is not caused by overfitting but to the fact that a deeper network has tons of parameters to learn

Solution:

Instead of just trying to learn H(x), we try to learn what to add or subtract to x, which is F(x). Since the output after the layer should be the same, it holds that

$$H(x) = F(x) + x$$

The layers will be used to fit the residual function: $F(x) = H(x) - x$ instead of $H(x)$ directly.

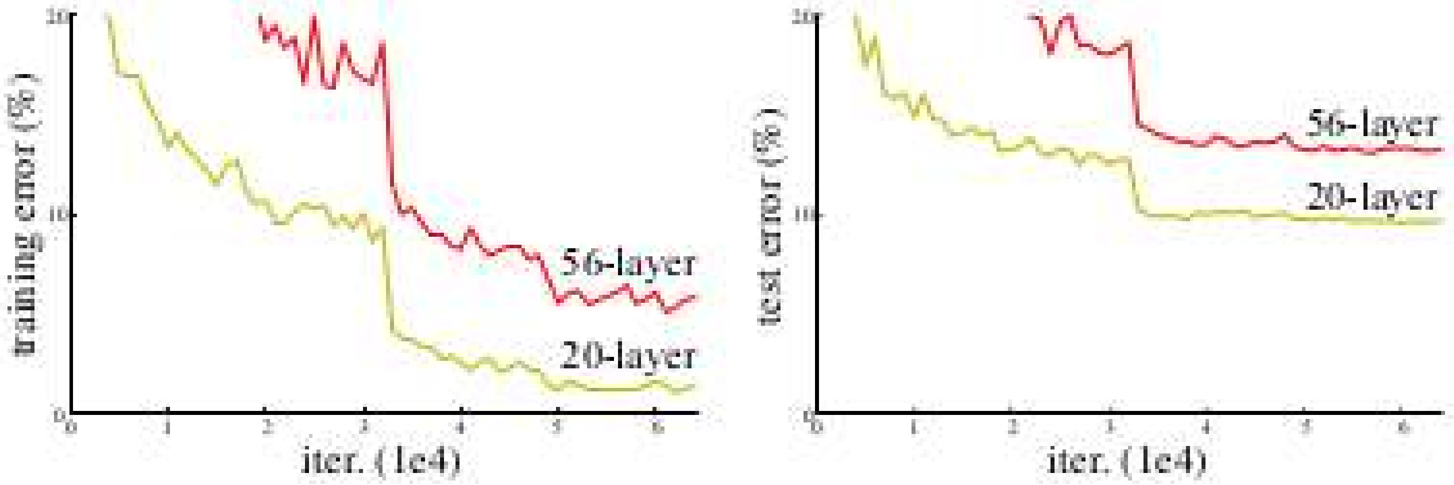
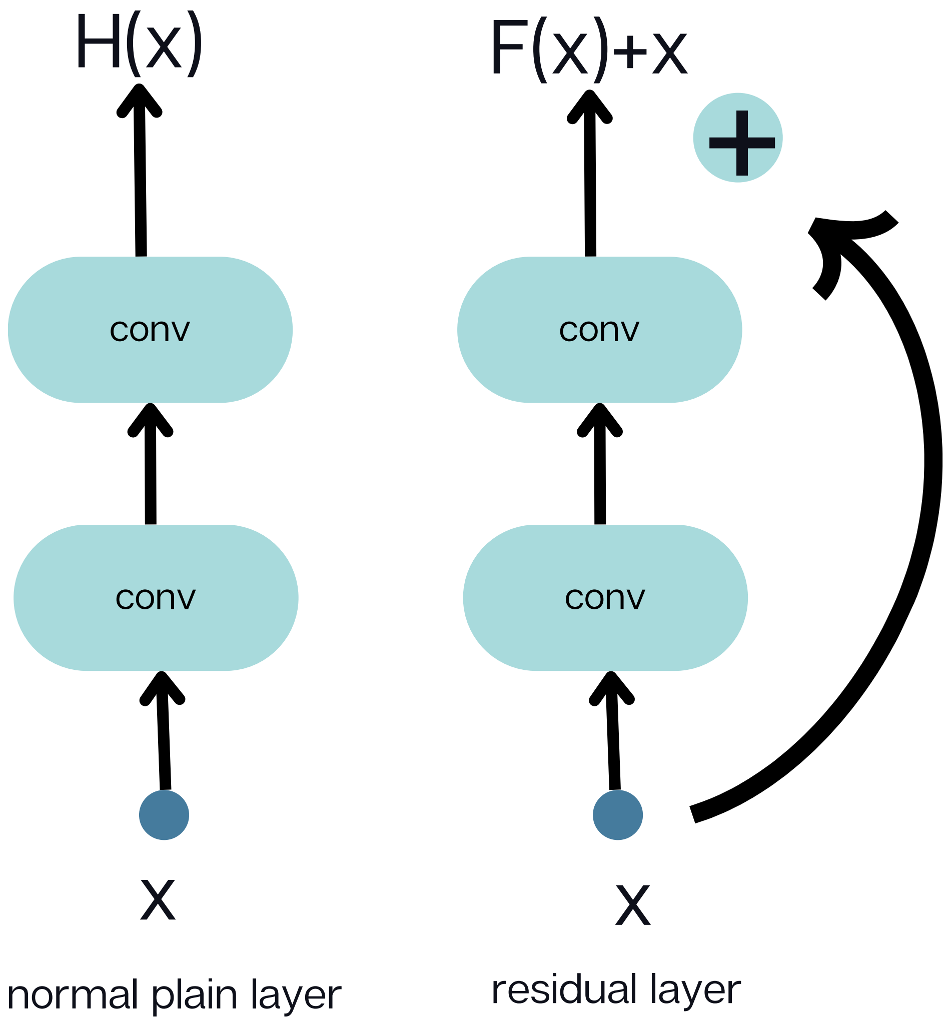
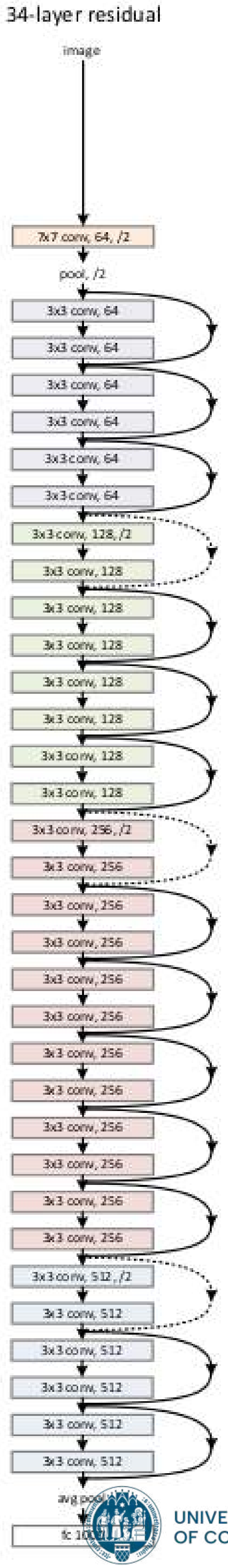


Figure 2.10: Training (left) and test error (right) on CIFAR10 with 20 and 2ith 56 layer plain networks. The deeper the network, the higher the test and training error. From the paper He et al., 2015



A residual network with 34 parameter layers (3.6 billion FLOPs). From the paper He et al., 2015



Full ResNet architecture:

- Stack residual blocks
- Every residual block has two 3x3 conv layers
- Periodically, double # of filters and downsample spatially using stride 2 (/2 in each dimension)
- Additional conv layer at the beginning
- No FC layers at the end (only FC 1000 to output classes)

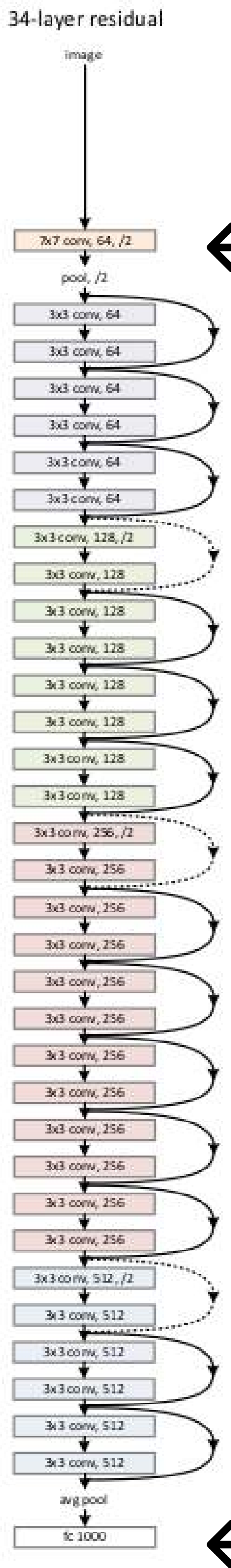
Experimental Results

- Able to train very deep networks without degrading (152 layers on ImageNet, 1202 on Cifar)

- Deeper networks now achieve lowing training error as expected

- Swept 1st place in all ILSVRC and COCO 2015 competitions

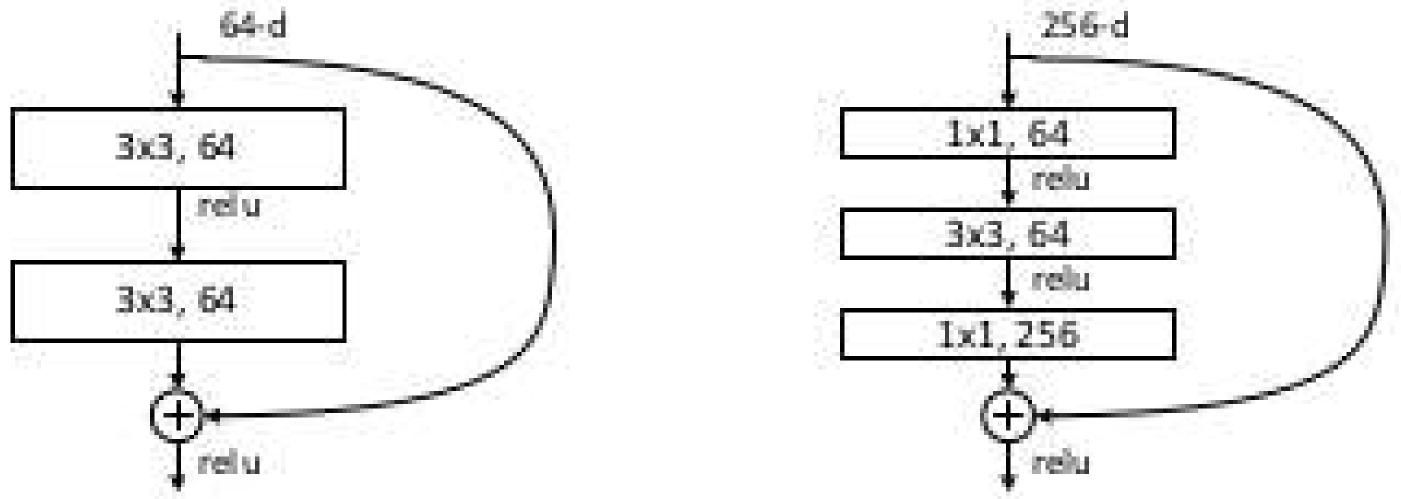
ILSVRC 2015 classification winner (3.6% top 5 error) -- better than “human performance”! (Russakovsky 2014)



begins with 7x7 convolutional layer

3x3 conv, 64 filters

3x3 conv, 128 filters, /2 spatially with stride 2



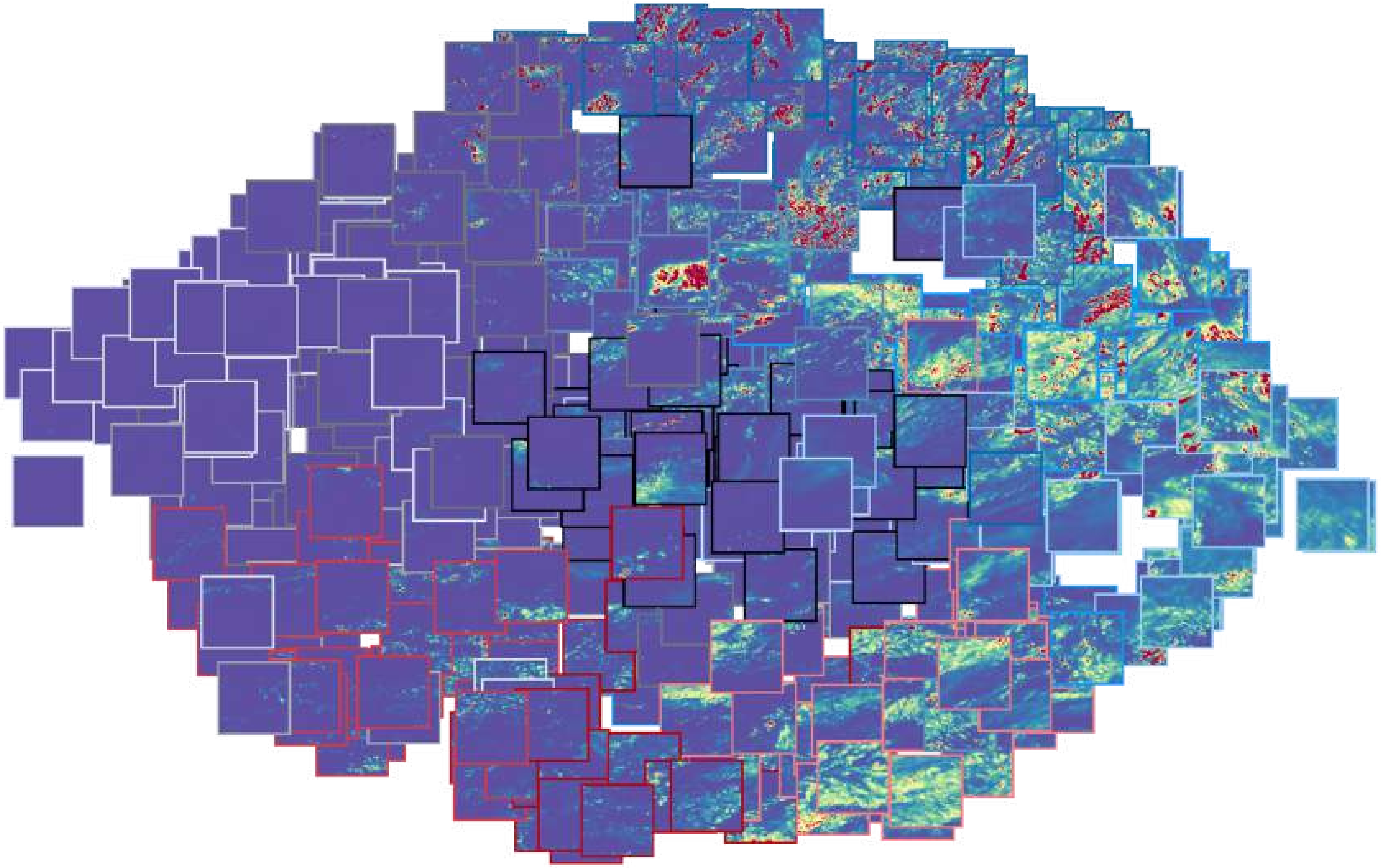
Global average pooling layer after last conv layer

No FC layers besides FC 1000 to output classes

Visualizing convolutional neural networks

Why visualizing?

because people think CNN are not interpretable



we will focus on the t-SNE representation methods often used with CNN for representing large datasets.

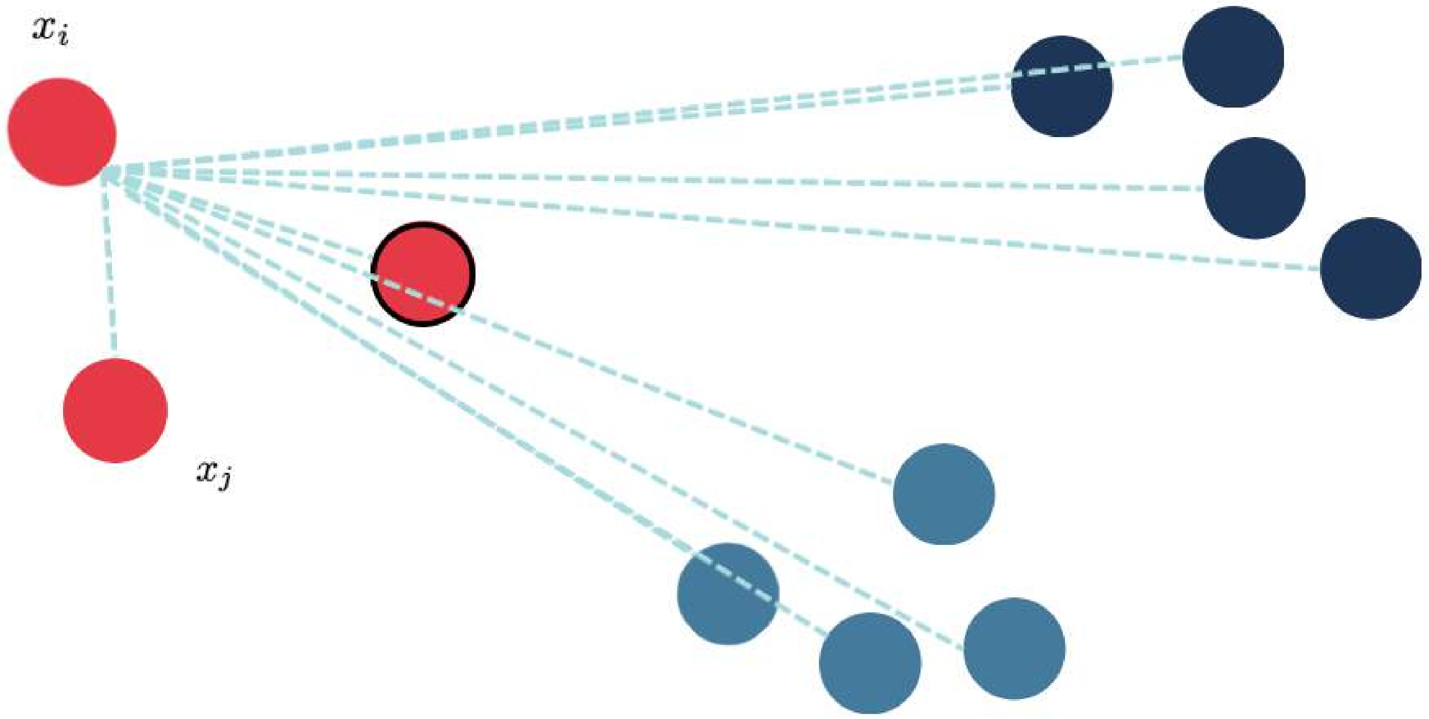
t-SNE. What is t-SNE, first of all?

t-SNE (t-distributed Stochastic Neighbor Embedding) is a method for **nonlinear dimensionality reduction** developed in 2008 by Geoffrey Hinton and Laurens Van der Maaten

it is used to understand high-dimensional data by projecting them on a 2 or 3 dimensional space.

How does the algorithm work?

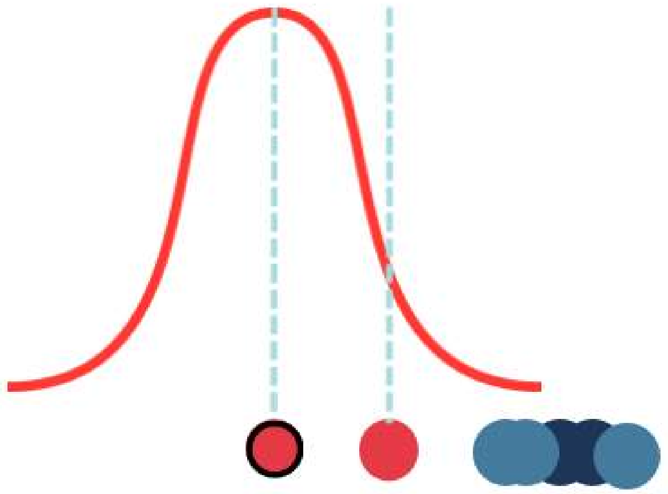
hypothesis: we have a **3 different classes**



1) STEP 1: build a probability distribution representing similarities between neighbours of classes.

what is a similarity? the conditional probability that the data point xi would pick xj as its neighbor.

As a first step, we **want to create a probability distribution that can represent similarities between neighbors of these classes**.



Euclidean distances of all other points from xi, are proportional to the Gaussian probability density centered around position xi of the given selected point.

Representation of the construction of a Gaussian probability distribution function of the distances between the points of a given multidimensional dataset, inspired from the article of Kemal Erdem on Medium on t-SNE representations.

To get better proportions we can normalize the probability density by the sum of all the projections, therefore the **conditional probability that the data point xi would pick xj as its neighbor** can be written as:

and it depends on the L2 distance between the points, as well as on a quantity **sigma**.

$$P_{j|i} = \frac{exp(-||x_i - x_j||^2/2\sigma_i^2)}{\sum_{k \neq i} exp(-||x_i - x_k||^2/2\sigma_i^2)}$$

Increasing sigma gets the Gaussian shape to become broader, contributing to better distinguish probabilities of neighboring points in the tail.

ok, but what is sigma?

The value for sigma is connected to a quantity called **perplexity**. Perplexity can be interpreted as a **guess about the number of neighbors for the central point of the cluster**, and it tells something about how to balance the attention between local and global aspects of the dataset.

$$Perp(Pi) = 2^{-\sum p_{i|j} log_2 p_{j|i}}$$

where $-\sum p_{i|j} log_2 p_{j|i}$ is the Shannon entropy.

The higher the perplexity, the higher is the variance of the distances of the points.

ok but how are sigma and the perplexity connected?

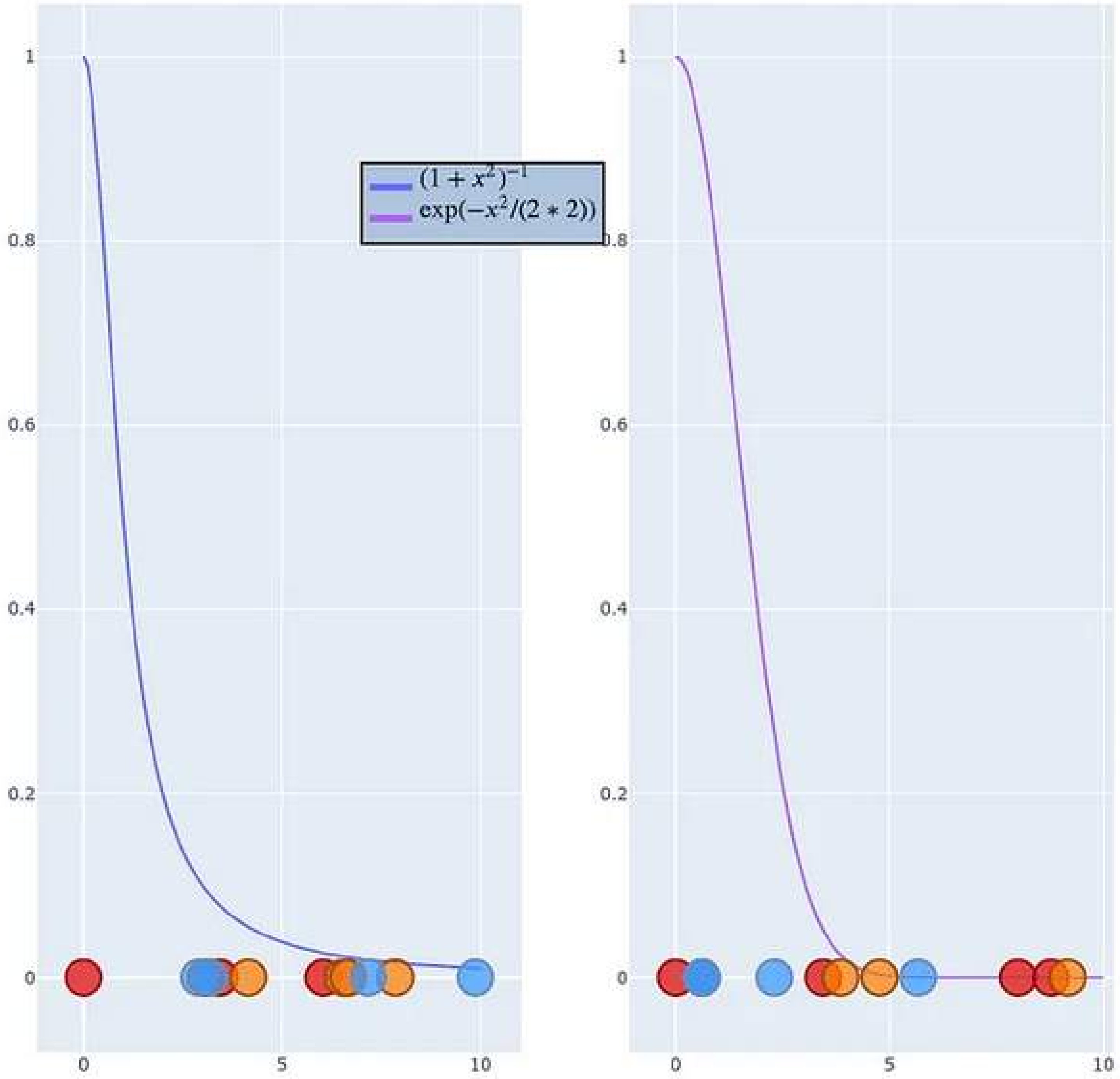
Sigma is connected to the perplexity because
SNE performs a binary search for the sigma value
that can reproduce a probability distribution with a **fixed perplexity** chosen from the user.

STEP 2: build low dimensional probability distribution.
create a low dimensional space with the same number of points as the original space, spread randomly on the new space. We want to find similar probability distributions of the Pj|i, but in this low dimensional space.

we use a **t-student distribution for describing the distances among the points in this new low-dimensional space**, because the tail of the t-distribution is more steep, and has a long tail, reducing the problem of squashing all points into a single point.

If we call yi the positions, we can write it as:

$$q_{ij} = \frac{(1 + ||y_i - y_j||^2)^{-1}}{\sum_{k \neq l}(1 + ||y_k - y_l||^2)^{-1}}$$



Comparison of gaussian and t-student distribution, from https://medium.com/towards-data-science/t-sne-clearly-explained-d84c537f53a

STEP 3: Minimization of the cost function to make qi|j similar to Pi|j

goal: make the probability distribution of the points qij in the low-dimensional space as similar as possible to the distribution Pi|j.

The cost function that does this job is the Kullback-Leiber (KL) divergence: KL divergence is a measure of how much two distributions are different from each other.

--> minimizing KL divergence calculated for Pi|j and qij will make qij as similar as possible to Pi|j and provide the optimal reduction of dimensions

$$KL(P||Q) = \sum_{i} \sum_{j} P_{ij} log \frac{p_{ij}}{q_{ij}}$$

STEP 3: Minimization of the cost function to make qi|j similar to Pi|j

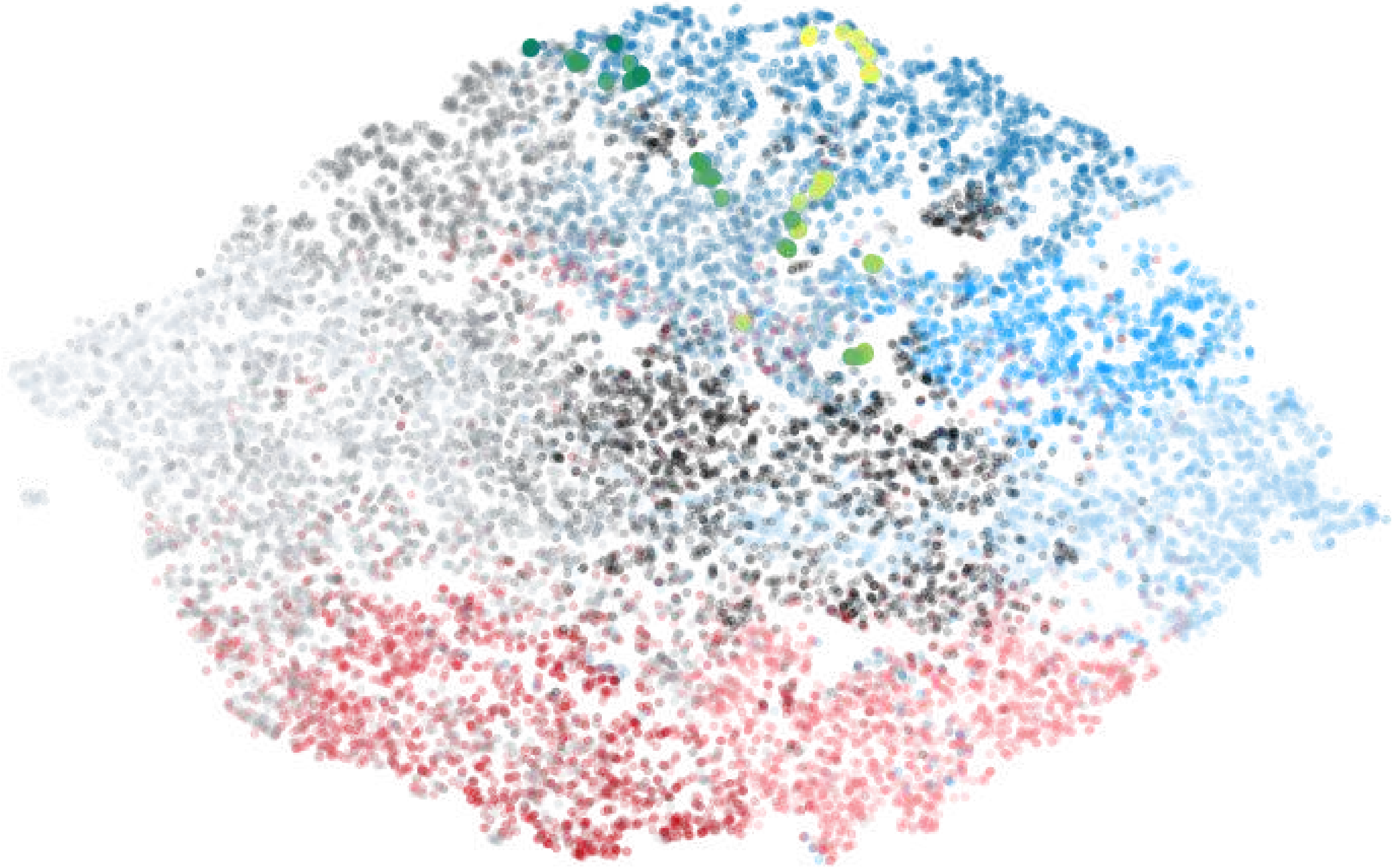
goal: make the probability distribution of the points qij in the low-dimensional space as similar as possible to the distribution Pi|j.

The cost function that does this job is the Kullback-Leiber (KL) divergence: KL divergence is a measure of how much two distributions are different from each other.

--> minimizing KL divergence calculated for Pi|j and qij will make qij as similar as possible to Pi|j and provide the optimal reduction of dimensions

$$KL(P||Q) = \sum_i \sum_j P_{ij} log \frac{p_{ij}}{q_{ij}}$$

By feeding the **feature array** coming out o**f the fully connected layer as input for Pi|j**, we can build a 2d distribution qij that looks like this:



Example of t-SNE representation of an input of satellite cloud optical thickness images (EXPATS research group)

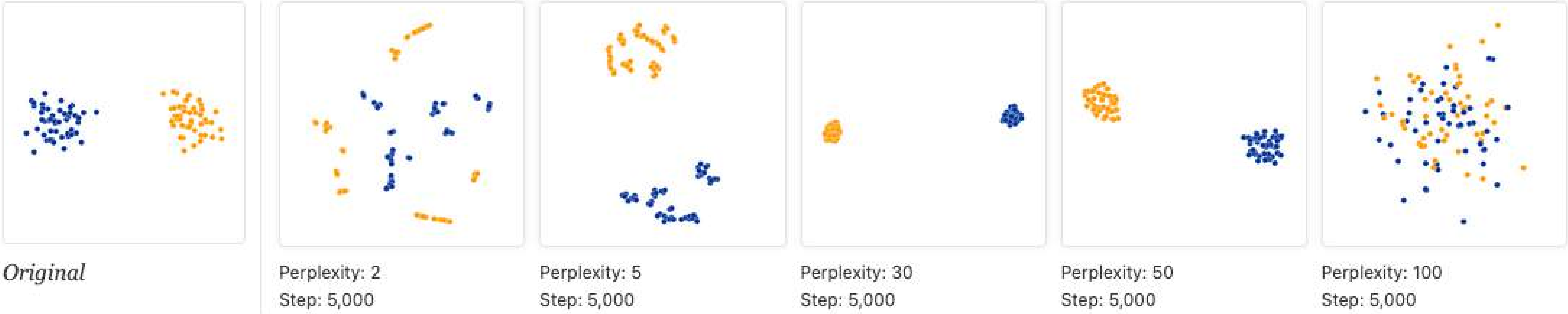
tSNE limitations

from Wattenberg, et al., "How to Use t-SNE Effectively", Distill, 2016. http://doi.org/10.23915/distill

“Although extremely useful for visualizing high-dimensional data, t-SNE plots can sometimes be mysterious or misleading.”

- **On perplexity values**: When perplexity ranges in the values suggested by the authors between 5 and 50, diagrams show clusters with different shapes, but for different values we can get unexpected behaviors. There’s **no fixed value that gives reasonable results** and **different datasets might require different number of iterations to converge**. Also, running multiple times with the same set of hyperparameters does not always give the same diagrams.

Figure from “How to Use t-SNE Effectively”. by Wattenberg, Viegas and Johnson, https://distill.pub/2016/misread-tsne/#citation reproduced under Creative Commons Attribution CC-BY 2.0



Outside that range for perplexity 5-50, things get a little weird.

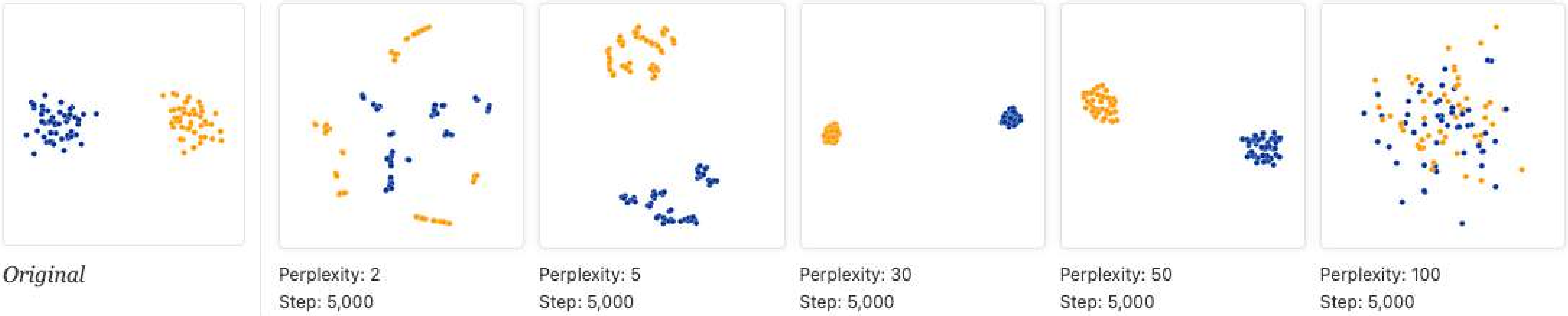
tSNE limitations

from Wattenberg, et al., "How to Use t-SNE Effectively", Distill, 2016. http://doi.org/10.23915/distill

“Although extremely useful for visualizing high-dimensional data, t-SNE plots can sometimes be mysterious or misleading.”

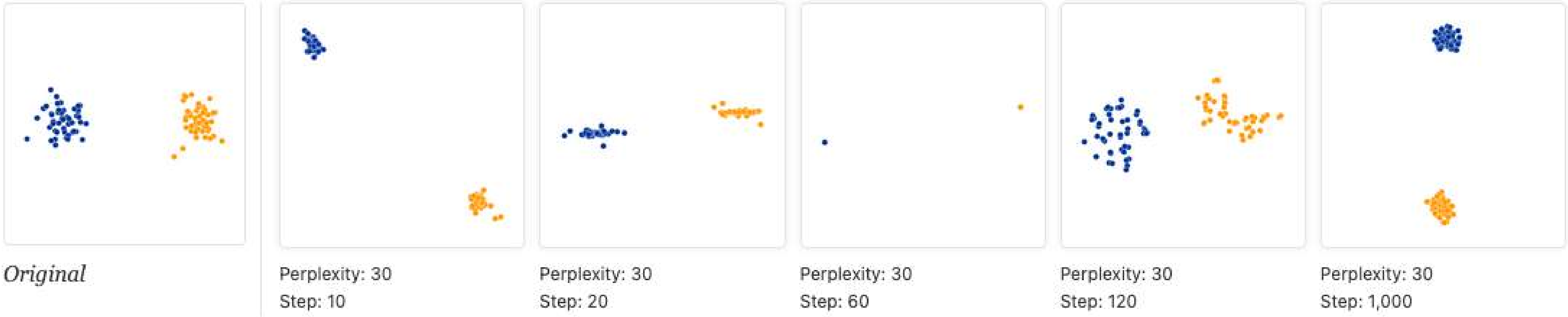
- **On perplexity values**: When perplexity ranges in the values suggested by the authors between 5 and 50, diagrams show clusters with different shapes, but for different values we can get unexpected behaviors. There’s **no fixed value that gives reasonable results** and **different datasets might require different number of iterations to converge**. Also, running multiple times with the same set of hyperparameters does not always give the same diagrams.

Figure from “How to Use t-SNE Effectively”. by Wattenberg, Viegas and Johnson, https://distill.pub/2016/misread-tsne/#citation reproduced under Creative Commons Attribution CC-BY 2.0



Outside that range for perplexity 5-50, things get a little weird.

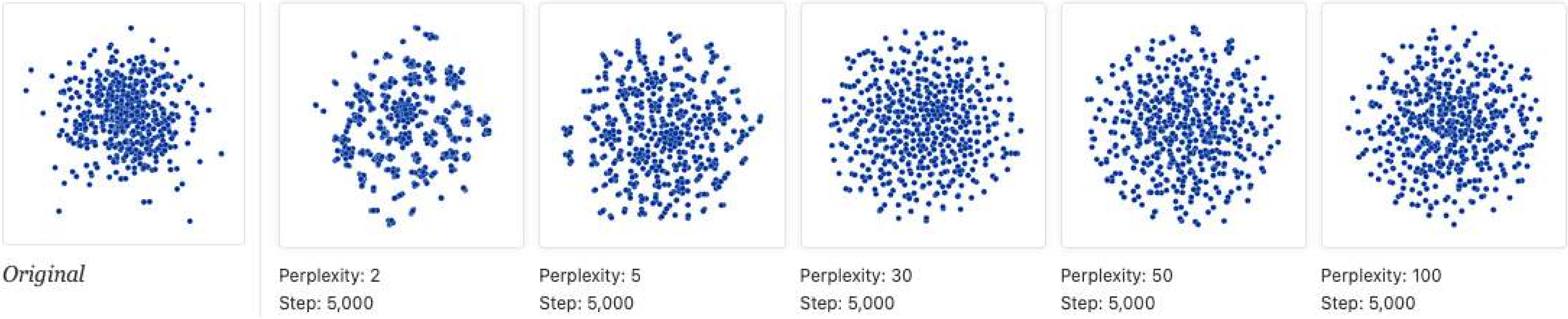
How much does the number of iterations affect the results? iterate until you get a stable configuration



The first four were stopped before stability. After 10, 20, 60, and 120 steps you can see layouts with seeming 1-dimensional and even pointlike images of the clusters. If you see a t-SNE plot with strange “pinched” shapes, chances are the process was stopped too early.

- **On random noise:** when representing with t-SNE a cloud of points generated randomly, depending on the perplexity t-SNE reproduces clusters, which aren’t meaningful. These clusters are just random noise of the-SNE plots. You see patterns in what is really just random data.

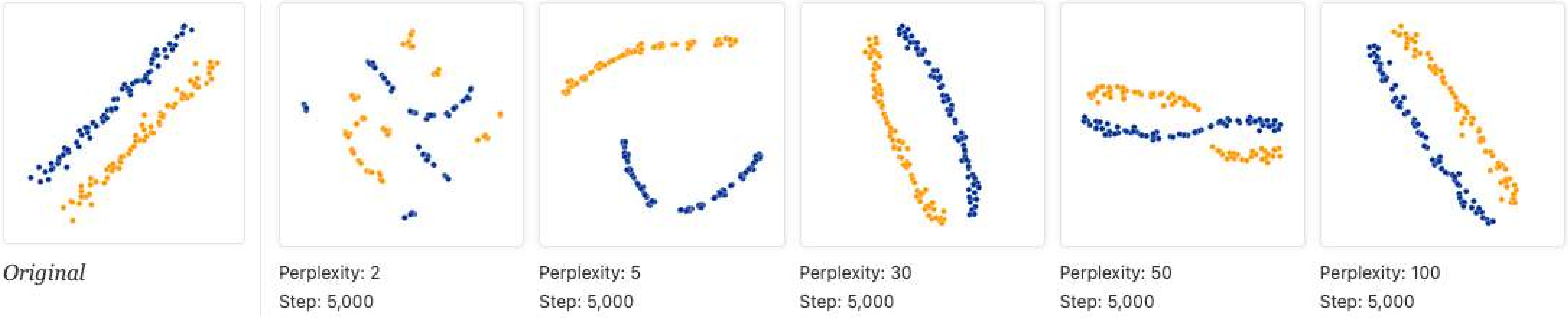
500 points drawn from a unit Gaussian distribution in 100 dimensions projected onto the first two coordinates.



perplexity 2: evident clusters

Figure from “How to Use t-SNE Effectively”. by Wattenberg, Viegas and Johnson, https://distill.pub/2016/misread-tsne/#citation reproduced under Creative Commons Attribution CC-BY 2.0

- **On shapes**: sometimes, shapes appear depending on the perplexity



t-SNE tends to expand denser regions of data. Since the middles of the clusters have less empty space around them than the ends, the algorithm magnifies them.

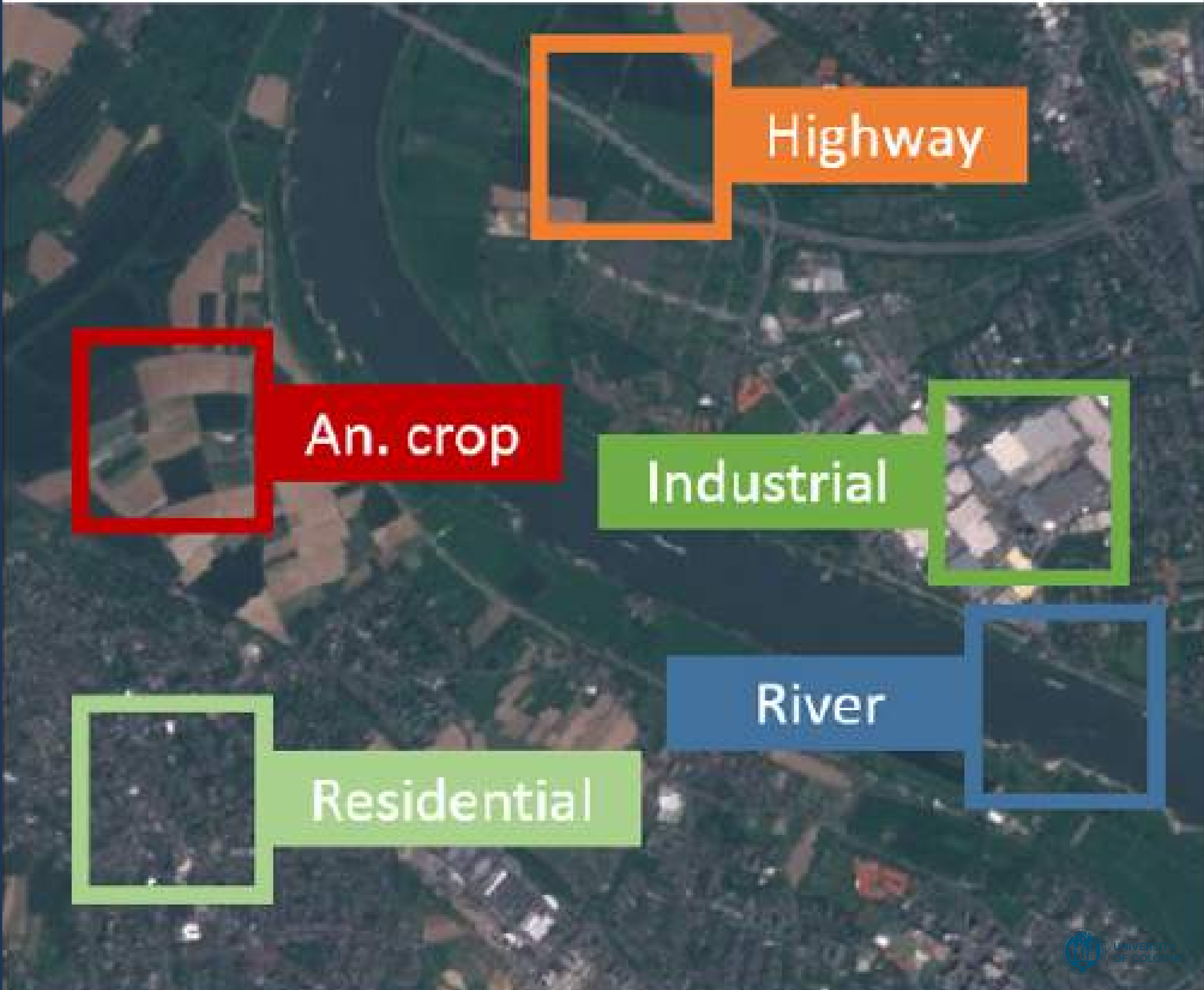
Even in the best cases, though, there’s a subtle distortion: the lines are slightly curved outwards in the t-SNE diagram.

Putting it all together: one example

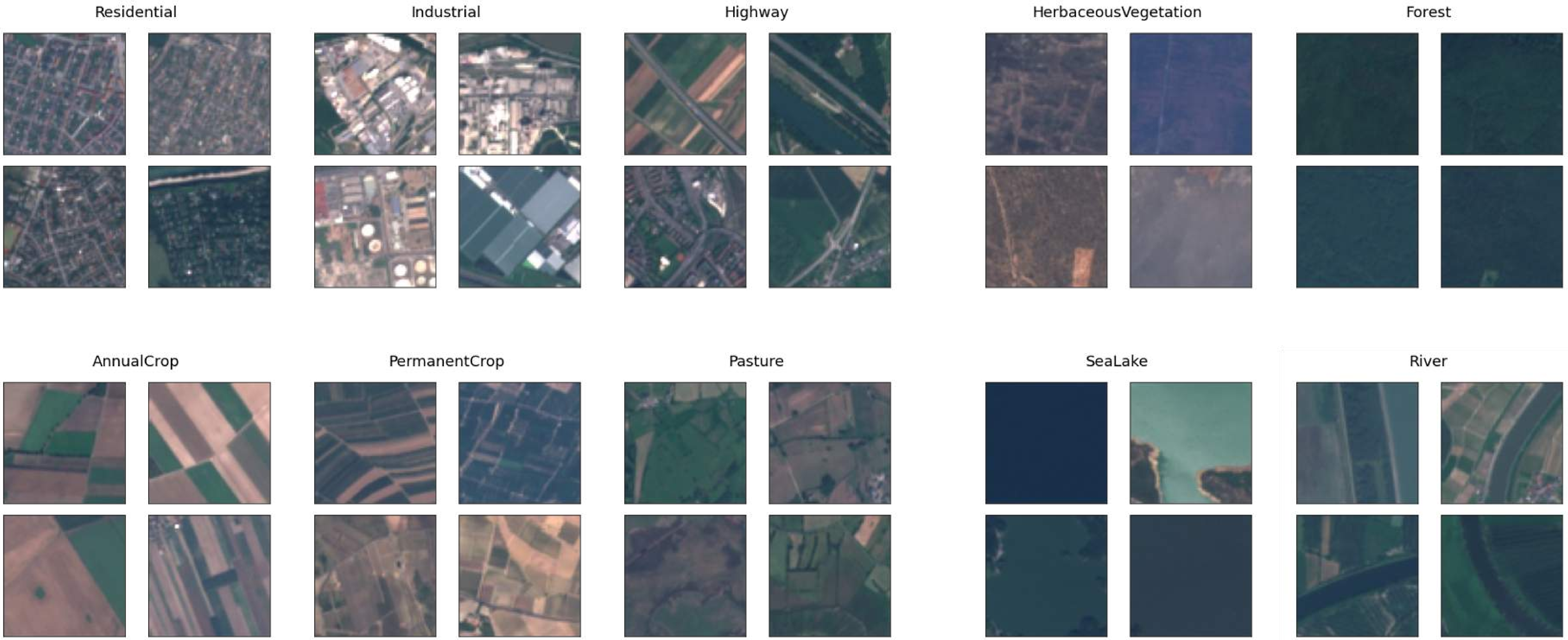
Supervised approach applied to EuroSAT

- benchmark dataset for land cover and land use classification
- 27 000 images (64x64 pixel), labelled & georeferenced, manually checked

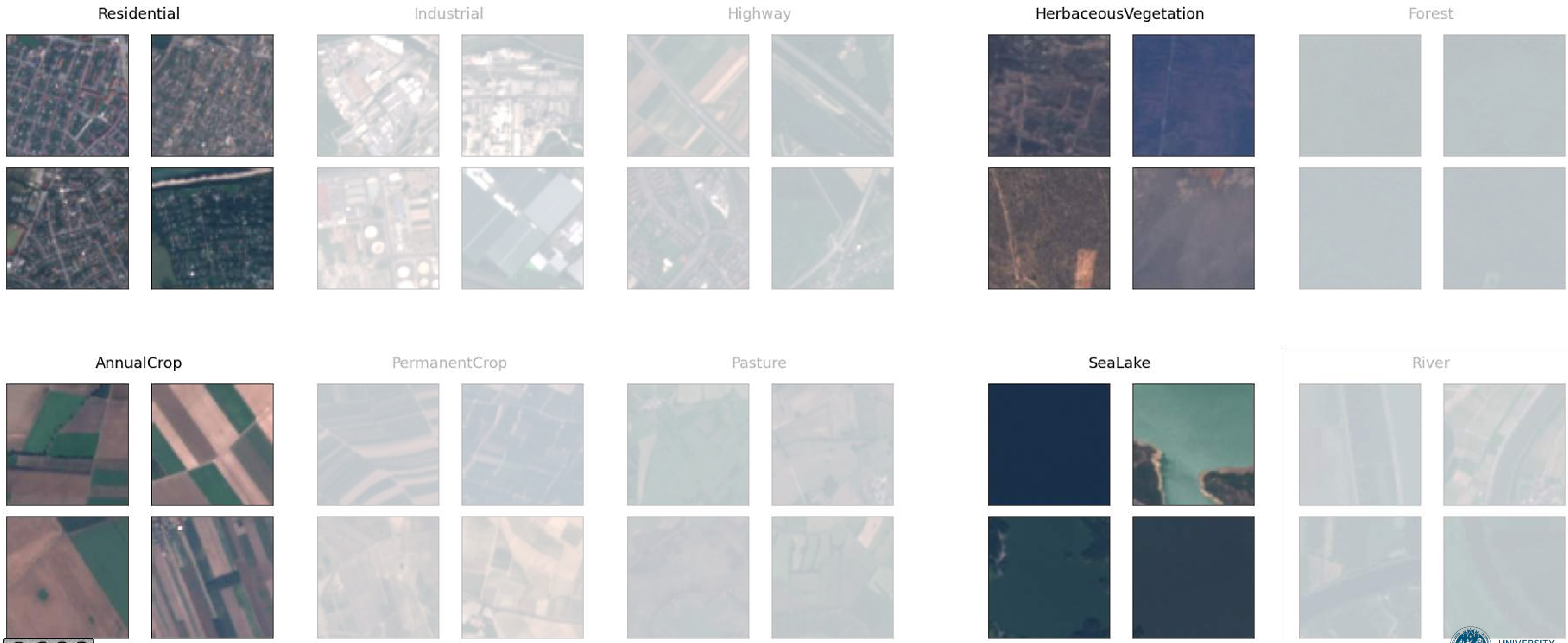
- Sentinel-2 satellite operated by ESA
- RGB from Multispectral Imager
- Spatial resolution ~ 10m
- Areas over cities covered by European Urban Atlas
- 34 countries, all year



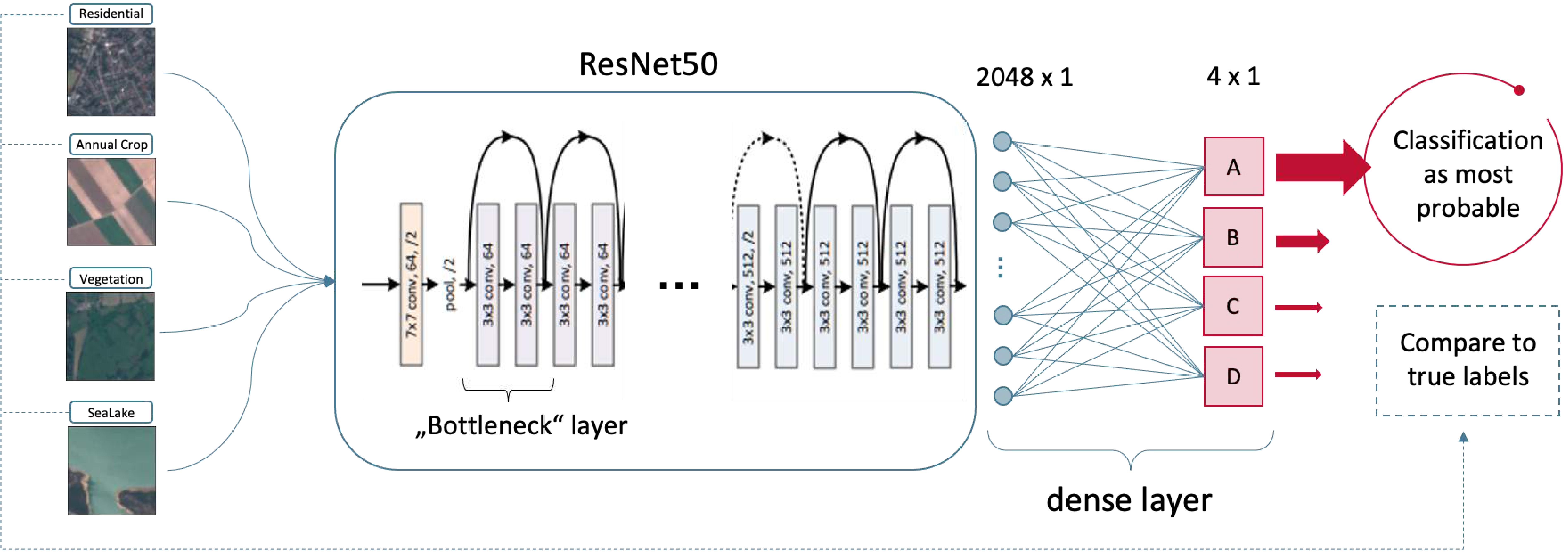
EuroSAT classes



Subset of images used (12000 for simplicity)



Architecture



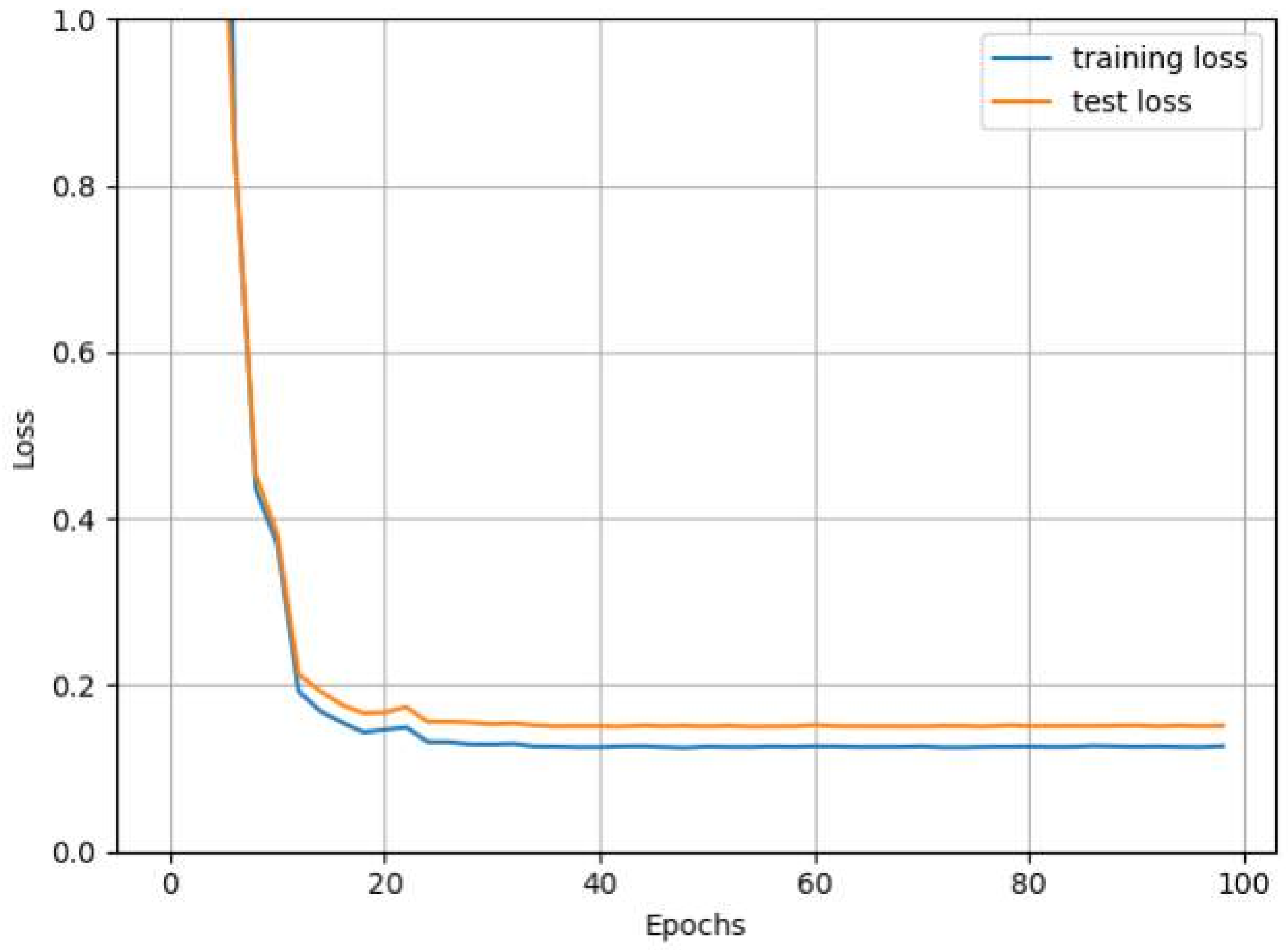
Training

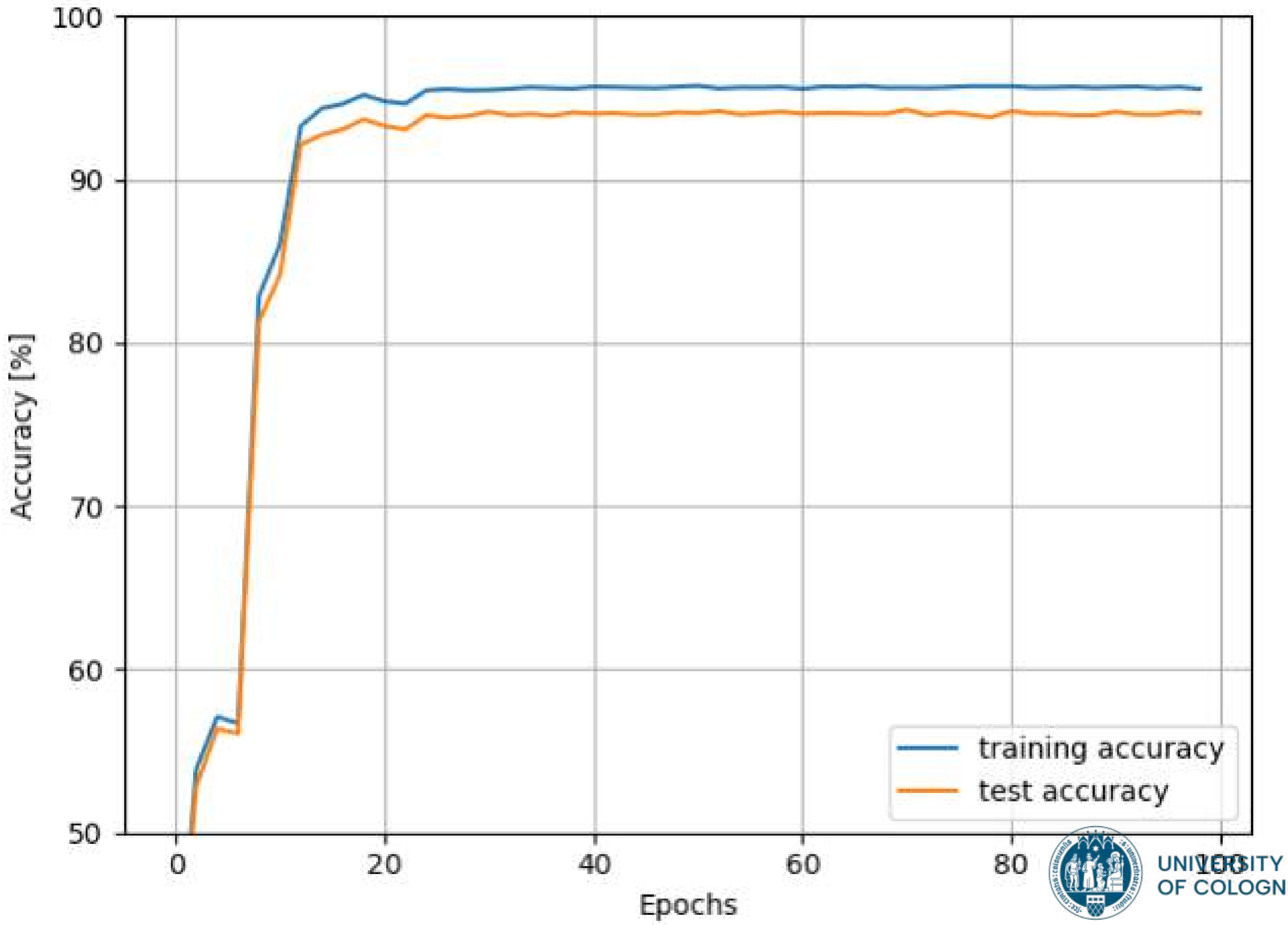
80% train, 20% test (normalized)

Hyperparameters chosen:

- Optimizer: stochastic gradient descent
- Scheduler: ReduceLROnPlateau
- Loss: cross-entropy

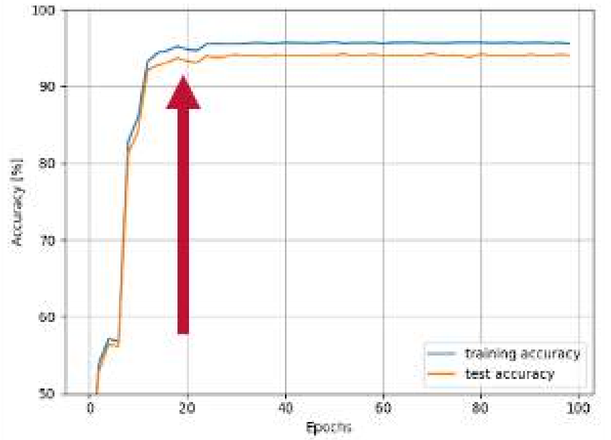
Train for 100 epochs





Test

Test accuracy = 93.3 %

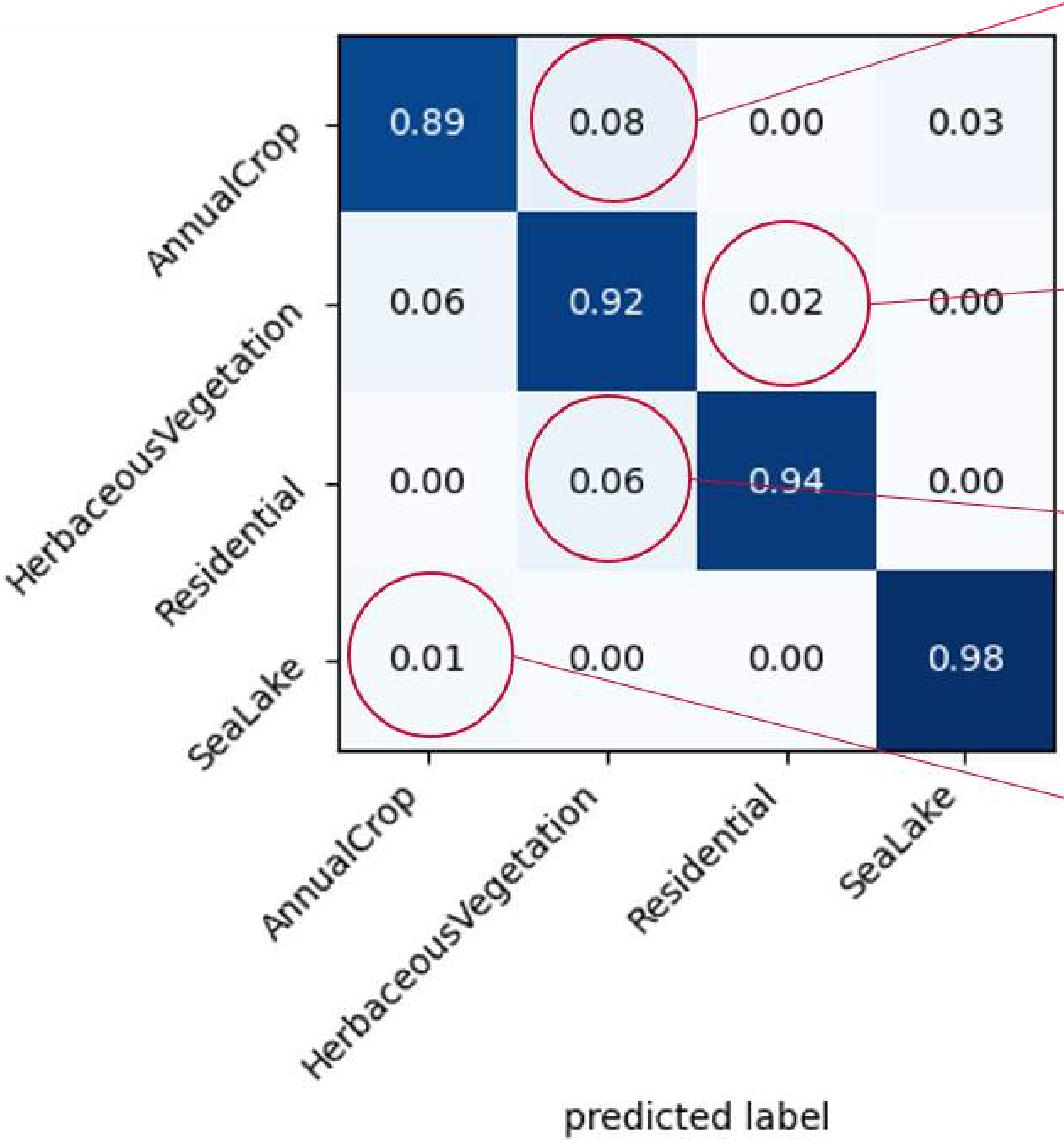


Residential

Annual Crop

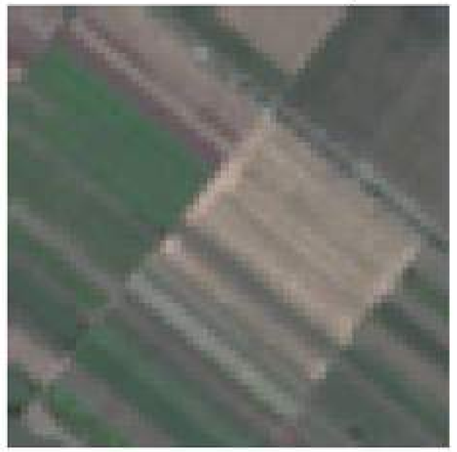
Vegetation

SeaLake



true label \ predicted label	AnnualCrop	HerbaceousVegetation	Residential	SeaLake
AnnualCrop	0.89	0.08	0.00	0.03
HerbaceousVegetation	0.06	0.92	0.02	0.00
Residential	0.00	0.06	0.94	0.00
SeaLake	0.01	0.00	0.00	0.98

P: HerbaceousVegetation
T: AnnualCrop



P: Residential
T: HerbaceousVegetation

P: HerbaceousVegetation
T: Residential

P: AnnualCrop
T: SeaLake

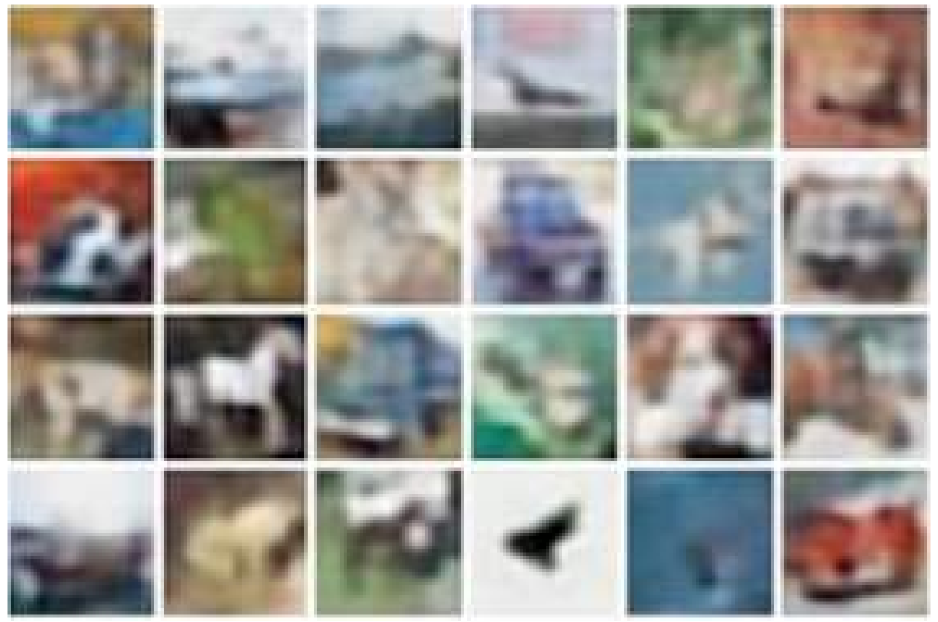
3

Generative models: auto-encoders

Generative models

Type of machine learning model that aims to learn the underlying patterns or distributions of data to generate new, similar data. Given a training dataset, it can generate new samples of the same data distribution.

Training data with a given distribution P_data(x)



Samples generated form a distribution P_model(x)

The generative model learns the distribution P_model(x) so that it is as similar as possible to P_data(x)

These models can do very different things:

- define and solve for P_model(x)
- learn a model that can sample from P_model(x) without explicitly defining P_model
- create artworks
- create generative models of time series of data that can be used for simulations and planning
- ...

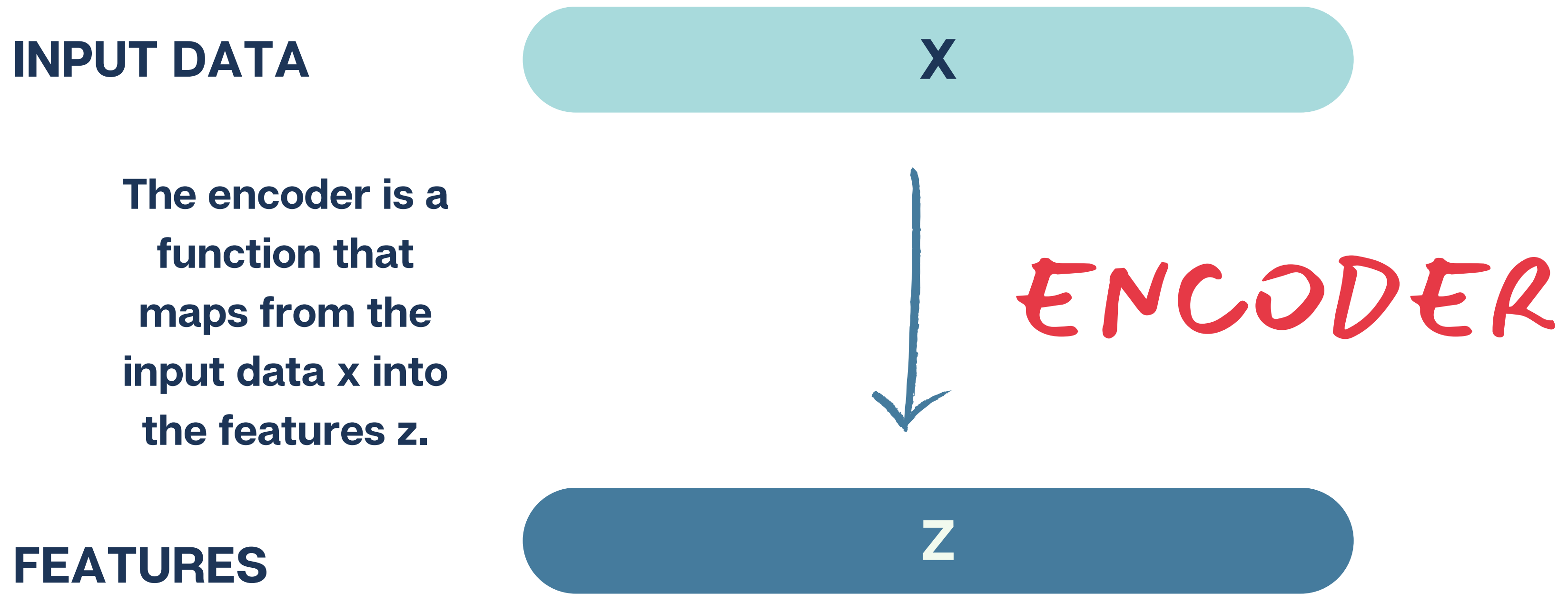
Autoencoders

Autoencoders are an **unsupervised approach** for learning a low dimensional feature representation from unlabeled training data.
Autoencoders **can reconstruct data**, and can **learn features to initialize a supervised model**

Autoencoders

Autoencoders are an **unsupervised approach** for learning a low dimensional feature representation from unlabeled training data. Autoencoders **can reconstruct data**, and can **learn features to initialize a supervised model**

Imagine that we have some input data x, and we want to learn from x some features that we will call z. Usually, z has a lower dimension than x (dimensionality reduction)



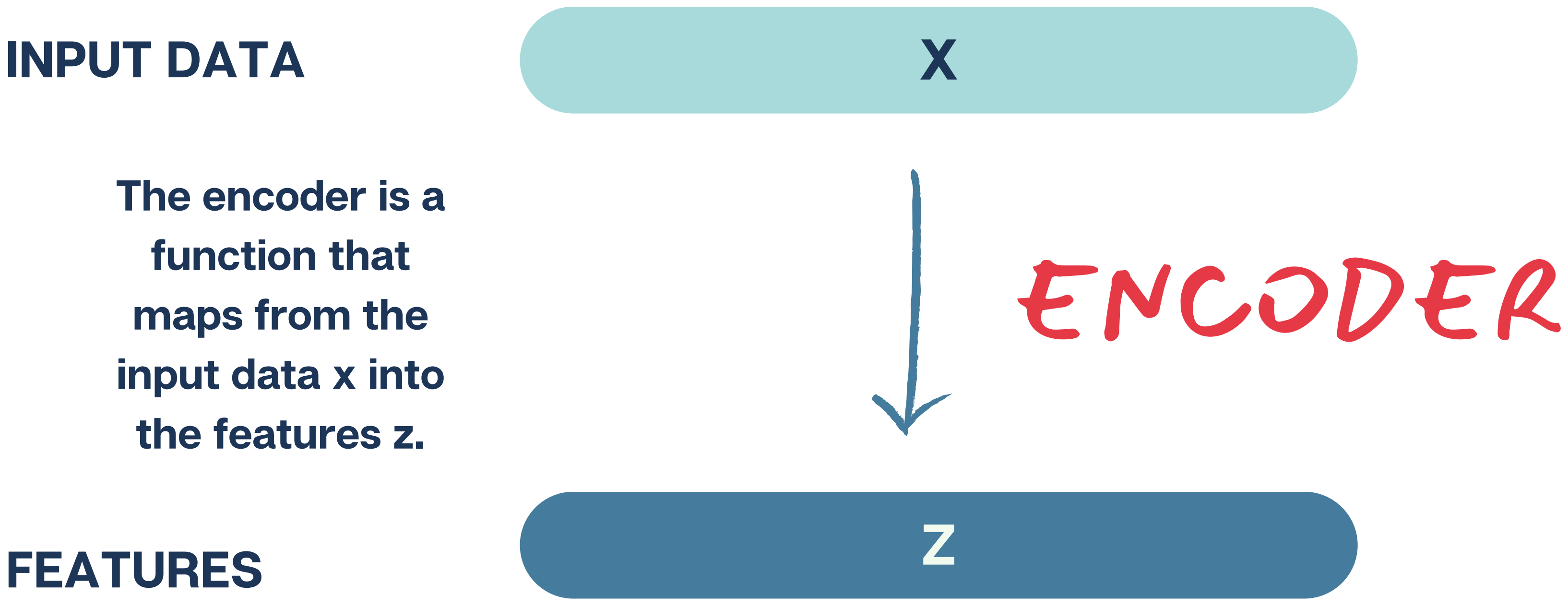
The encoder is a function that maps from the input data x into the features z.

z should consist of features that can capture meaningful factors of variations in the data (good features) and this is why they are less than x

Autoencoders

Autoencoders are an **unsupervised approach** for learning a low dimensional feature representation from unlabeled training data. Autoencoders **can reconstruct data**, and can **learn features to initialize a supervised model**

Imagine that we have some input data x, and we want to learn from x some features that we will call z. Usually, z has a lower dimension than x (dimensionality reduction)



The encoder is a function that maps from the input data x into the features z.

z should consist of features that can capture meaningful factors of variations in the data (good features) and this is why they are less than x

Training

The training should be such that the features we obtain can be used to reconstruct the original data. This is the reason for the name “auto-encoding”, that means encoding themselves.

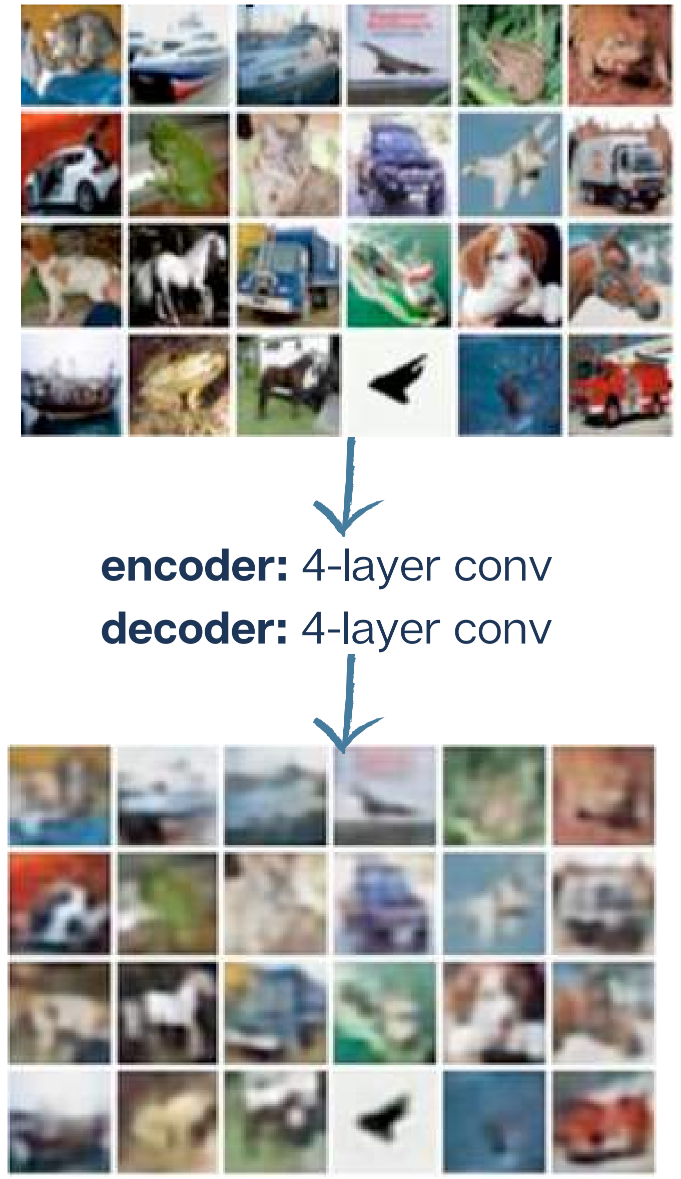
Encoder architecture

Typically the encoder architecture has changed with time:
first --> **linear systems** with the addition of the sigmoid nonlinearity,
then --> **deep fully connected architecture**
finally --> a **relu activation with a CNN**

Stanford course on convolutional neural network for computer vision (https://tinyurl.com/courselinkccomputervision)

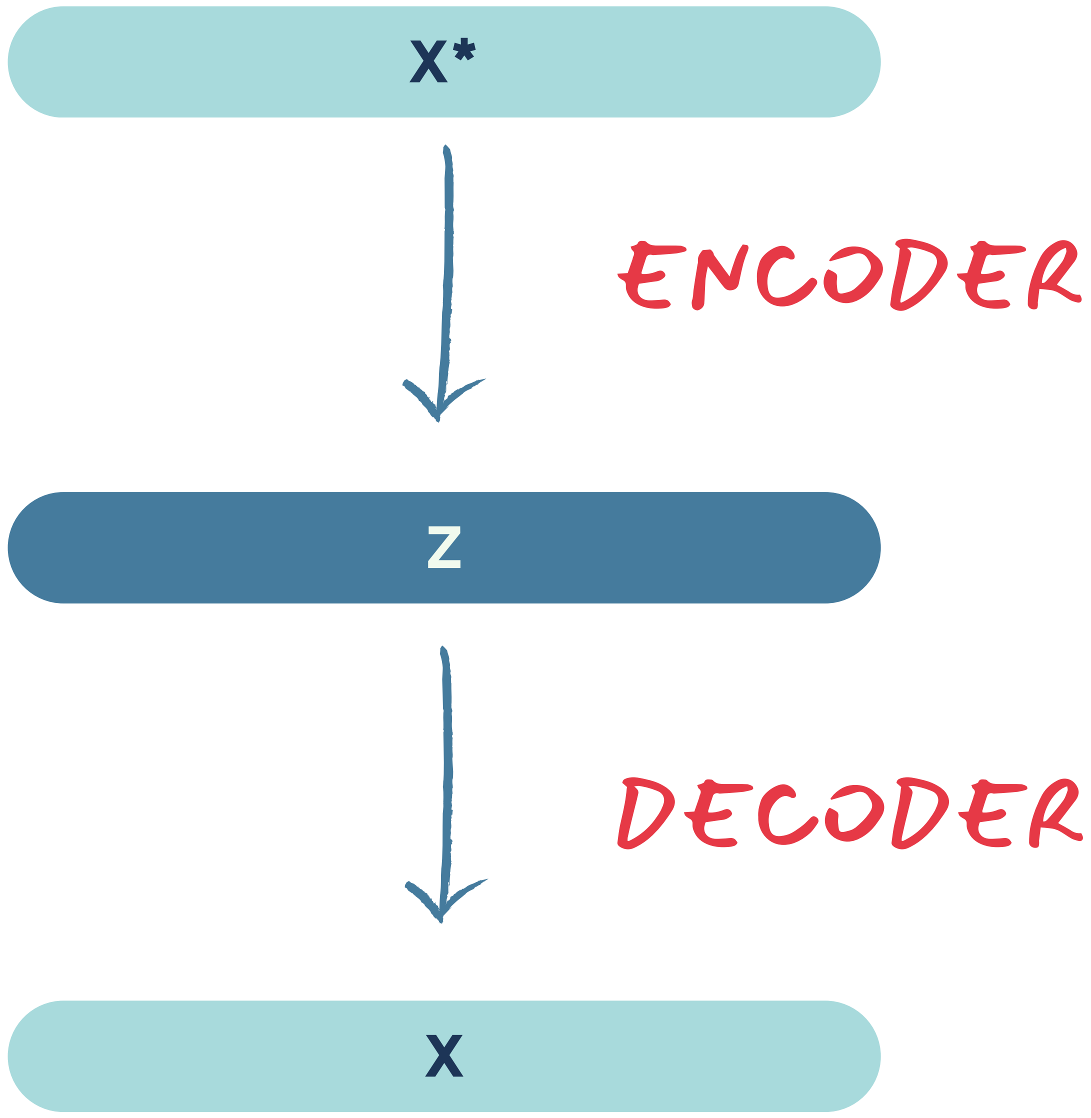
The decoder part

The decoder is a second network that receives the features that were produced based on the input and outputs something that has the same dimensionality of x, so something similar to x.



encoder: 4-layer conv
decoder: 4-layer conv

Source: Examples from lecture series cs231_n from Stanford University, (https://tinyurl.com/courselinkccomputervision)



INPUT DATA

X*

ENCODER

FEATURES

Z

DECODER

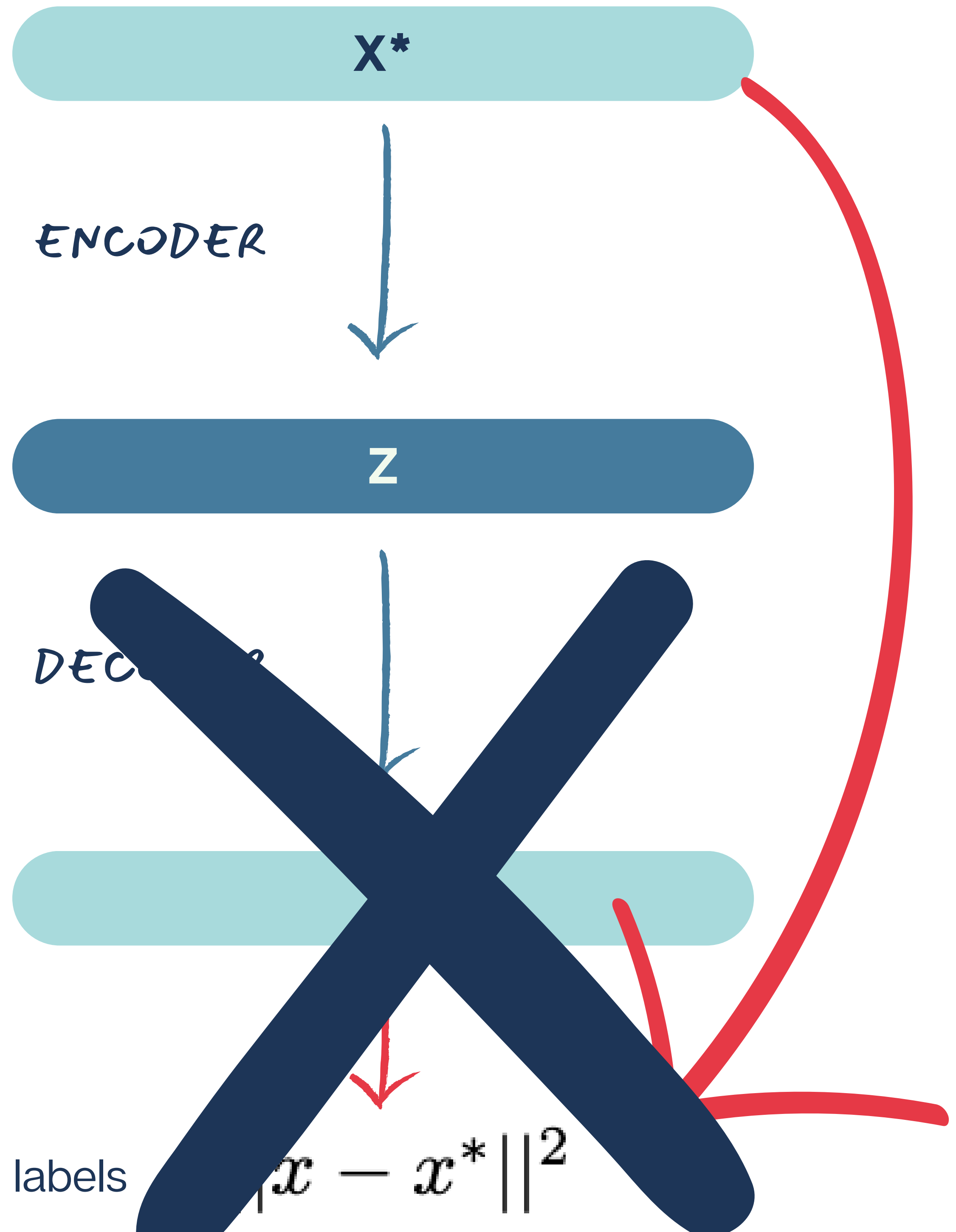
RECONSTRUCTED INPUT DATA

X

For the decoder, we are using the same type of network architectures used in the encoders, i.e. CNN for most of the time

Training

We then train such features in such a way to be used to reconstruct the original data. No external labels are used to train this network. Instead, we use a **loss function which is similar to a L2 loss distance**, that acts at pixel scale, trying to make the pixels of the reconstructed image to be the same as the ones of the input data.



INPUT DATA

FEATURES

RECONSTRUCTED INPUT DATA

After training, we can throw away the decoder. It was used to produce our reconstruction and provided the input to compute the loss function

Value: use a lot of untrained data to learn good general feature representations. It can be used also to initialize a supervised learning problem when we don't have enough labelled data.

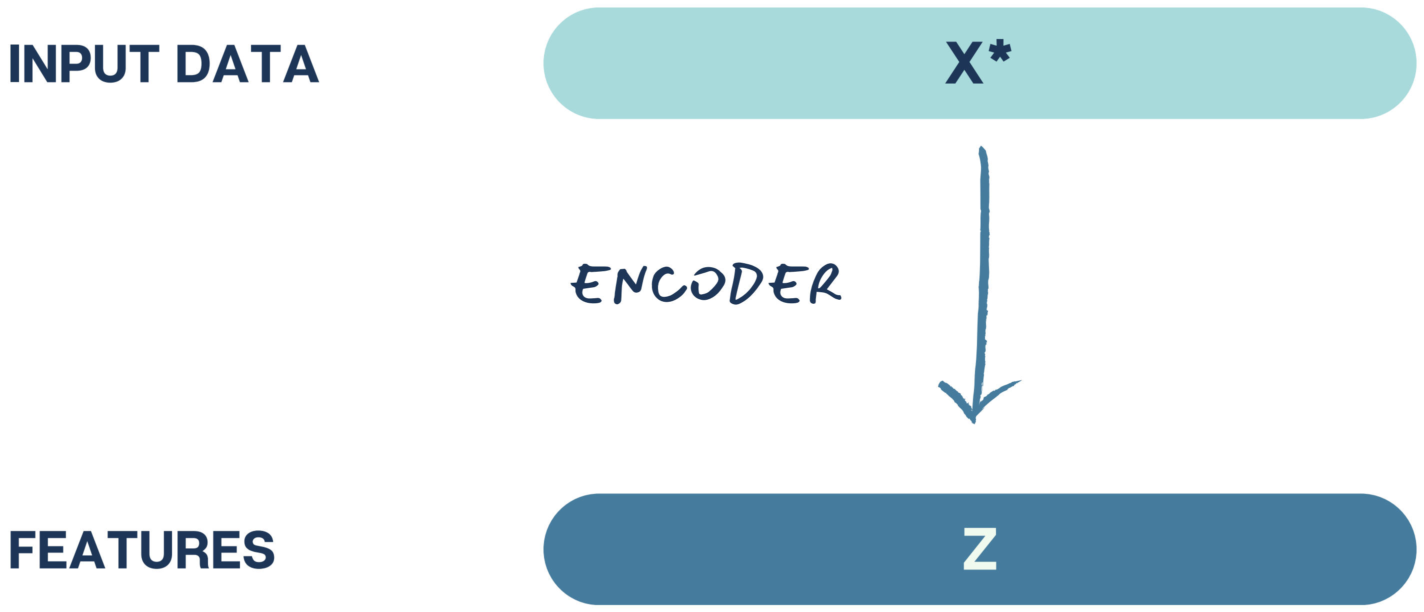
This loss function does not use labels

L2 LOSS FUNCTION

Training

We then train such features in such a way to be used to reconstruct the original data. No external labels are used to train this network. Instead, we use a **loss function which is similar to a L2 loss distance**, that acts at pixel scale, trying to make the pixels of the reconstructed image to be the same as the ones of the input data.

The encoder can be used to produce a feature mapping with which we can initialize a supervised model.
We can also connect the features to an additional classifier network on top of the encoder, to provide a class label for a classification problem.



INPUT DATA

X*

ENCODER

FEATURES

Z

Value: use a lot of untrained data to learn good general feature representations. It can be used also to initialize a supervised learning problem when we don't have enough labelled data.

4

Application of CNNs: Cloud classification

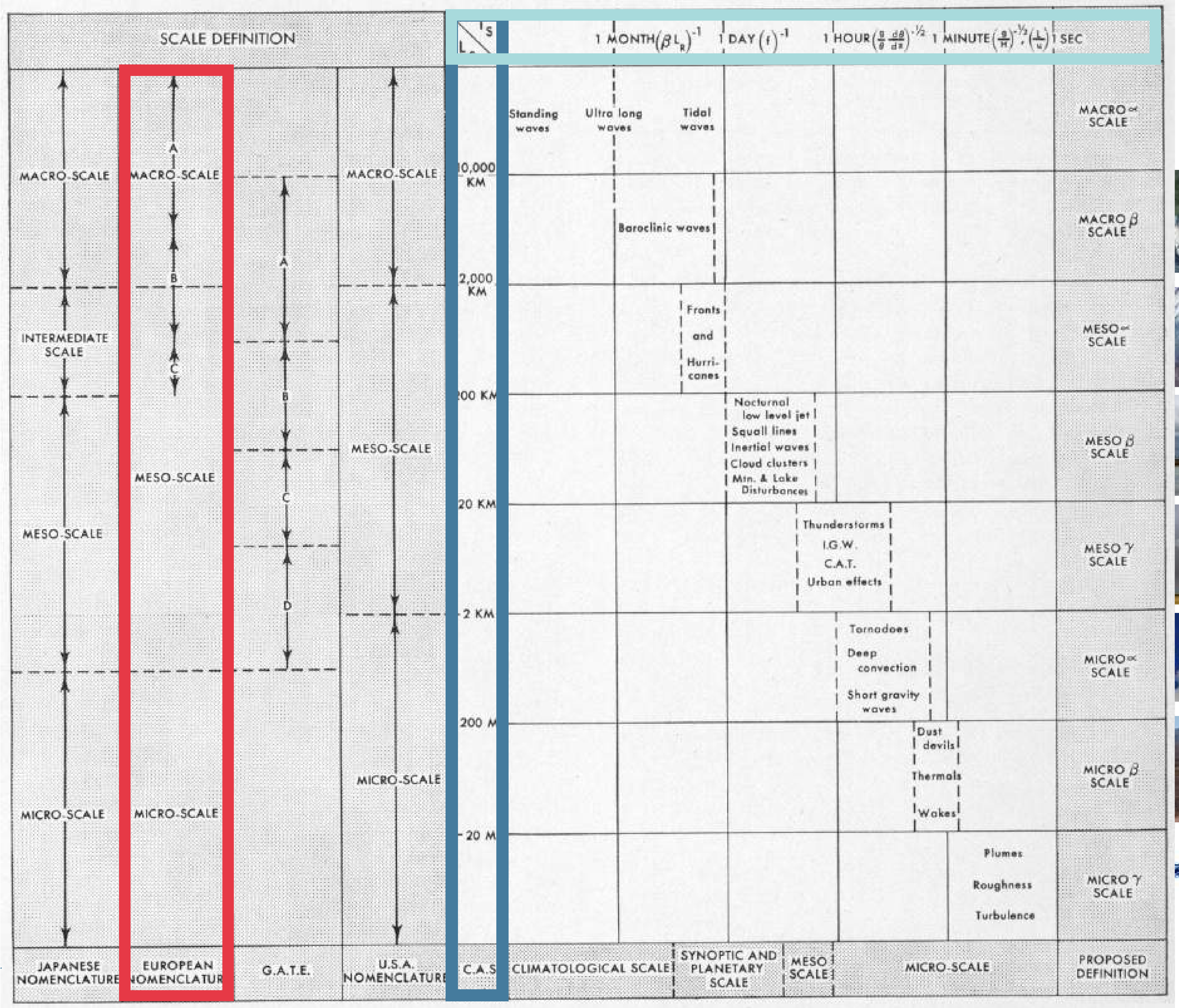
A year of weather 2022
Share
01 January 2022
Watch on YouTube

Is cloud mesoscale organization relevant?

Meteorological scales (1975)

A large number of atmospheric processes occur at scales in between the scale of meters and the scale of thousands of Km.

With the term mesoscale, we refer to all the states in between the micro scale and the macro scale.



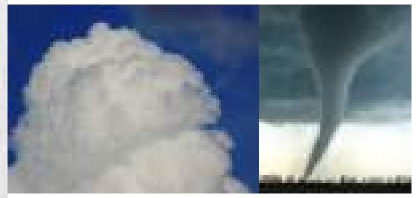
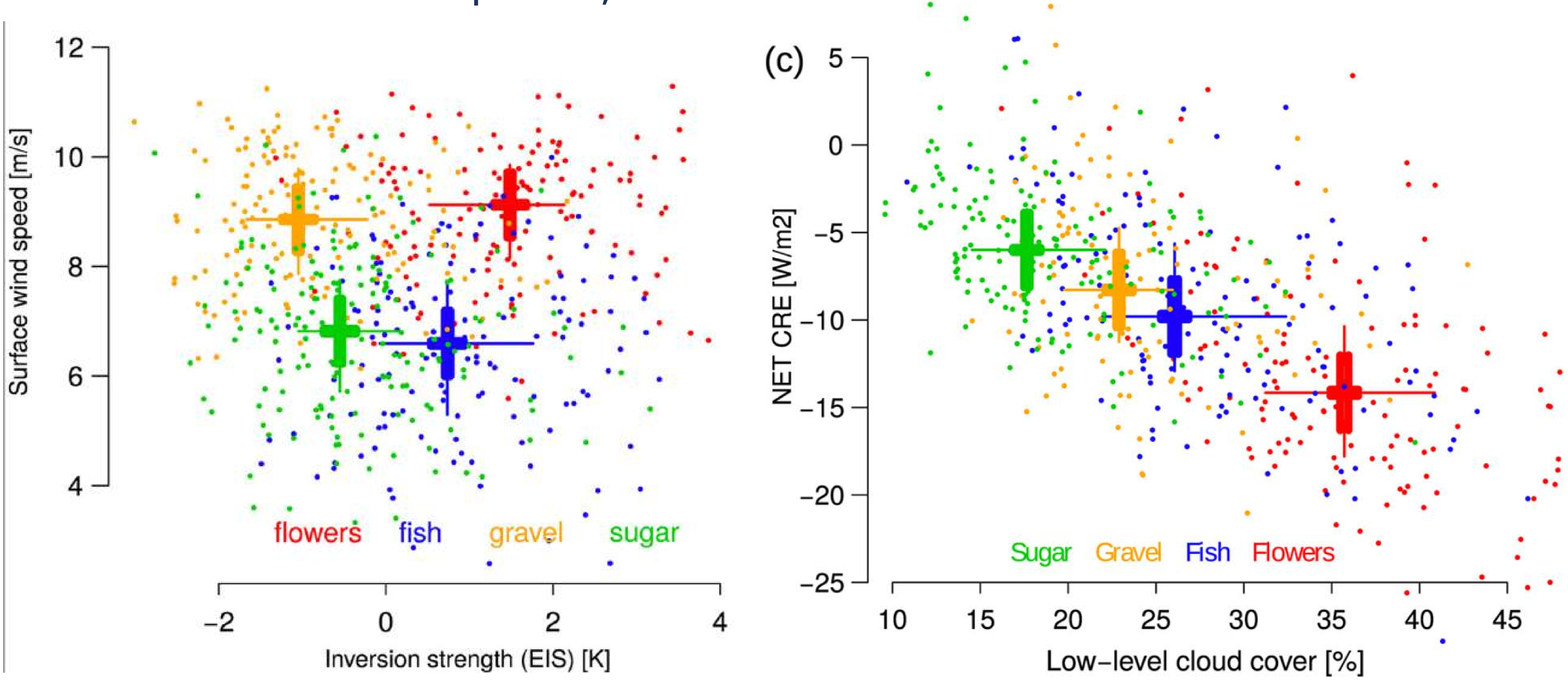


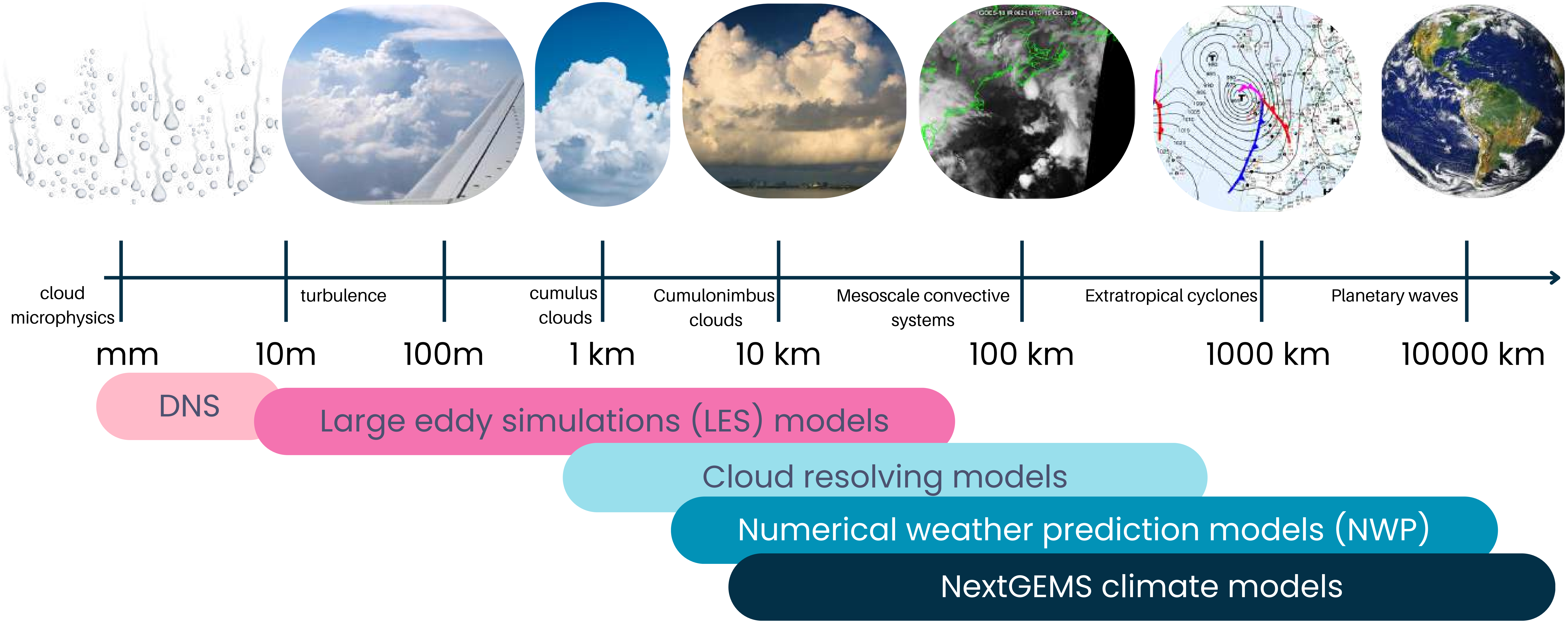
Figure 1 from A Rational Subdivision of Scales for Atmospheric Processes. Bulletin of the American Meteorological Society, 56(5), 527–530. http://www.jstor.org/stable/26216020

Different cloud mesoscale patterns have different properties and radiative responses, which matter for climate



Bony, S., Schulz, H., Vial, J., & Stevens, B. (2020). Sugar, gravel, fish and flowers: Dependence of mesoscale patterns of trade-wind clouds on environmental conditions. Geophysical Research Letters, 47, e2019GL085988.

and **numerical models** should **represent** such **processes** and their interactions



cloud microphysics

turbulence

cumulus clouds

Cumulonimbus clouds

Mesoscale convective systems

Extratropical cyclones

Planetary waves

mm

10m

100m

1 km

10 km

100 km

1000 km

10000 km

DNS

Large eddy simulations (LES) models

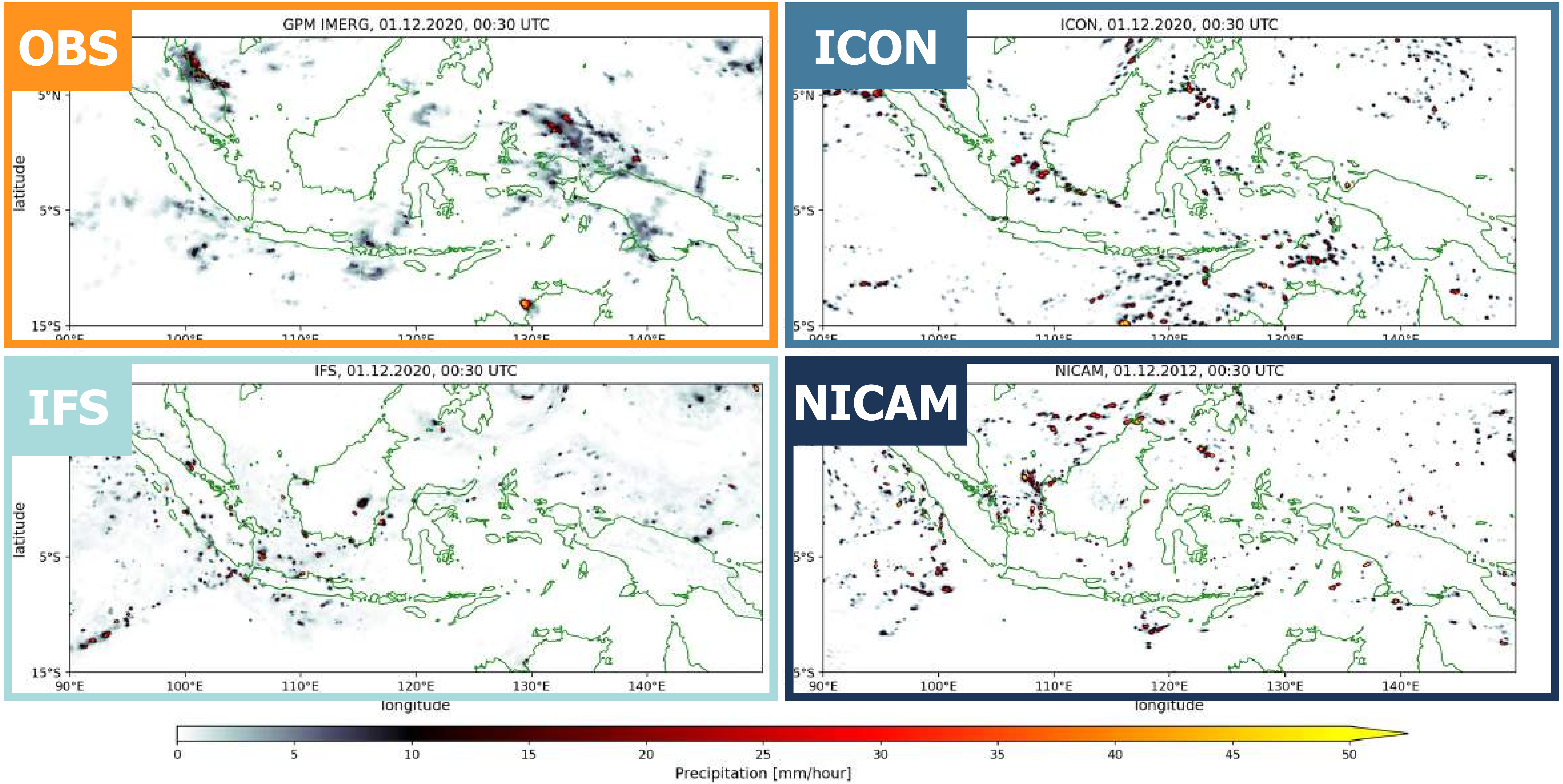
Cloud resolving models

Numerical weather prediction models (NWP)

NextGEMS climate models

Are they good in that?

Models resolving convection **do not reproduce** the same **aggregation** we see in the observations at the **mesoscale**


GPM IMERG, 01.12.2020, 00:30 UTC

ICON, 01.12.2020, 00:30 UTC

IFS, 01.12.2020, 00:30 UTC

NICAM, 01.12.2012, 00:30 UTC

Figure: Becker, T., Takasuka, D., and Bao, J.: Characteristics of precipitating convection and moisture-convection relationships in global km-scale simulations, EGU General Assembly 2024, Vienna, Austria, 14–19 Apr 2024, EGU24-17683, https://doi.org/10.5194/egusphere-egu24-17683, 2024.,

paper in preparation “"On the influence of moisture-convection relationships on precipitating convection in global km-scale simulations” from Takasuka, Bao and Becker

from Becker et al., in prep,

Does mesoscale convective aggregation matter?

is crucial in the dynamics of tropical convection.

contributes to the formation of large-scale weather systems like mesoscale convective complexes

is crucial for accurate forecasting in NWP and climate models

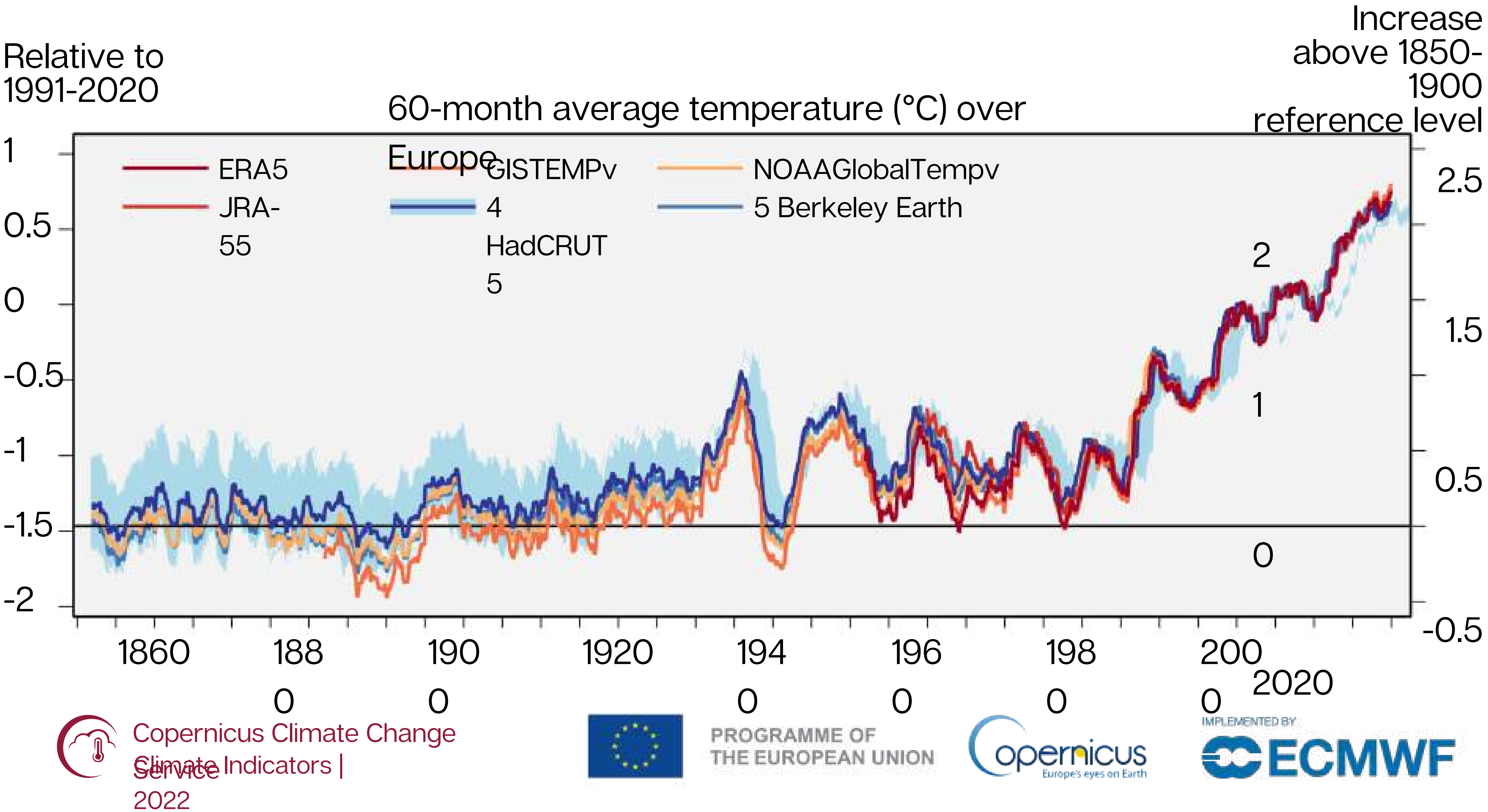
it influences the mean state of the atmosphere by affecting the distribution of convective heating.

What forms of organization do we have in nature?
Can we reproduce them?

What forms of organization do we have in nature?
Can we reproduce them?

not really, so we have a difficulty in representing these clouds in climate models.

The Earth is warming...



Global average near-surface temperature for centred running 60-month periods, relative to the average for the 1991–2020 reference period (left-hand axis) and as an increase above the 1850–1900 level (right-hand axis), according to six datasets. The average temperature for 1991-2020 from ERA5 is 14.4°C. Data sources: ERA5 (C3S/ECMWF), JRA-55 (JMA), GISTEMPv4 (NASA), HadCRUT5 (Met Office Hadley Centre), NOAAGlobalTempv5 (NOAA) and Berkeley Earth. Credit: C3S/ECMWF. From https://climate.copernicus.eu/climate-indicators/temperature

Since 1850—1900, an increase in surface air temperature of around

Globe

+1.2°C ↗

Europe

+2.2°C ↗

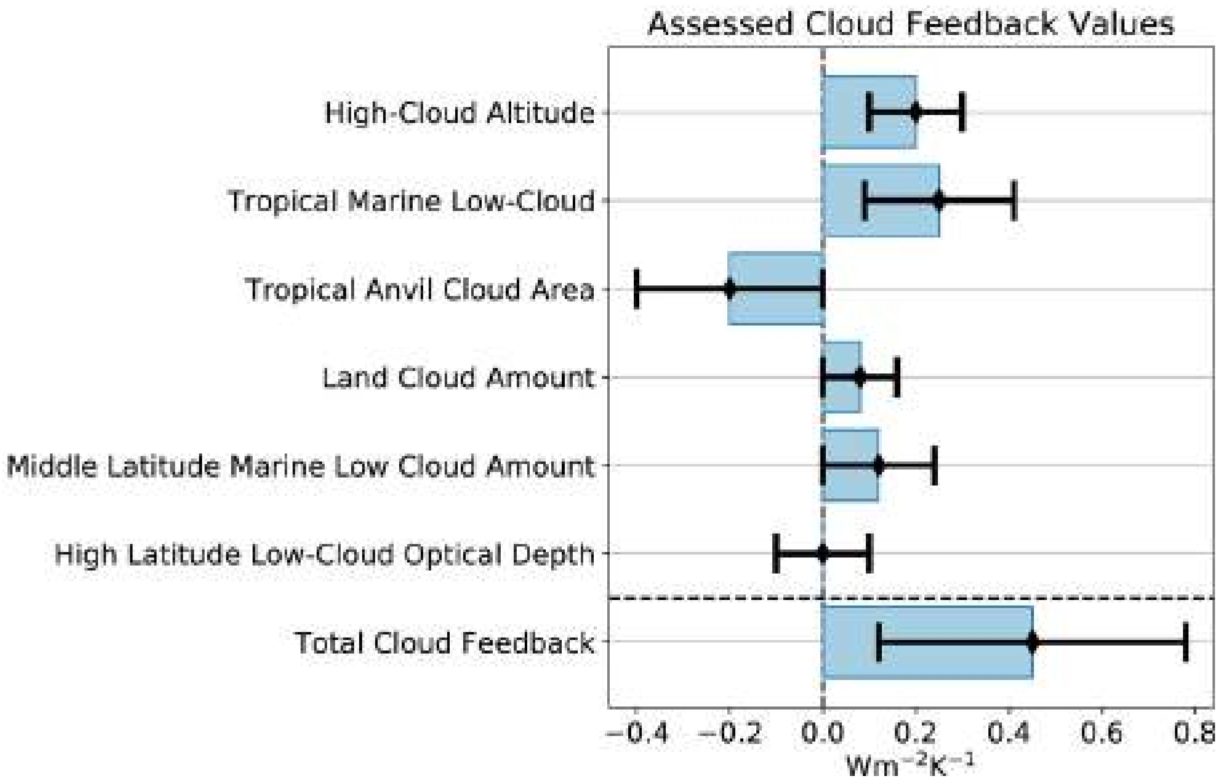
Arctic

+3°C ↗

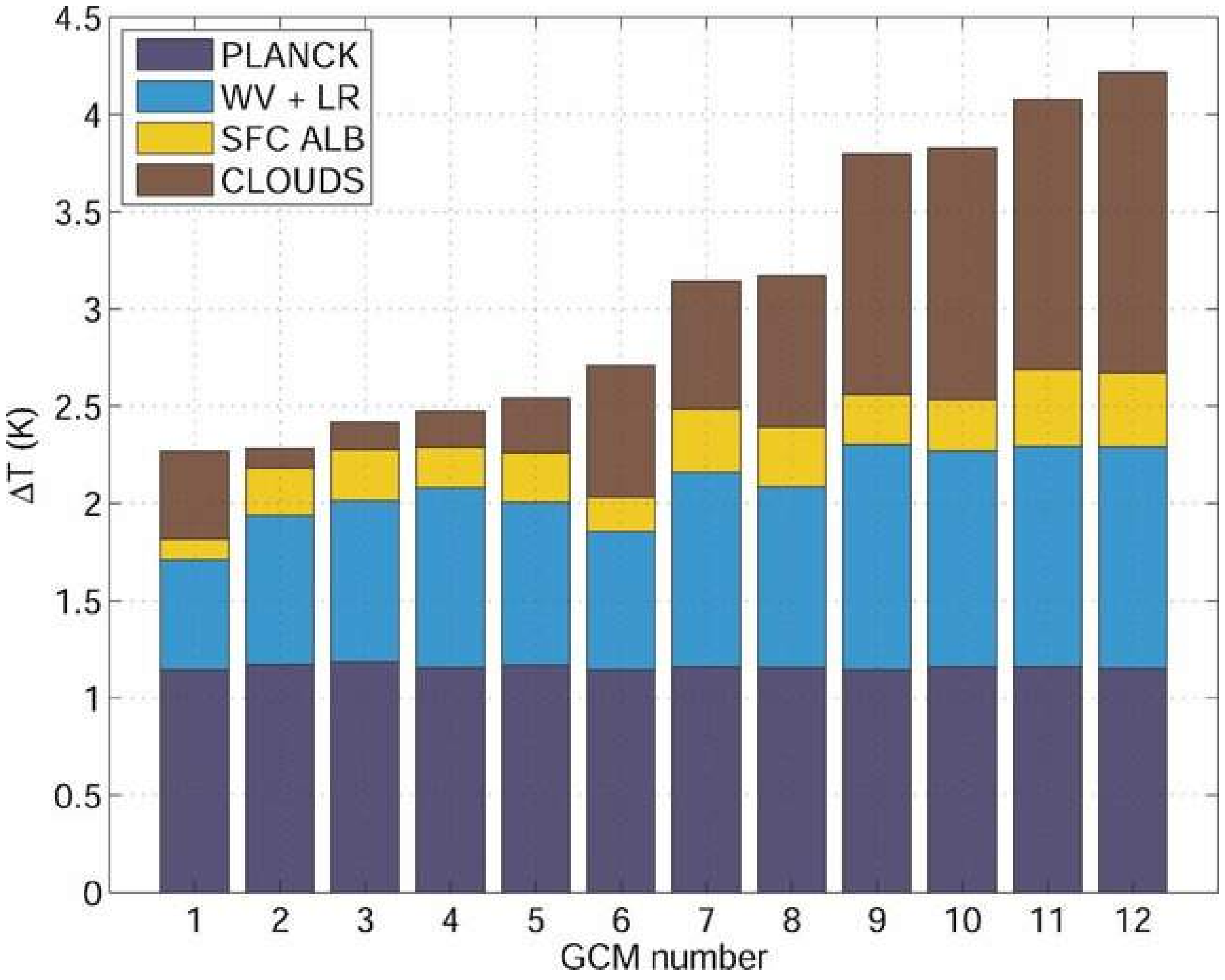
(For latest five-year averages)

...and climate models struggle to predict how clouds will respond to climate change.

Differing predictions, between different climate models, for how these clouds will respond to a warming climate account for most of the variation in climate sensitivity between models (Bony & Dufresne, 2005; Medeiros et al., 2008; Vial et al., 2013; Webb et al., 2006)

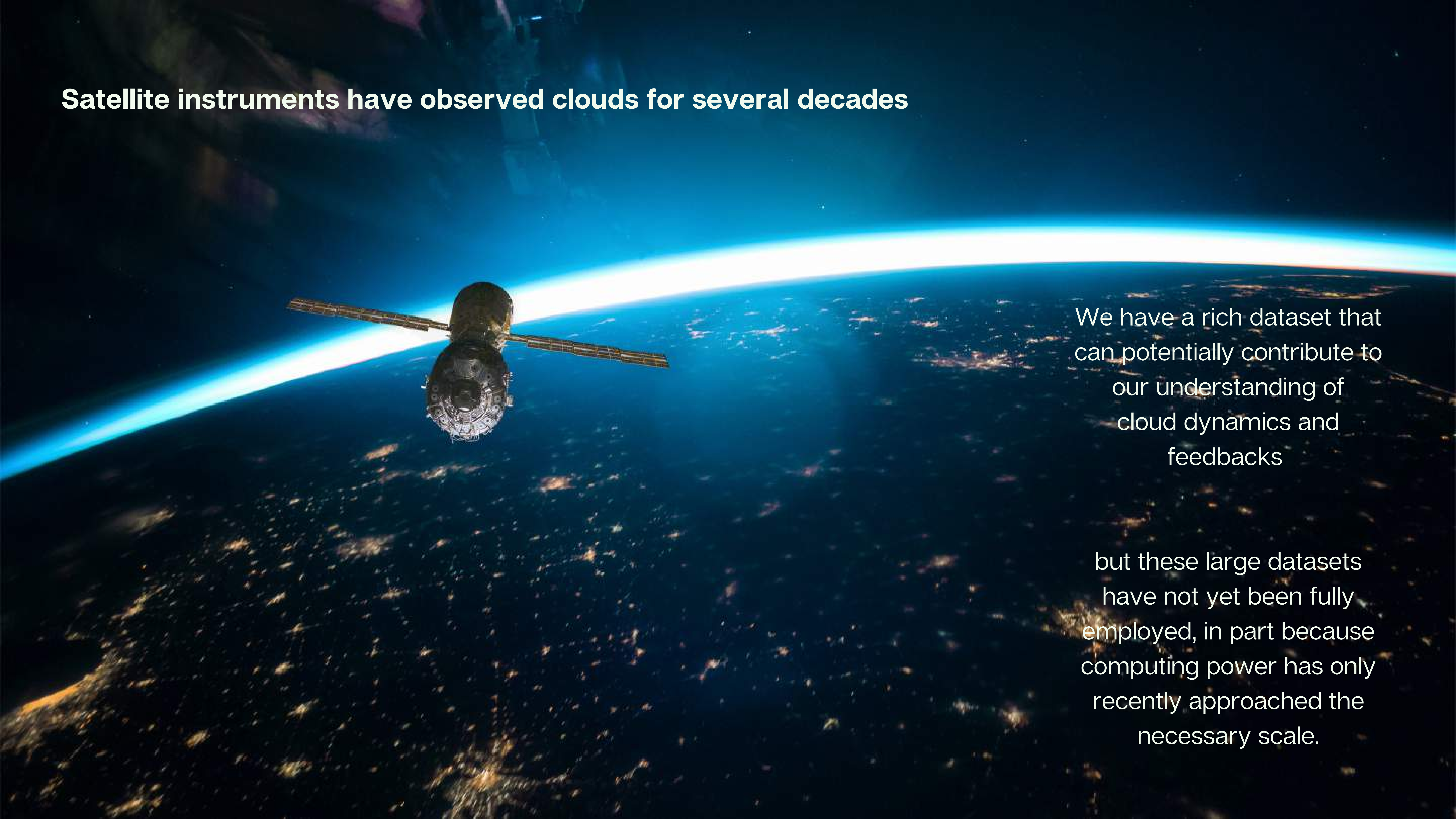


*Assessed values of individual cloud feedbacks and the total cloud feedback based upon process evidence. For individual cloud feedbacks, maximum likelihood values are shown by black diamonds and the widths of blue rectangles, with 2 times t**he 1-sigma likelihood values shown by the width of the black uncertainty bars**. For the total cloud feedback, the mean value of the PDF is shown by a black diamond and the width of the accompanying blue rectangle, with 2 times the PDF standard deviation shown by the width of the black uncertainty bar. From Sherwood, S. C., Webb, M. J., Annan, J. D., Armour, K. C., Forster, P. M., Hargreaves, J. C., et al. (2020). An assessment of Earth's climate sensitivity using multiple lines of evidence. Reviews of Geophysics, 58, e2019RG000678. https://doi.org/10.1029/2019RG000678*



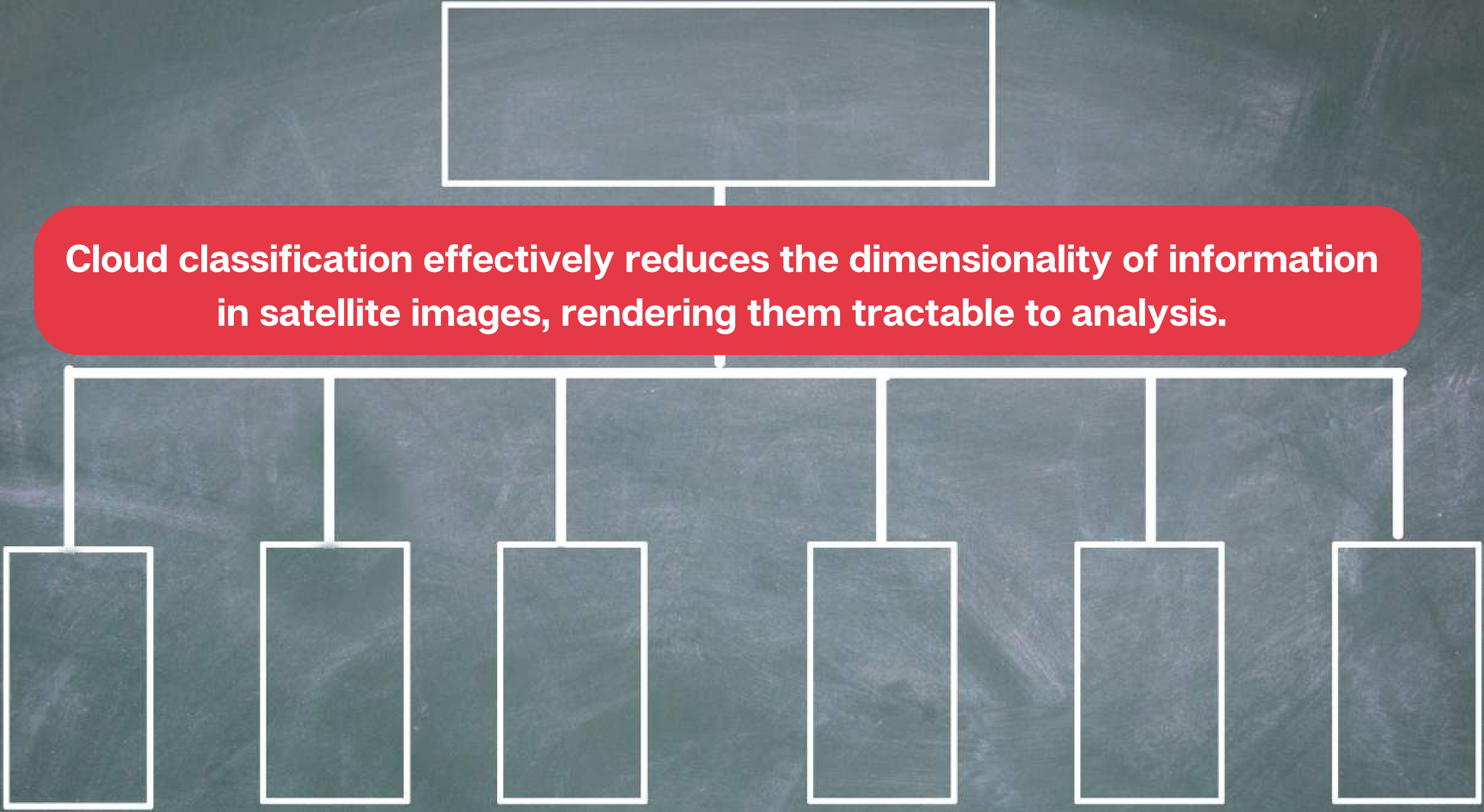
From Dufresne, J., and S. Bony, 2008: An Assessment of the Primary Sources of Spread of Global Warming Estimates from Coupled Atmosphere–Ocean Models. J. Climate, 21, 5135–5144, https://doi.org/10.1175/2008JCLI2239.1.

Satellite instruments have observed clouds for several decades



We have a rich dataset that can potentially contribute to our understanding of cloud dynamics and feedbacks

but these large datasets have not yet been fully employed, in part because computing power has only recently approached the necessary scale.



Cloud classification effectively reduces the dimensionality of information in satellite images, rendering them tractable to analysis.

5

Ways of learning

Supervised learning

Data (x,y)

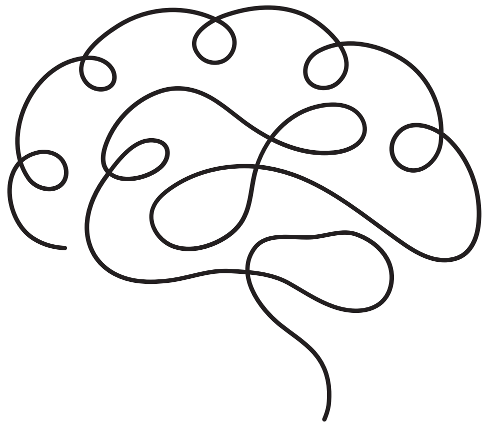
x is data

y is label

Golden retriever

Goal

Learn a function that can map x into y

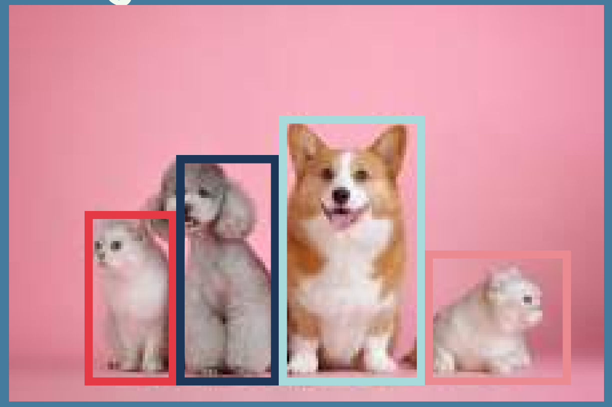


Examples

Classification

→ CAT

Object detection

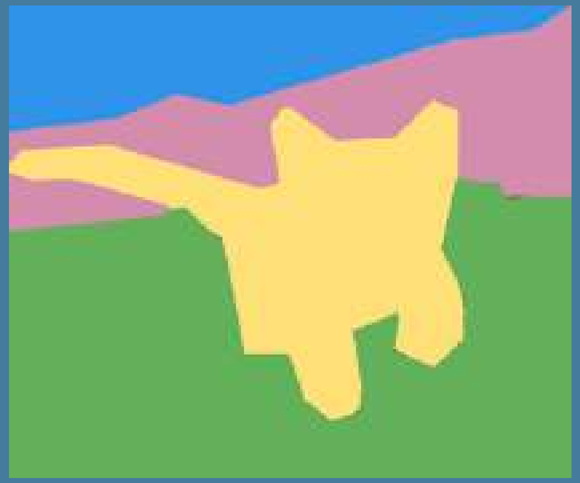


CAT, DOG, DOG, CAT

Image captioning

A dog is playing with a ball on the floor

Image segmentation



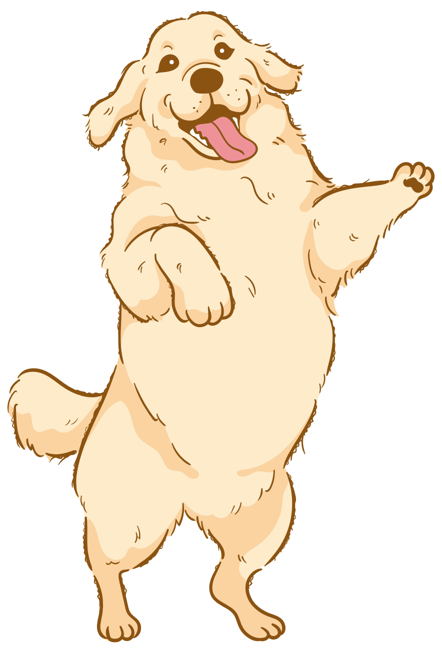
GRASS, CAT, SKY

Source: Examples from lecture series cs231_n from Stanford University (https://www.youtube.com/redirect?event=video_description&redir_token=QUFFLUhqbXhQMVMwcXQ3bGgwdE9FdVBQemJVNzRlb0lUQXxBQ3Jtc0ttbjAtWEtHUF9KUXhsYjM4SDNRQlhBeFM1NG1vMkE3YjhzQVNsV1FXQlRQZk9WN0dra0NuWVRMS0pVdzJxS2FlemtxLS00b0xDenpzREJER2doTnN0dklJYjl5V3hVSWFuQVA0UWo2NWJhcTZLVmVZTQ&q=http%3A%2F%2Fcs231n.stanford.edu%2Fslides%2F2017%2Fcs231n_2017_lecture13.pdf&v=5WoItGTWV54)

Unsupervised learning

Data

Only data, no labels



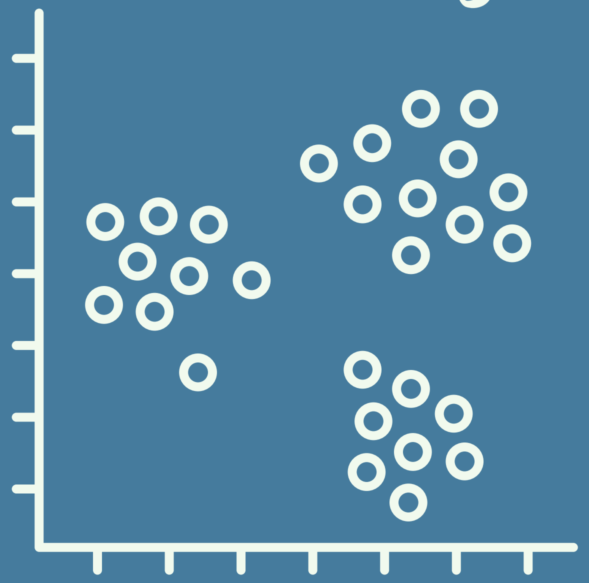
x is data

Goal

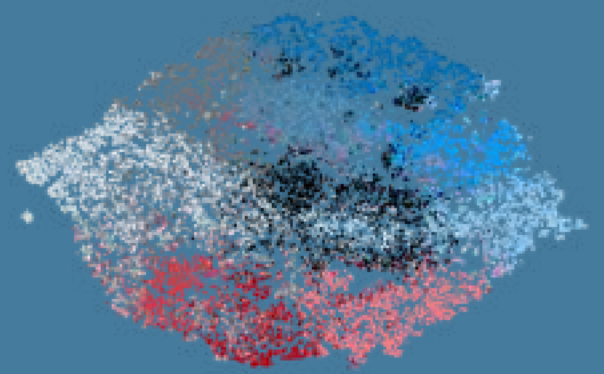
Learn underlying hidden structures in the data

Examples

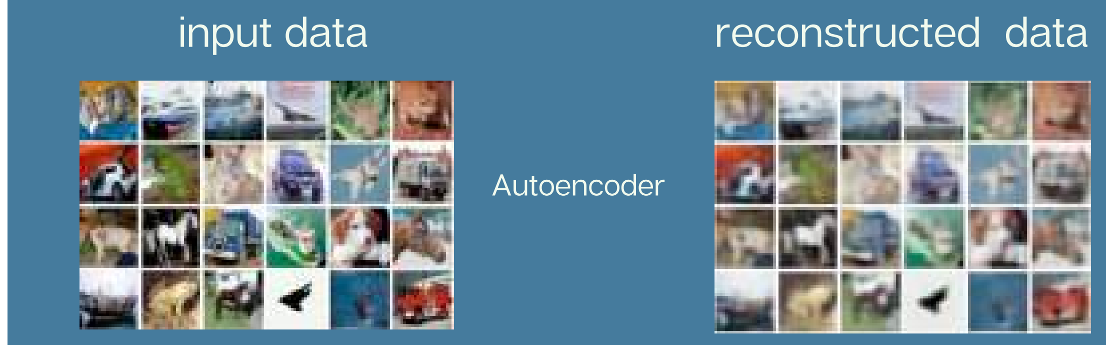
Clustering

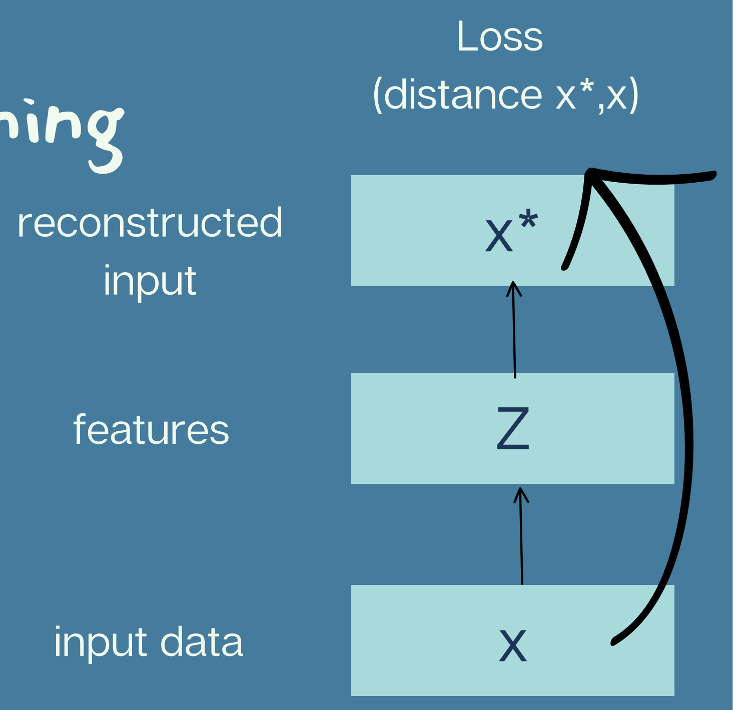


PCA and dimensionality reduction



Feature learning





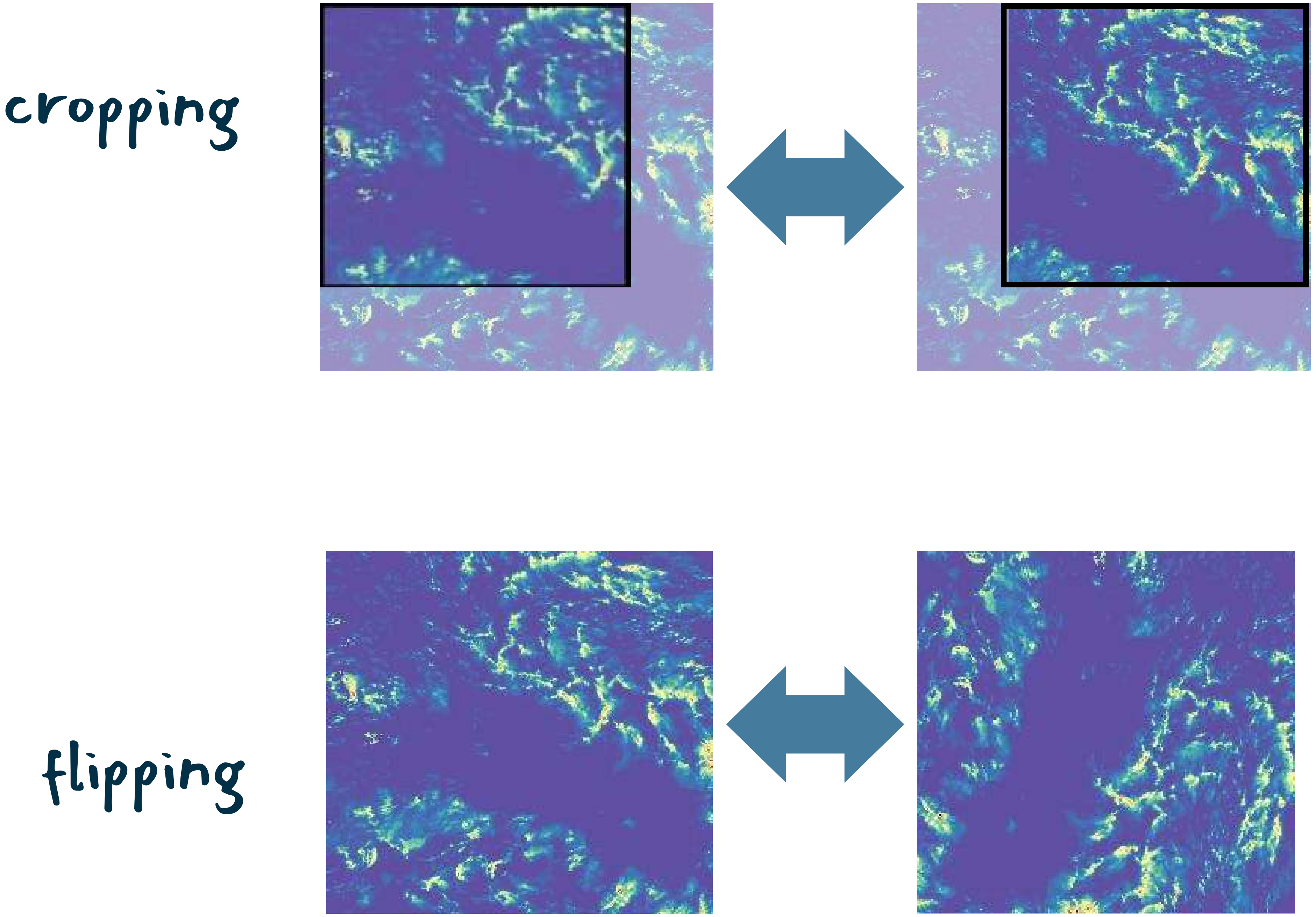
Self-supervised learning

In a self-supervised approach we don't have labels, but we ask the model to learn that the augmentation combinations created from our data are "similar", since they are different "versions" of the same image

Learning principle Contrastive learning: learn general features of a dataset without labels by teaching which data points are **similar** or **different** in a comparison among pairs

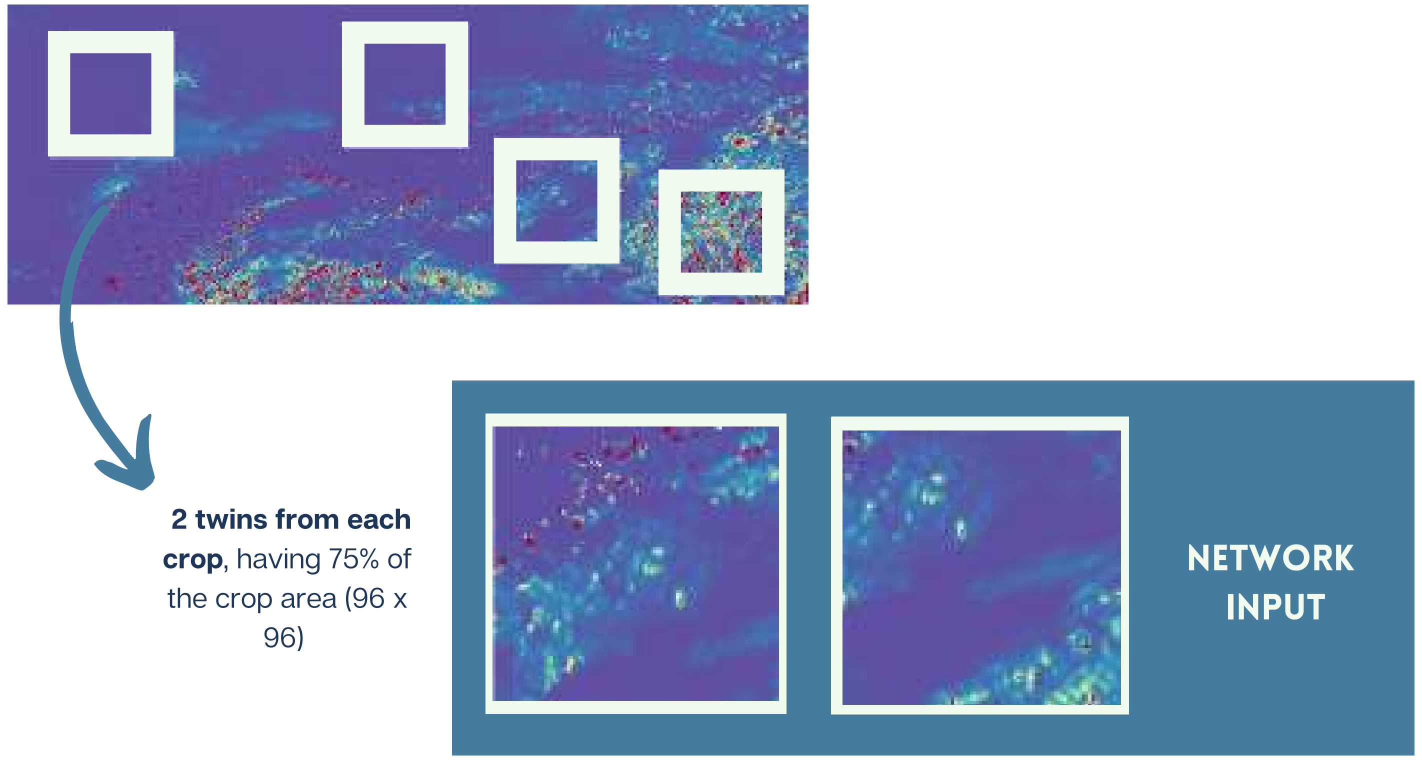
In this process, the unsupervised problem is transformed into a supervised problem by auto-generating the labels. (pseudolabels from the data itself)

To make use of the huge quantity of unlabeled data, it is crucial to set the right learning objectives to get supervision from the data itself.



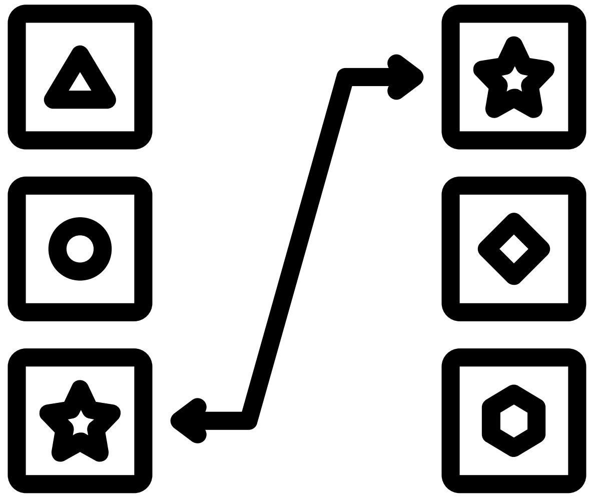
Self-supervised methods design a learning task that does not rely on human annotations.

1) from one image, you take two different crops



2 twins from each crop, having 75% of the crop area (96 x 96)

2) you build two representations of those crops and you try to match them



3) there are many ways to operate the matching: **contrastive learning**, **matching representations**, **cluster assignment**

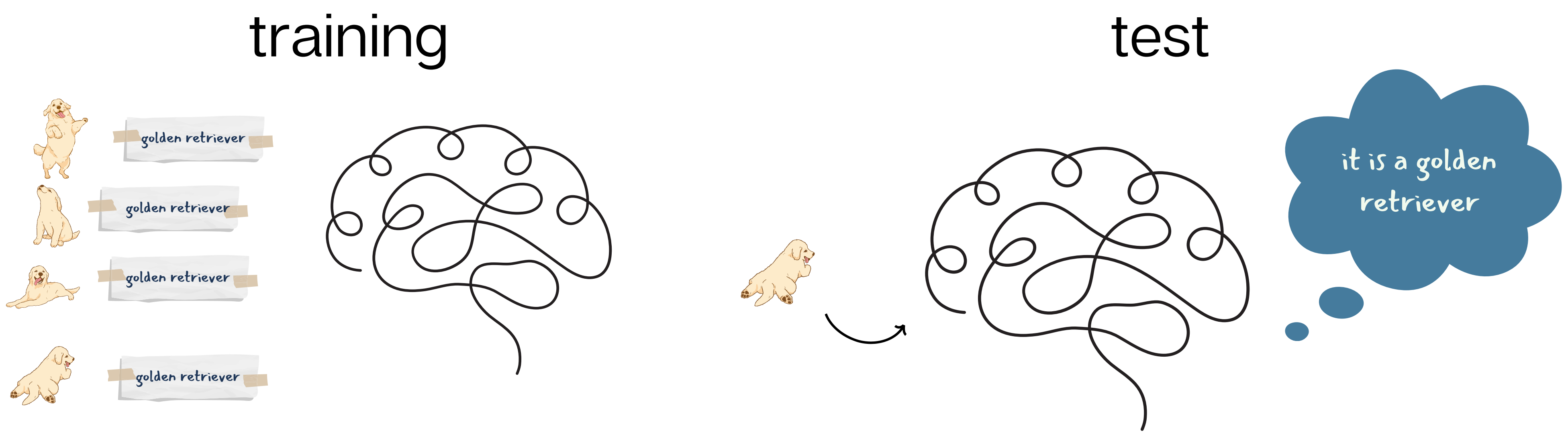
All in all, it is about learning invariance with respect to data augmentation and cropping.

Inspired from the tutorial on computer vision self-supervised approaches, contribution from Mathilde Caron (https://www.youtube.com/watch?v=MdD4UMshl1Q&t=13629s)

Summarizing

Supervised learning

training neural networks on **labeled data** for a specific task.

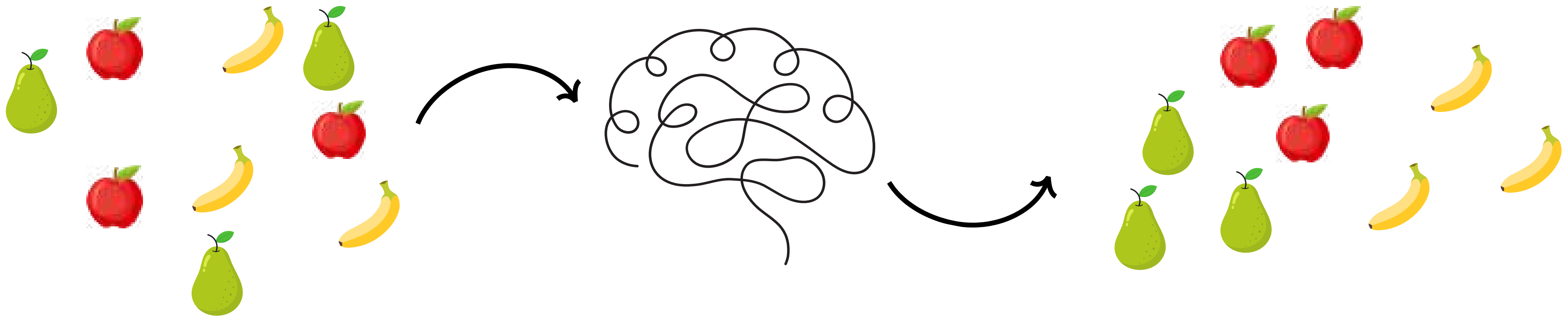


training

test

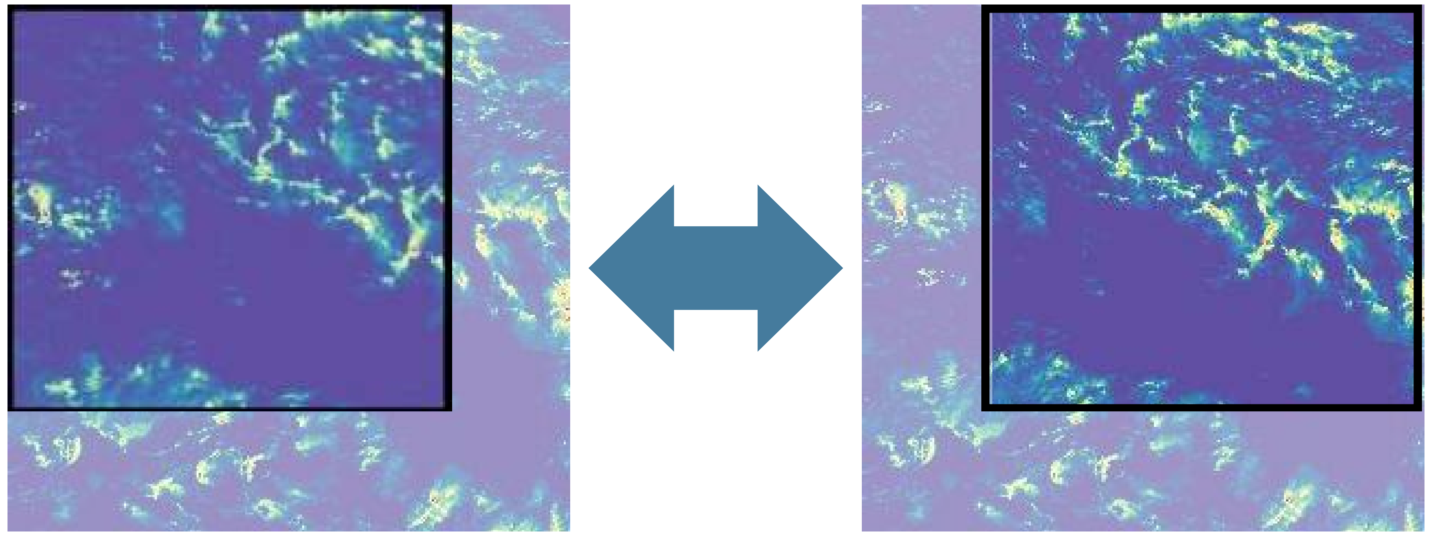
Unsupervised learning

find **implicit** patterns in the data **without being explicitly trained on labeled data**. Unlike supervised learning, it does not require annotations and a feedback loop for training. For e.g. clustering.



Self-supervised learning

predict any unobserved or hidden part (or property) **of the input from any observed** or unhidden part **of the input**. It is also known as **predictive** or **pretext** learning.



6

Cloud classification approaches: an overview

Overview of cloud classification algorithms

SUPERVISED METHODS

Stevens et al., 2019: mesoscale patterns in trade winds

Rasp et al., 2019: crowdsourcing and deep learning to explore mesoscale organization of shallow convection

UNSUPERVISED METHODS

Denby, 2019: Discovering the importance of mesoscale cloud organization through unsupervised classification

Kurihana et al. 2022: Cloud classification with unsupervised deep learning

SELF-SUPERVISED METHODS

Chatterjee et al., 2022: Understanding Cloud Systems' Structure and Organization Using a Machine's Self-Learning Approach

Overview of cloud classification algorithms

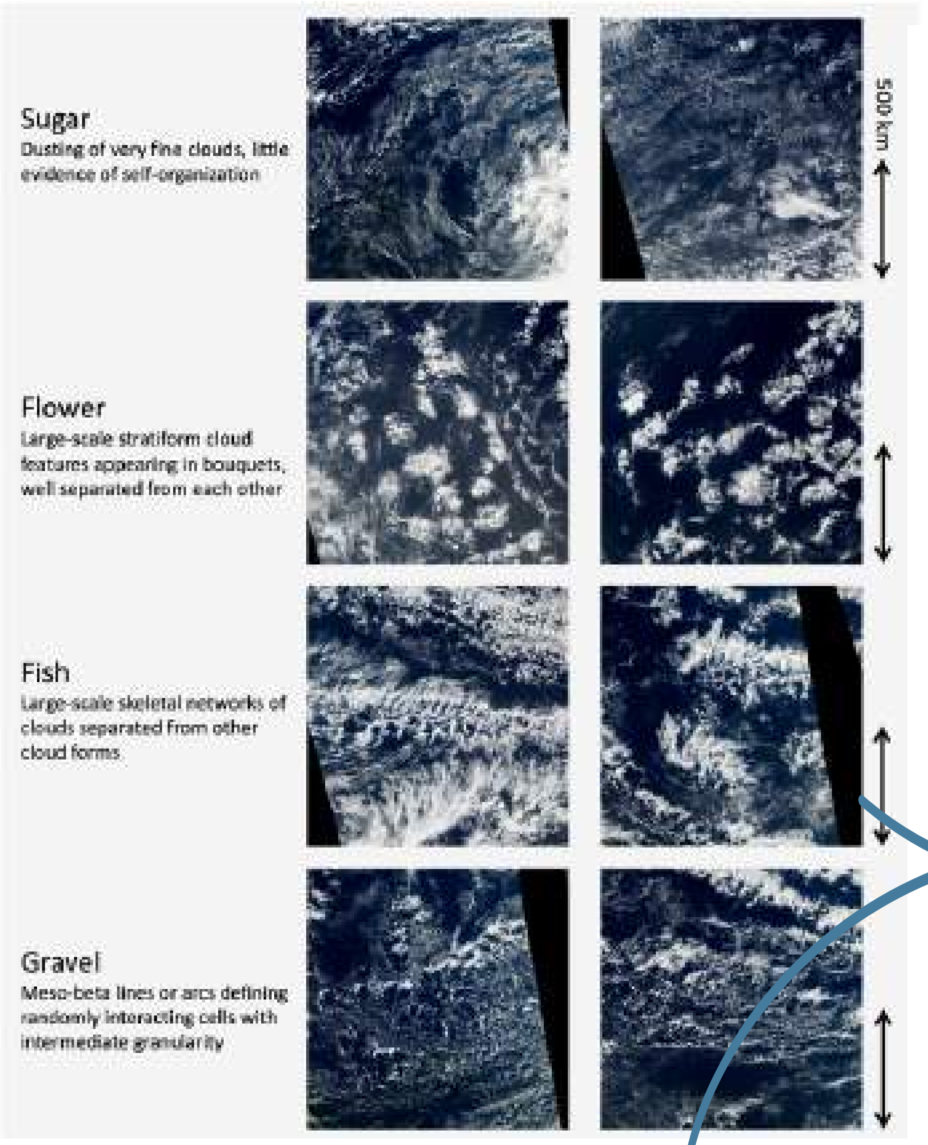
SUPERVISED METHODS

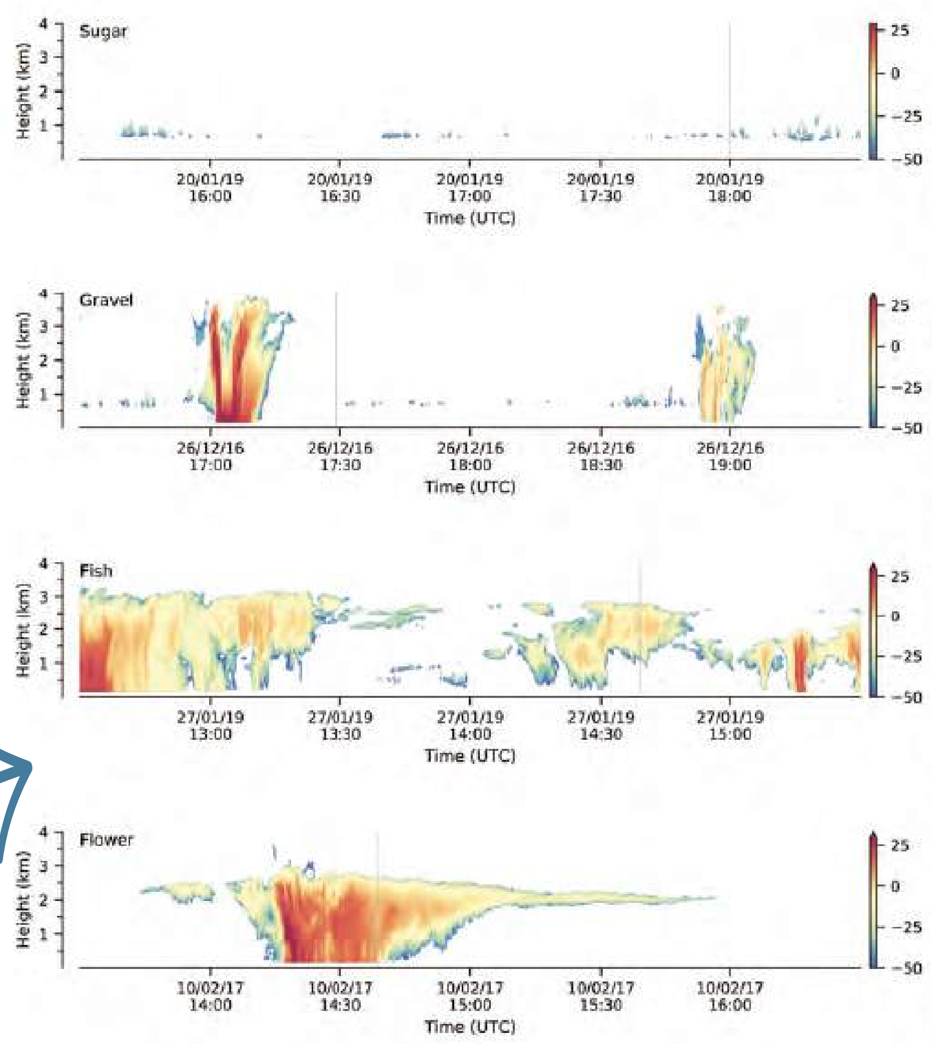
Stevens et al., 2019: mesoscale patterns in trade winds

Rasp et al., 2019: crowdsourcing and deep learning to explore mesoscale organization of shallow convection

Human identified cloud patterns: first step

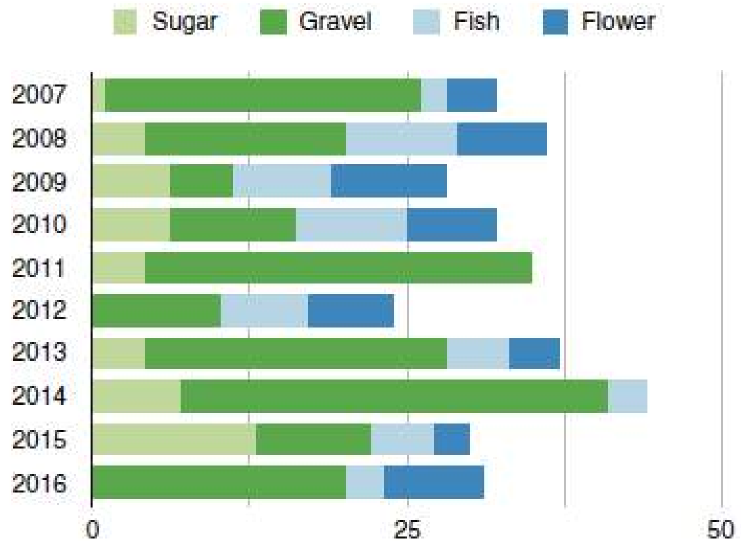
Twelve trained atmospheric scientists gathered to explore to what extent patterns of mesoscale variability could be visually and subjectively identified in satellite imagery of clouds (MODIS) in the winter trades of the North Atlantic.

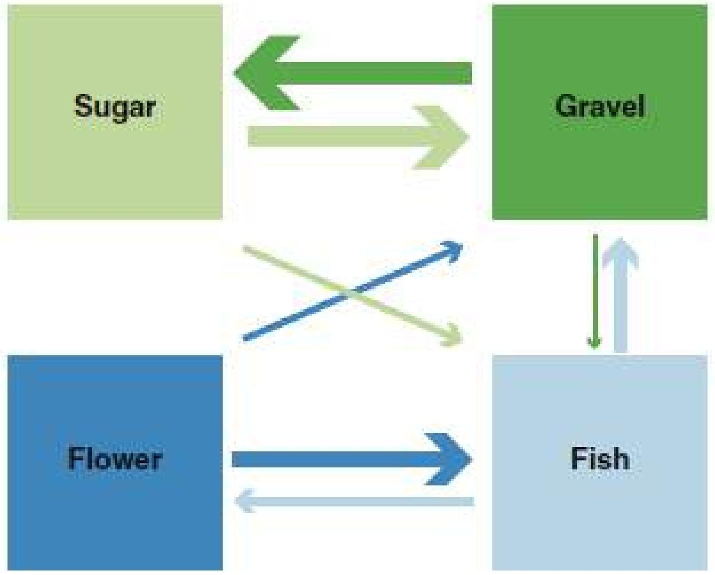

Sugar
Dusting of very fine clouds, little evidence of self-organization
Flower
Large-scale stratiform cloud features appearing in bouquets, well separated from each other
Fish
Large-scale skeletal networks of clouds separated from other cloud forms
Gravel
Meso-beta lines or arcs defining randomly interacting cells with intermediate granularity
500 km



Differences in patterns are associated with differences in the structure of the cloud field as also visualised by its radar presentation, with Fish being most associated with deeper clouds and precipitation.

1. 4 modes of organization: sugar, flower, fish and gravel.
2. The majority (4 of 6) of the labellers agreed on one of these four labels with a probability (p = .4) much larger than would be expected by randomly assigning six (p = .052).
3. Almost all of the images (more than 90%) exhibited features sufficient for at least one person to say that a particular pattern dominated the image.





Stevens et al., 2019, Rasp et al., 2020

Human and machine identified cloud patterns: Rasp et al., 2019

WHAT THEY DO:

On cloud labeling days at two institutes, **67 scientists** screened 10,000 satellite images on a crowd-sourcing platform and classifed almost 50,000 mesoscale cloud clusters.

This dataset is then used as a **training dataset (labeled)** for deep learning algorithms that automate the pattern detection and create global climatologies of the four patterns.

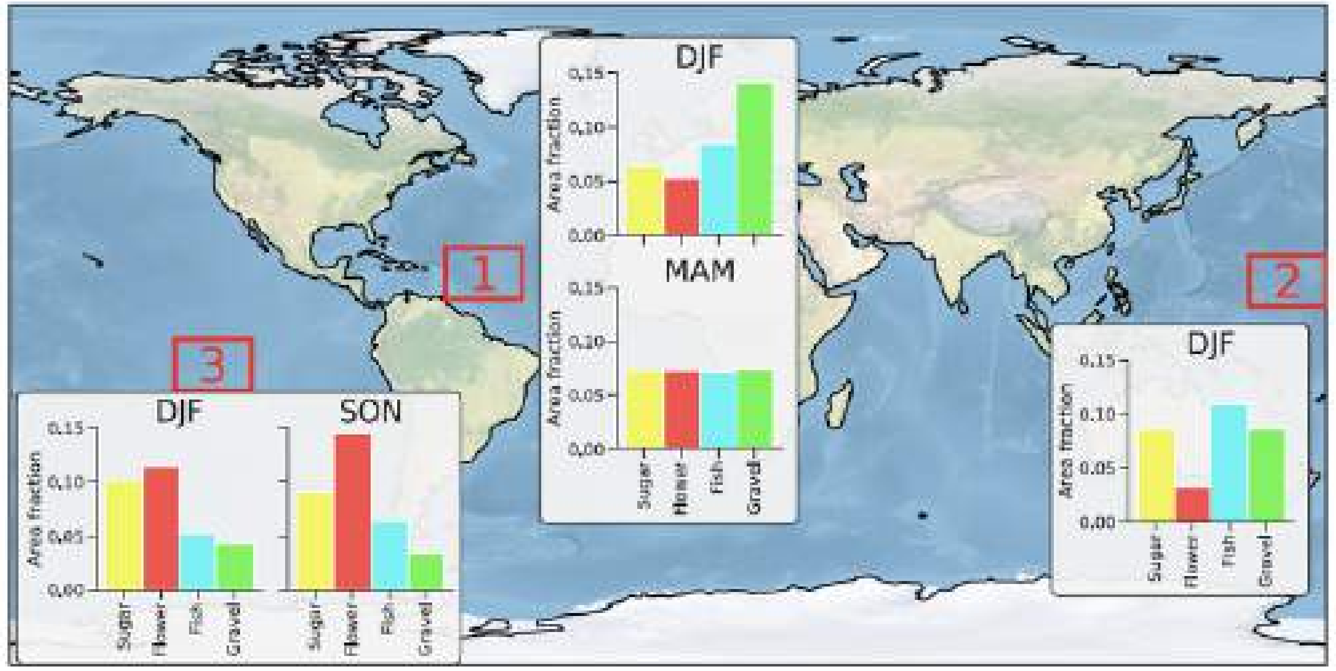


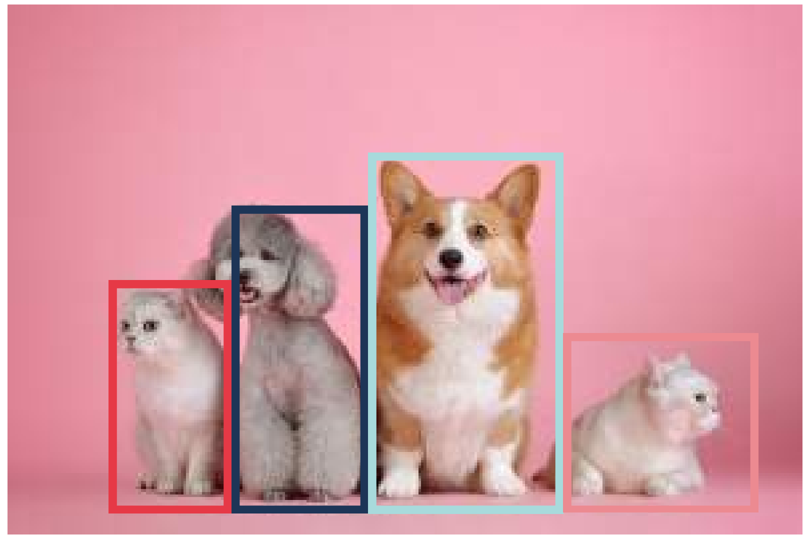
Figure 2: *World map showing the three regions selected for the Zooniverse project. Bar charts are showing which fraction of the image area was classified into one of the four regions by the human labelers. Note that the areas do not add up to one. The remaining fraction was not classified.*

Rasp et al., 2019

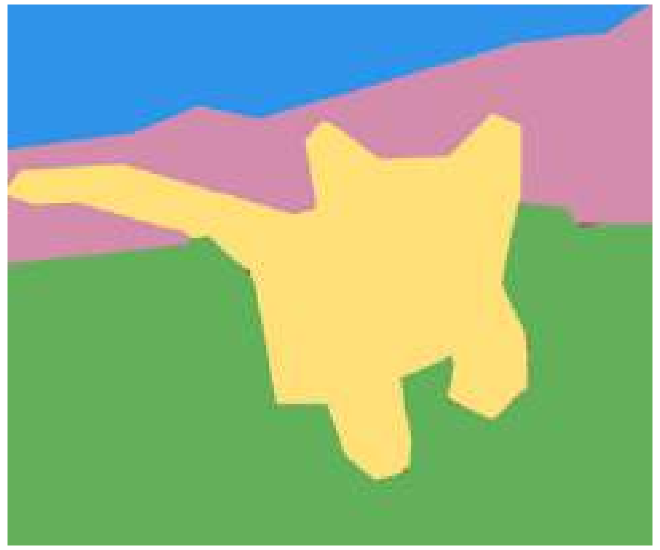
They applied **2 pattern recognition tasks**:

1) **object detection** (drawing boxes around images)

2) **image segmentation** (classification of every pixel of the image)



CAT, DOG, DOG, CAT



GRASS, CAT, SKY

Human and machine identified cloud patterns

WHAT THEY DO:
On cloud labeling days at two institutes, 67 scientists screened 10,000 satellite images on a crowd-sourcing platform and classifed almost 50,000 mesoscale cloud clusters.

This dataset is then used as a **training dataset (labeled)** for deep learning algorithms that automate the pattern detection and create global climatologies of the four patterns.

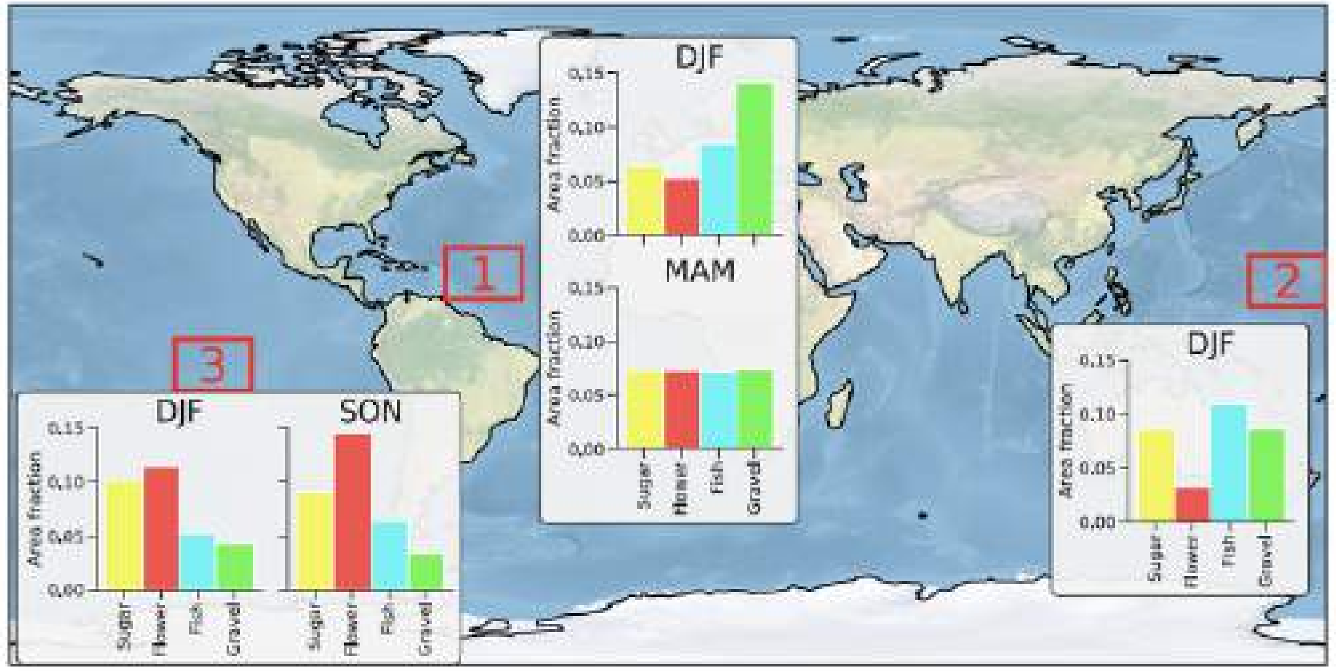
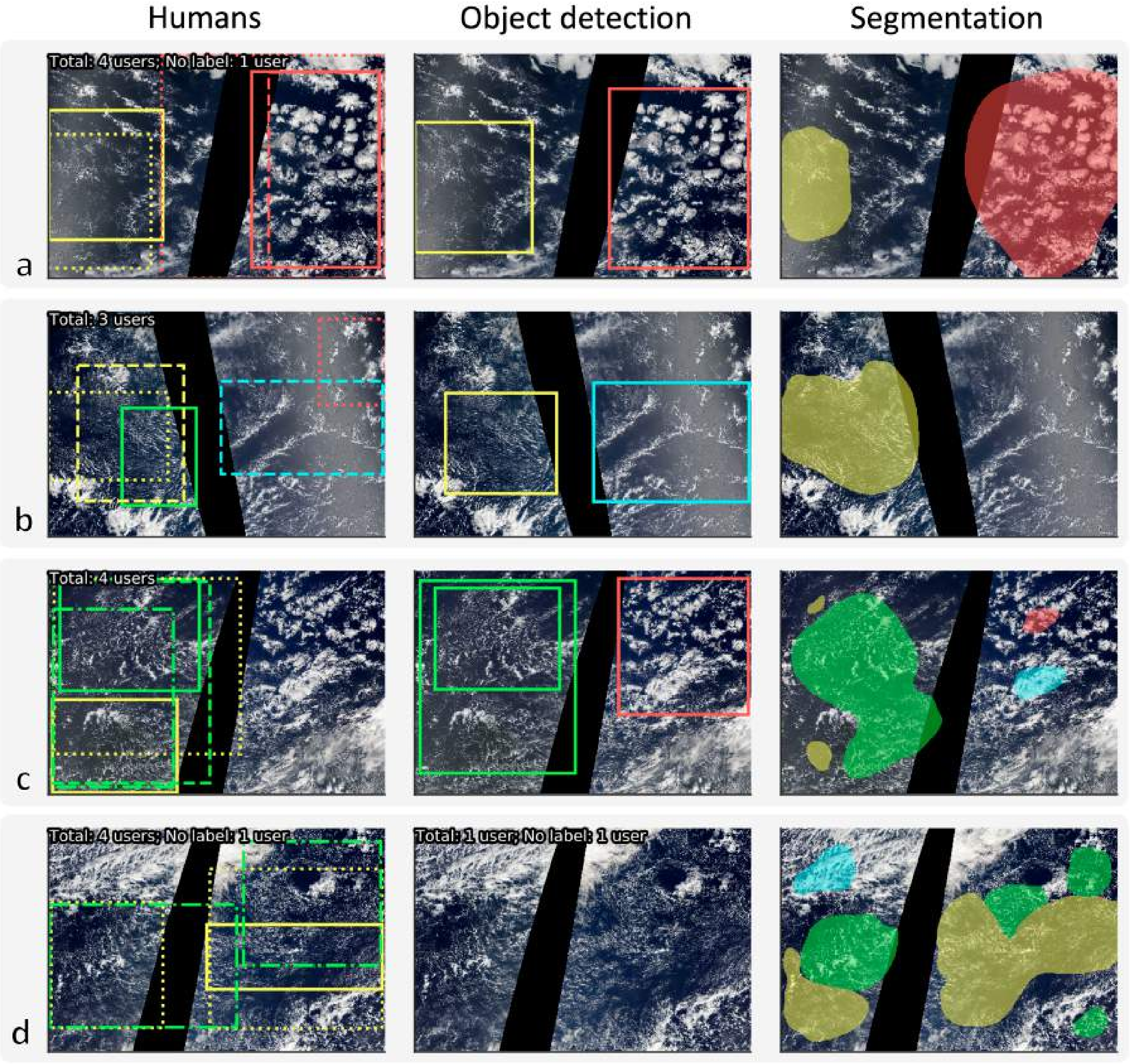


Figure 2: *World map showing the three regions selected for the Zooniverse project. Bar charts are showing which fraction of the image area was classified into one of the four regions by the human labelers. Note that the areas do not add up to one. The remaining fraction was not classified.*

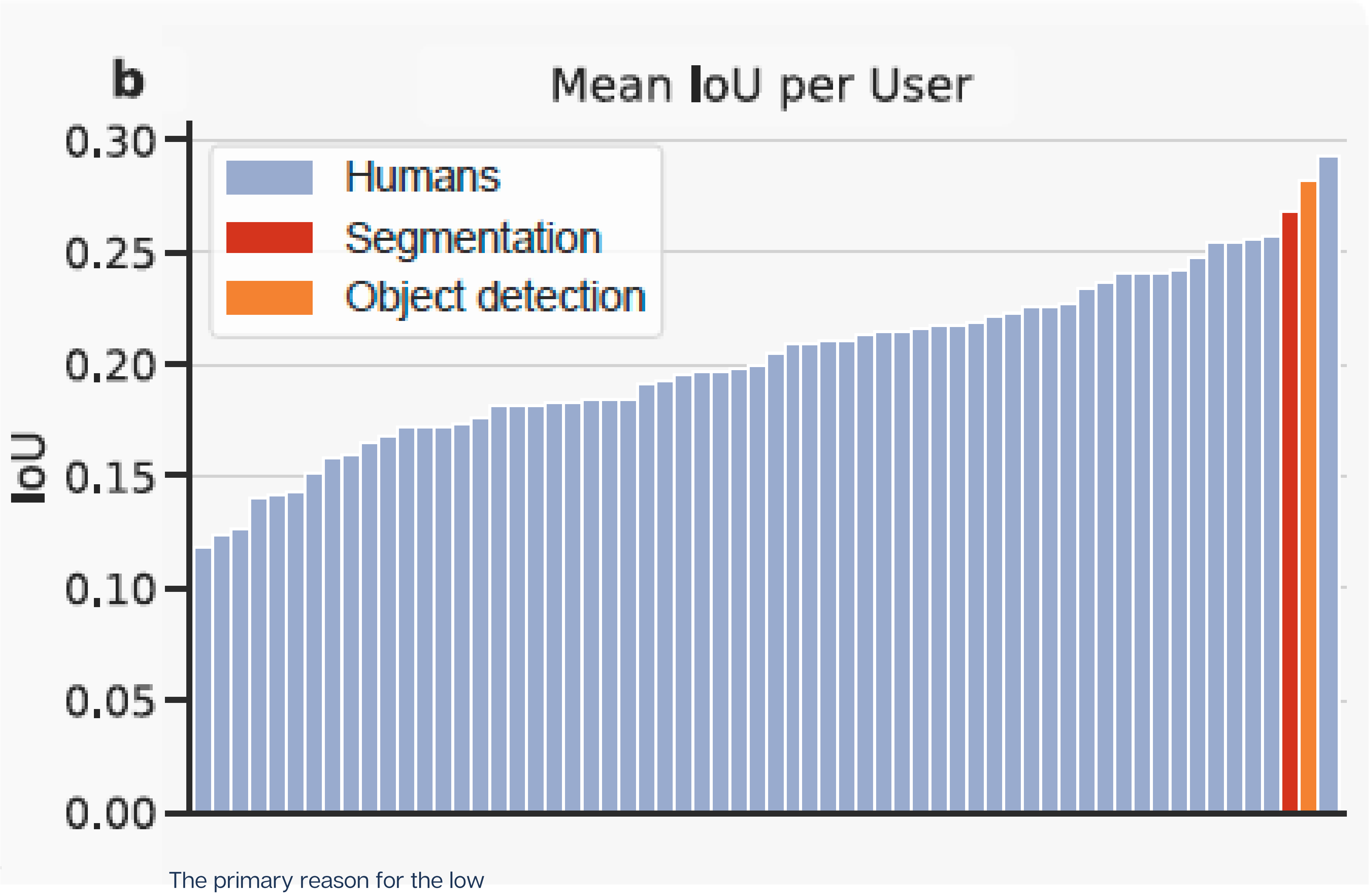


Humans
Object detection
Segmentation
Total: 4 users; No label: 1 user
a
Total: 3 users
b
Total: 4 users
c
Total: 4 users; No label: 1 user
Total: 1 user; No label: 1 user
d

Figure 6: *Human and machine learning predictions for four images from the validation set. Note that images a) and b) are also shown in Fig. 3.*

Rasp et al., 2019

Quantifying the agreement between classifications (Humans vs Machine)



The primary reason for the low mean IoU score are zero values, which arise from some users detecting a feature while others did not.

Intersect over Union (IoU) score, also called the Jaccard index.

Given two sets, A and B it is defined as the ratio of their intersection to their union, i.e., I = A\B divided by U = A U B.

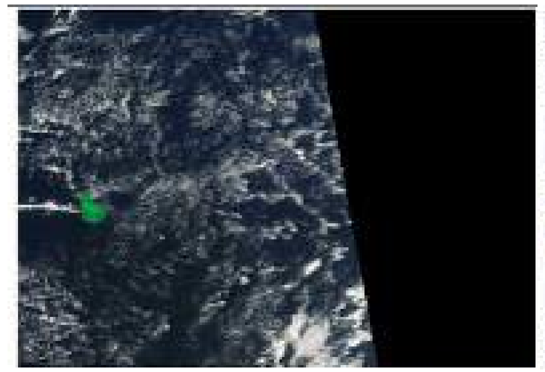
IoU = 1: perfect overlap
IoU = 0: no overlap

Both algorithms show a large agreement with the human labels for a random validation dataset. The fact that the scores are higher than the mean inter-human IoU directly reflects the fact that the algorithms tend to produce less noisy predictions.

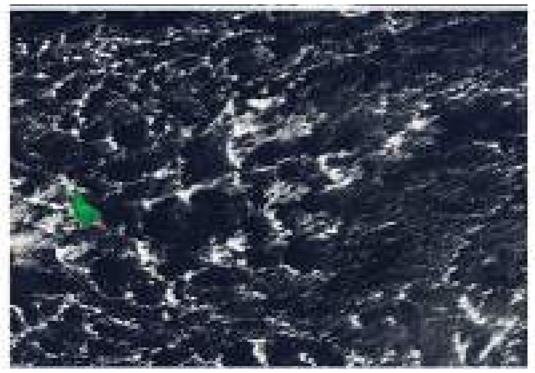
Limitations of human labeled approach

Stevens et al., 2020

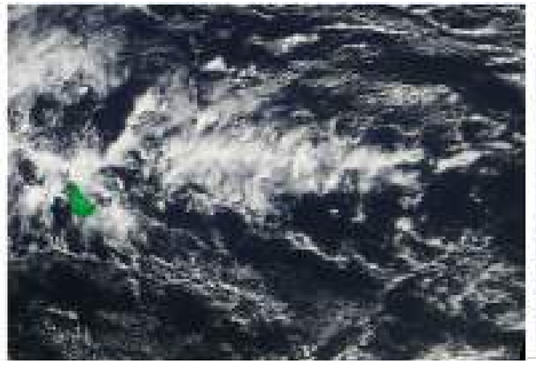
sugar



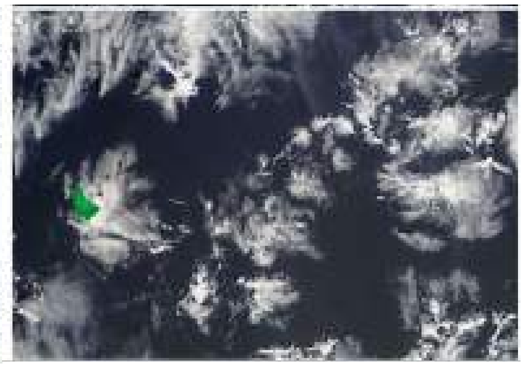
gravel



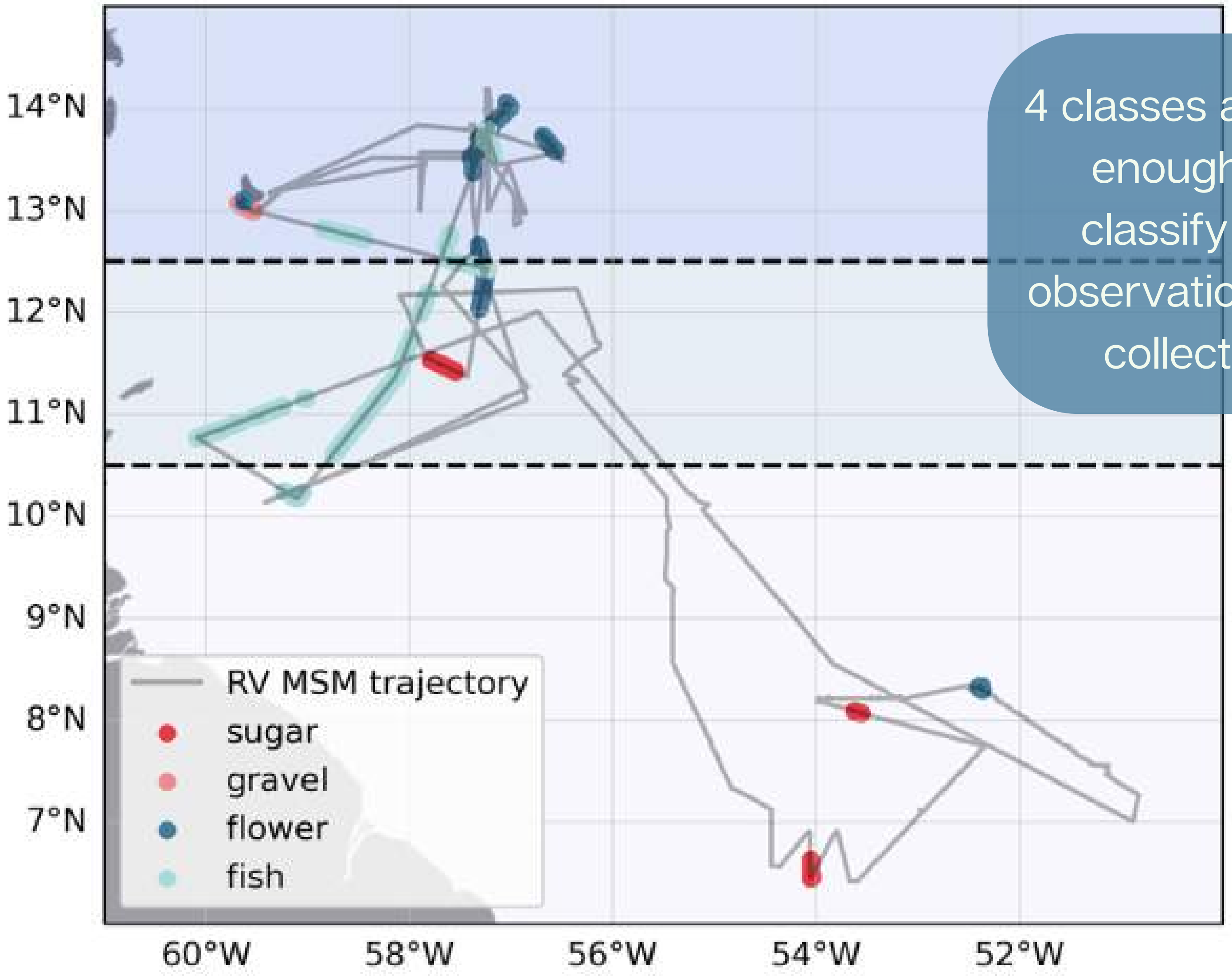
fish



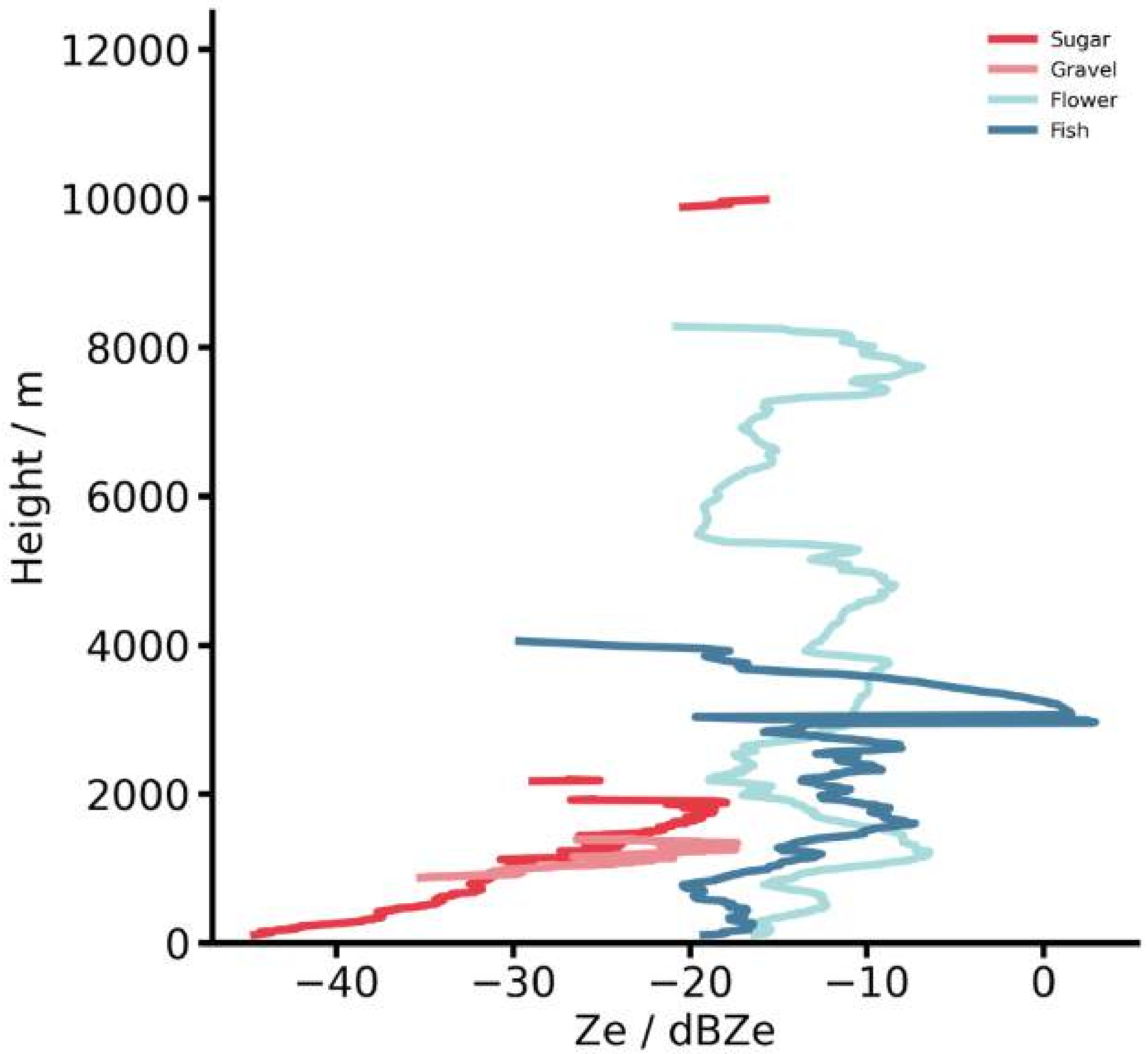
flower



they are sometimes associated with misclassifications



4 classes are not enough to classify the observations we collected



some misclassification occurred during the campaign

from Acquistapace et al., 2024, in prep

Limitations of human labeled approach

Human label classes **occur in less than 50% of the cases** (Schulz et al., 2021)

Subjective approach can really capture a complete representation of the cloud organization? Or will it privilege somehow what our eyes are able to recognize?

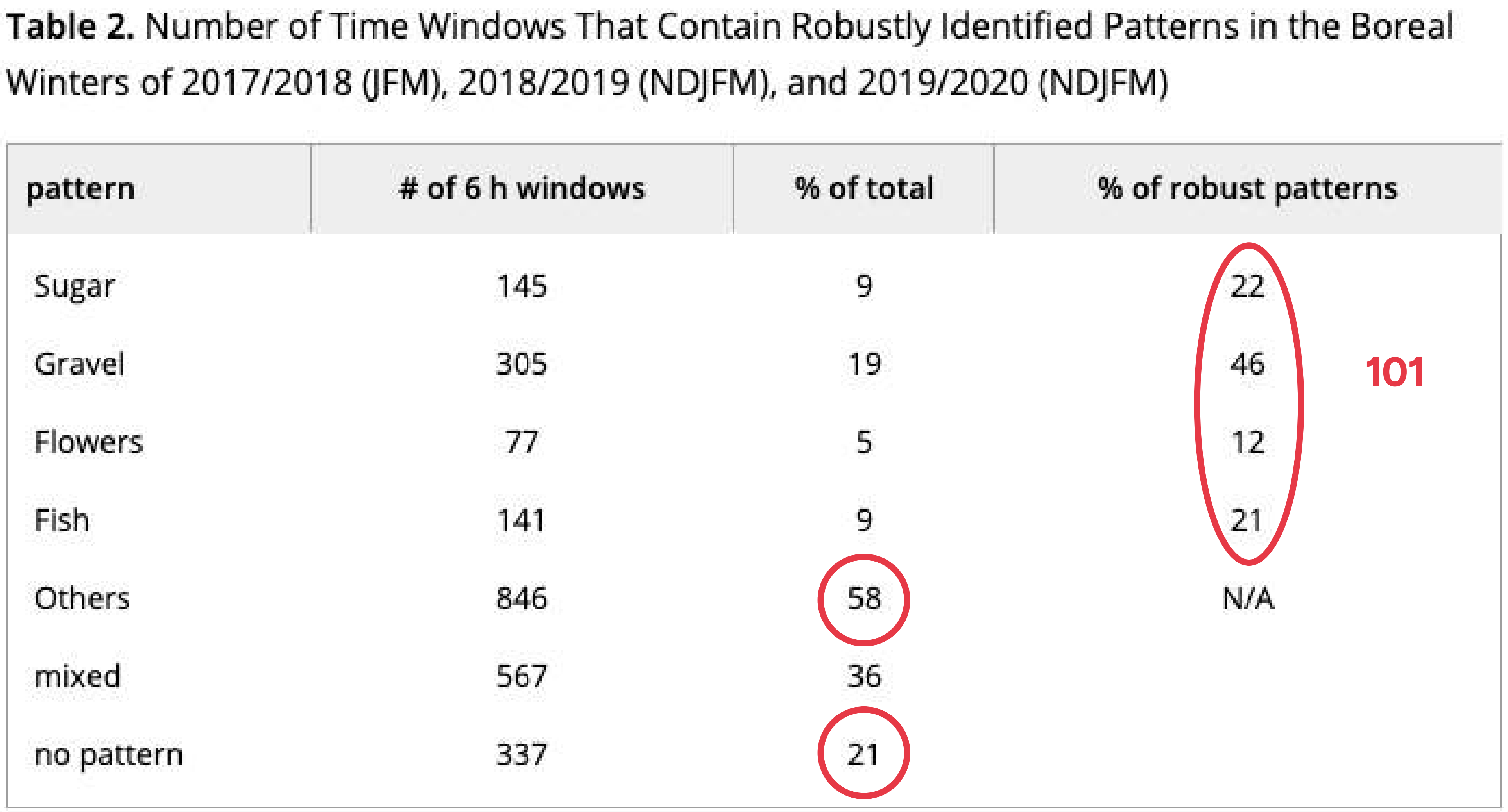


Table 2. Number of Time Windows That Contain Robustly Identified Patterns in the Boreal Winters of 2017/2018 (JFM), 2018/2019 (NDJFM), and 2019/2020 (NDJFM)

pattern	# of 6 h windows	% of total	% of robust patterns
Sugar	145	9	22
Gravel	305	19	46
Flowers	77	5	12
Fish	141	9	21
Others	846	58	N/A
mixed	567	36	
no pattern	337	21	

101

Table from Schulz, H., Eastman, R., & Stevens, B. (2021). Characterization and evolution of organized shallow convection in the downstream North Atlantic trades. Journal of Geophysical Research: Atmospheres, 126, e2021JD034575. Schulz et al., 2021:

Overview of cloud classification algorithms

UNSUPERVISED METHODS

Denby, 2019:
Discovering the importance of mesoscale cloud organization through unsupervised classification

Kurihana et al. 2022:
Cloud classification with unsupervised deep learning

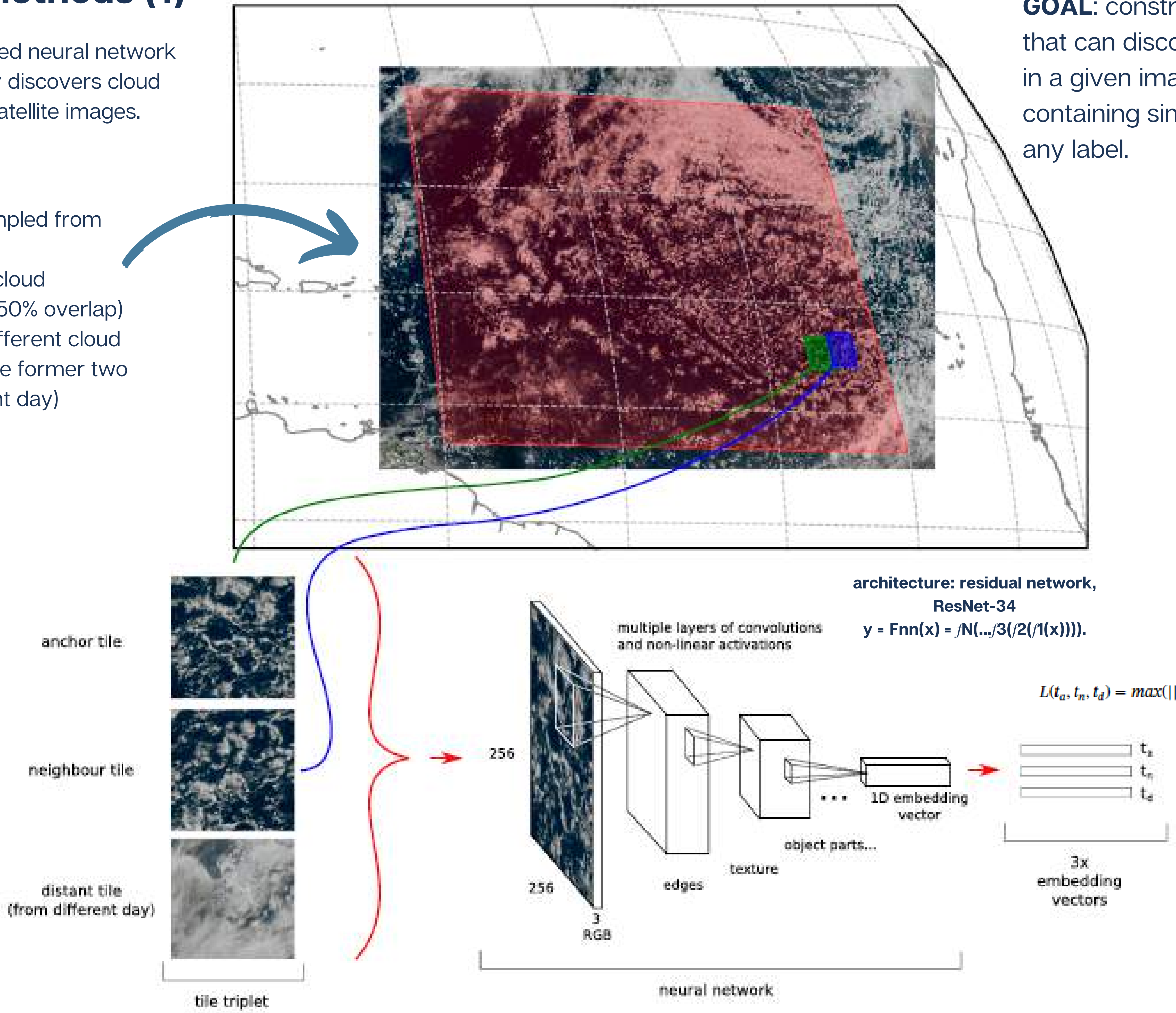
Unsupervised methods (1)

Denby, 2019: first unsupervised neural network model which autonomously discovers cloud organization regimes in satellite images.

GOAL: construct a computational model that can discover which structures exist in a given image and group images containing similar structures without any label.

The tiles of the triplet are sampled from satellite imagery:

- two contain very similar cloud structures (same image, 50% overlap)
- the third contains very different cloud structure compared to the former two (random location, different day)



Input images: 256 x 256 RGB composites using the channels 1, 2, and 3 of GOES.

Tiles size: 200 km (capturing any cloud structure up to beta mesoscale.

Dataset:

- Four satellite images per day,
- domain Tropical Atlantic,
- 3 months of data period nov 2018 to Jan 2019 (Boreal winter)

architecture: residual network, ResNet-34

$y = F_{nn}(x) = f_N(\ldots f_3(f_2(f_1(x))))$.

The loss function encourages similar embeddings from the anchor and the neighbor and penalizes similar embedding from the anchor and the distant tile. It is based on L2 norm distances.

$$L(t_a, t_n, t_d) = max(||F_{nn}(t_a) - F_{nn}(t_n)||_2 - ||F_{nn}(t_a) - F_{nn}(t_d)||_2, 0) + m,$$

m is the hyperparameter called margin, regulating the distance between anchor and distant tile. m set to 1.

Neural network without supervision discovers different types of cloud structures and groups images with similar features together in the embedding space.

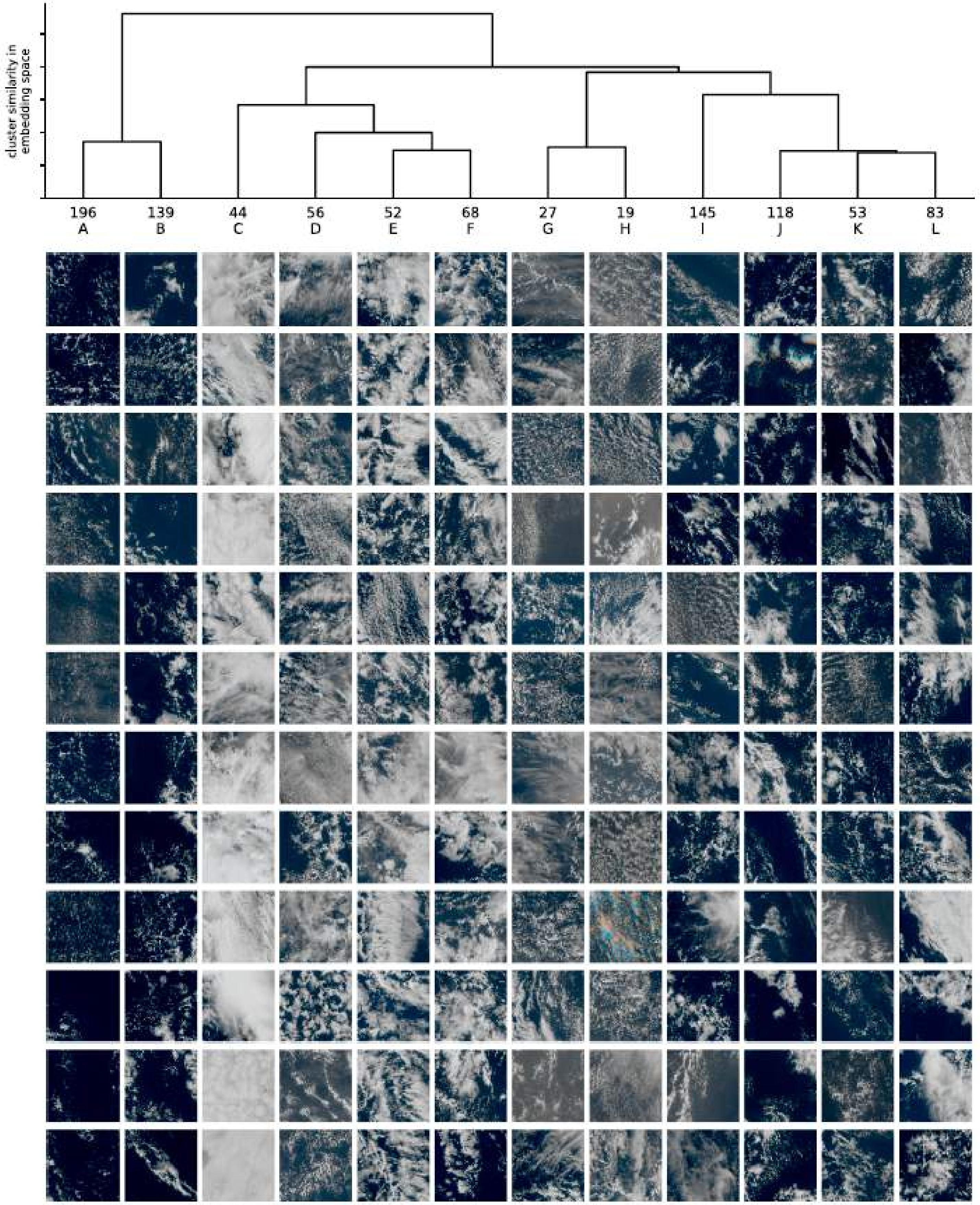


Figure 2. Demonstration of clustering for embeddings produced by the trained neural network. Top: dendrogram showing hierarchical structure of the clustering, with height representing intra-cluster variance in embedding distance in a cluster resulting from a specific merge of two child clusters. Only the clusters present before the last 12 merges are showed for brevity. Bottom: 12 random tile examples from each cluster belonging to the leaf immediately above in the dendrogram. Each leaf node cluster is annotated with the number of tiles in that cluster and a label to aid discussion. The persistence in the dendrogram indicates for example clusters A and B are much more similar to each other than to any other cluster. A number of visibly distinct structures have been identified, for example, scattered small clouds (A, B) cellular structures (C, D, G, and H in order of scale) and larger cloud cellular (D–F) and broken (I–L) cloud structures.

measures the intra-cluster variance

- length of the vertical vertices connecting each merge indicates the similarity between clusters

(larger intra-cluster variance meaning larger variance in distance between points in the embedding space)

This is a dendrogram, and displays the outcome of a hierarchical clustering based on a given metric (Ward metric here).

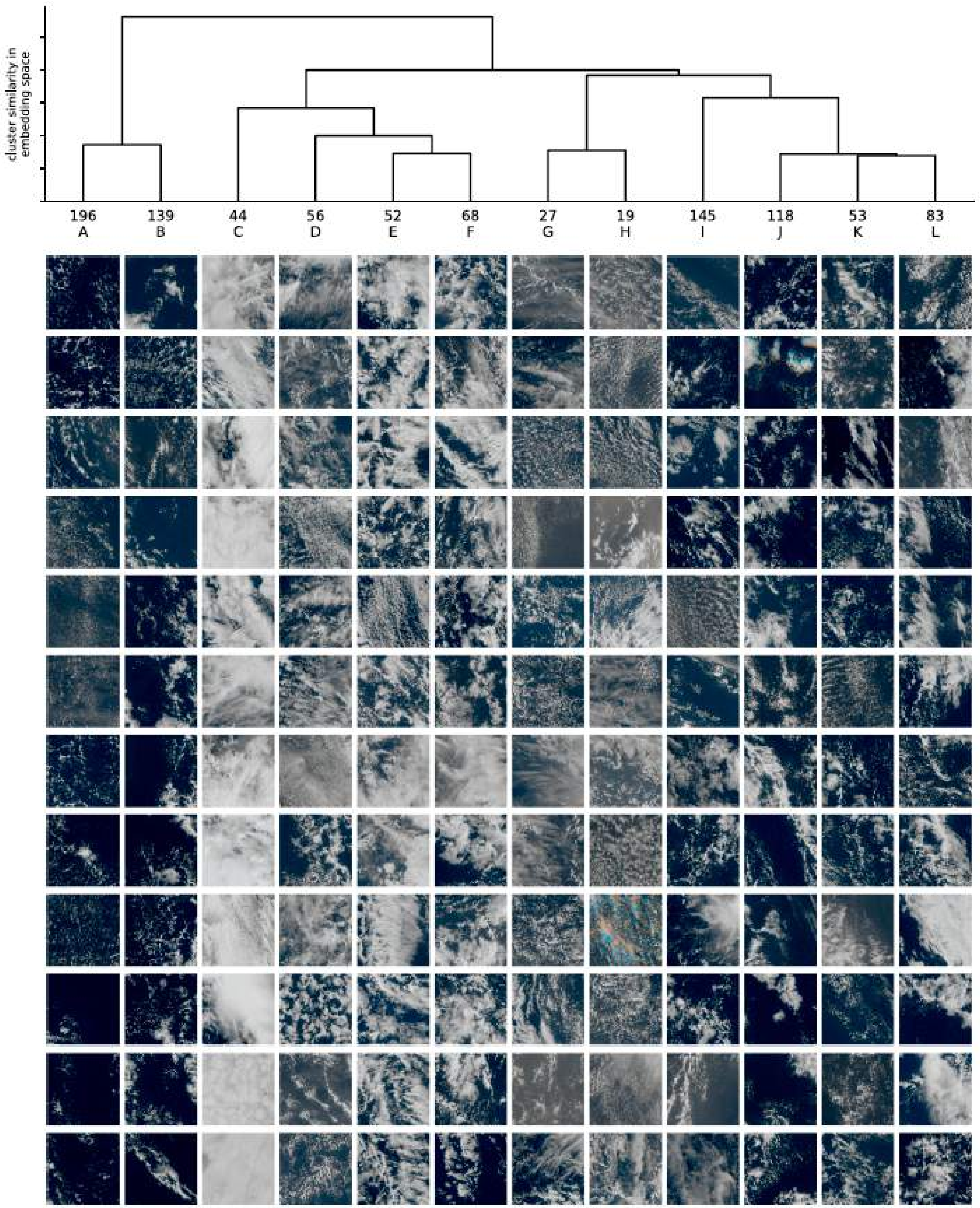


Figure 2. Demonstration of clustering for embeddings produced by the trained neural network. Top: dendrogram showing hierarchical structure of the clustering, with height representing intra-cluster variance in embedding distance in a cluster resulting from a specific merge of two child clusters. Only the clusters present before the last 12 merges are showed for brevity. Bottom: 12 random tile examples from each cluster belonging to the leaf immediately above in the dendrogram. Each leaf node cluster is annotated with the number of tiles in that cluster and a label to aid discussion. The persistence in the dendrogram indicates for example clusters A and B are much more similar to each other than to any other cluster. A number of visibly distinct structures have been identified, for example, scattered small clouds (A, B) cellular structures (C, D, G, and H in order of scale) and larger cloud cellular (D–F) and broken (I–L) cloud structures.

Neural network without supervision discovers different types of cloud structures and groups images with similar features together in the embedding space.

A, B, G, and H contain smaller cloud structures as compared to C–F

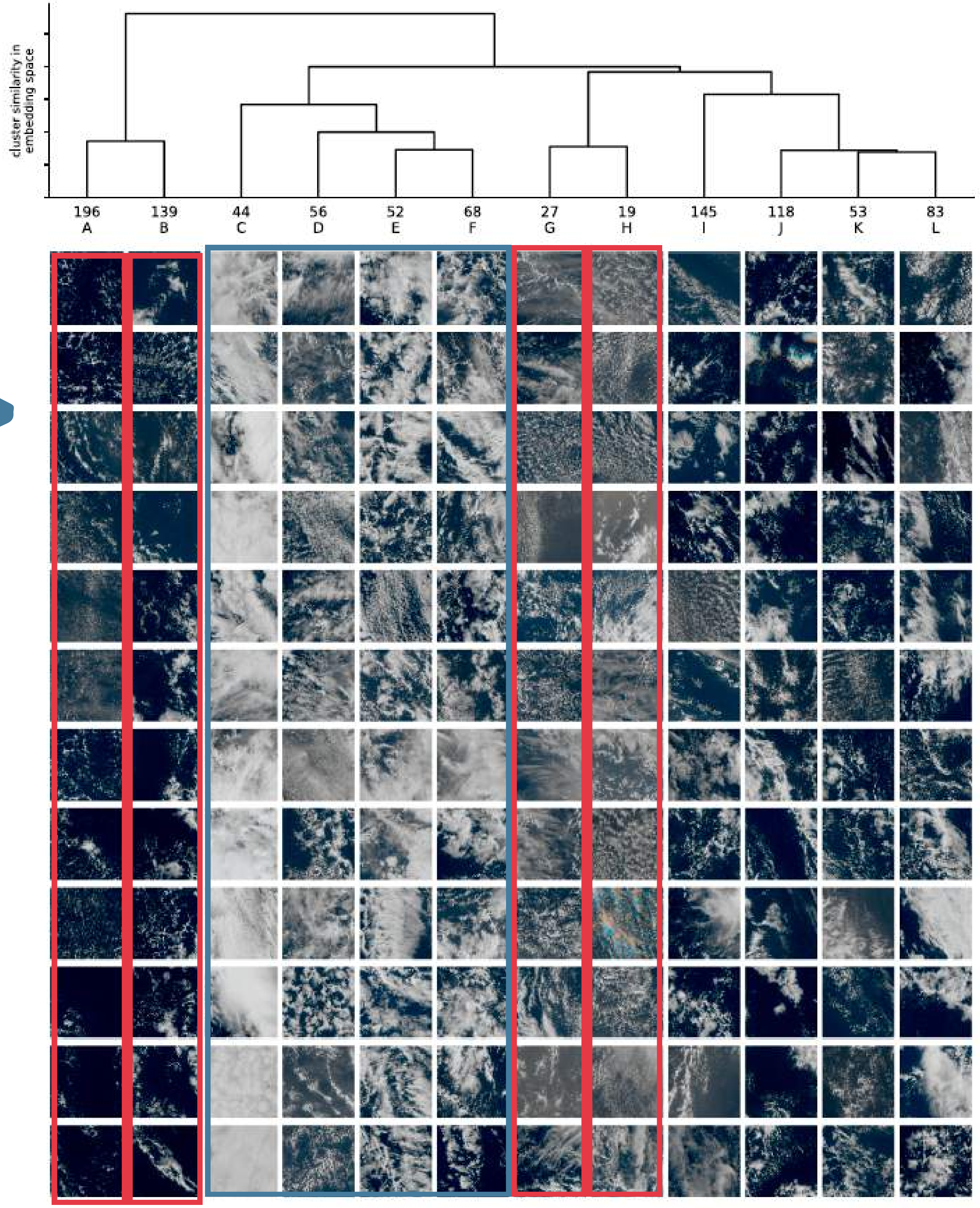


Figure 2. Demonstration of clustering for embeddings produced by the trained neural network. Top: dendrogram showing hierarchical structure of the clustering, with height representing intra-cluster variance in embedding distance in a cluster resulting from a specific merge of two child clusters. Only the clusters present before the last 12 merges are showed for brevity. Bottom: 12 random tile examples from each cluster belonging to the leaf immediately above in the dendrogram. Each leaf node cluster is annotated with the number of tiles in that cluster and a label to aid discussion. The persistence in the dendrogram indicates for example clusters A and B are much more similar to each other than to any other cluster. A number of visibly distinct structures have been identified, for example, scattered small clouds (A, B) cellular structures (C, D, G, and H in order of scale) and larger cloud cellular (D–F) and broken (I–L) cloud structures.

A, B, G, and H contain smaller cloud structures as compared to C–F

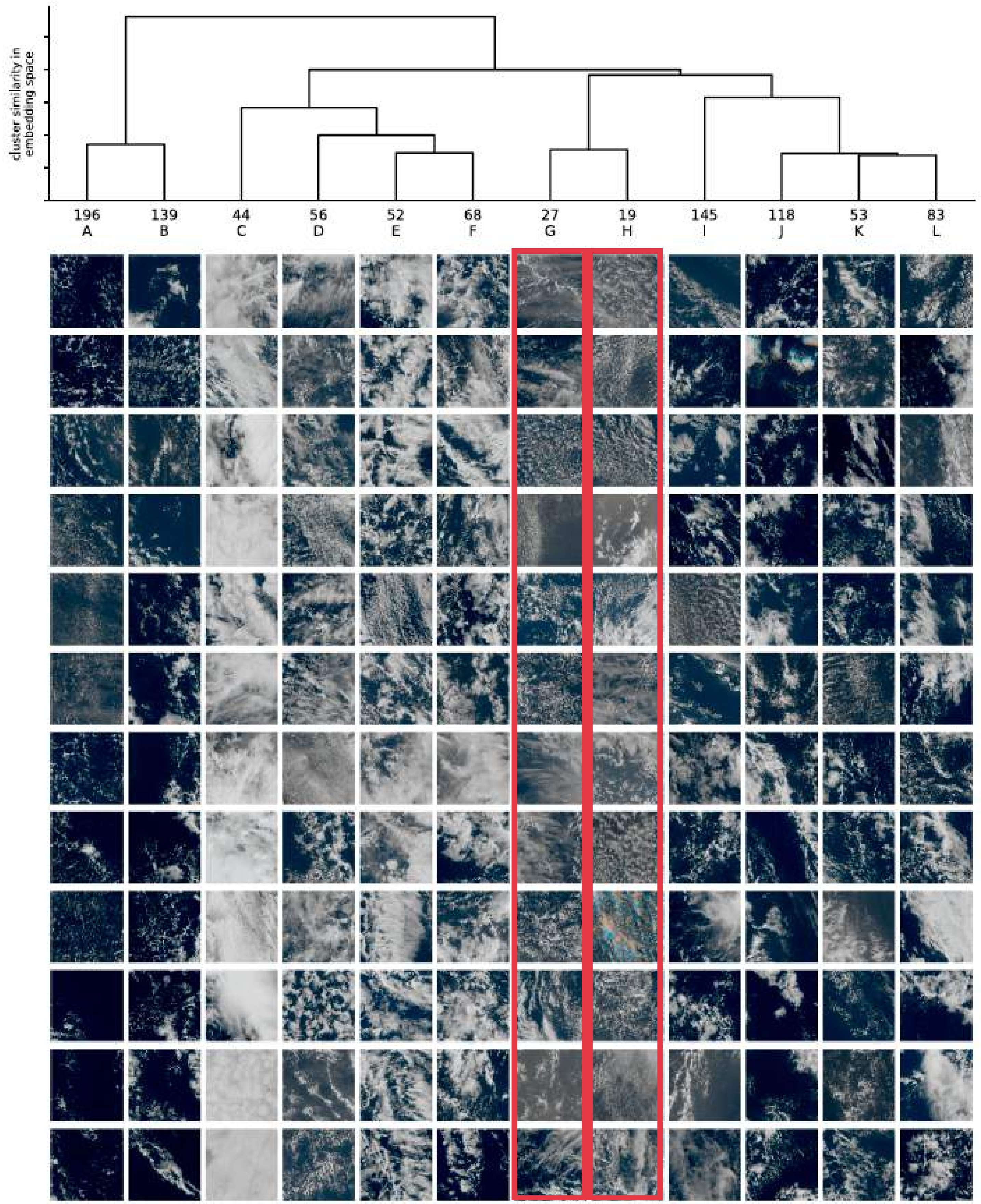


Figure 2. Demonstration of clustering for embeddings produced by the trained neural network. Top: dendrogram showing hierarchical structure of the clustering, with height representing intra-cluster variance in embedding distance in a cluster resulting from a specific merge of two child clusters. Only the clusters present before the last 12 merges are showed for brevity. Bottom: 12 random tile examples from each cluster belonging to the leaf immediately above in the dendrogram. Each leaf node cluster is annotated with the number of tiles in that cluster and a label to aid discussion. The persistence in the dendrogram indicates for example clusters A and B are much more similar to each other than to any other cluster. A number of visibly distinct structures have been identified, for example, scattered small clouds (A, B) cellular structures (C, D, G, and H in order of scale) and larger cloud cellular (D–F) and broken (I–L) cloud structures.

another feature used by the model is the presence or absence of a light-gray shading (which is likely high-level dust being swept from the coast of Africa) seen in G and H.

A, B, G, and H contain smaller cloud structures as compared to C–F

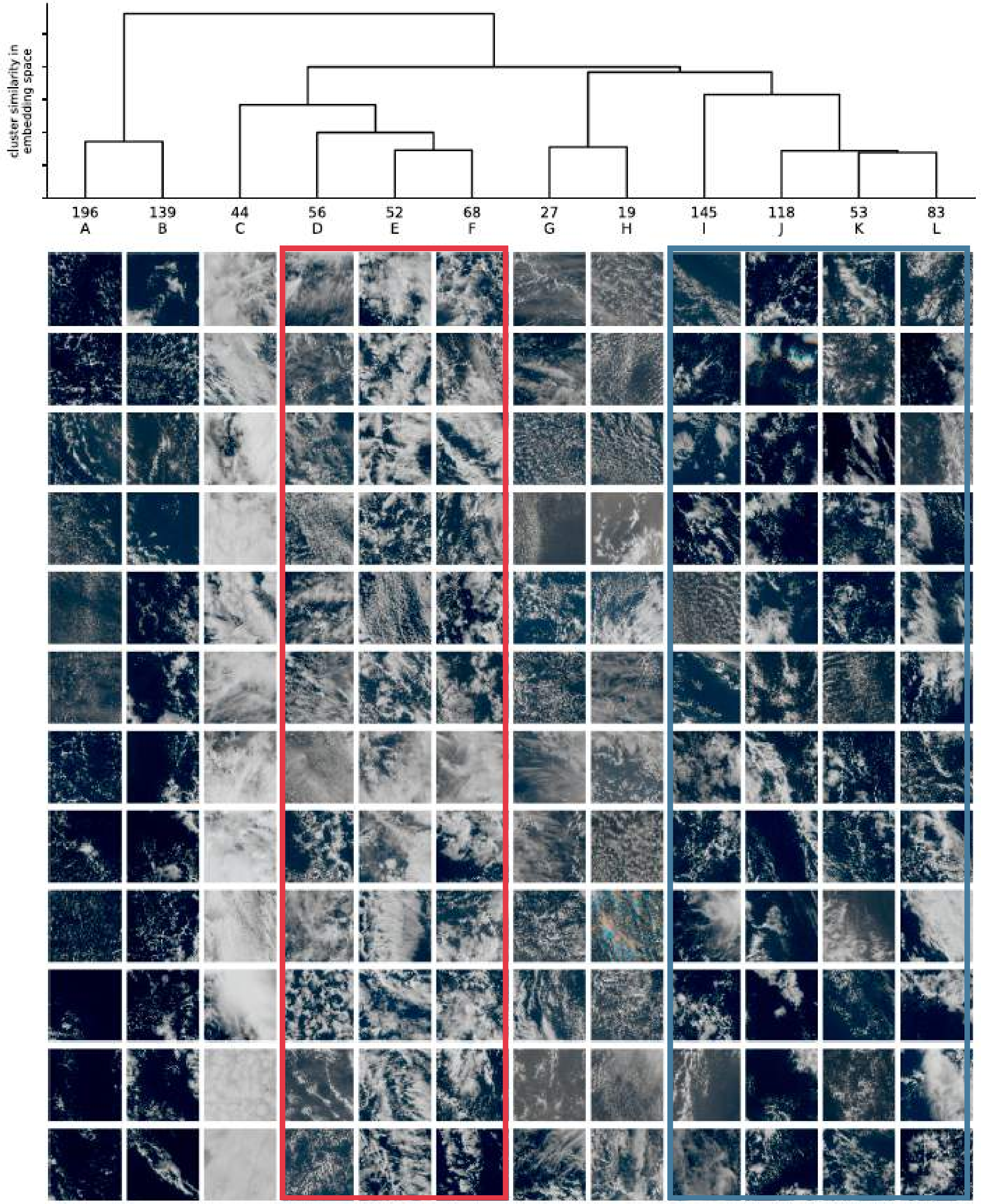


Figure 2. Demonstration of clustering for embeddings produced by the trained neural network. Top: dendrogram showing hierarchical structure of the clustering, with height representing intra-cluster variance in embedding distance in a cluster resulting from a specific merge of two child clusters. Only the clusters present before the last 12 merges are showed for brevity. Bottom: 12 random tile examples from each cluster belonging to the leaf immediately above in the dendrogram. Each leaf node cluster is annotated with the number of tiles in that cluster and a label to aid discussion. The persistence in the dendrogram indicates for example clusters A and B are much more similar to each other than to any other cluster. A number of visibly distinct structures have been identified, for example, scattered small clouds (A, B) cellular structures (C, D, G, and H in order of scale) and larger cloud cellular (D–F) and broken (I–L) cloud structures.

another feature used by the model is the presence or absence of a light-gray shading (which is likely high-level dust being swept from the coast of Africa) seen in G and H.

D–F contain more cellular structures as compared to the broken larger clouds in I–L.

A, B, G, and H contain smaller cloud structures as compared to C–F

I–LA and B are distinguished from G and H by the latter containing a more regular cellular pattern of clouds where clouds in A and B are more scattered.

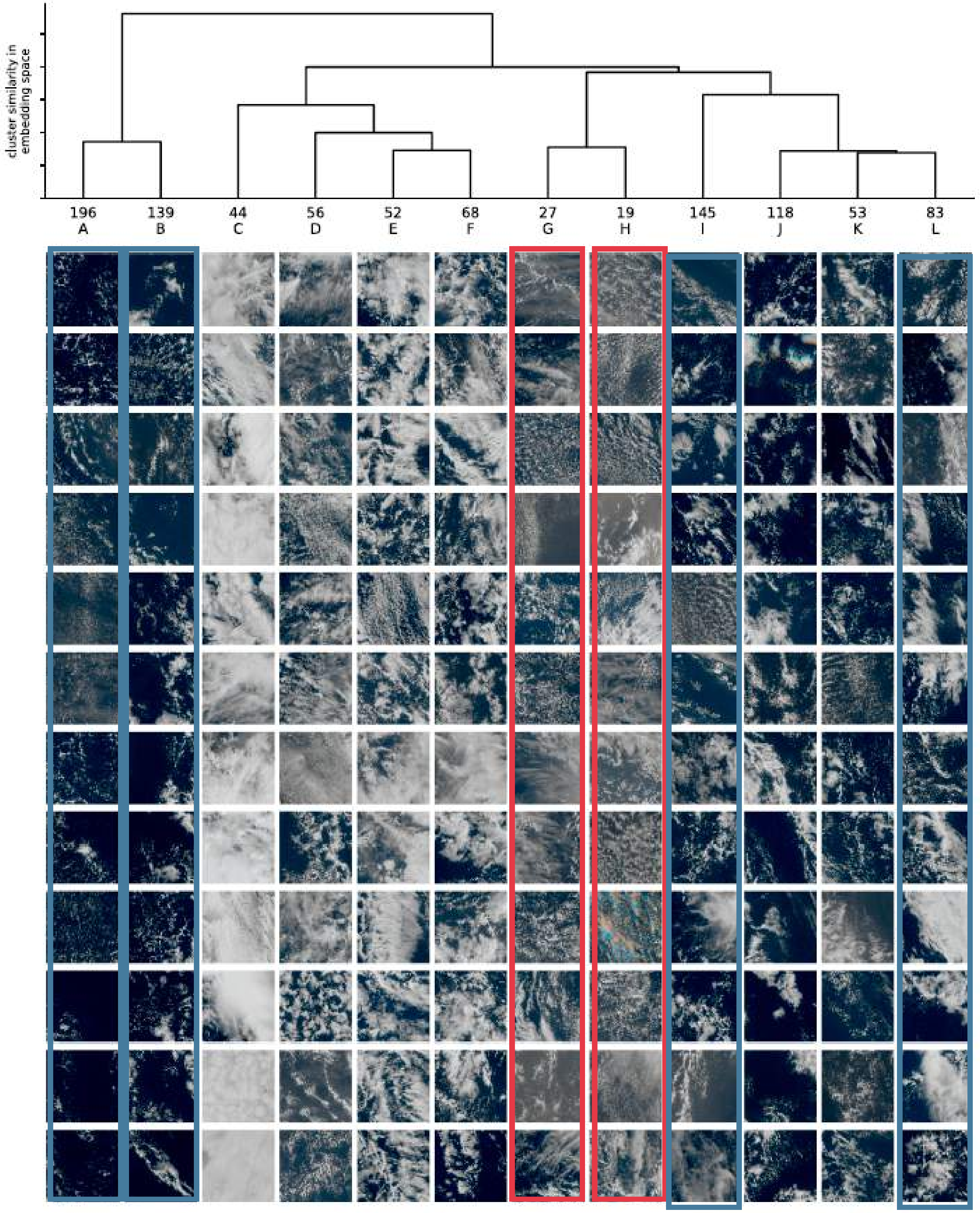


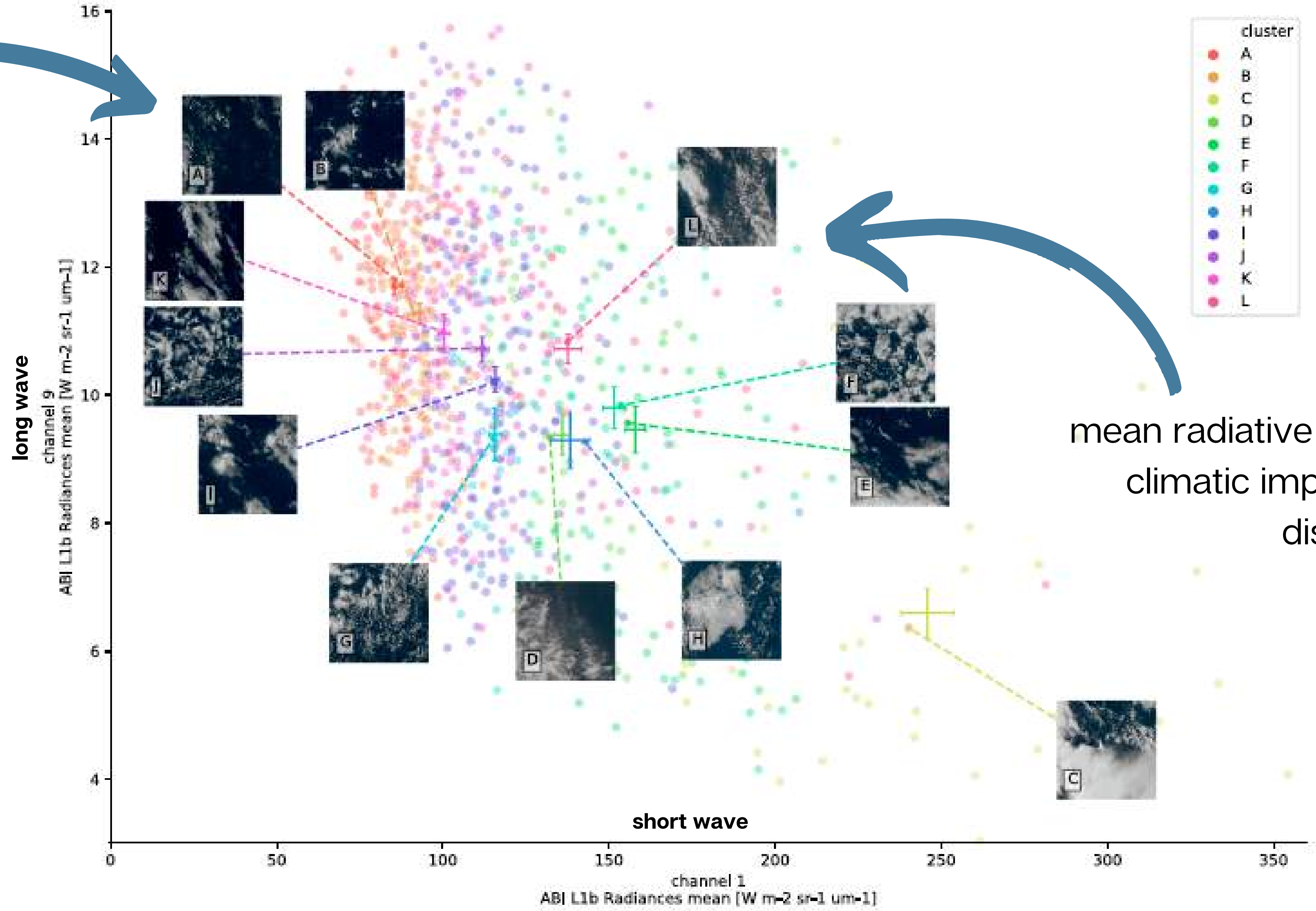
Figure 2. Demonstration of clustering for embeddings produced by the trained neural network. Top: dendrogram showing hierarchical structure of the clustering, with height representing intra-cluster variance in embedding distance in a cluster resulting from a specific merge of two child clusters. Only the clusters present before the last 12 merges are showed for brevity. Bottom: 12 random tile examples from each cluster belonging to the leaf immediately above in the dendrogram. Each leaf node cluster is annotated with the number of tiles in that cluster and a label to aid discussion. The persistence in the dendrogram indicates for example clusters A and B are much more similar to each other than to any other cluster. A number of visibly distinct structures have been identified, for example, scattered small clouds (A, B) cellular structures (C, D, G, and H in order of scale) and larger cloud cellular (D–F) and broken (I–L) cloud structures.

another feature used by the model is the presence or absence of a light-gray shading (which is likely high-level dust being swept from the coast of Africa) seen in G and H.

D–F contain more cellular structures as compared to the broken larger clouds in I–L.

Cloud impact on the Earth's radiation budget: different clusters have different cloud radiative properties in outgoing short-wave and long-wave radiation

tile nearest to the mean



mean radiative properties (and so climatic impact) are clearly distinct.

with **increasing cloud cover** there is in general a **decreasing** amount of **long-wave** emissions to space and an **increase** of **short-wave** emissions.

Figure 3. Per-tile mean radiance of channels 1 (in short-wave range) and 9 (in long-wave range) for all 1000 tiles in *study* set together with the per-cluster mean and error (standard error of the mean) across all tiles in each cluster produced by hierarchical clustering (figure 2) colored by cluster. Each cluster is annotated with the nearest tile. The clusters show a clear separation between different cloud structures, more cloudy tiles having less long-wave and more short-wave emission is radiated into space, and each cluster has distinct radiative properties.

Denby, 2019

Iorg - DMB space

The clusters of cloud types identified by the neural network occupy different parts of the Iorg - DMB space, and the **cluster means are clearly separated**, the neural network has learnt distinct cloud regimes.

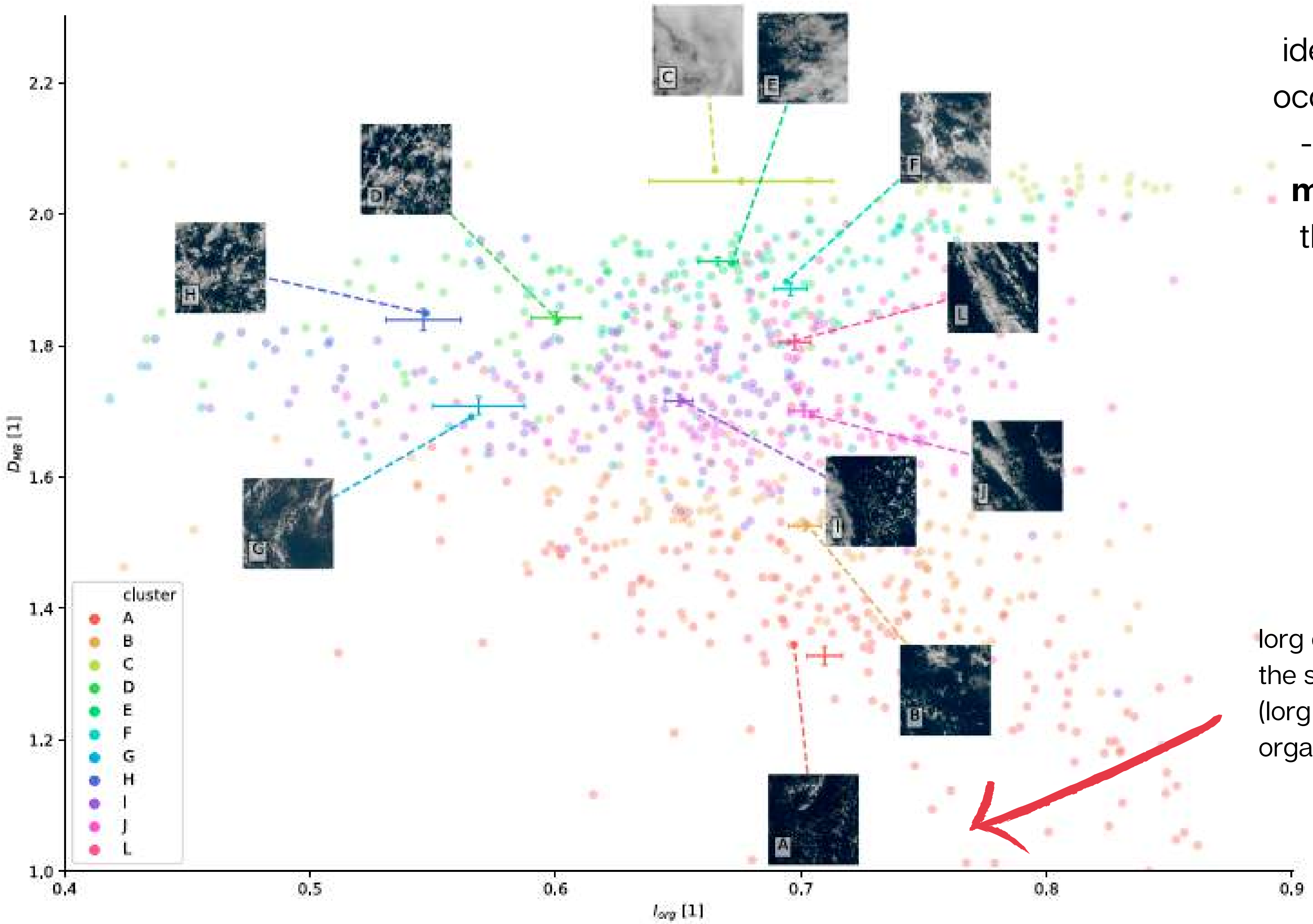


Figure 4. Cloud organization (I_{org}) and fractal dimension (D_{MB}) for all 1,000 tiles in *study* set produced by hierarchical clustering (figure 2), colored by cluster, together with per-cluster mean and standard error in the mean. Each cluster is annotated with nearest tile.

DMB quantifies the spatial form of clouds by measuring the fractal dimension, expected to be between that of a line (DMB = 1) and a plane (DMB = 2), by counting the number of successively smaller boxes that cover the cloud mask.

Iorg quantifies whether the spatial distribution of clouds is regular (Iorg < 0.5), random (Iorg = 0.5), or organized (Iorg>0.5),

Denby, 2019

Unsupervised methods (2)

Kiruhana et al., 2022: learning cloud features directly from radiances produced by MODIS

Architecture

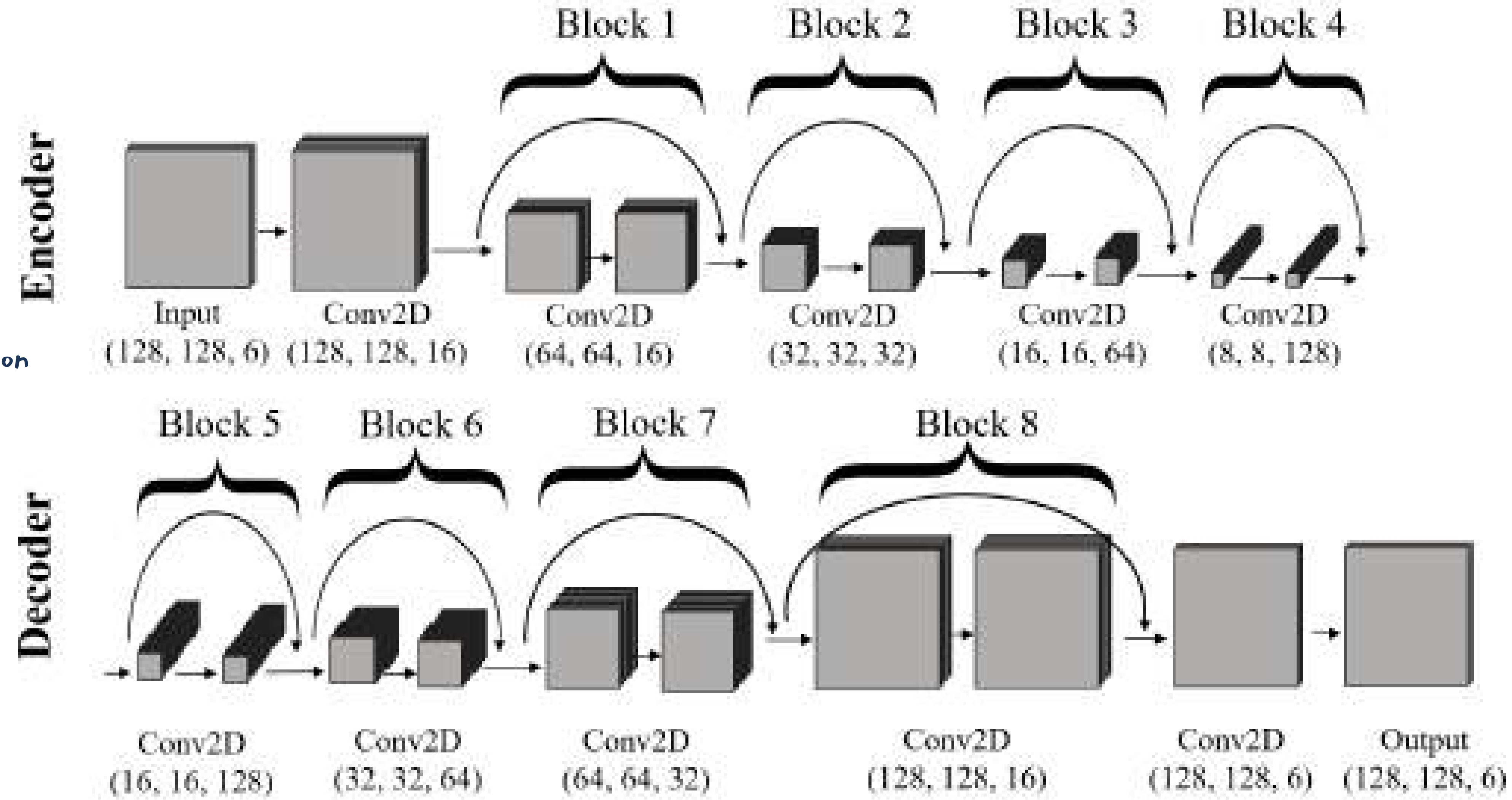
Deep convolutional autoencoder, that in learning optimizes the function L

$$\min L(x) = \min ||x - F(x)||_p$$

image

map to representation

The loss combines four metrics: L1, L2, Sobel filter and multiscale similarity index (MSSIM)



In encoder, width and stride halved at each block, depth doubled.

In decoder, tranformations reversed.

Unsupervised methods (2)

Kiruhana et al., 2022: learning cloud features directly from radiances produced by MODIS

Architecture

Deep convolutional autoencoder, that in learning optimizes the function L

$$\min L(x) = \min ||x - F(x)||_p$$

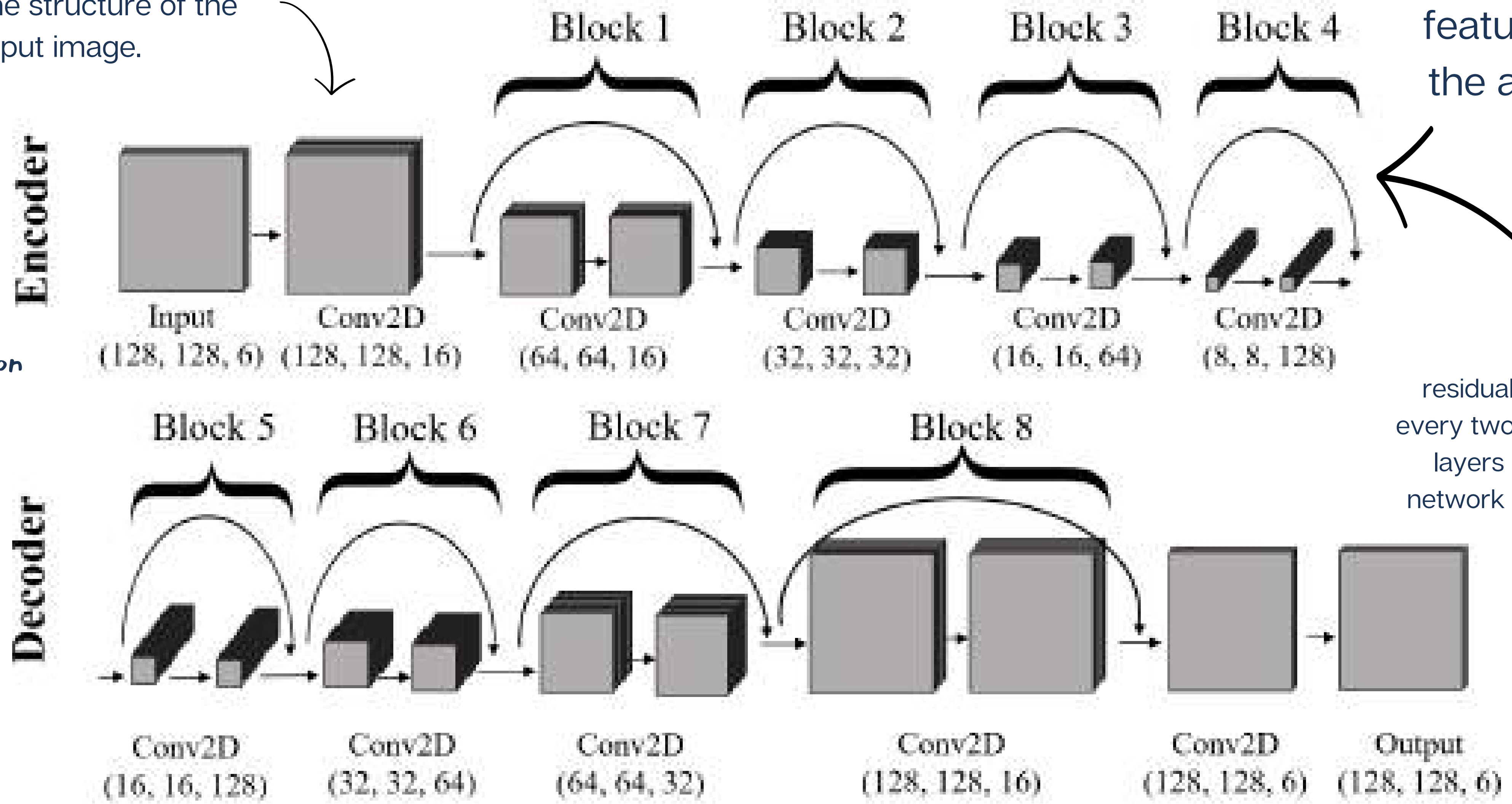
image

map to representation

The loss combines four metrics: L1, L2, Sobel filter and multiscale similarity index (MSSIM)

Convolutional layer included to preserve the structure of the input image.

leaky REIu activated with residual connections



filter kernel operation extracts local features and parses the activation layer.

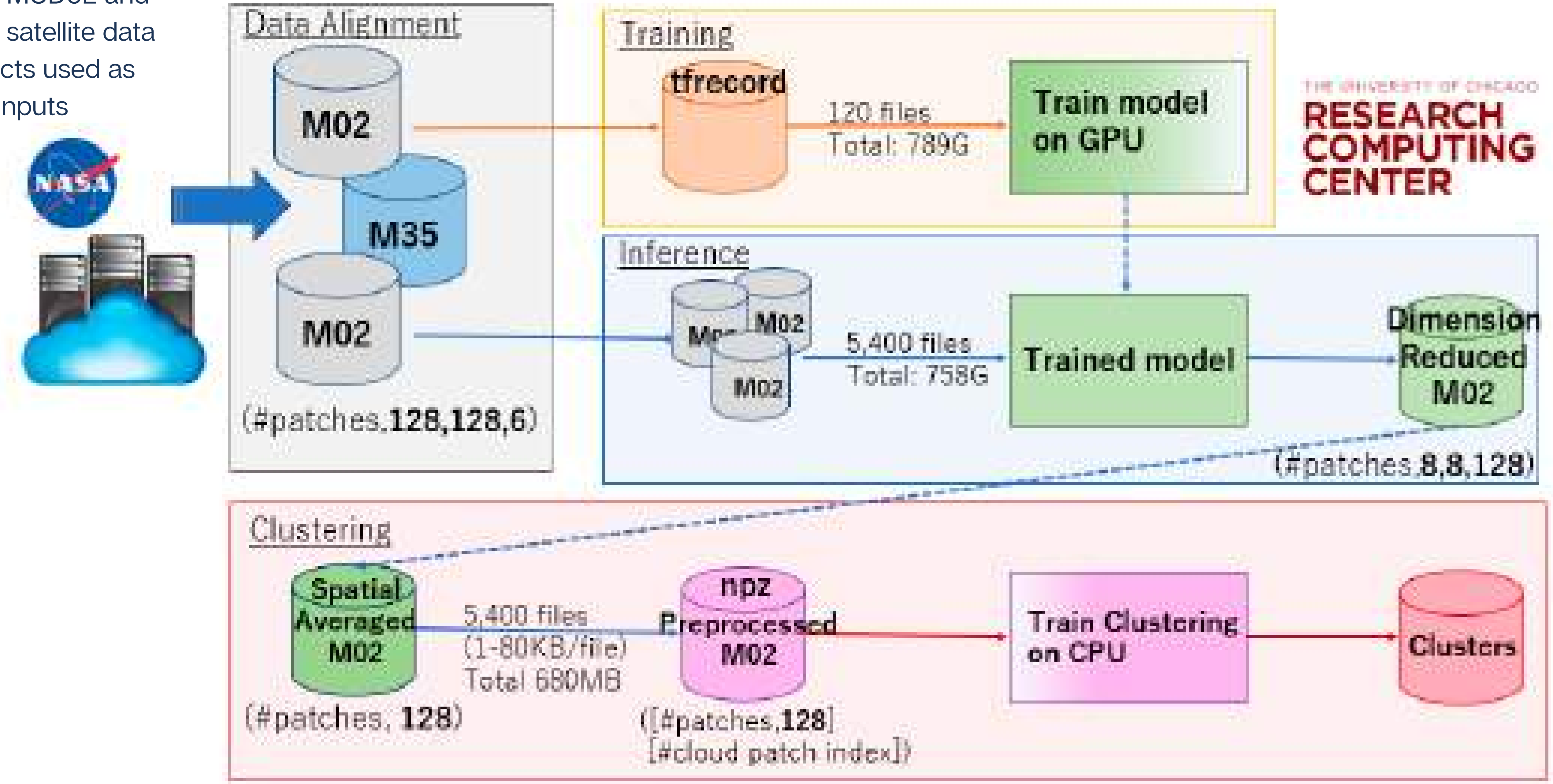
In encoder, width and stride halved at each block, depth doubled.

residual connection every two convolutional layers to improve network performance

In decoder, tranformations reversed.

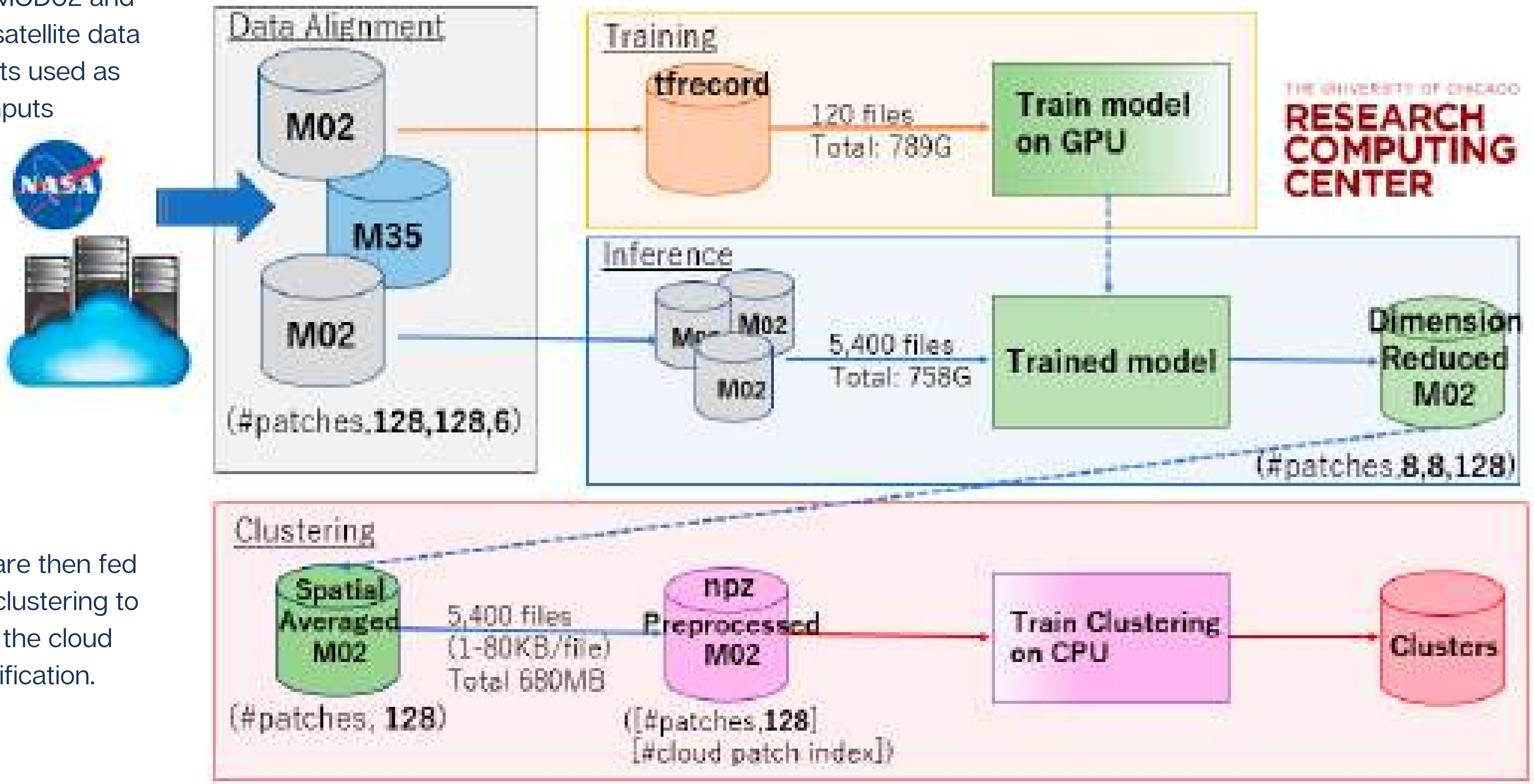
Model framework

M02 and M35 are the MOD02 and MOD35 satellite data products used as inputs



Model framework

M02 and M35
are the MOD02 and
MOD35 satellite data
products used as
inputs

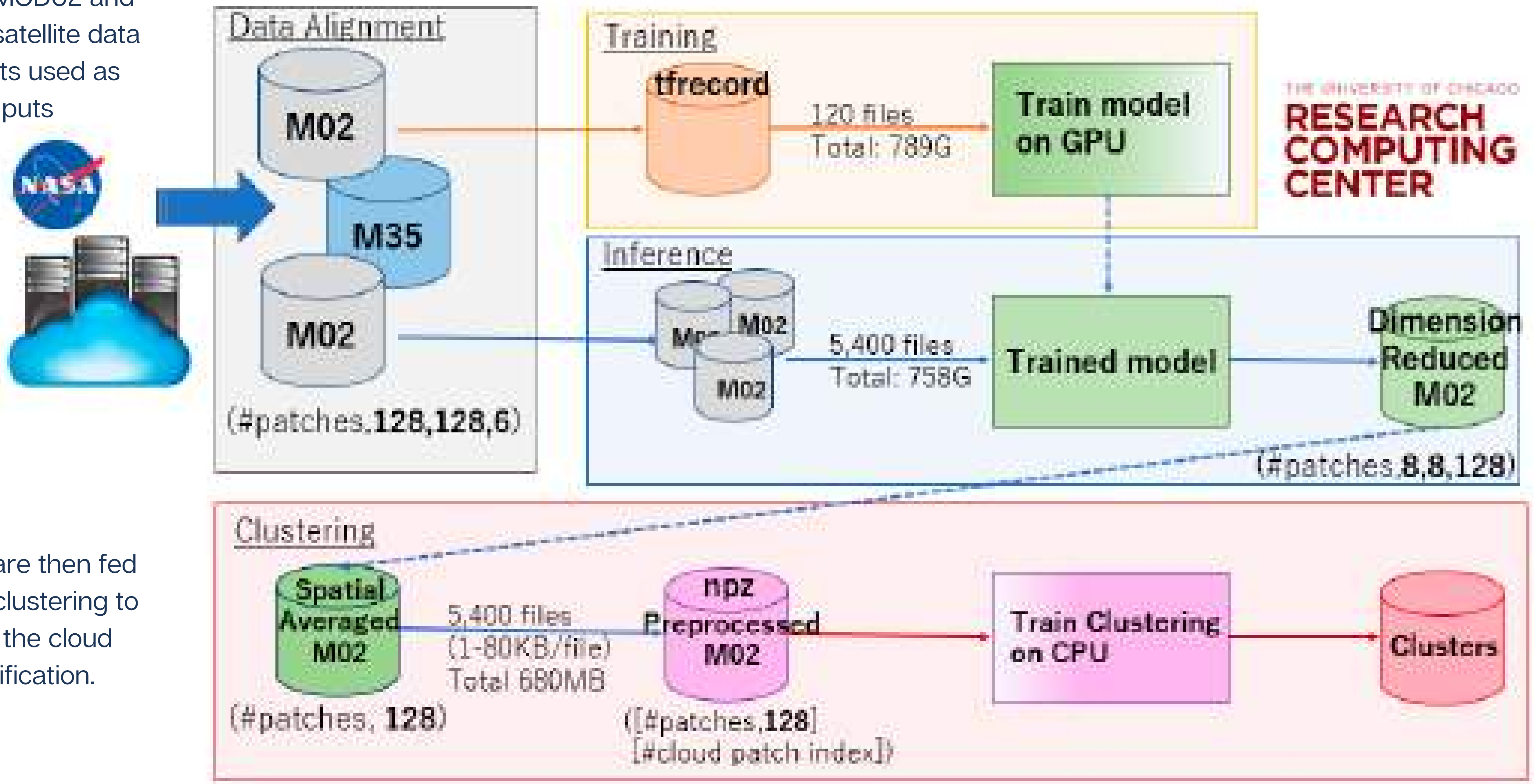


The encoder decoder
identifies the
dimensionality
reduced features....

...which are then fed
into the clustering to
obtain the cloud
classification.

Model framework

M02 and M35 are the MOD02 and MOD35 satellite data products used as inputs



The encoder decoder identifies the dimensionality reduced features....

...which are then fed into the clustering to obtain the cloud classification.

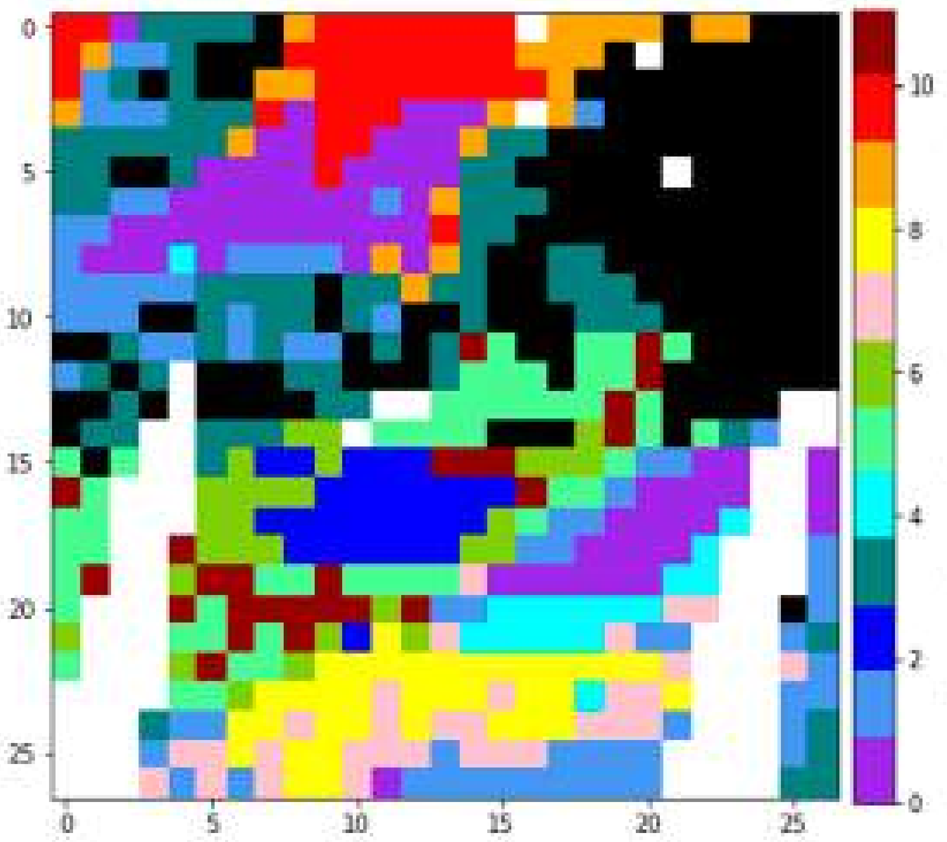
They use hierarchical agglomerative clustering (HAC) to merge data points by minimizing cluster variance

$$\delta \text{dist}(X_A, X_B) = \frac{n_A n_B}{n_A + n_B}||C_A - C_B||^2$$

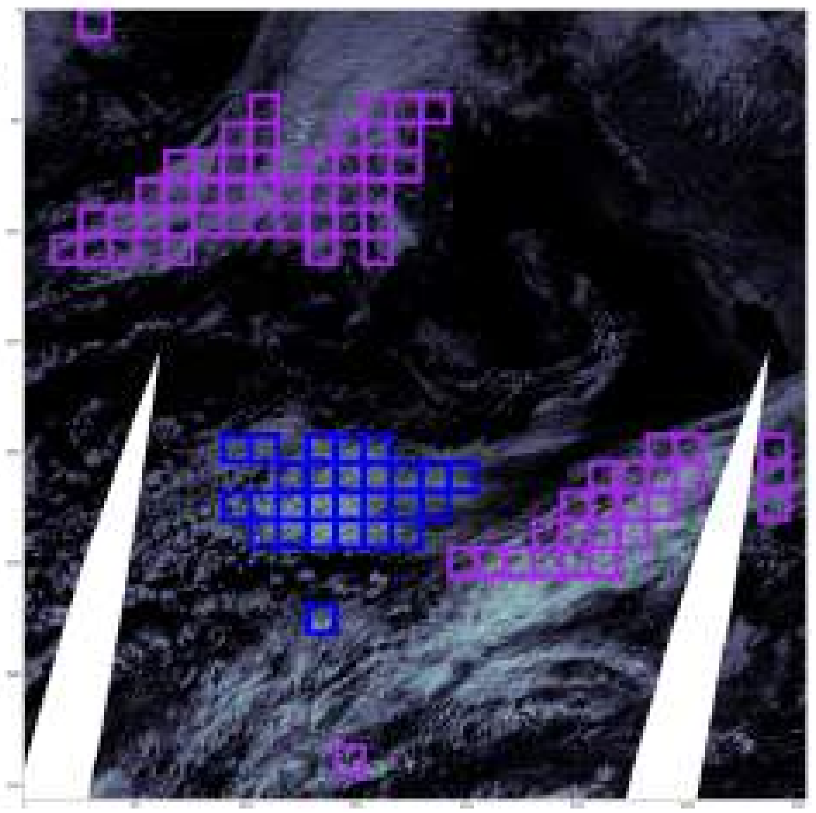
distance between cluster A and B is the squared distance between centroids of the merged clusters CA and CB weighted by the sum of the number of patches in A and B.

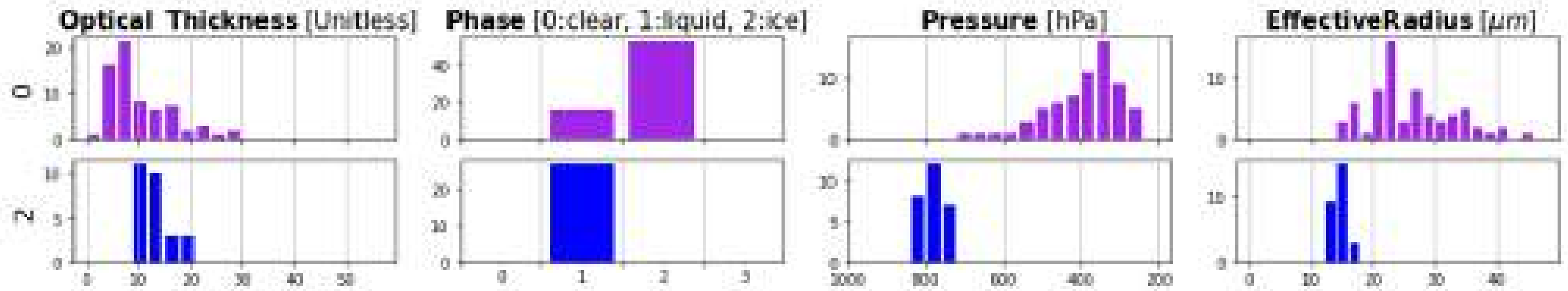
Results

ENCODER - DECODER CLUSTERING



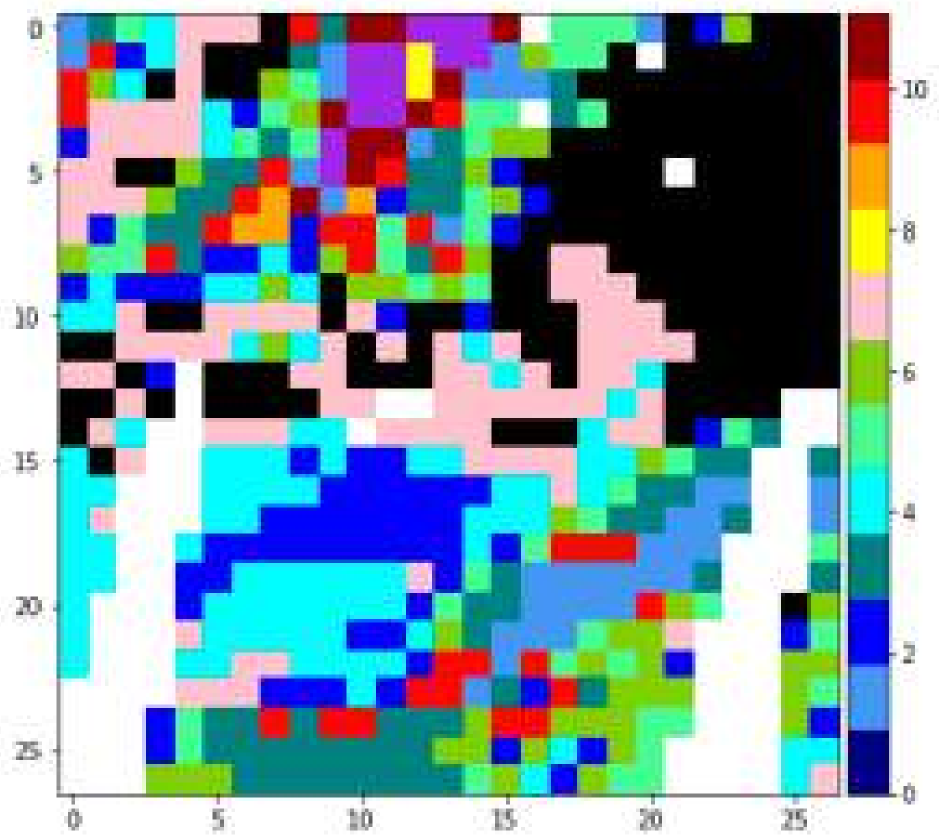
Visible image with cluster 0 (violet) and 2 (blue)

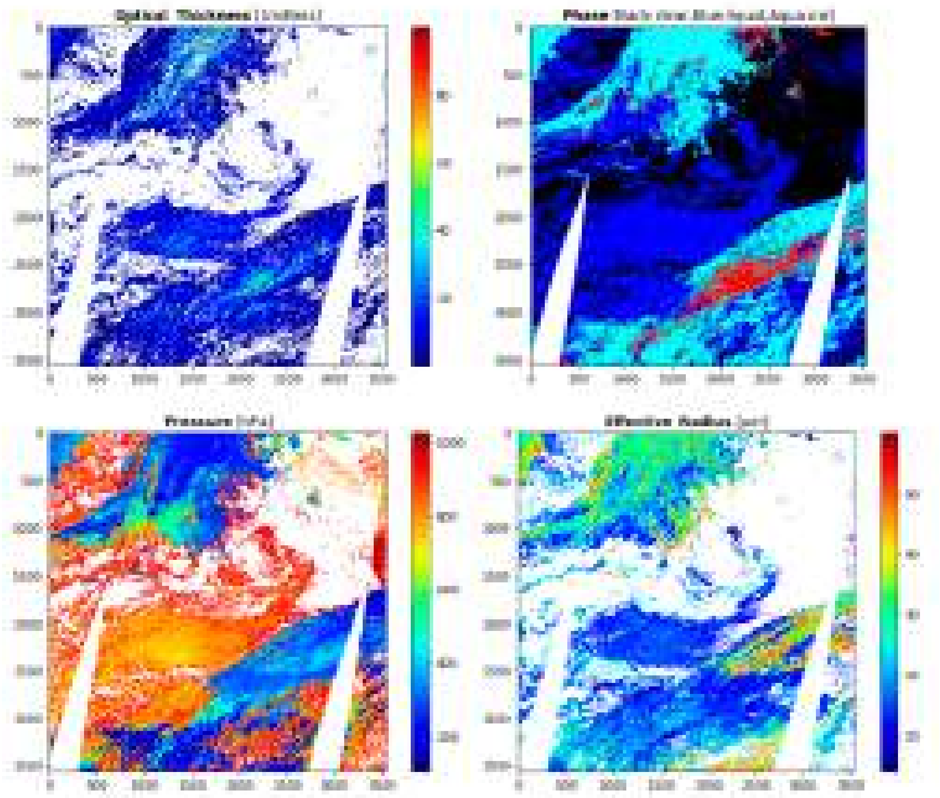




white indicates no data or invalid data; black indicates patches with <30% cloud pixels.

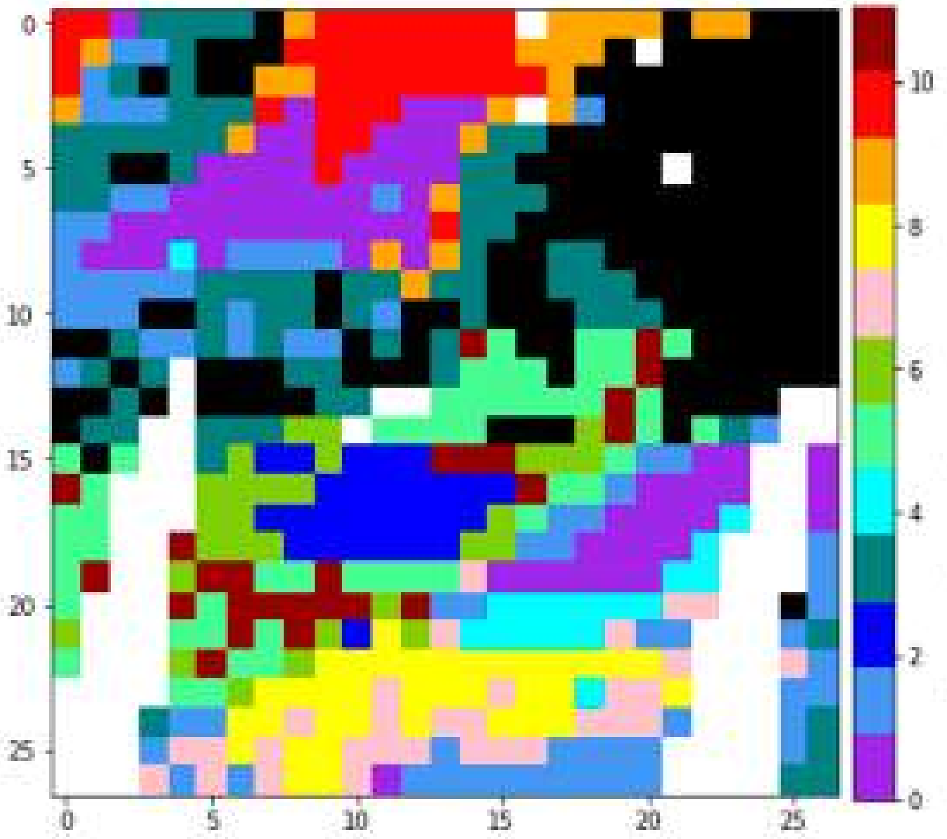
PATCH MEAN VALUES OF MODIS CHANNEL CLUSTERING



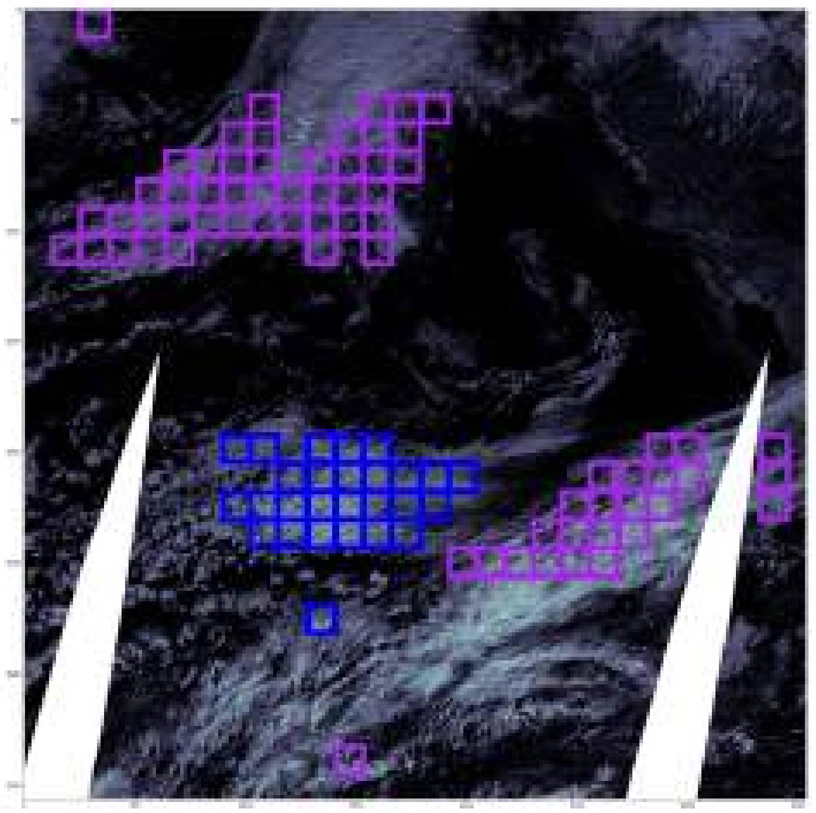


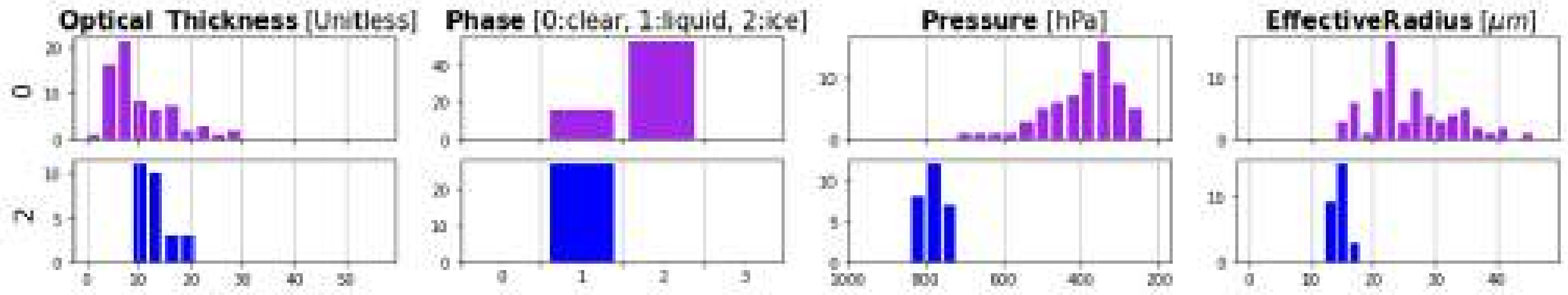
Results

ENCODER - DECODER CLUSTERING



Visible image with cluster 0 (violet) and 2 (blue)

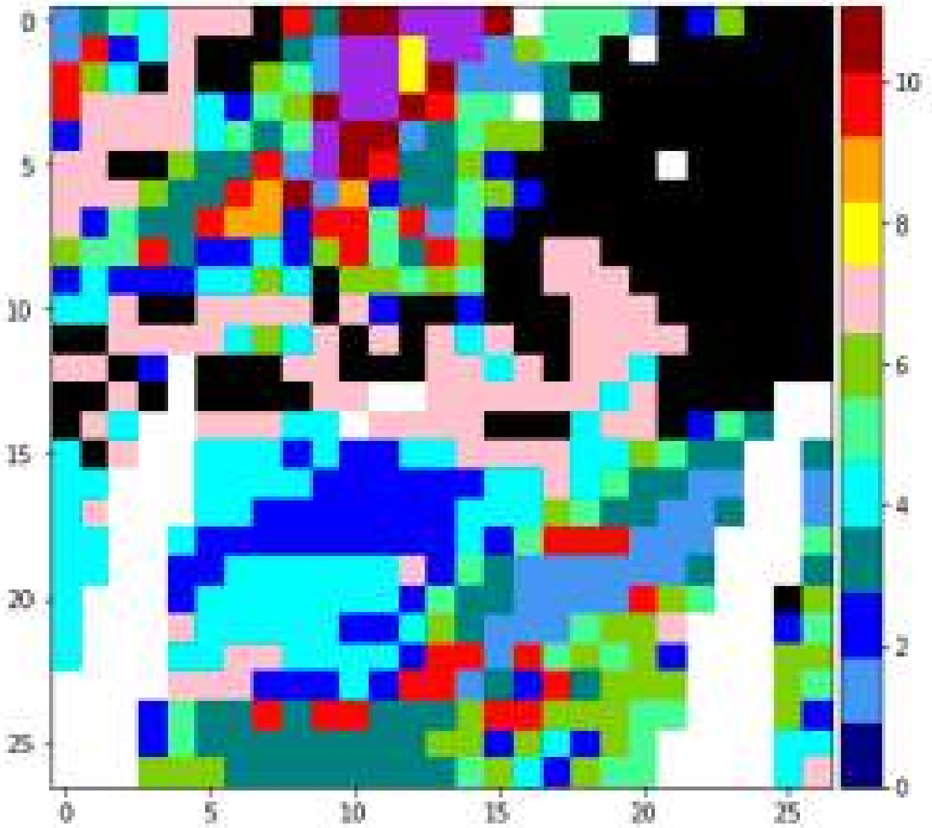


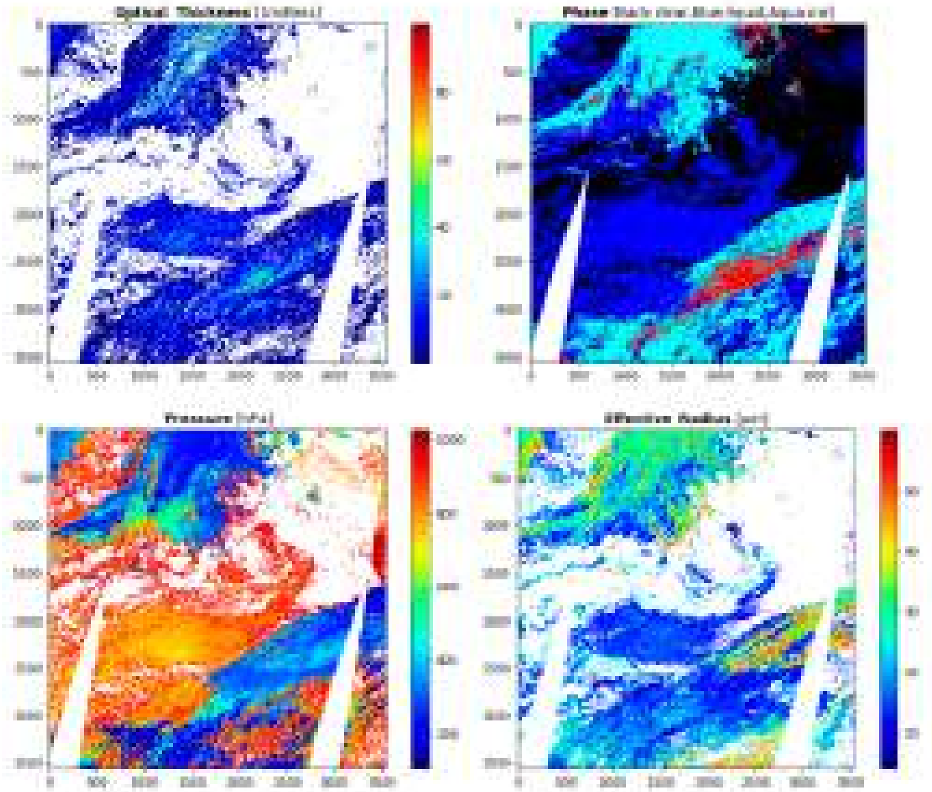


white indicates
no data or invalid data;
black indicates patches
with <30% cloud
pixels.

Cluster 2 is stratocumulus
and cluster 0 is cirrus
clouds

PATCH MEAN VALUES OF MODIS CHANNEL CLUSTERING

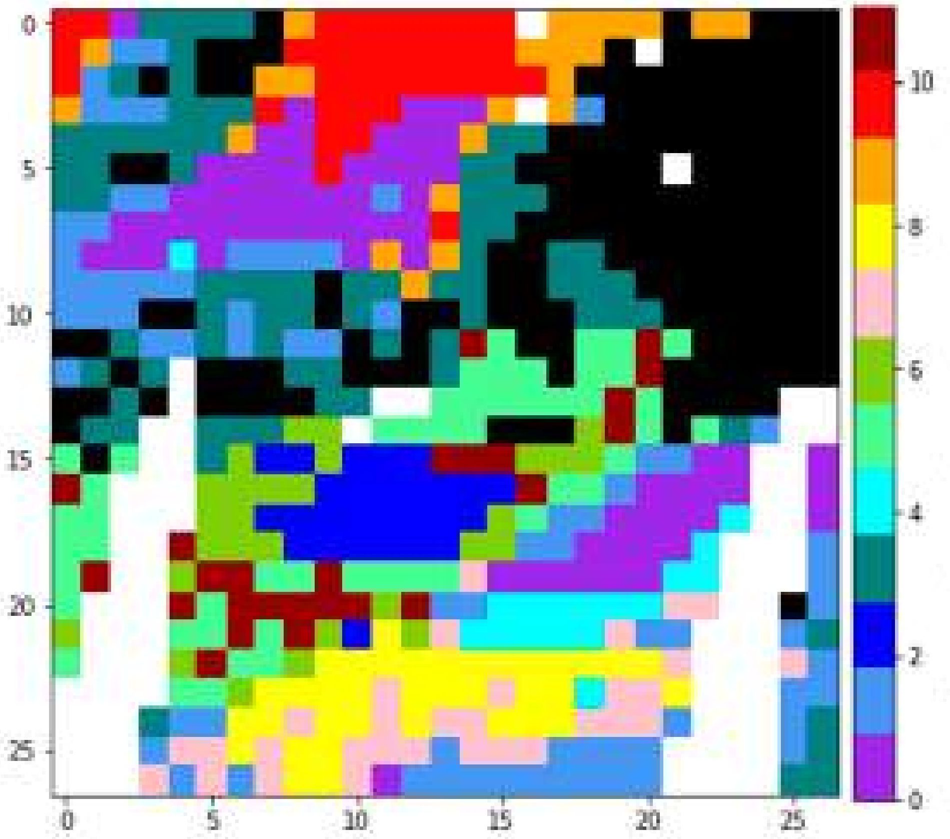




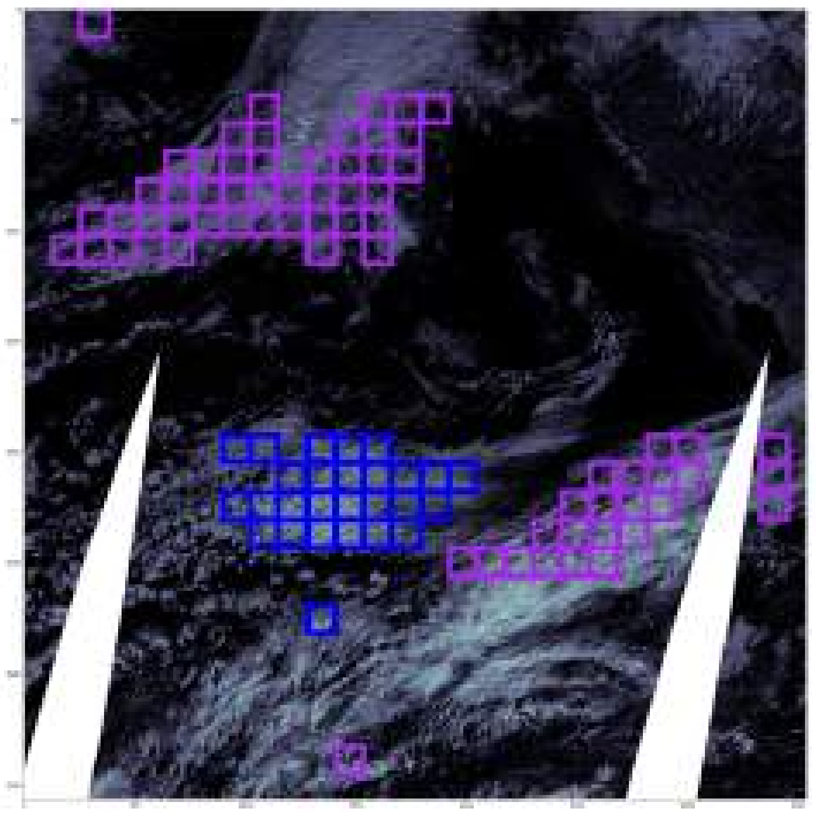
clustering via
autoencoder produces
classes that are spatially
more
cohesive and that better
capture important
physical
transitions

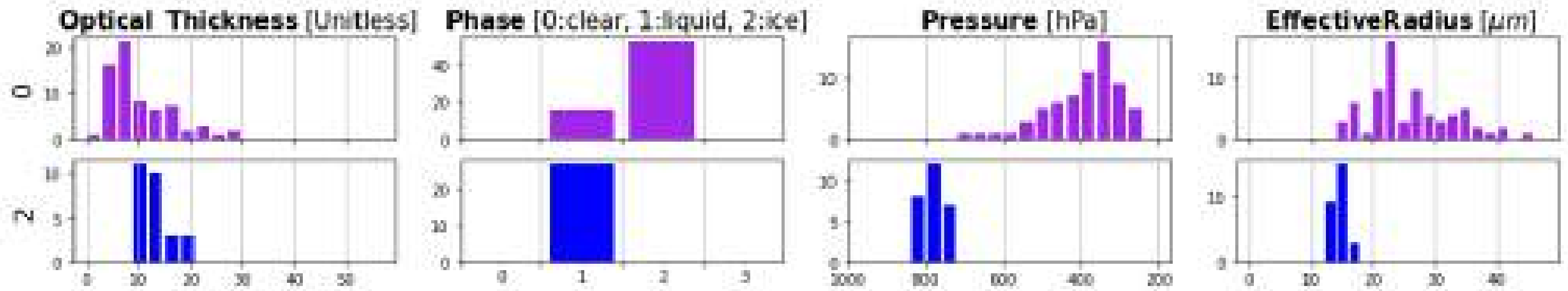
Results

ENCODER - DECODER CLUSTERING



Visible image with cluster 0 (violet) and 2 (blue)



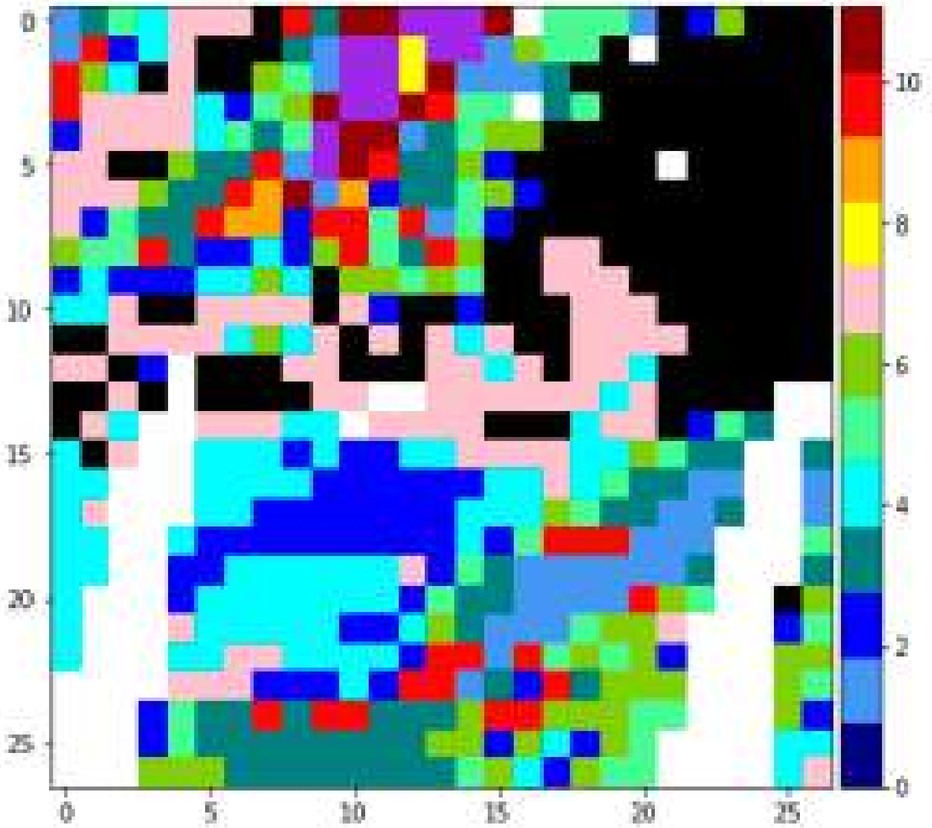


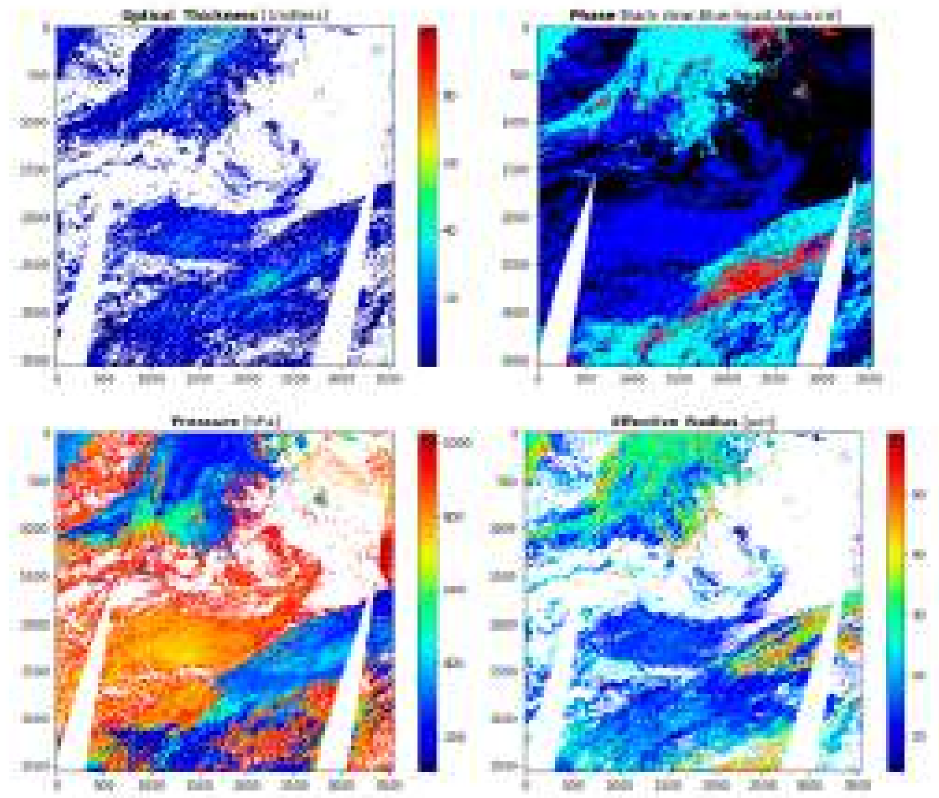
Optical Thickness [Unitless]
Phase [0:clear, 1:liquid, 2:ice]
Pressure [hPa]
EffectiveRadius [μm]
0
2

white indicates no data or invalid data; black indicates patches with <30% cloud pixels.

Cluster 2 is stratocumulus and cluster 0 is cirrus clouds

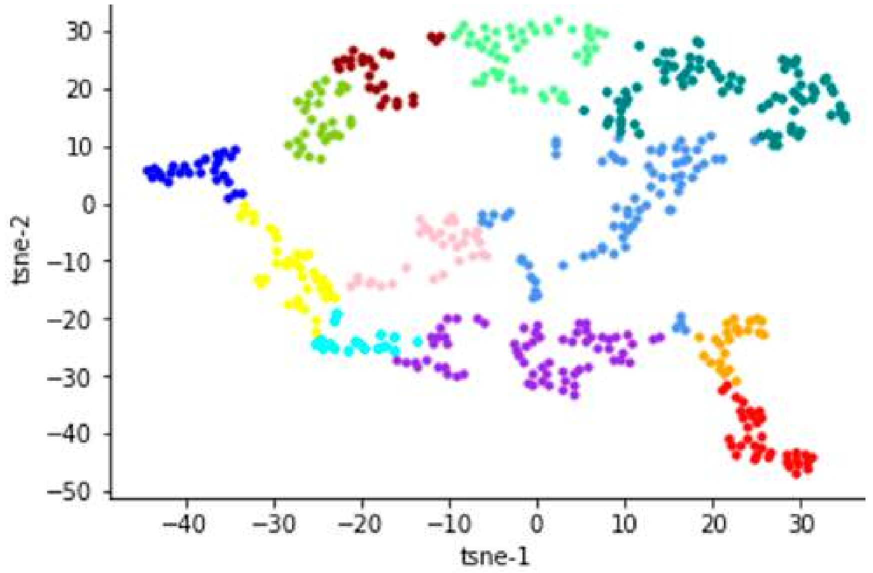
PATCH MEAN VALUES OF MODIS CHANNEL CLUSTERING





clustering via autoencoder produces classes that are spatially more cohesive and that better capture important physical transitions

T-sne representation of autoencoder clusters shows cohesive and distinct patch clusters



Side thoughts on crop image sizes, channels and domain sizes

SUPERVISED

Stevens, 2019: around 1000x2000 km, visible images from MODerate-resolution Imaging Spectroradiometer (MODIS)

Rasp, 2019: Terra and Aqua MODIS visible images from NASA Worldview subregions a larger image (larger than Stevens)

UNSUPERVISED

Kiruhana, 2022: 128 x 128 km, 6 selected bands radiances and MODIS products

Denby, 2019: 256 x 256 pixels corresponding to 200 km size, RGB composite with visible channels

Only around 50% of the images are classified, what do we do with the rest?

Single or multiple inputs?
Visible channels or
multi-frequency radiances ?

Side thoughts on crop image sizes, channels and domain sizes

SUPERVISED

Stevens, 2019: around 1000x2000 km, visible images from MODerate-resolution Imaging Spectroradiometer (MODIS)

Rasp, 2019: Terra and Aqua MODIS visible images from NASA Worldview subregions a larger image (larger than Stevens)

UNSUPERVISED

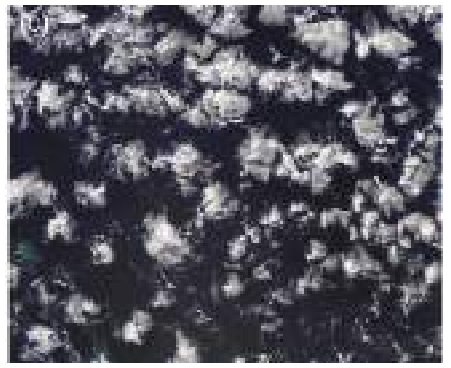
Kiruhana, 2022: 128 x 128 km, 6 selected bands radiances and MODIS products

Denby, 2019: 256 x 256 pixels corresponding to 200 km size, RGB composite with visible channels

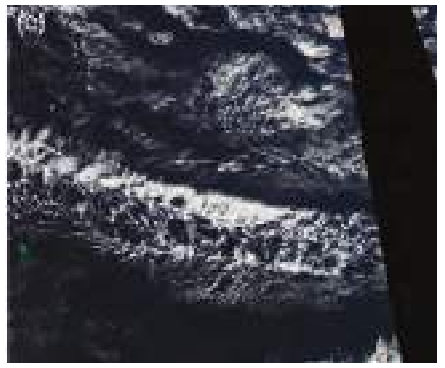
Only around 50% of the images are classified, what do we do with the rest?

Single or multiple inputs? Visible channels or multi-frequency radiances?

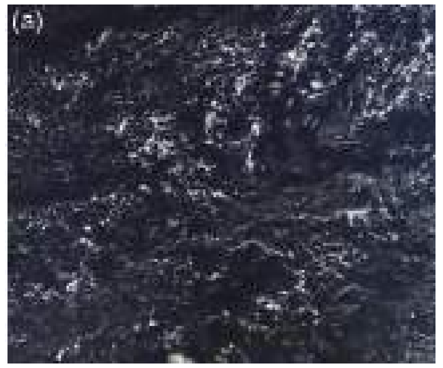
Are these crops large enough to capture all mesoscale pattern organizations?



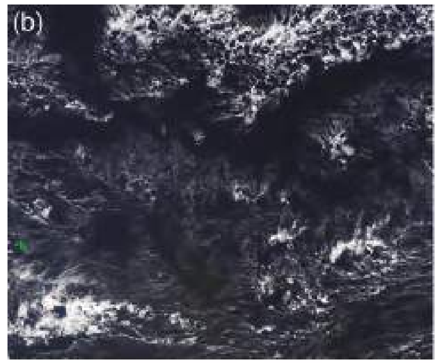
flower (200-200 km)



fish (200-2000 km)



gravel (20-100 km)



sugar

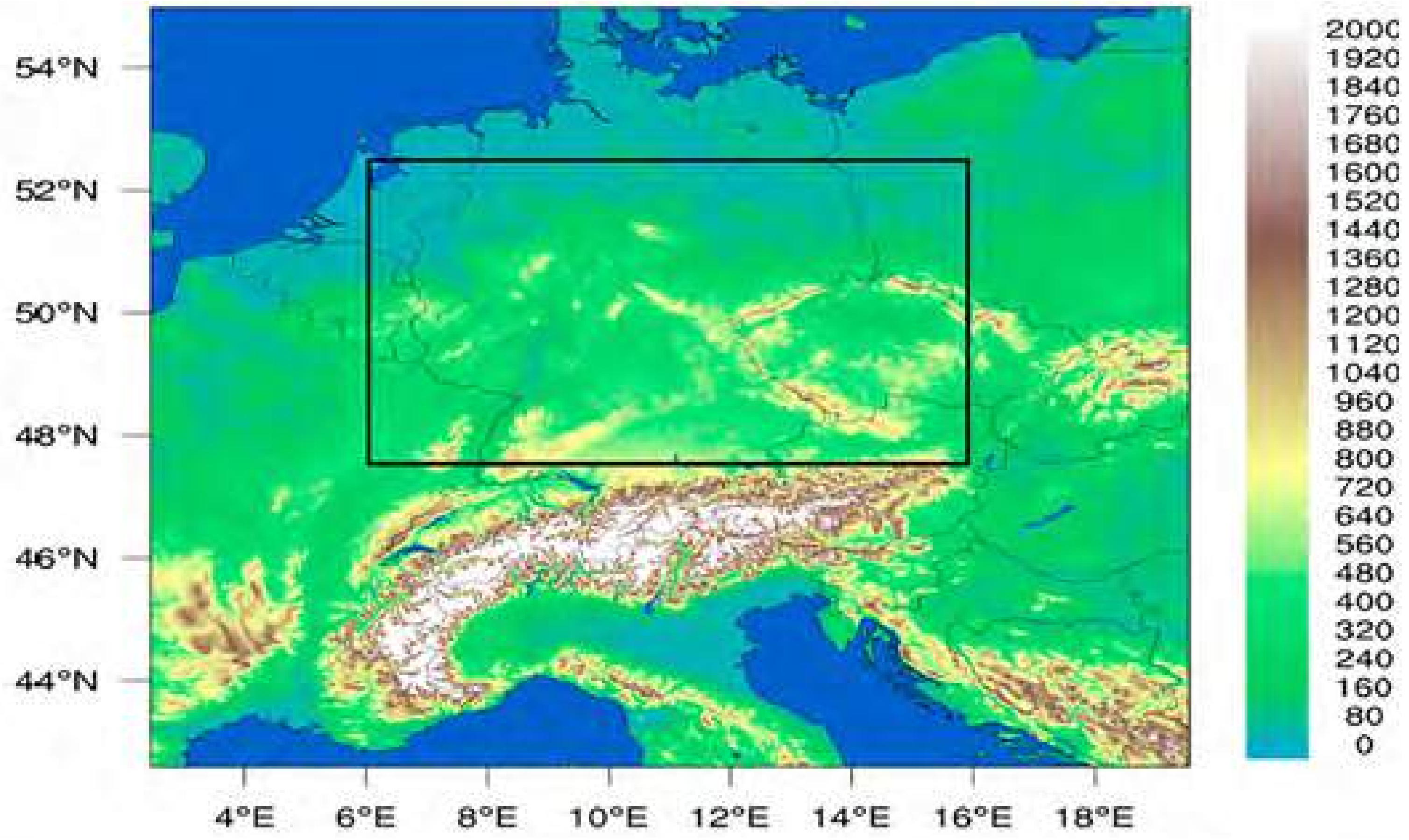
and L2 distances for measuring differences among images?

Overview of cloud classification algorithms

SELF SUPERVISED METHODS

Chatterjee et al., 2022:
Understanding Cloud Systems' Structure and Organization Using a Machine's Self-Learning Approach

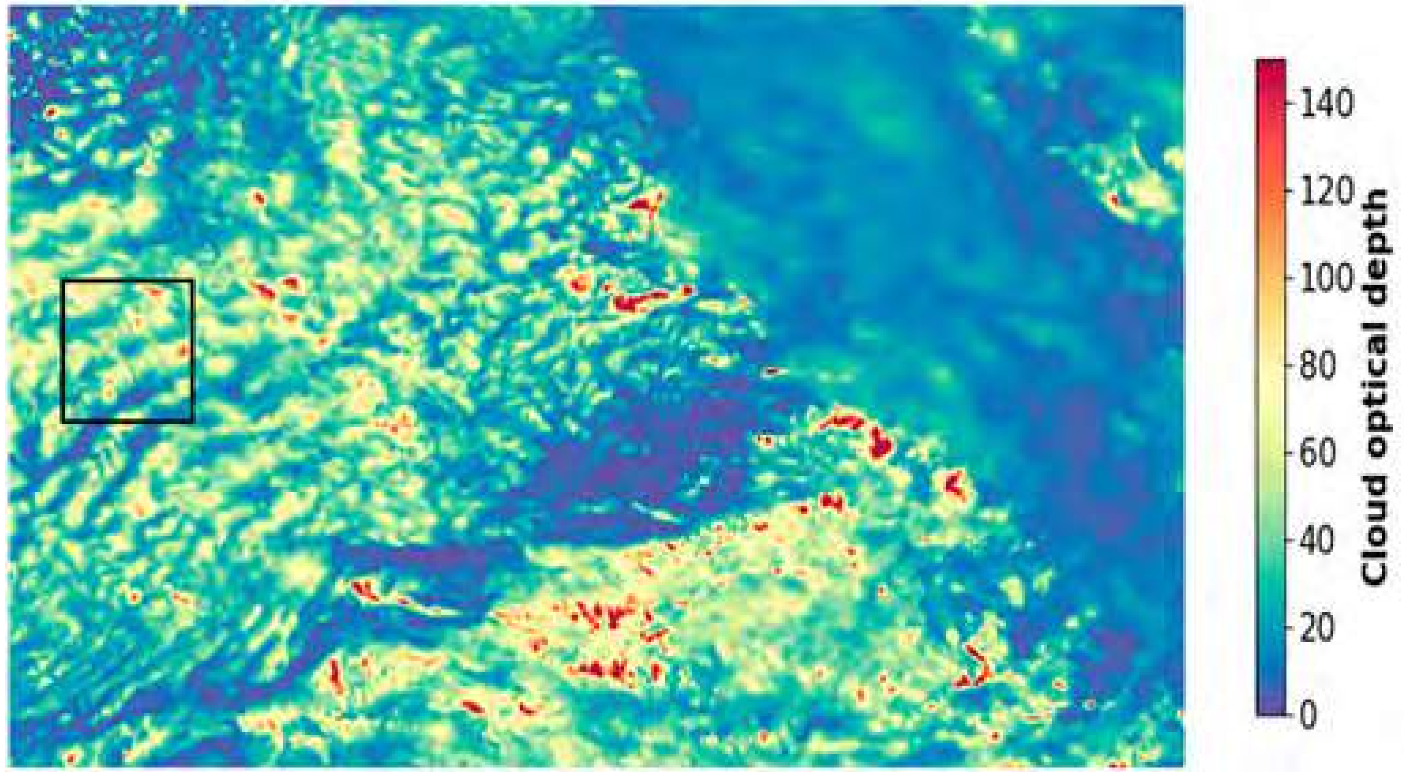
We leave the tropics and we move to Europe

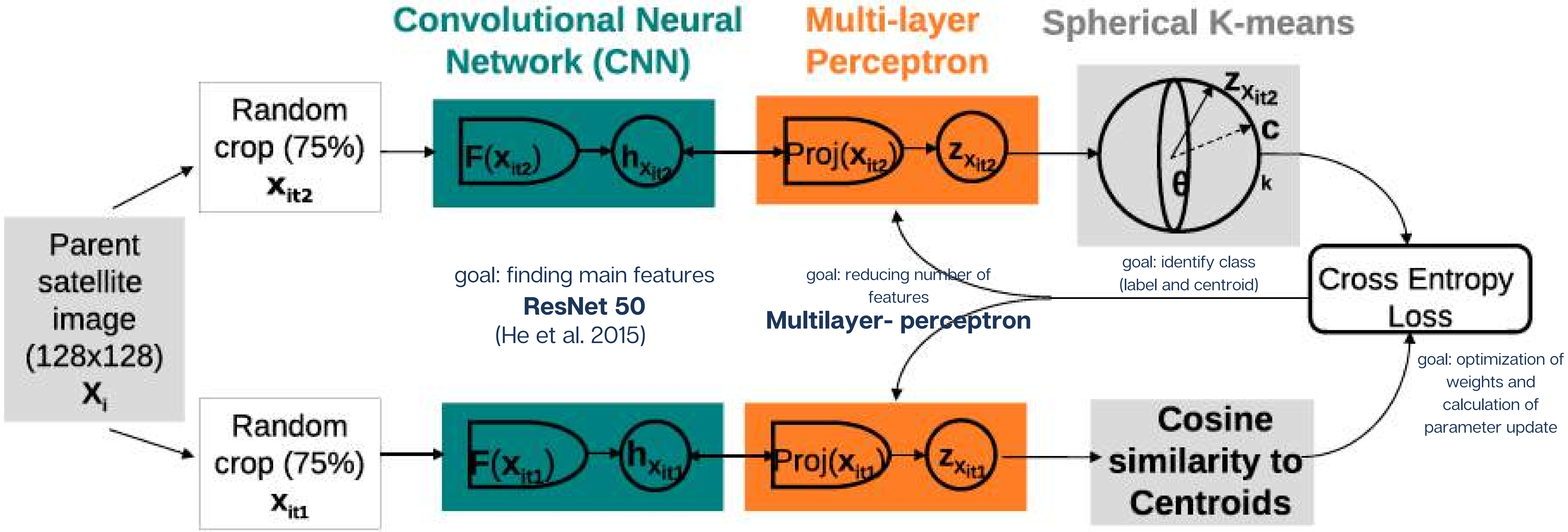


Four images are randomly cropped out of 128 x128 pixels at every time step

Instead of visible channel combinations or radiances, we use a validated product:

Cloud optical thickness





Convolutional Neural Network (CNN)

Multi-layer Perceptron

Spherical K-means

goal: finding main features

ResNet 50

(He et al. 2015)

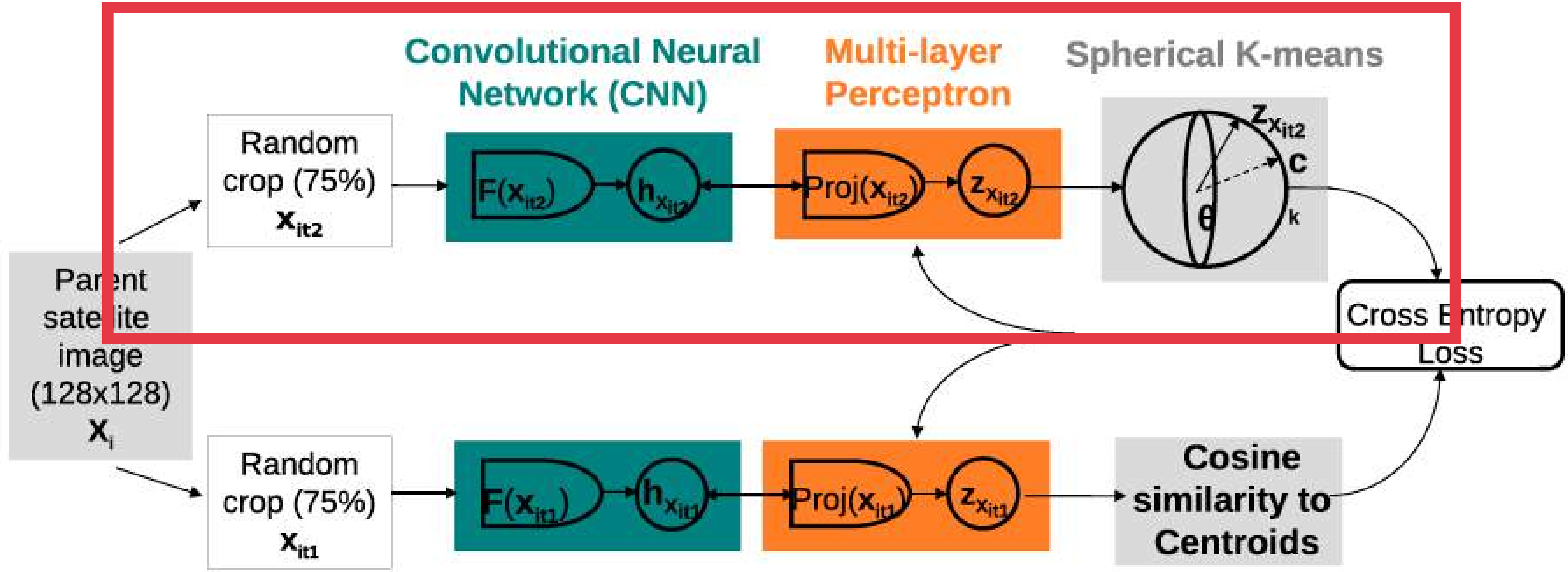
goal: reducing number of features

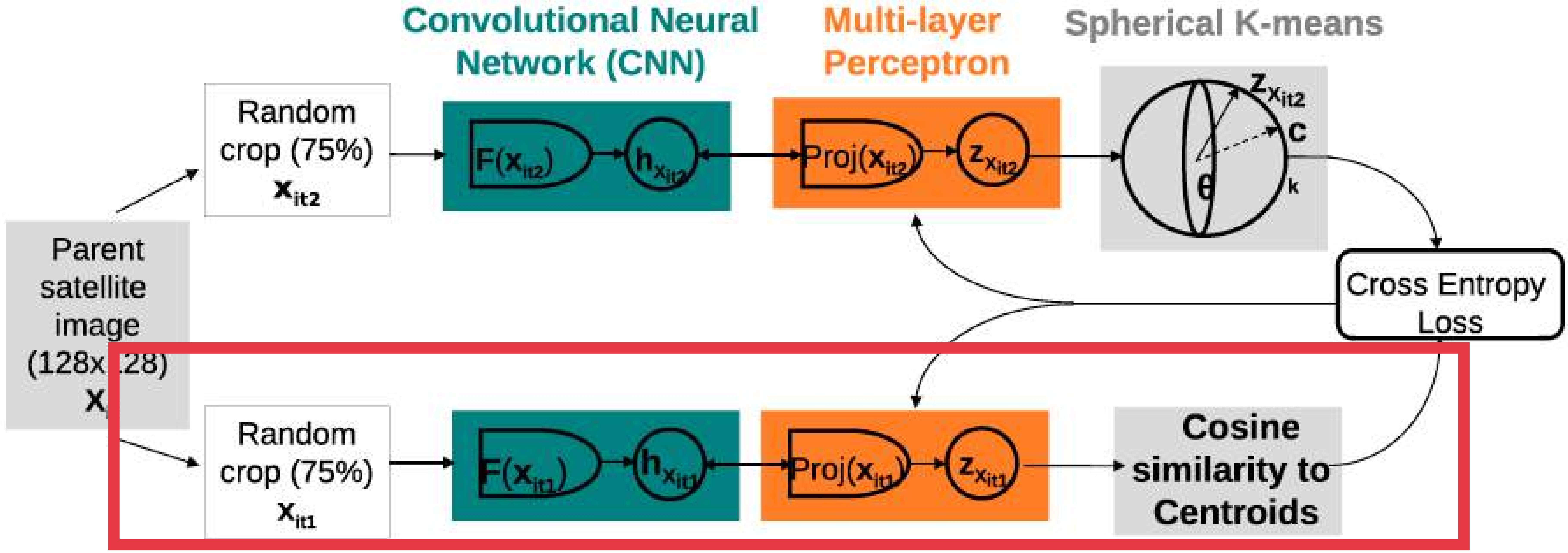
Multilayer- perceptron

goal: identify class (label and centroid)

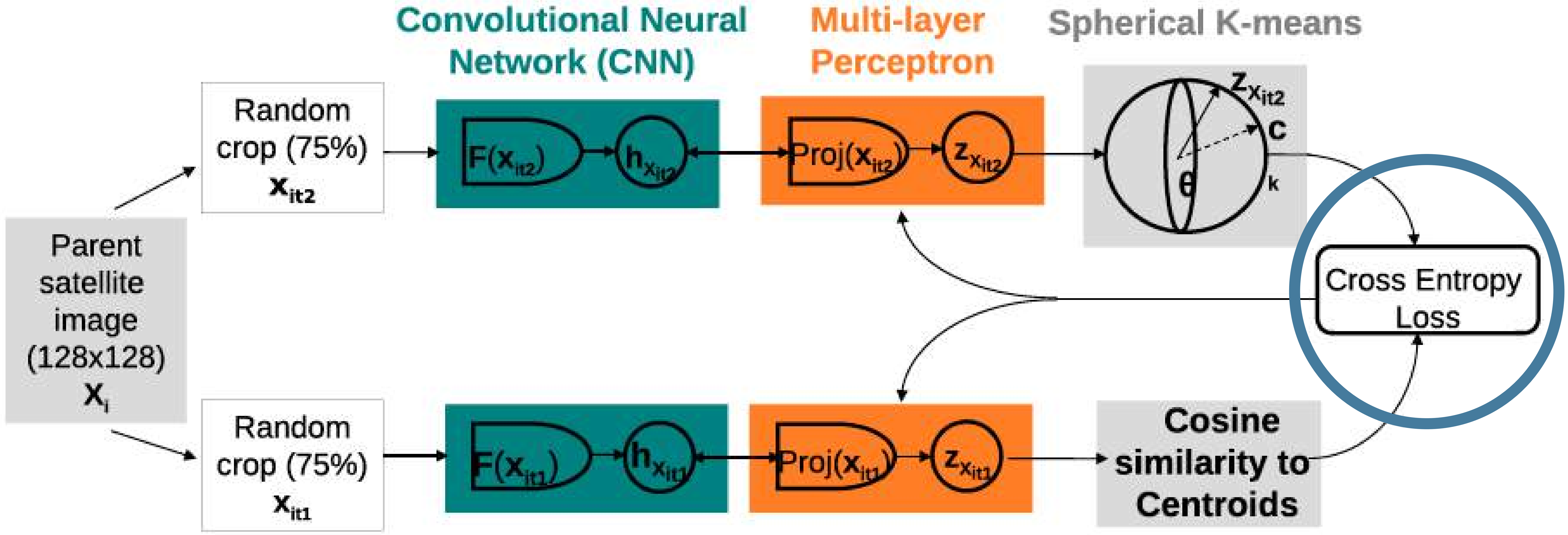
goal: optimization of weights and calculation of parameter update

Through the upper branch, the network **assigns a pseudolabel qs** corresponding to a **centroid Cs** to the feature vector Z obtained from the image xit2.





In the lower branch we also obtain a feature array z, which is compared to the one from the other branch by inserting it in the cost function together with the output (Cs, qs) of the upper branch



When the feature vectors of the two branches capture similar information about the original image, the loss becomes lower and higher when they diverge. That is how the network branches are encouraged to focus on the image characteristics, which progressively makes the feature vectors similar.

Training and classification

The seven centroids in show distinct COD patterns

Centroid 2 is associated mainly with clear-sky conditions

Centroid 3: optically thin clouds

Centroids with extended cloud fields with mean cloud fraction higher than 94%

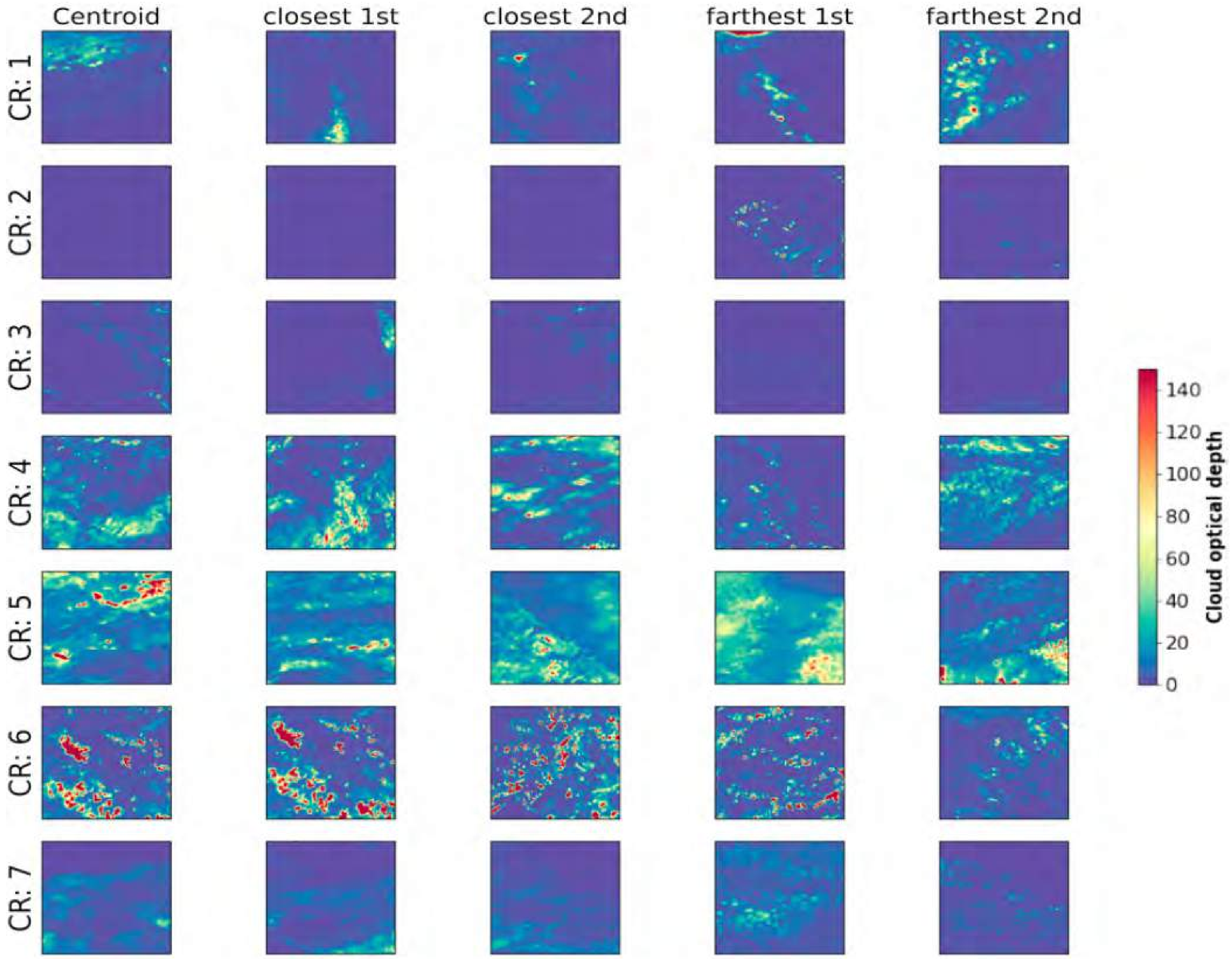
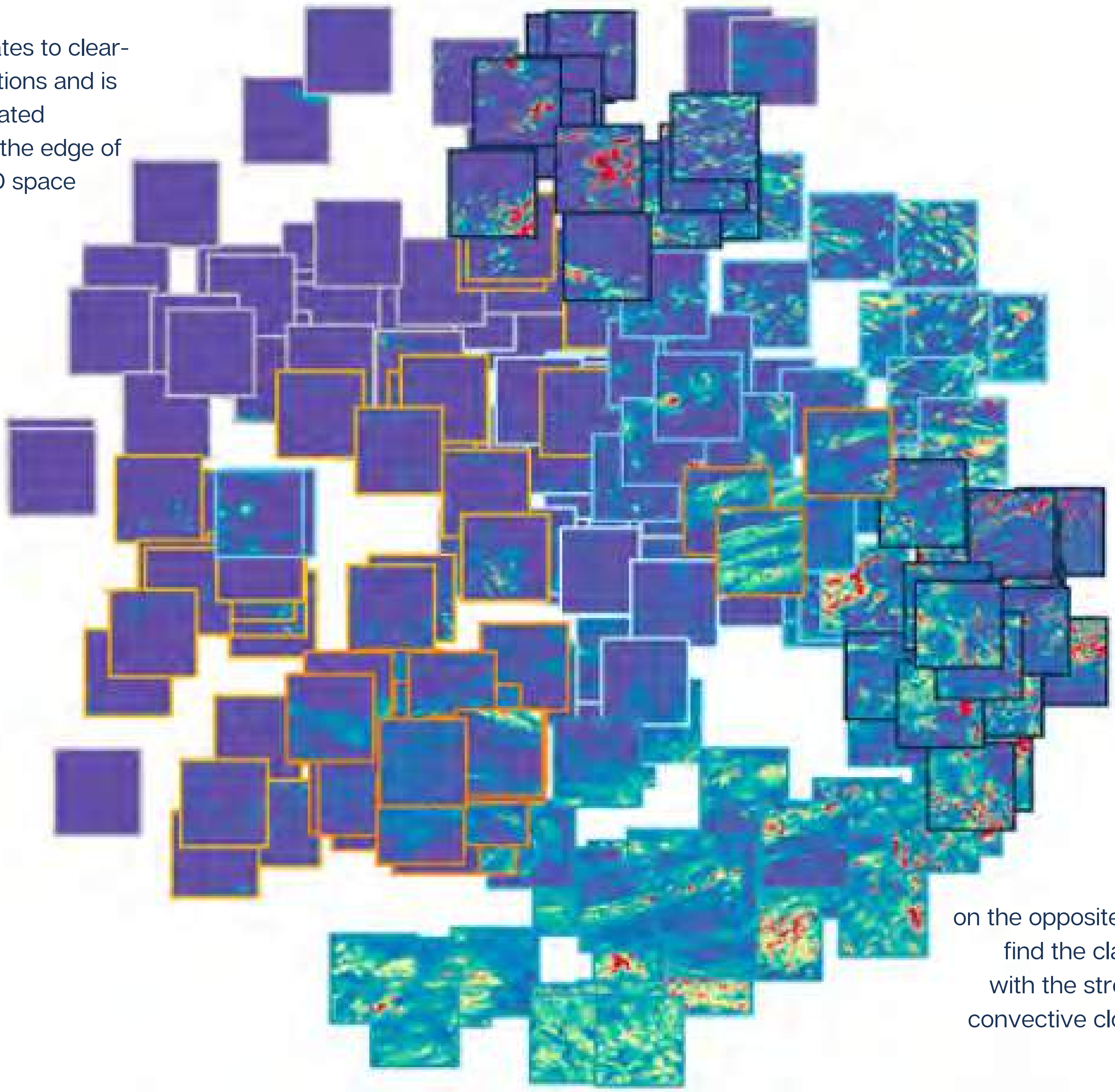


FIG. 4. Each row shows five 128 × 128 COD images belonging to a certain cloud regime (CR) over central Europe for (left) centroid, (left center) closest first, (center) closest second, (right center) second farthest, and (right) farthest; the color scale is shown for COD reference purposes.

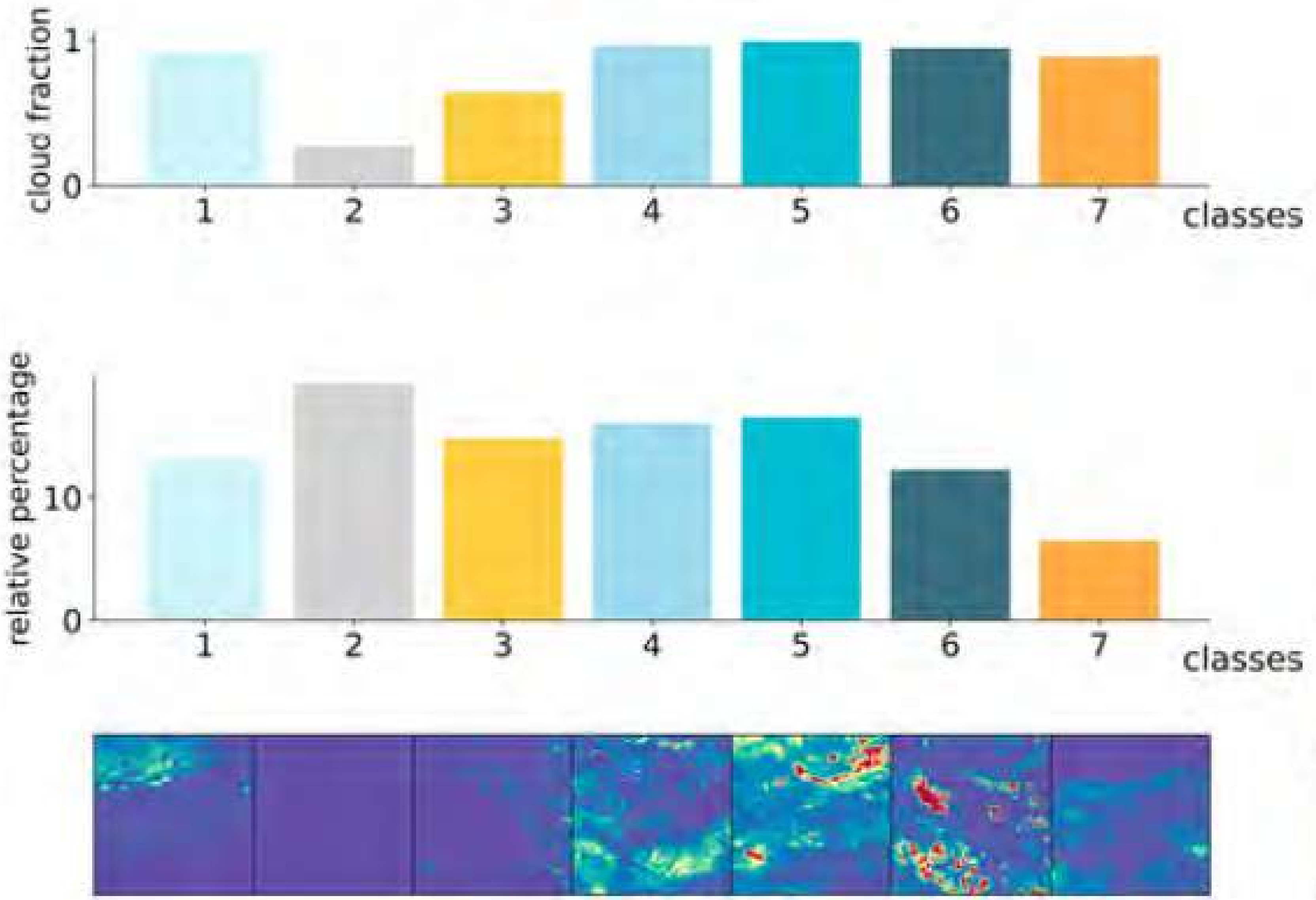
Optimum classification

distinct cloud patterns occupy distant areas

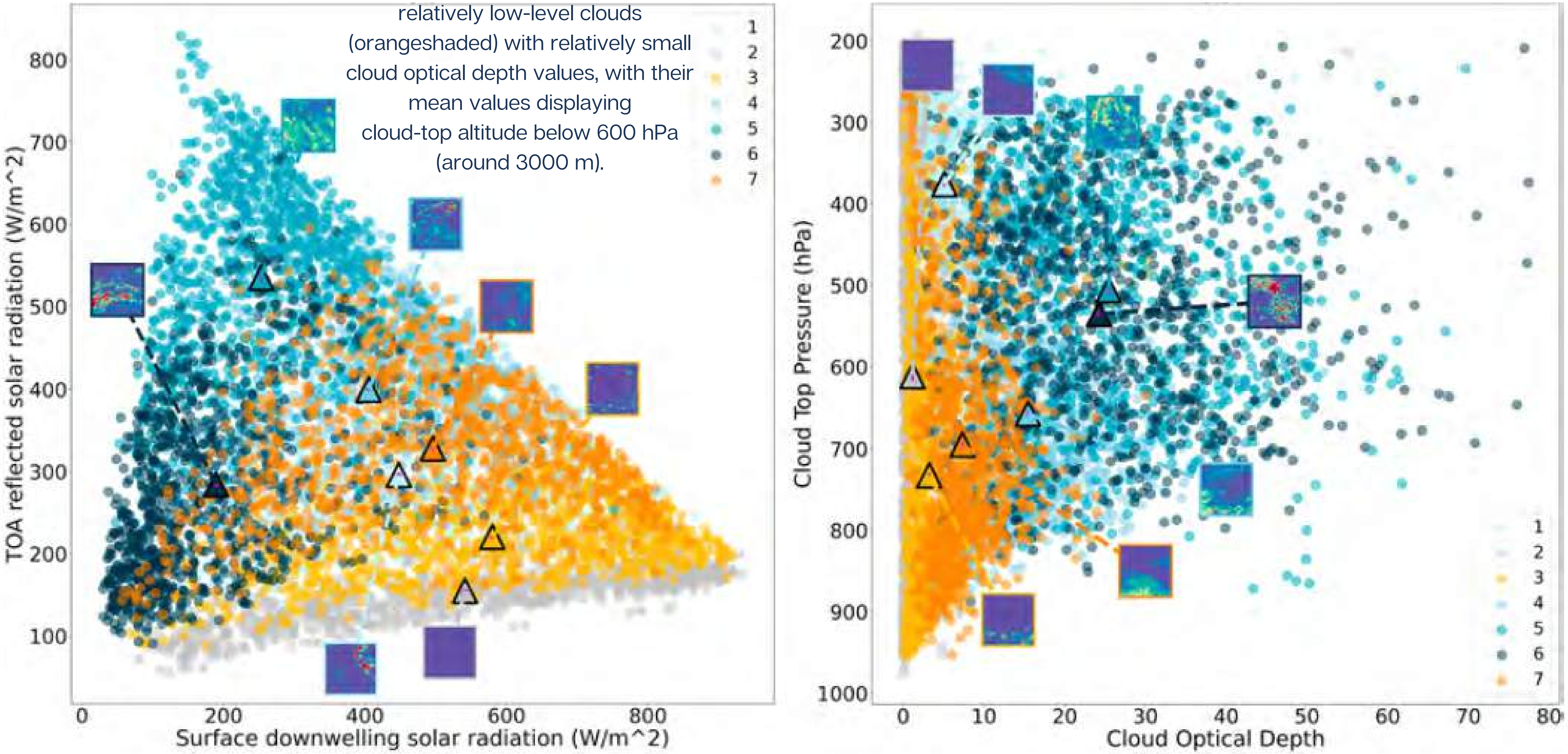
class 2 relates to clear-sky conditions and is located
clearly on the edge of the 2D space



on the opposite edge we find the classes
with the strongest convective cloudiness



Physical properties of clouds

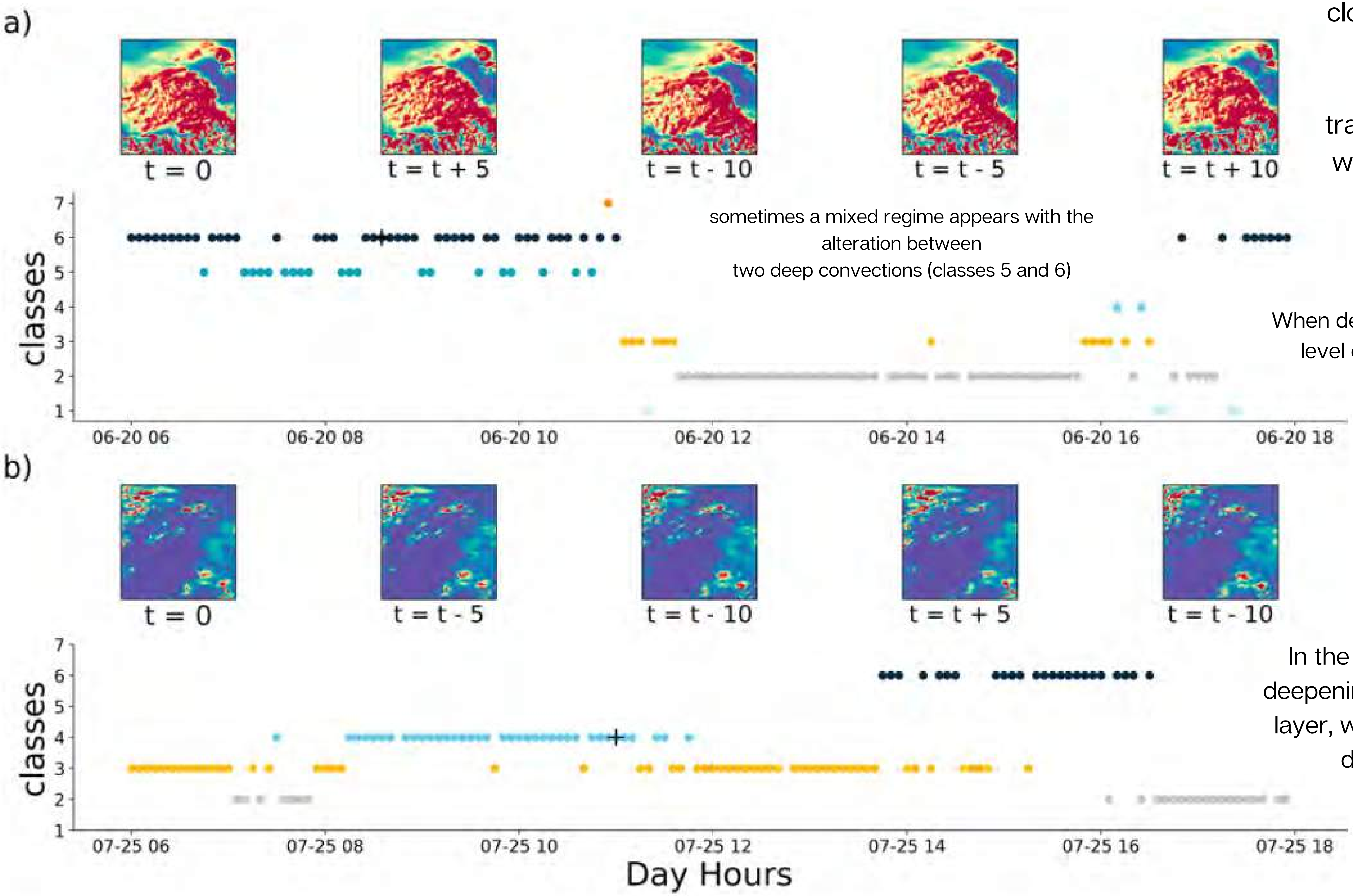


Classes 3 and 7 correspond to relatively low-level clouds (orangeshaded) with relatively small cloud optical depth values, with their mean values displaying cloud-top altitude below 600 hPa (around 3000 m).

Classes 4, 5, and 6 (blue-shaded dots in Fig. 7) all describe overcast situations with cloud fractions of more than 94%. For these classes, average COD values are larger than 10, and they span a broad region of CTP values above 800 hPa

FIG. 7. (a) Mean TOA TRS vs mean SDS given by dots of 1000 randomly selected images of the 128 × 128 configuration, with color indicating the class to which they belong. Triangles indicate the mean position of the classes. The images associated with the triangles are the closest to the mean position of the 1000 images. The crosses in the triangles represent the error bars of each cluster group for the radiative properties on *x* and *y*. (b) As in (a), but in the 2D space given by COD (abscissa) and CTP (ordinate).

We can study the temporal development of specific cloud regimes and exploit the 5-min temporal resolution of the dataset to observe the transformation of cloud systems with time at a higher resolution.



sometimes a mixed regime appears with the alteration between two deep convections (classes 5 and 6)

When deep convection decreases, then low level cloud and then clear sky appear.

In the morning you can also see the deepening of the atmospheric boundary layer, with lower clouds substituted by deeper convective clouds.

that’s it for this course!