Clarifications about yesterday?

- Training dataset
- Training loss
- Model
- Supervised learning / Unsupervised learning
- Regression / Classification
- Perceptron / Feed-forward model / Fully connected NN
- weights, biases
- backpropagation
- gradient descent

Lecture #2: Implementation of Deep Learning models; Convolutional Neural Networks (CNNs)

Implementation of Deep Learning models

What is a DL model?

- DL models are feedforward neural networks (NN) in general
- Deep NN are NN with a significant number of layers
- A Multi Layer Perceptron (MLP) is a NN where neurons in one layer are all connected to all neurons of the next layer. It is also called Dense Neural Network (DNN)

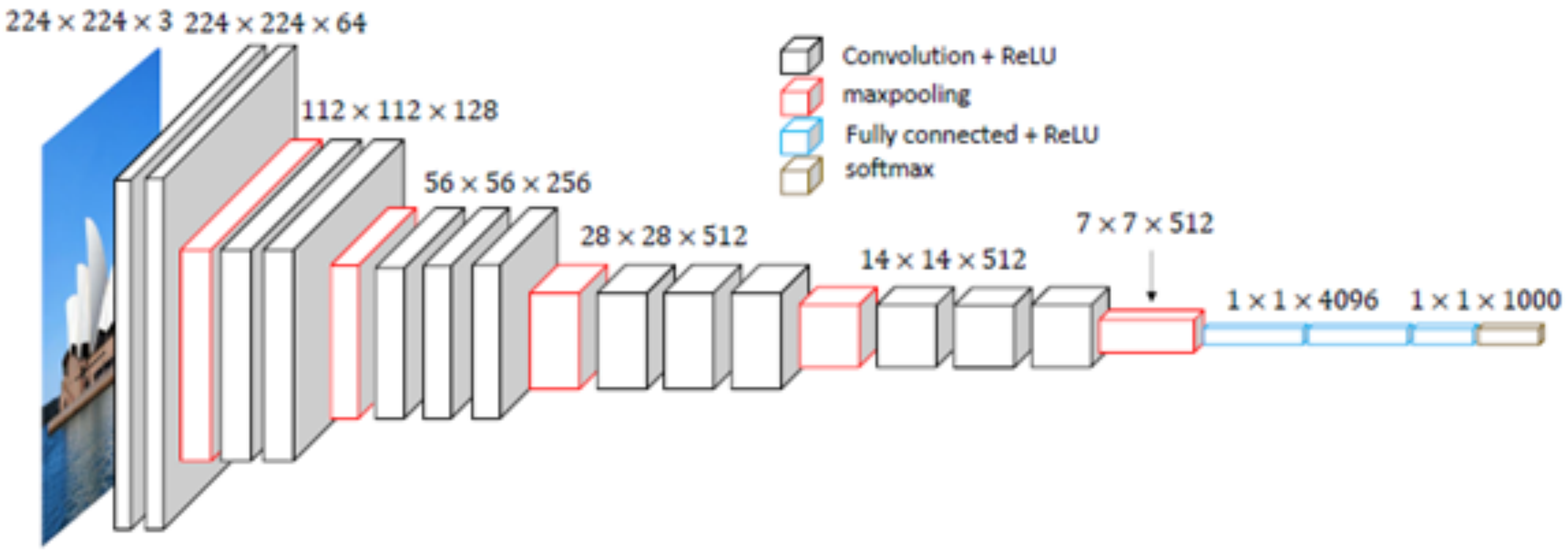


Figure 2: The architecture of VGG16 model .

Guidelines to implement DL models

1. Problem formulation (inputs/outputs)
2. Data collection (cf. supervised vs. non-supervised)
3. Definition of performance metrics
4. Selection of neural architectures (at least 2 models)
5. Selection of a training loss
6. Split dataset into training / validation / test datasets
7. Train the selected models from the training dataset and save the best models onto the validation dataset
8. Benchmark the performance of the trained models onto the test dataset
9. Update/iterate 4-5-6-7-8

Splitting the dataset

- Your dataset is ready. You use it (all of it) to train your model. What is the problem with this?

Dataset

Splitting the dataset

- A fraction of the dataset is not used for training, only for testing the model.

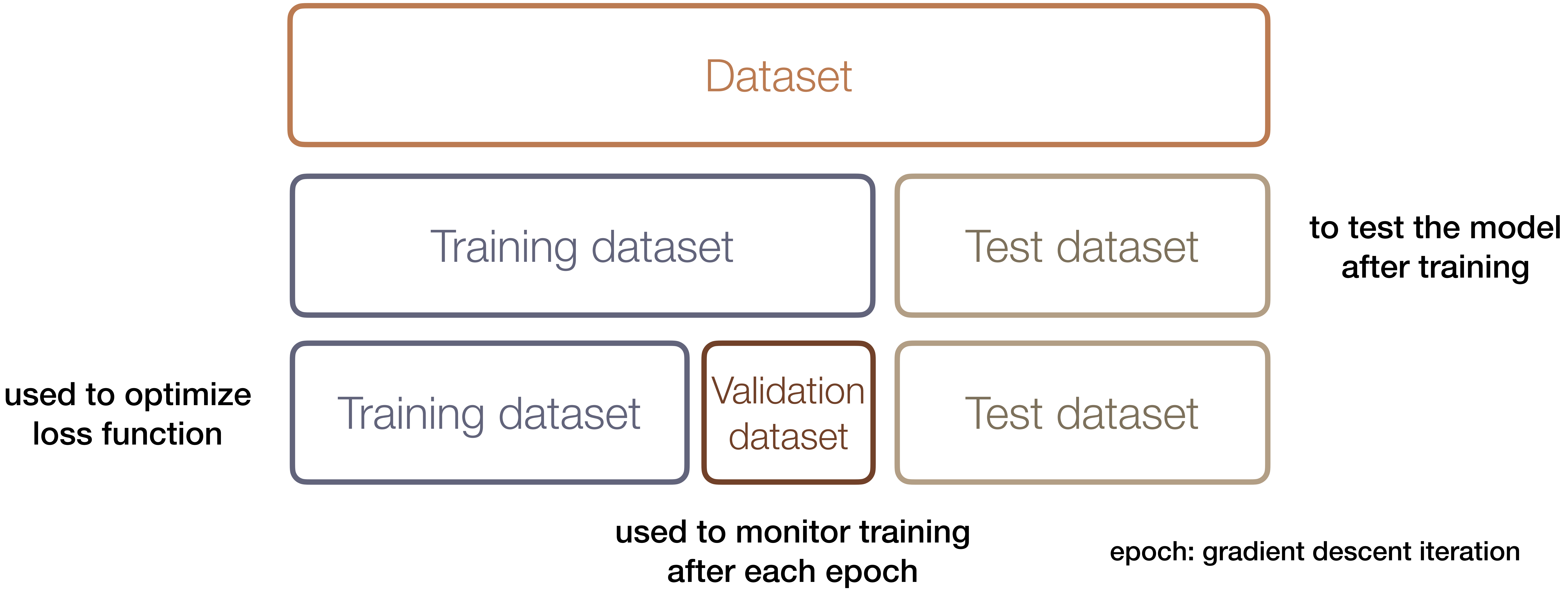
to test the model after training

Overfitting

- Quite often, training leads to a model being too specific to the training data: it performs poorly with test data.

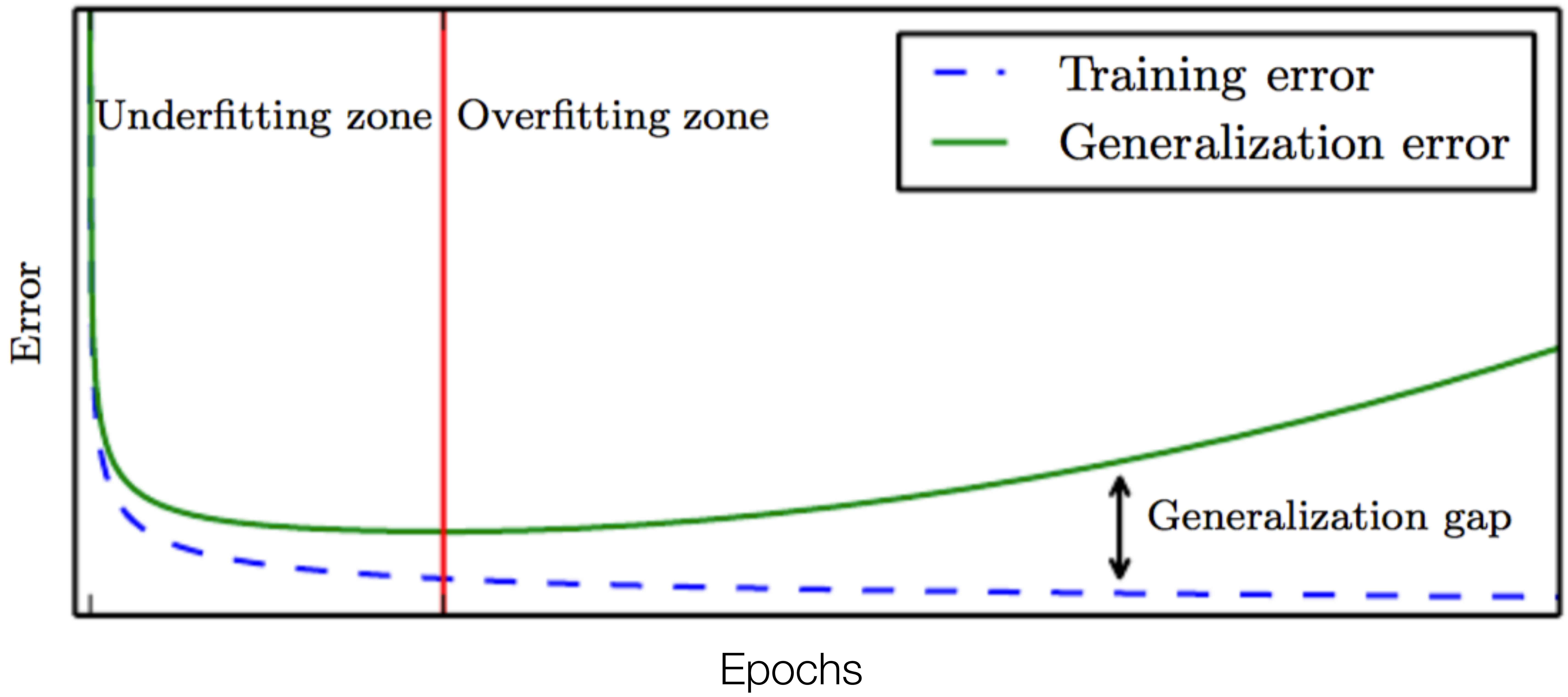
Splitting the dataset and avoid overfitting

- For this reason, the training dataset is further divided in training and validation datasets.



epoch: gradient descent iteration

Splitting the dataset and avoid overfitting



Early stopping: stopping iterations when the validation curve starts to grow.

Other options to minimize risks of overfitting

- Reduce model capacity, i.e. the number of parameters. i.e. simplify the model;

Other options to minimize risks of overfitting

- Reduce model capacity, i.e. the number of parameters. i.e. simplify the model;
- Weight **regularization**:

$$\tilde{J}(\theta) = J(\theta) + \alpha\,\Omega(\theta)$$

$$\Omega(\theta) = ||w||_1 = \sum_i |w_i| \qquad \text{or} \qquad \Omega(\theta) = \frac{1}{2}||w||_2^2$$

Other options to minimize risks of overfitting

- Reduce model capacity, i.e. the number of parameters. i.e. simplify the model;
- Weight regularization;
- **Dropout** layers:

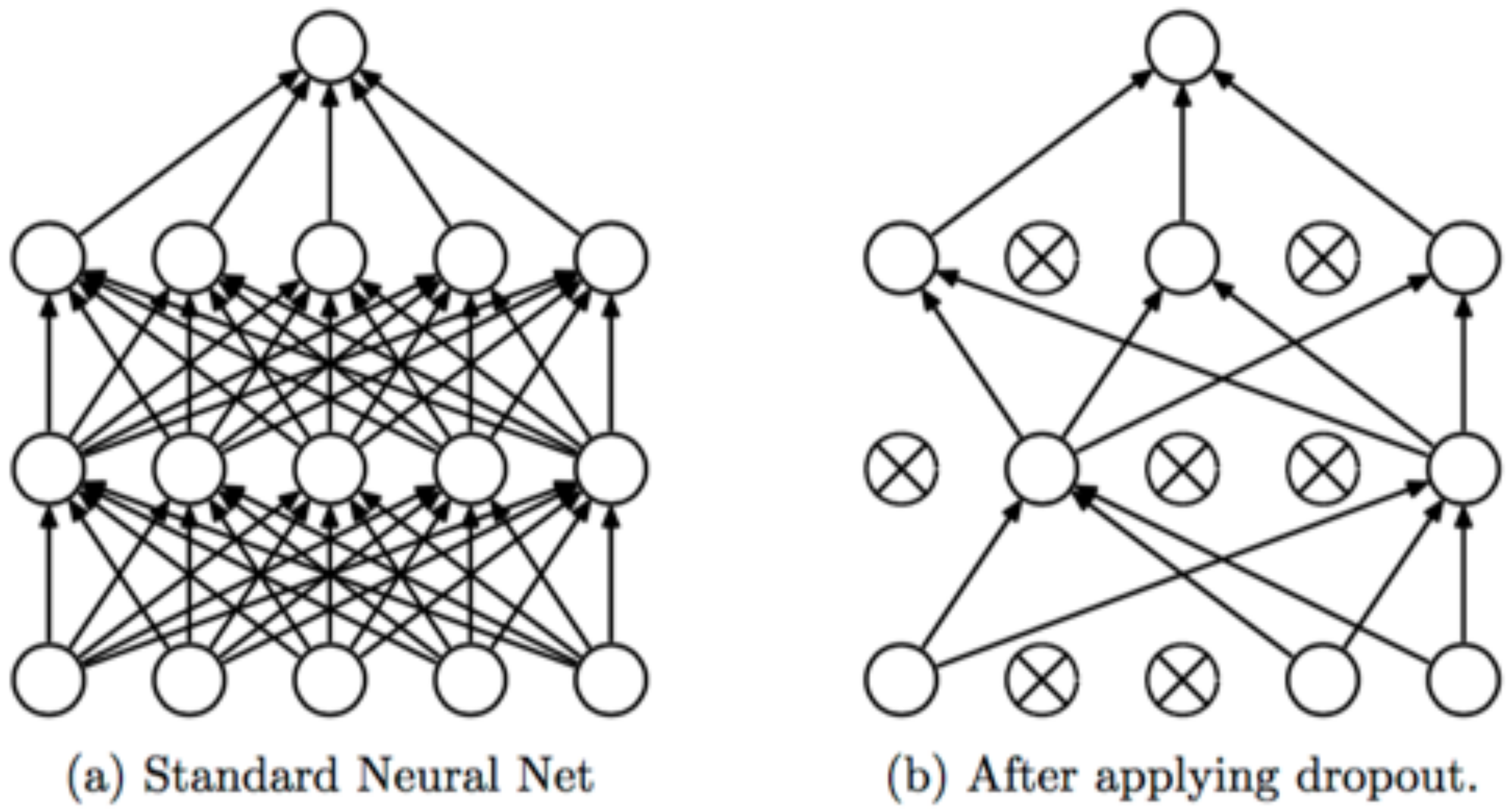


Figure 1: Dropout Neural Net Model. **Left**: A standard neural net with 2 hidden layers. **Right**: An example of a thinned net produced by applying dropout to the network on the left. Crossed units have been dropped.

Srivastava et al, 2014

Other options to minimize risks of overfitting

- Reduce model capacity, i.e. the number of parameters. i.e. simplify the model;
- Weight regularization;
- Dropout layers;
- Use more training data:
    - New data (from a physical model?)

Other options to minimize risks of overfitting

- Reduce model capacity, i.e. the number of parameters. i.e. simplify the model;
- Weight regularization;
- Dropout layers;
- Use more training data:
    - New data (from a physical model?)
    - **Data augmentation**: apply geometric transformation to images (translation, rotation, zoom…) or add noise

Other options to minimize risks of overfitting

- Reduce model capacity, i.e. the number of parameters. i.e. simplify the model;
- Weight regularization;
- Dropout layers;
- Use more training data:
    - New data (from a physical model?)
    - **Data augmentation**: apply geometric transformation to images (translation, rotation, zoom…) or add noise
    - Cross-validation methods

K-fold cross-validation

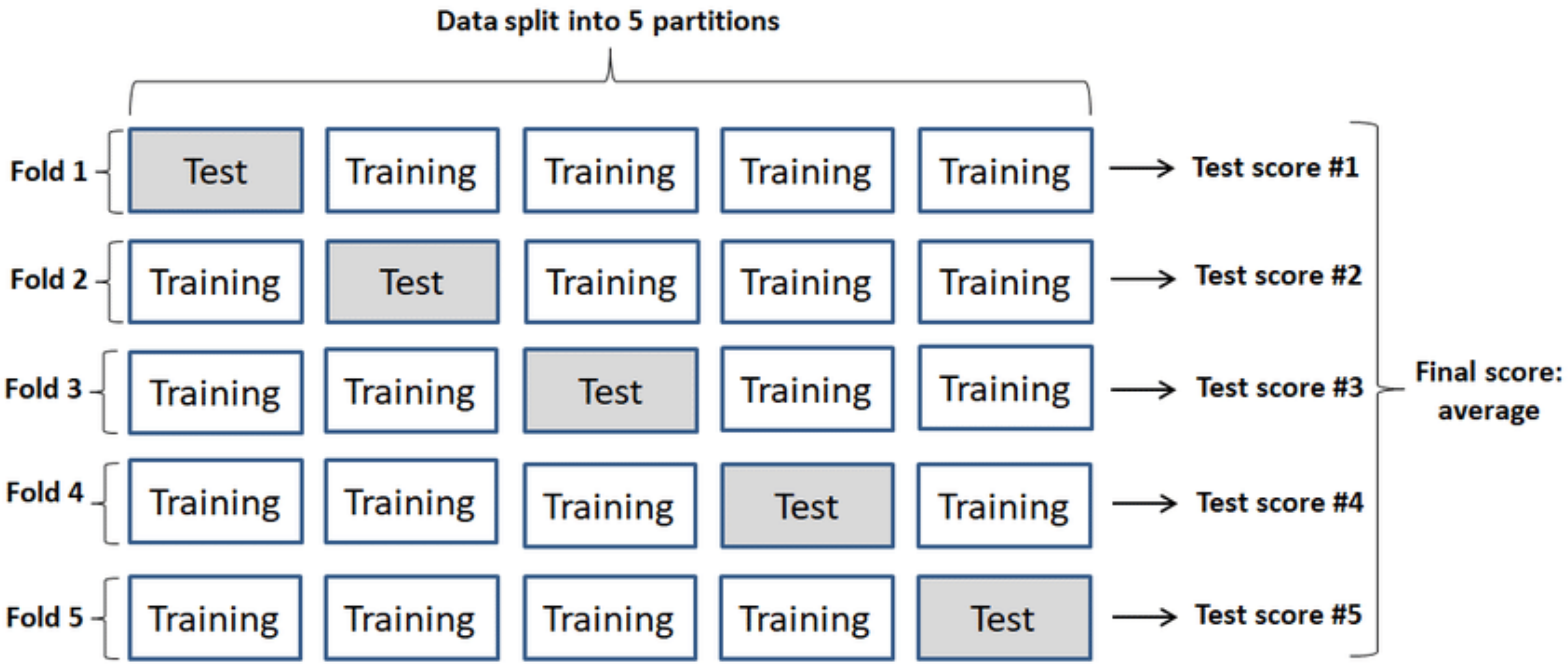


Image: Phung and Rhee, 2019

Data preprocessing

- Vectorization: organization of the dat in numerical arrays of appropriate shapes (can be a pain with text, images of different sizes, etc)
- Normalization: to avoid to give more importance to high-valued quantities (centimeters) than low-valued quantities (kilometers)…

Metrics

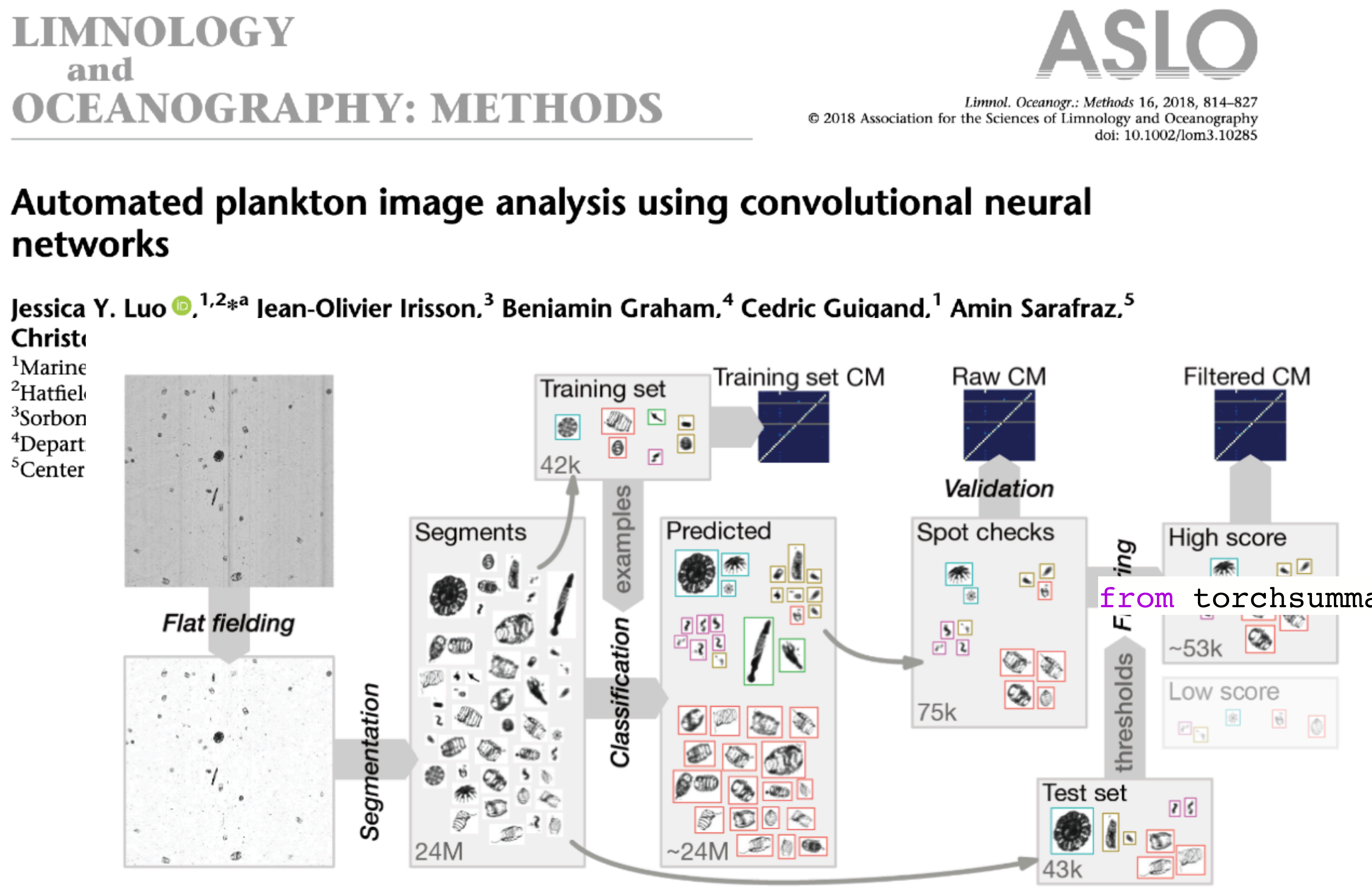
- Metrics must be chosen to provide a quantitative assessment of model performance.
- They are used to measure model performance with test data.
- It is also used with training and validation datasets to monitor overfitting.
- If a metric is differentiable, it can be used as loss function.

Hands on!

- https://github.com/CIA-Oceanix/DLOA2023/blob/main/lectures/notebooks/notebook_MNIST_classification_MLP_CNN_students.ipynb

Convolutional Neural Networks (CNN)

CNNs in OA sciences: in fast development


LIMNOLOGY
and
OCEANOGRAPHY: METHODS

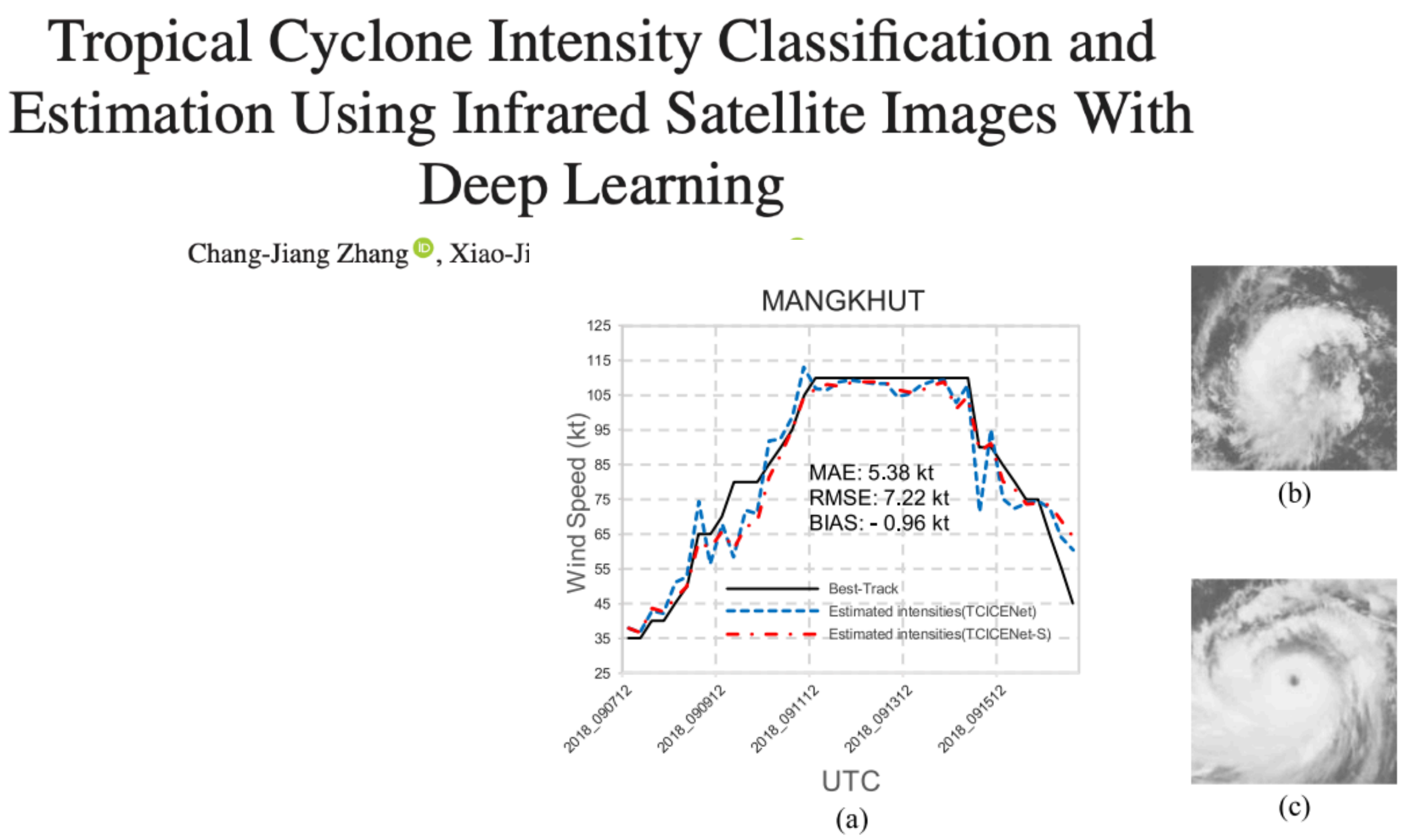
ASLO

Limnol. Oceanogr.: Methods 16, 2018, 814–827
© 2018 Association for the Sciences of Limnology and Oceanography
doi: 10.1002/lom3.10285

Automated plankton image analysis using convolutional neural networks

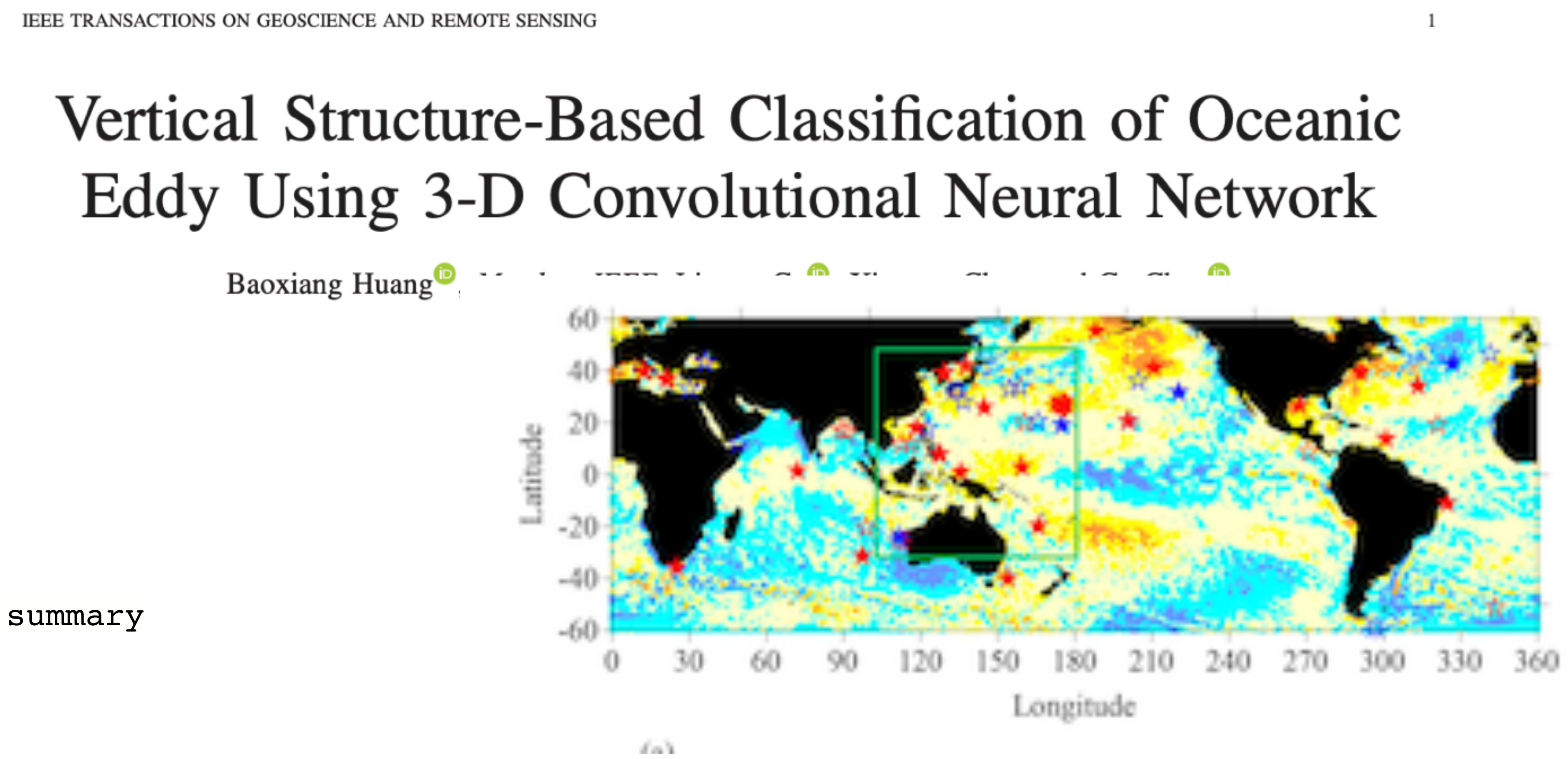
Jessica Y. Luo,[1,2*a] Jean-Olivier Irisson,[3] Benjamin Graham,[4] Cedric Guigand,[1] Amin Sarafraz,[5]


```
from torchsummary import summary
```


Tropical Cyclone Intensity Classification and Estimation Using Infrared Satellite Images With Deep Learning

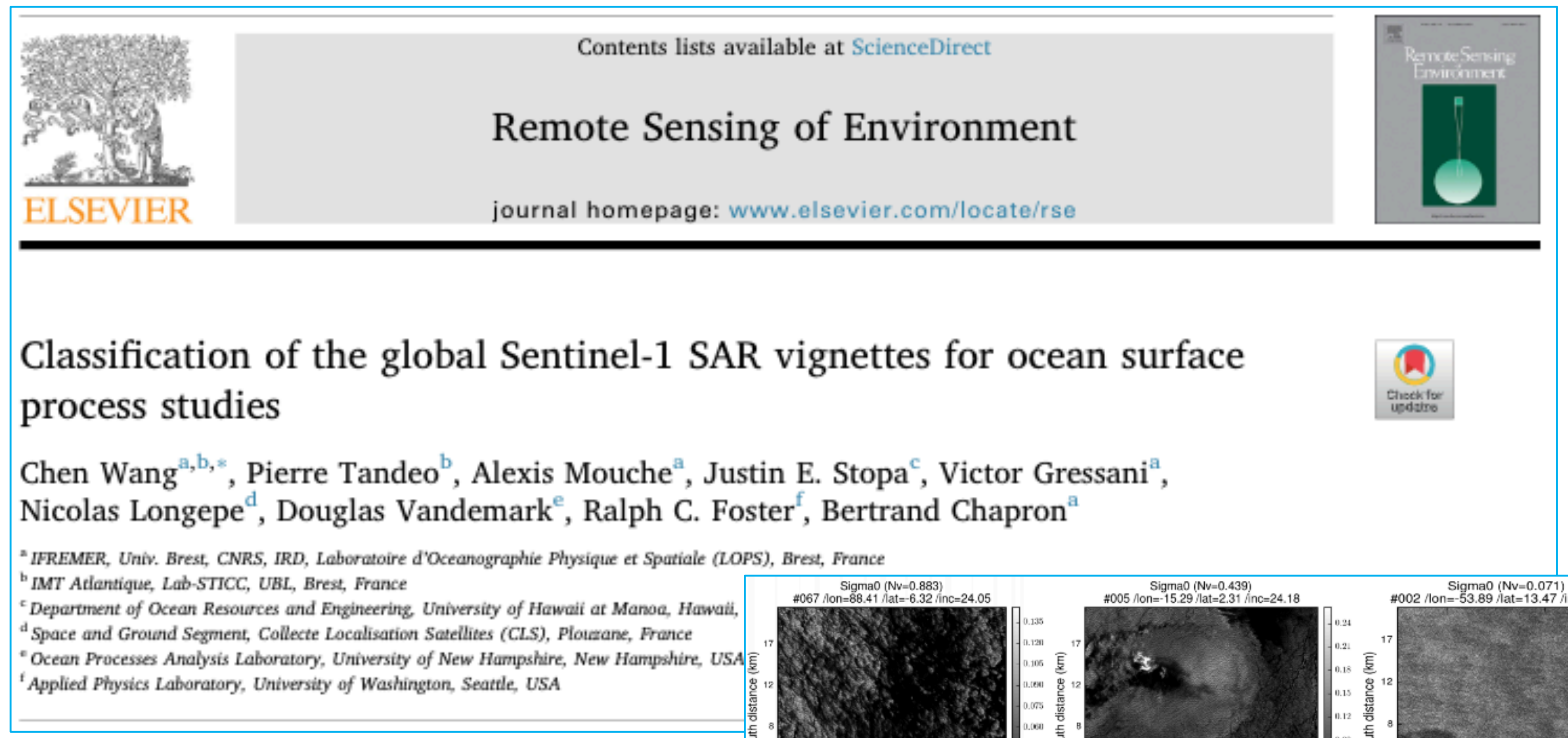
Chang-Jiang Zhang, Xiao-Ji

(a) (b) (c)


IEEE TRANSACTIONS ON GEOSCIENCE AND REMOTE SENSING 1

Vertical Structure-Based Classification of Oceanic Eddy Using 3-D Convolutional Neural Network

Baoxiang Huang


Contents lists available at ScienceDirect

Remote Sensing of Environment

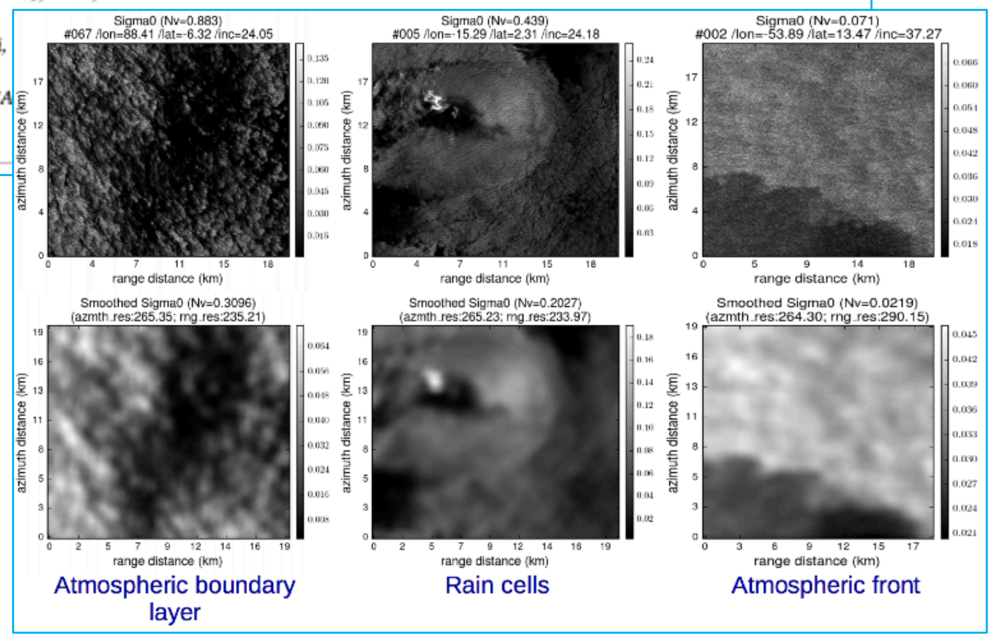
journal homepage: www.elsevier.com/locate/rse

ELSEVIER

Classification of the global Sentinel-1 SAR vignettes for ocean surface process studies

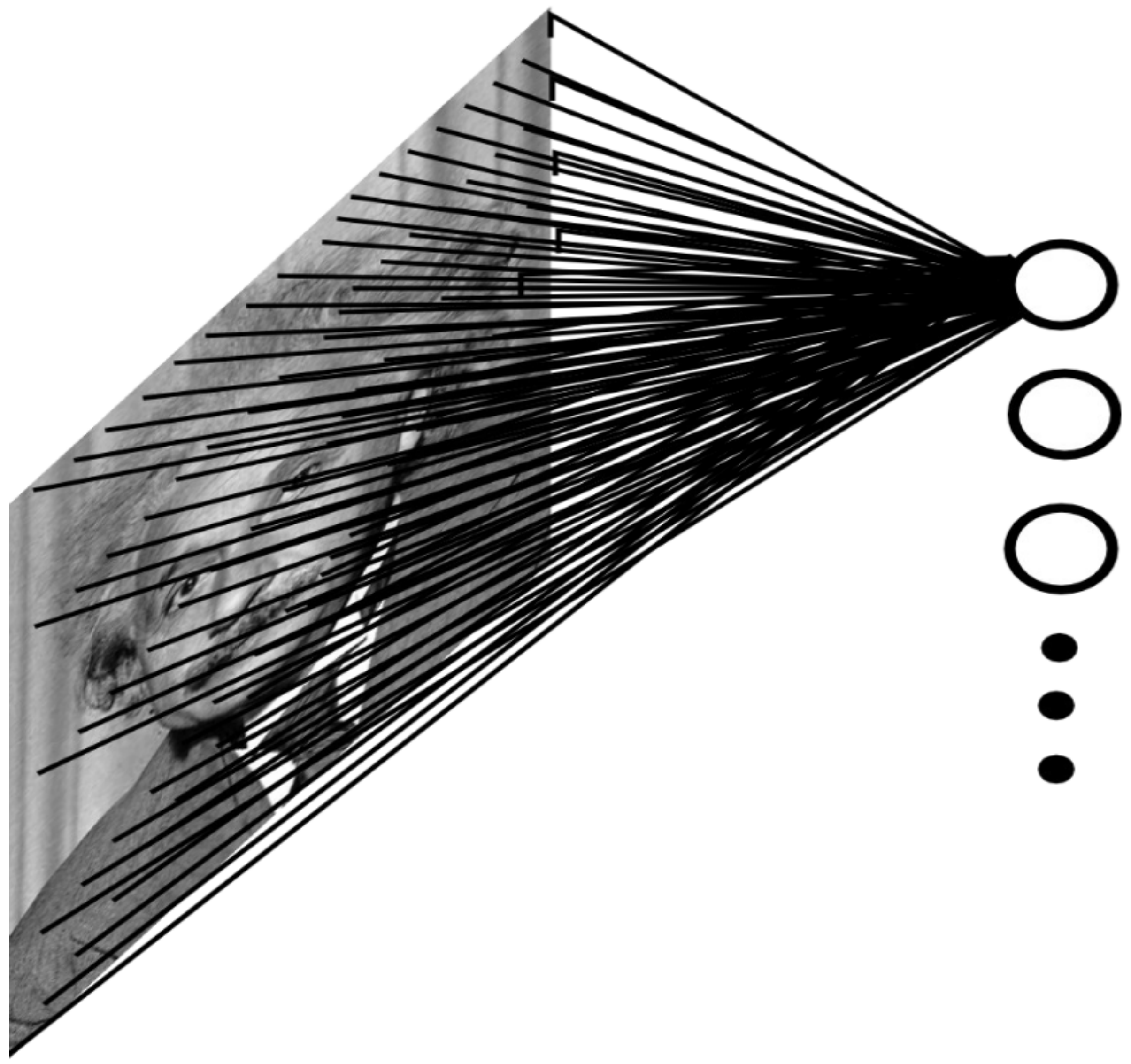
Chen Wang[a,b,*], Pierre Tandeo[b], Alexis Mouche[a], Justin E. Stopa[c], Victor Gressani[a], Nicolas Longepe[d], Douglas Vandemark[e], Ralph C. Foster[f], Bertrand Chapron[a]

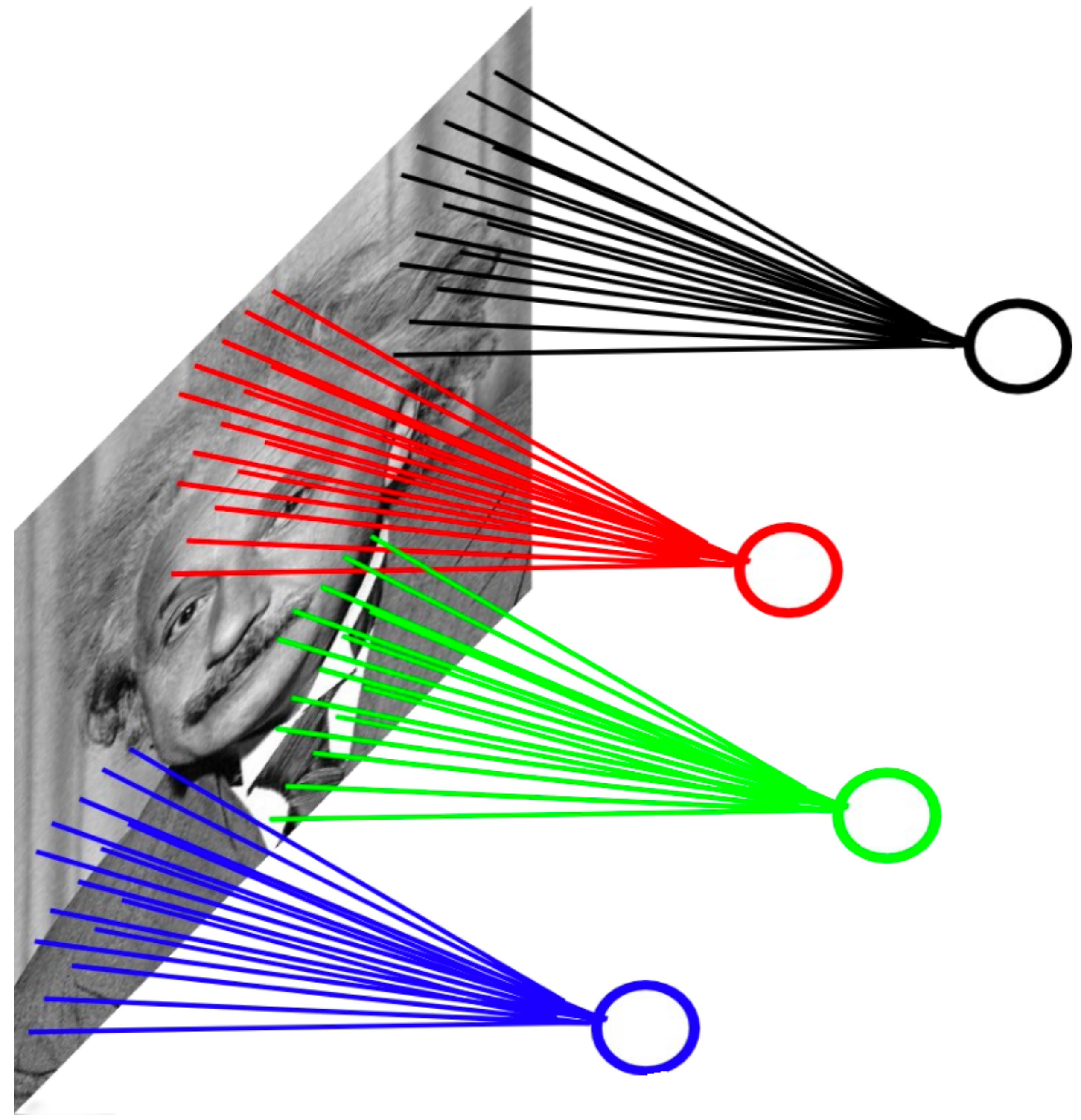
[a] IFREMER, Univ. Brest, CNRS, IRD, Laboratoire d'Oceanographie Physique et Spatiale (LOPS), Brest, France
[b] IMT Atlantique, Lab-STICC, UBL, Brest, France
[c] Department of Ocean Resources and Engineering, University of Hawaii at Manoa, Hawaii,
[d] Space and Ground Segment, Collecte Localisation Satellites (CLS), Plouzane, France
[e] Ocean Processes Analysis Laboratory, University of New Hampshire, New Hampshire, USA
[f] Applied Physics Laboratory, University of Washington, Seattle, USA



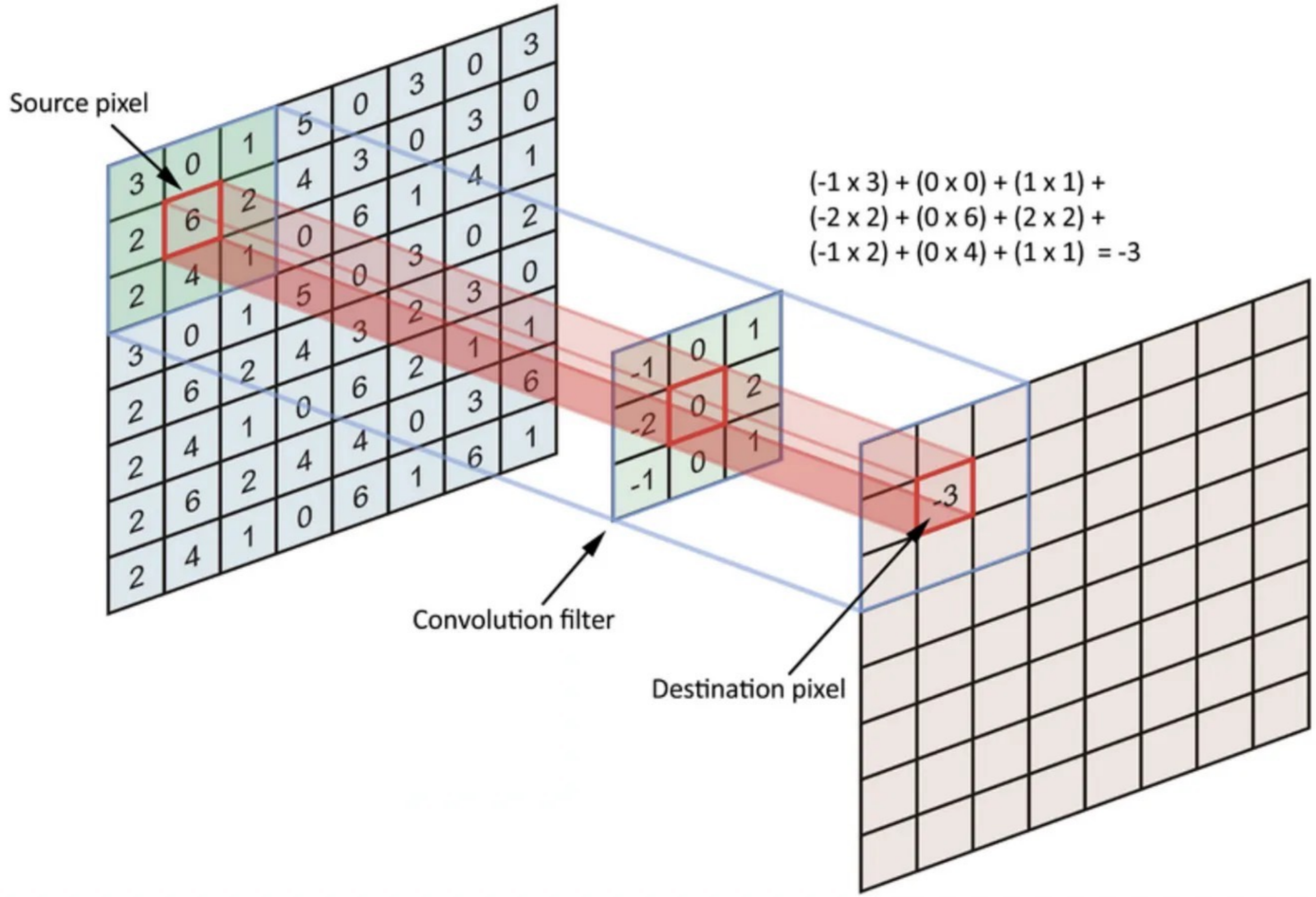
From Dense NN to CNN

- With images as inputs, a DNN architecture makes a direct connection between image pixels distant from each other.
- In many cases, it is unlikely that distant pixels carry relevant connections.
- The idea of limiting the connections between neighbouring pixels leads to the idea of CNN.



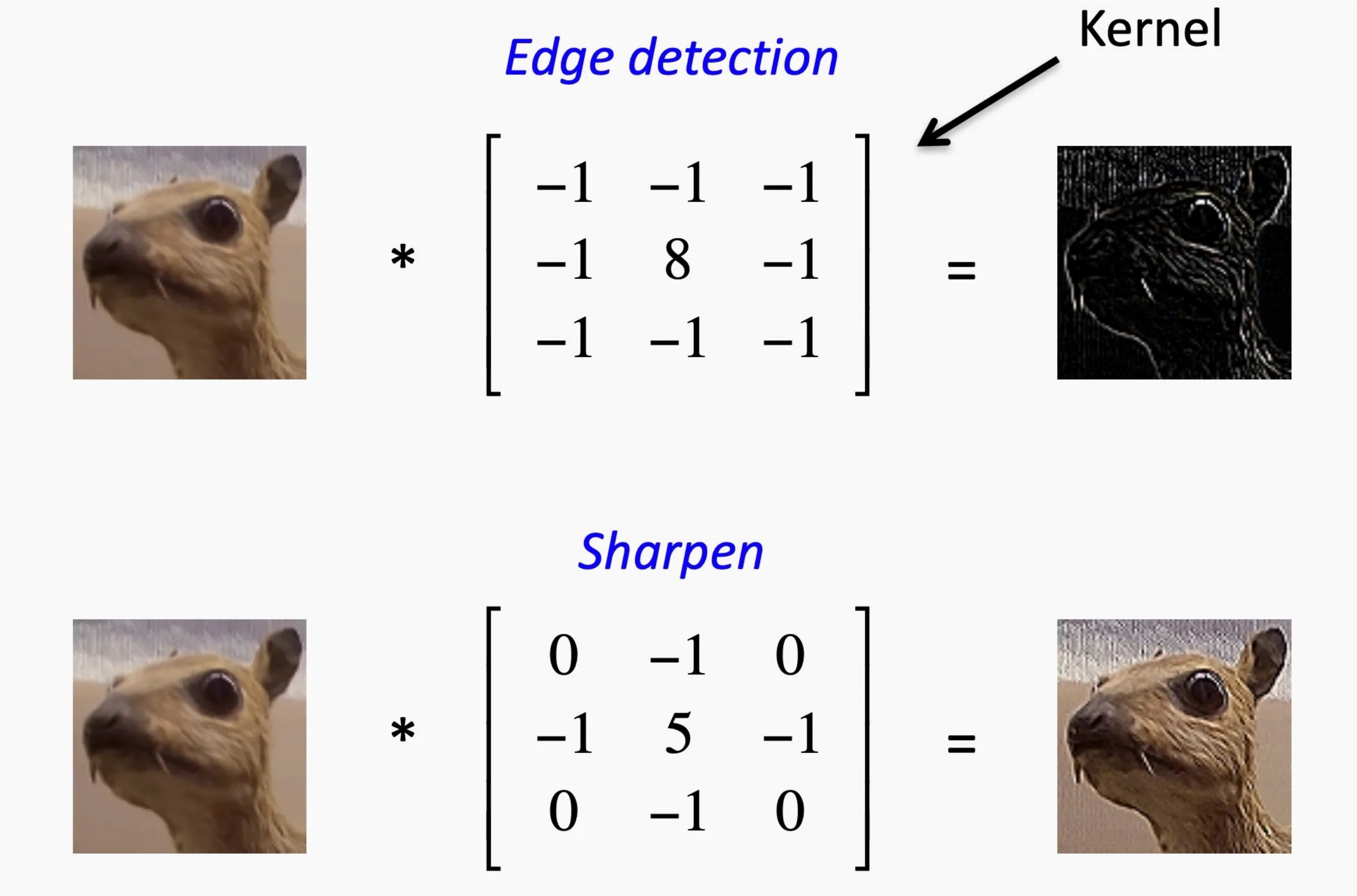


The core transformation: Convolution



https://towardsdatascience.com/simple-introduction-to-convolutional-neural-networks-cdf8d3077bac

Examples of standard convolution kernels in image processing



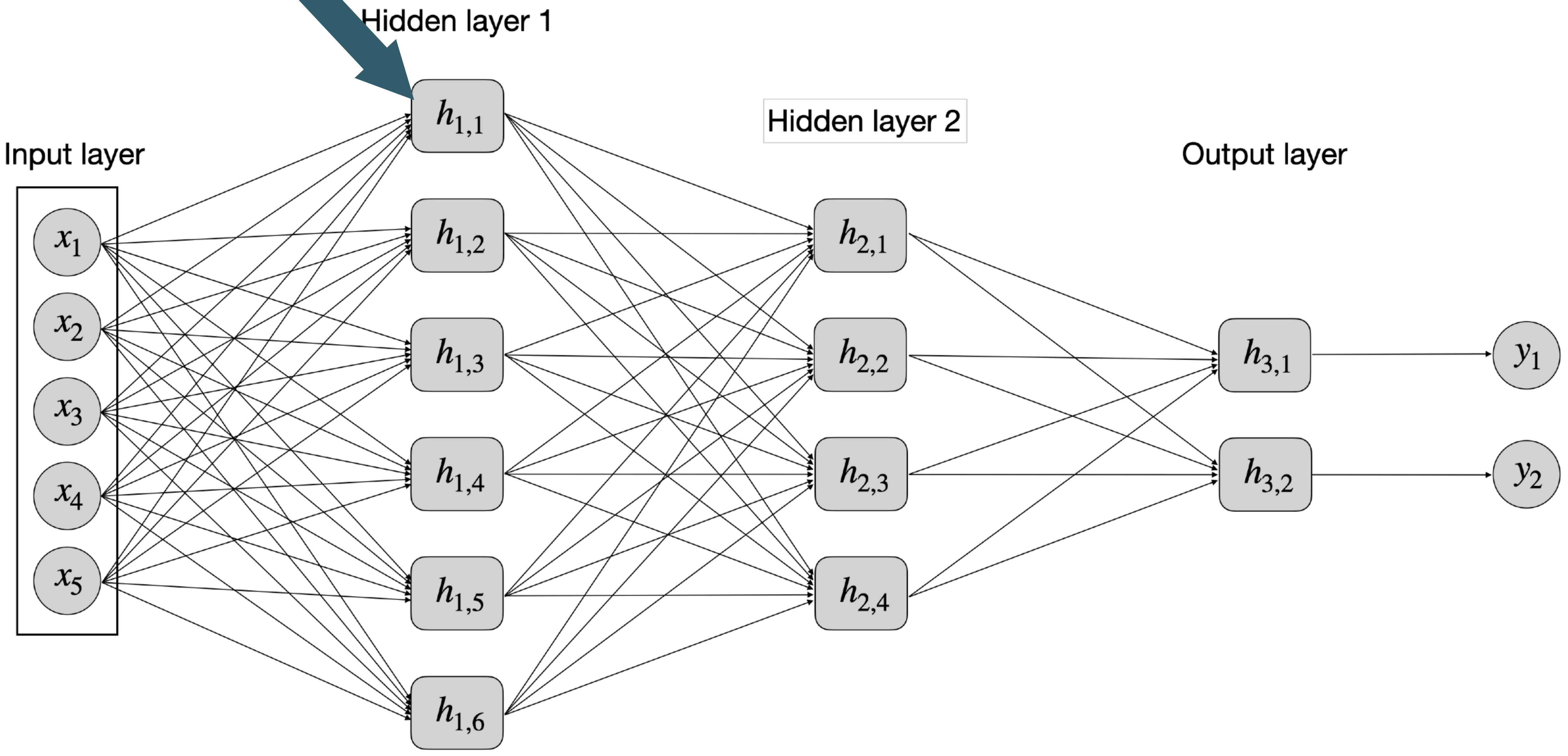
and a notebook in the course repo.

https://towardsdatascience.com/simple-introduction-to-convolutional-neural-networks-cdf8d3077bac

Convolutional neurons

Perceptron: 16385 parameters

Input: image with
$128 \times 128 = 16384$
pixels (values)

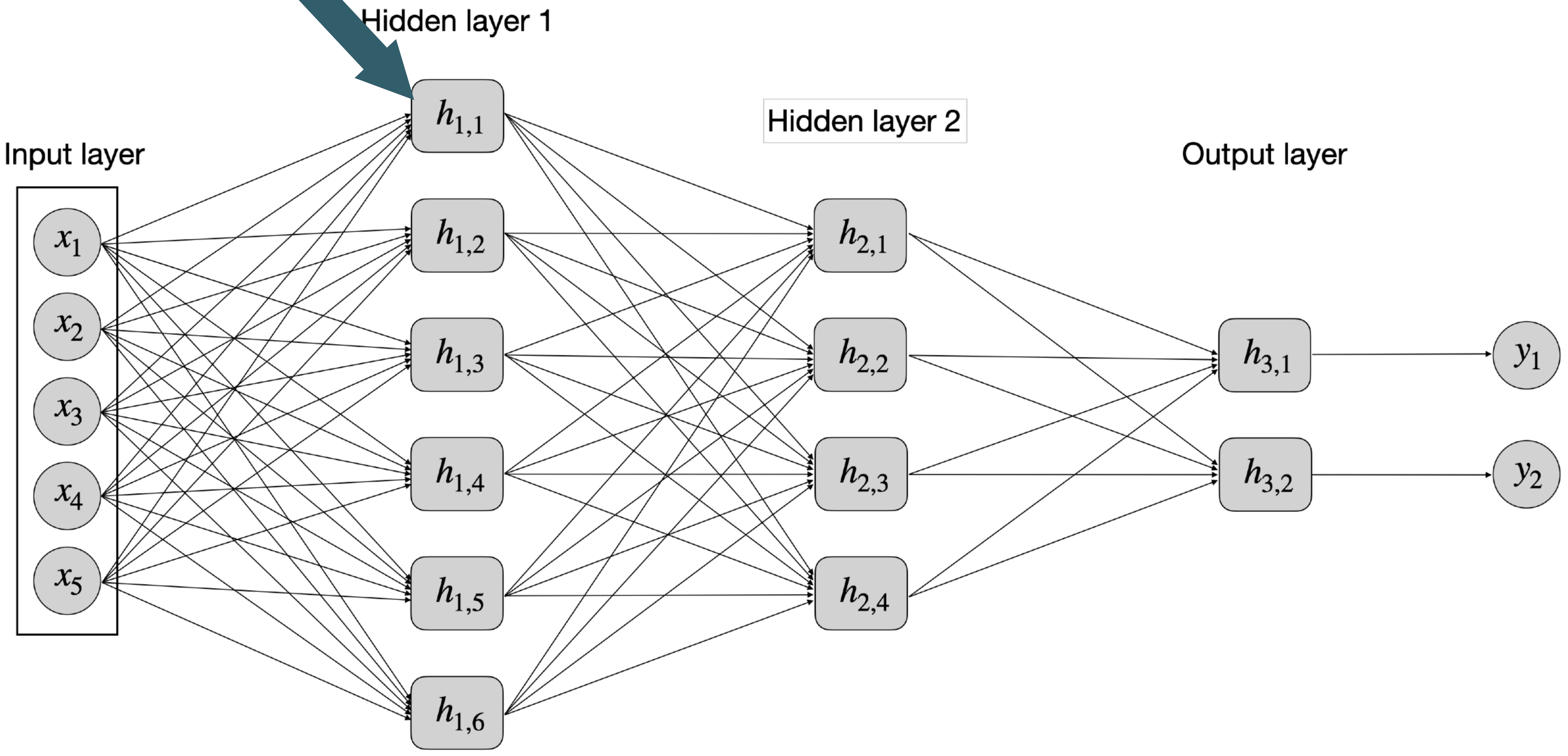


Convolutional neurons

~~Perceptron: 16385 parameters~~

Convolutional neuron: 10 parameters

Input: image with
$128 \times 128 = 16384$
pixels (values)



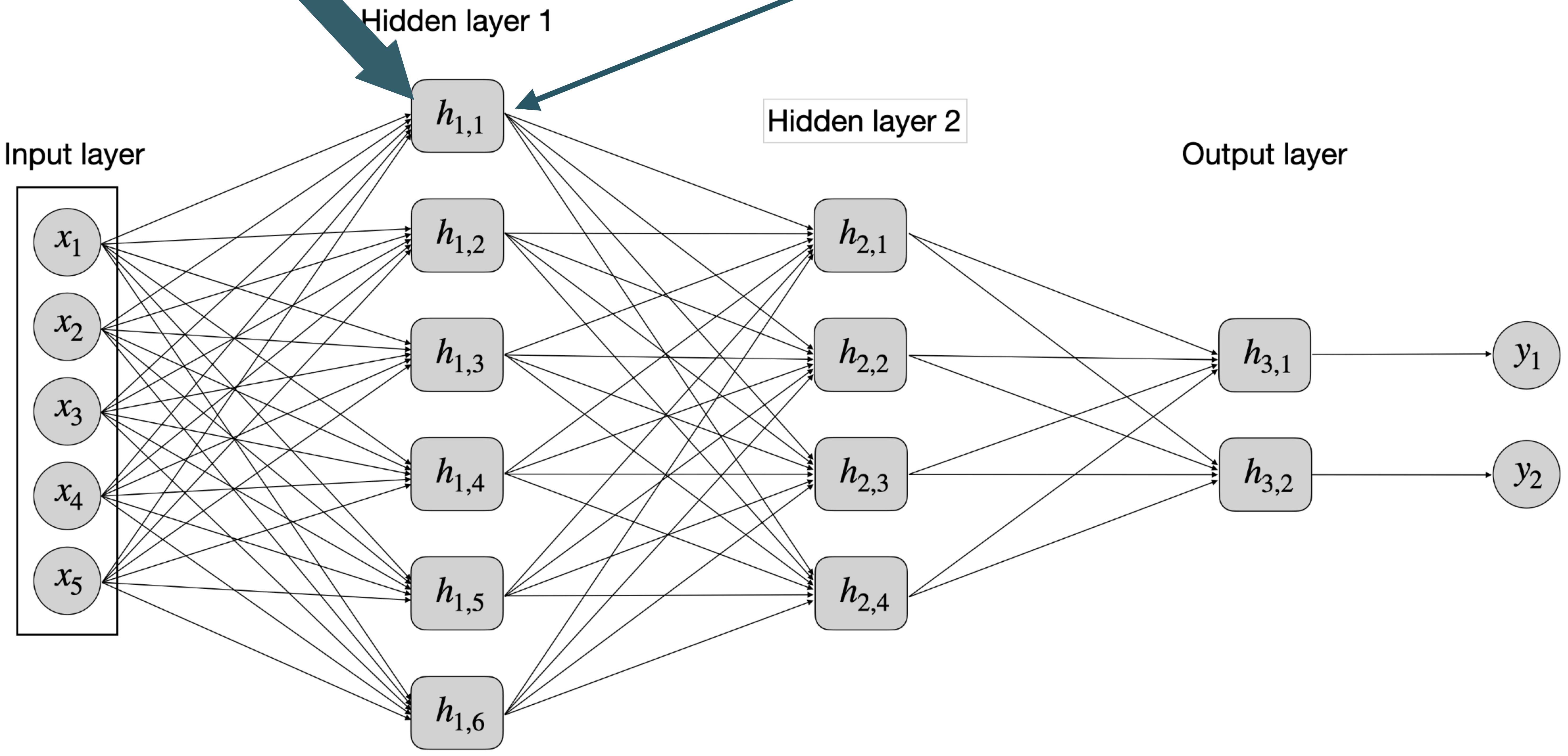
Convolutional neurons

~~Perceptron: 16385 parameters~~

Convolutional neuron: 10 parameters

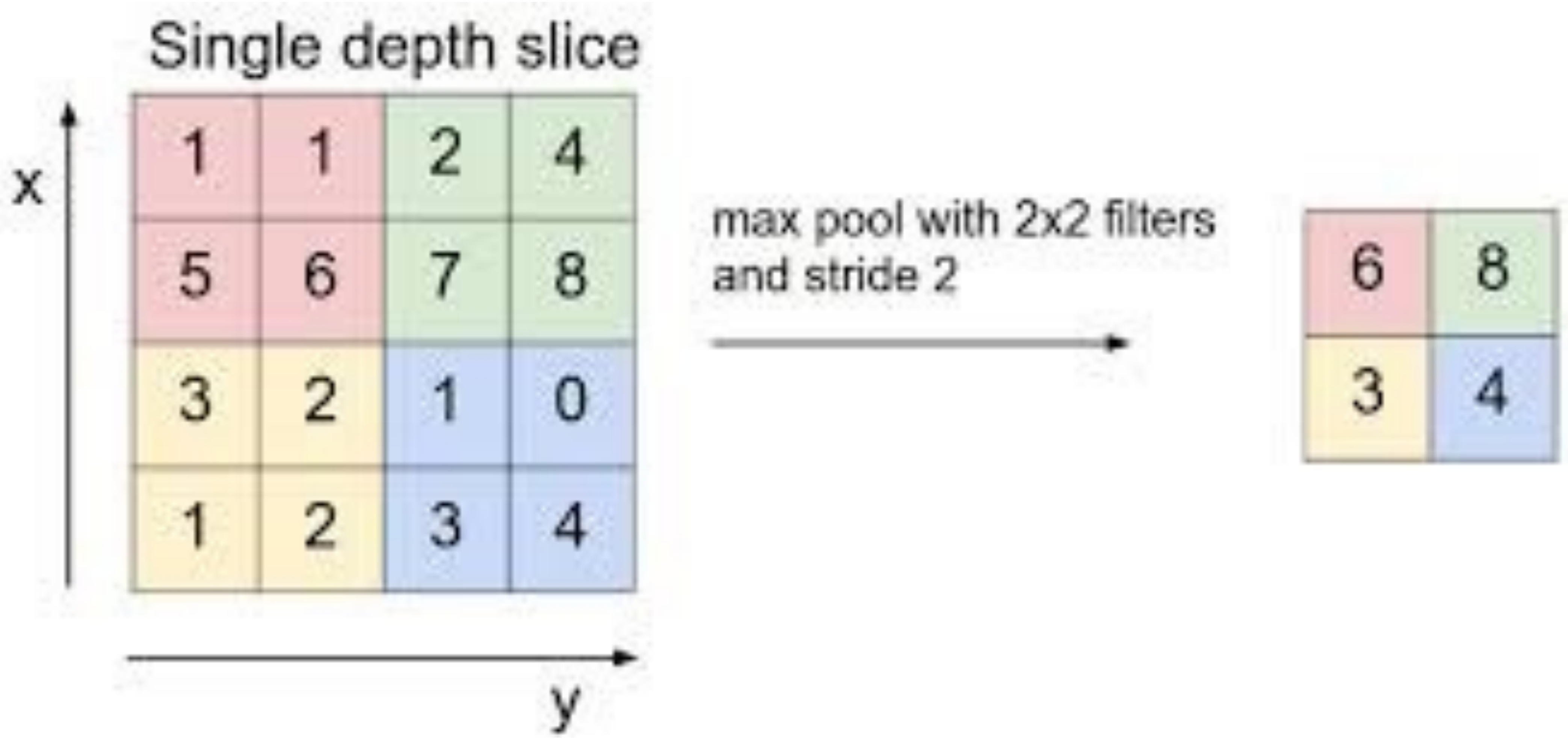
An activation function is still applied (ReLU, quite often)

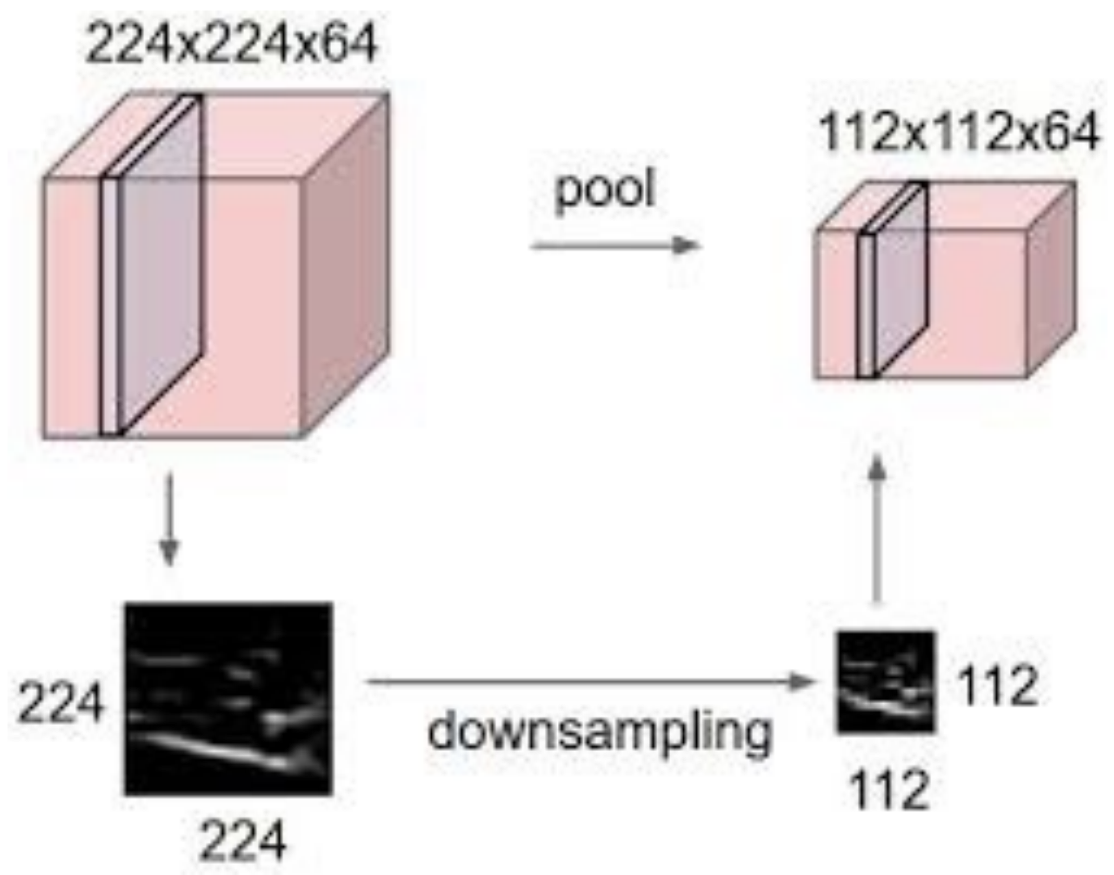
Input: image with $128 \times 128 = 16384$ pixels (values)



Pooling layers

- A convolution transforms a feature map (result of a convolution) but does not reduce its size (or poorly)
- Pooling layers are meant to reduce the feature maps size efficiently.
- The most commonly used is **max-pooling:**





Notion of stride

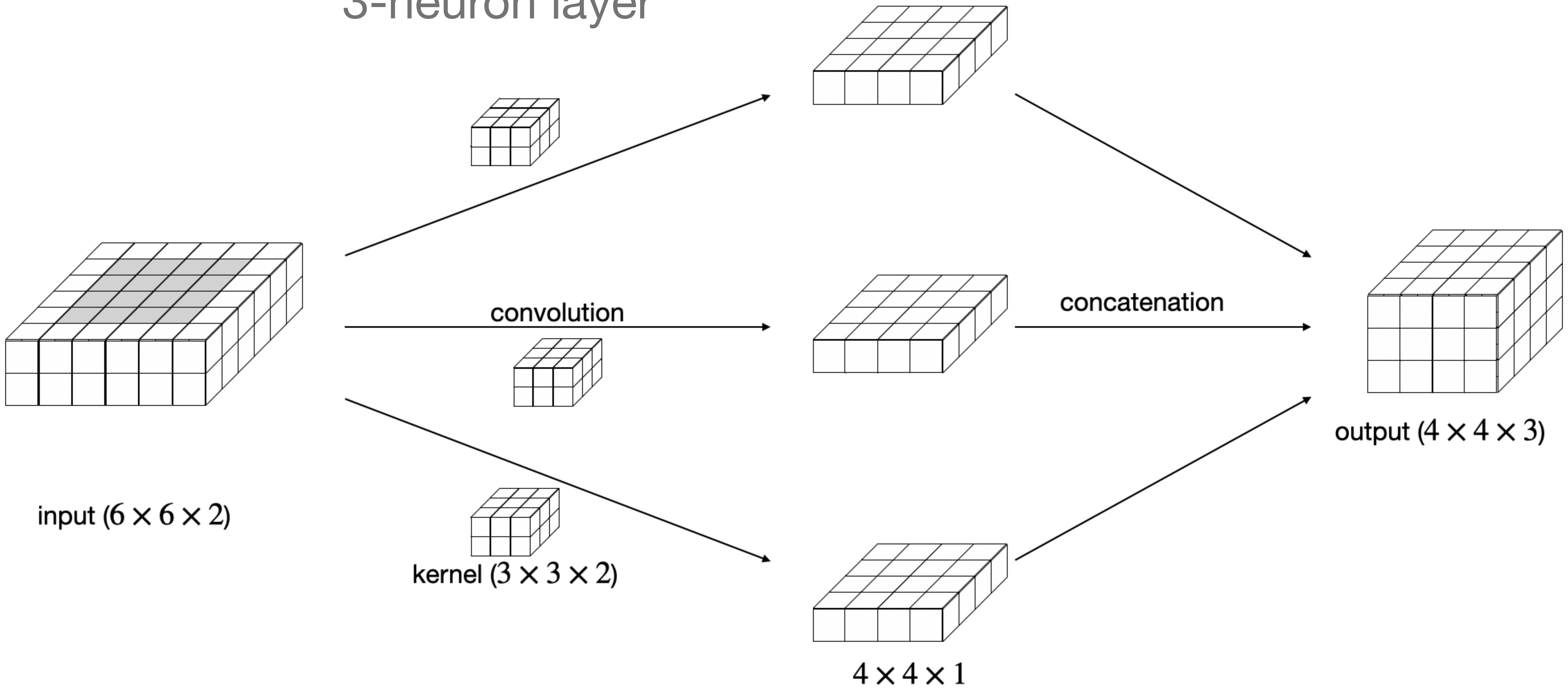
- A stride is a step with which a kernel is sliding along a feature map.
- It concerns convolution and max-pooling.
- In many applications, the stride for convolution is 1 (kernel size of 3x3 or 5x5), and 2 for max-pooling (kernel of 2x2)

Handling multiple channels

- Images can come with multiple channels (color images, multi-spectral data)

3-neuron layer

Image with 2 channels



Representation and anatomy of a CNN

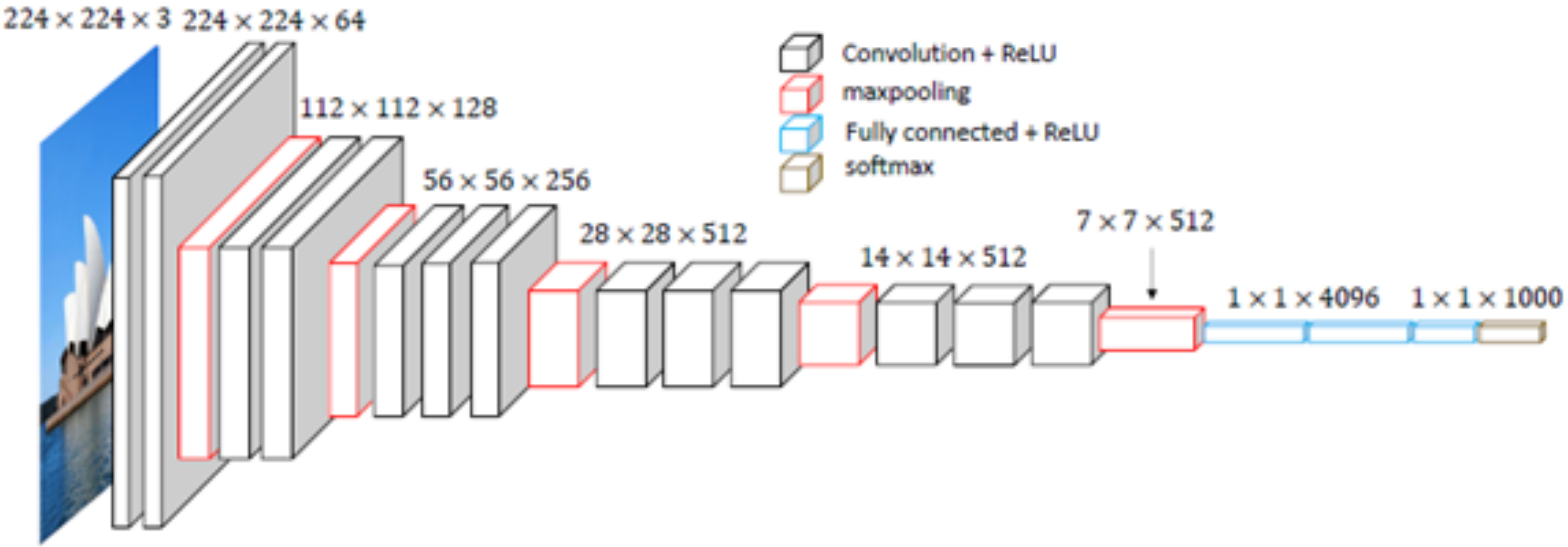


Figure 2: The architecture of VGG16 model .

Representation and anatomy of a CNN

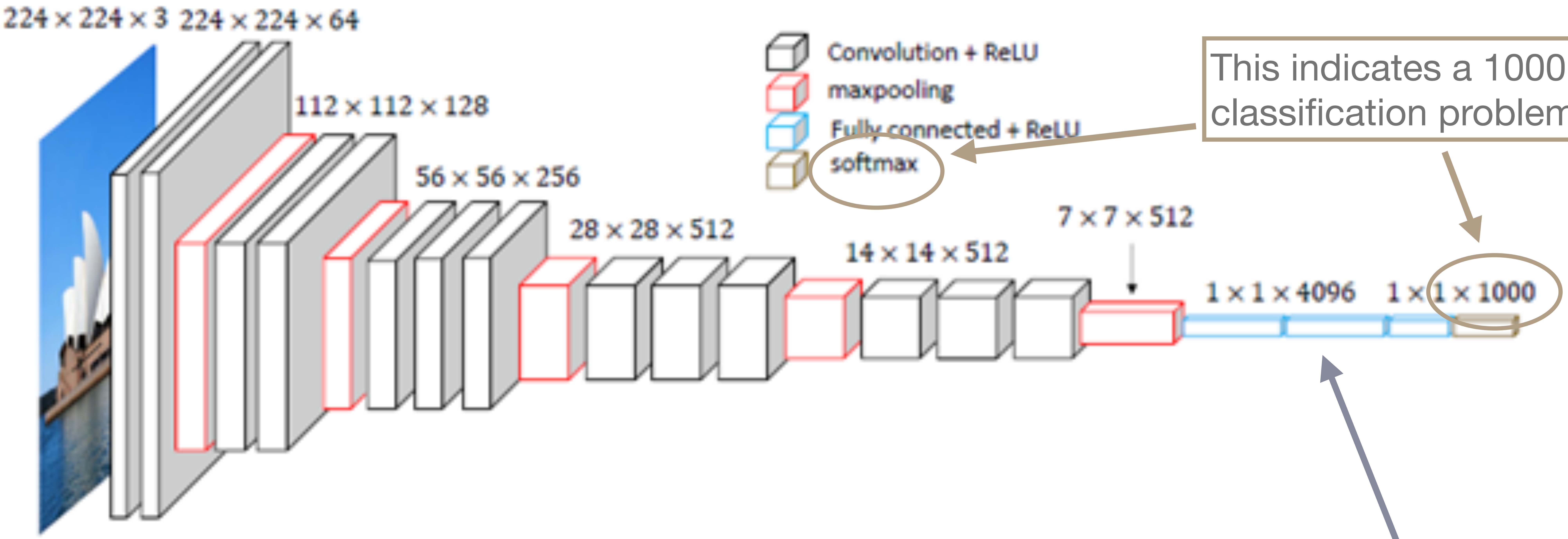


Figure 2: The architecture of VGG16 model .

This indicates a 1000-label classification problem.

A dense layer can be set up for appropriate problems (classification, essentially)

Representation and anatomy of a CNN

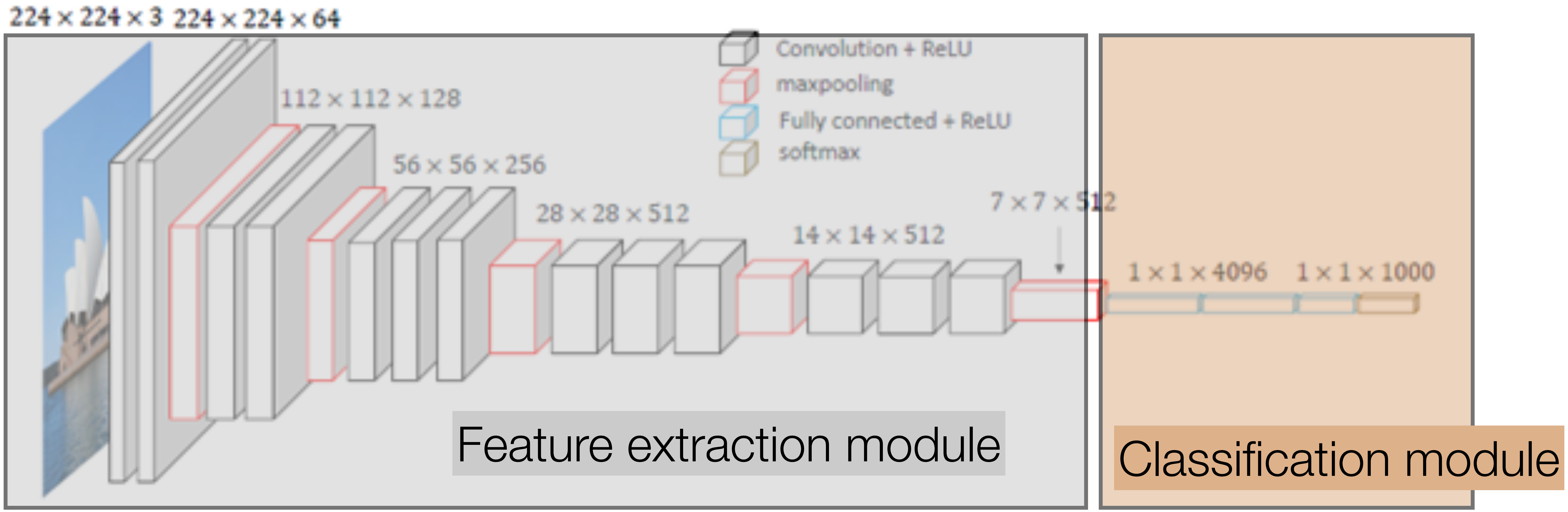


Figure 2: The architecture of VGG16 model .

General idea: the first layers involve generic feature extraction step and the last block can be regarded as a dataset-specific classification block.

Representation and anatomy of a CNN

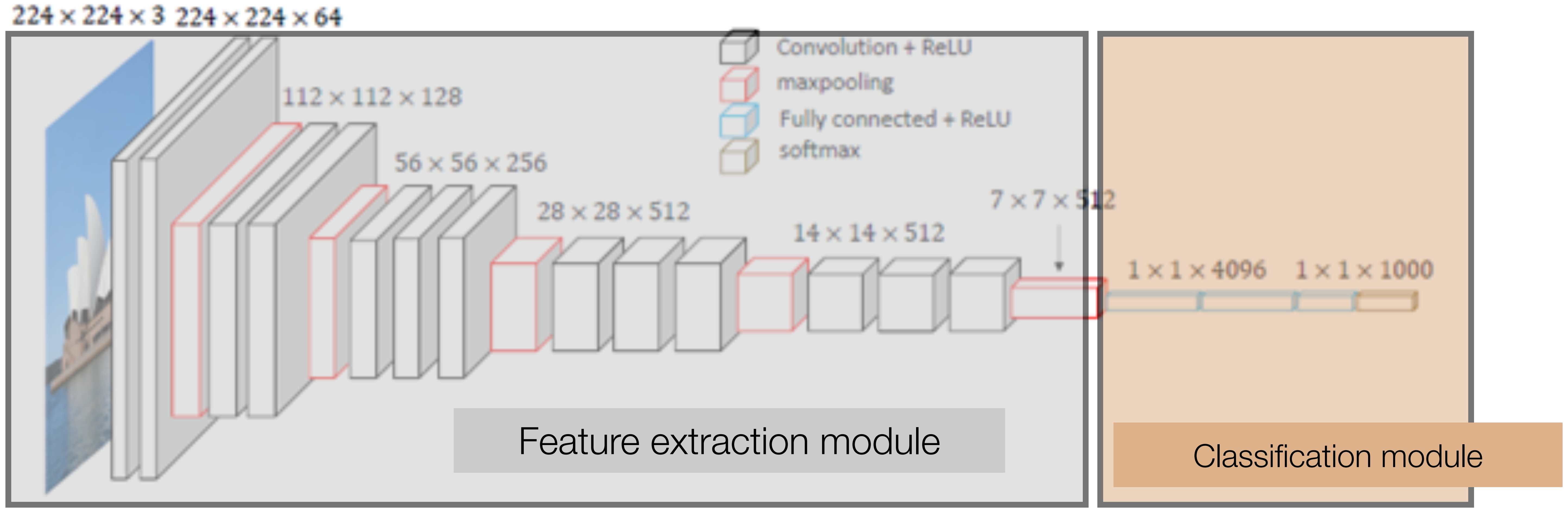


Figure 2: The architecture of VGG16 model .

This motivates the use (fine-tuning) of pre-trained models. "Transfer learning" consists of share a part of the learning process between different models.

Hands on!

- A hands-on explanation of convolution: https://github.com/CIA-Oceanix/DLOA2023/blob/main/lectures/notebooks/2022-09-09-convolutions.ipynb
- https://github.com/CIA-Oceanix/DLOA2023/blob/main/lectures/notebooks/notebook_MNIST_classification_MLP_CNN_TransferLearning_students.ipynb