Computer vision and machine learning for the material scientist (CVML) 2021
Lecture 9

Convolutional Neural Networks (CNNs)
for semantic segmentation

26/02/2021

João P C Bertoldo
joao.bertoldo@mines-paristech.fr

BIG MECA

CENTRE DES MATERIAUX PIERRE-MARIE FOURT

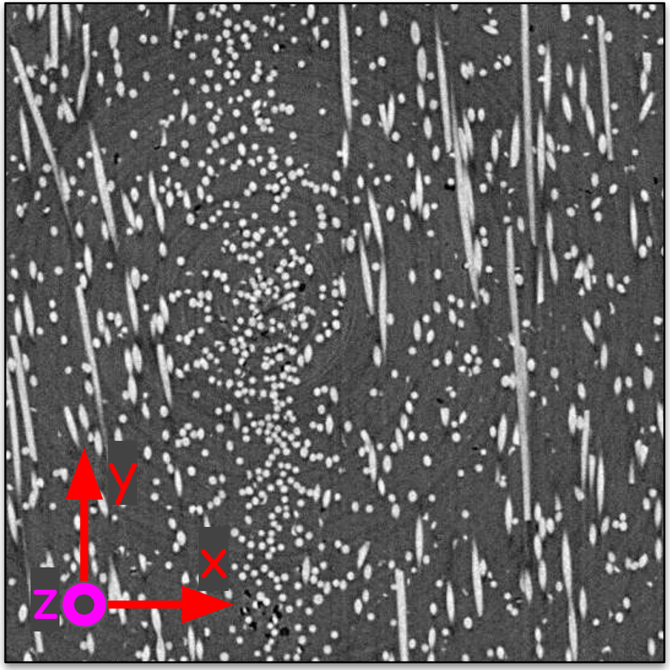
MINES ParisTech

PSL

teaser

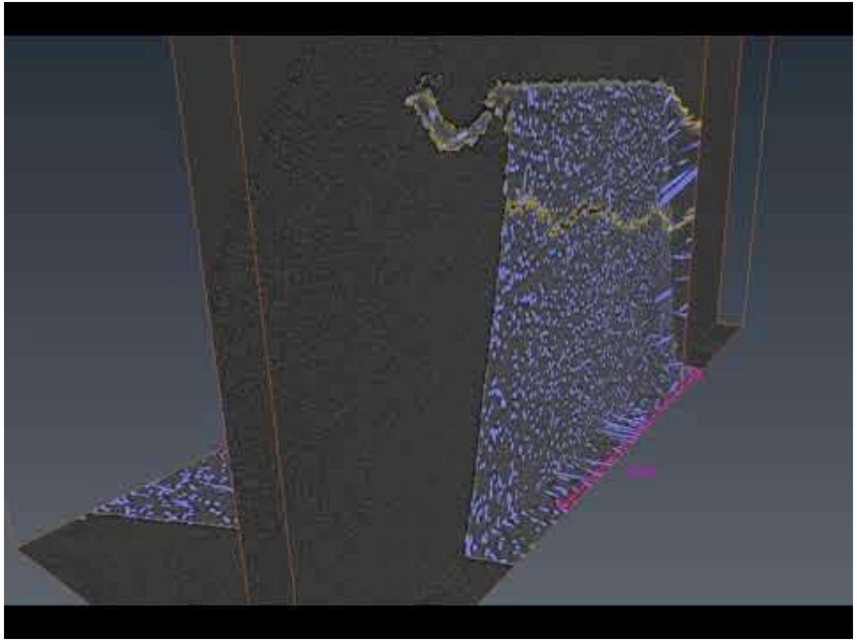
what we have

a stack of gray images



what we want

the 3d structure



blue: fibers

yellow: crack

link to the video

3D tomography of a glass fiber-reinforced composite. Left: raw data. Right: raw data with superposed segmentation.

this presentation

goals

- put together pieces learned during the week
- showcase an application example

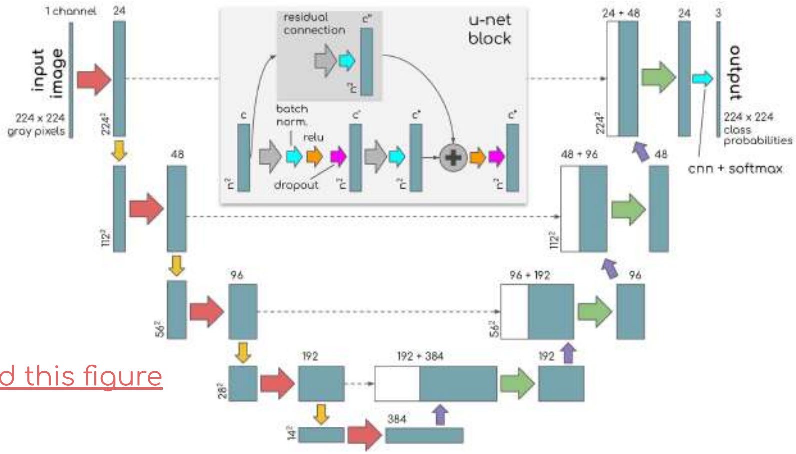
focus

- a neural network architecture and its components

context

- the (machine) learning problem
- data provenance

Monday	Tuesday	Wednesday	Thursday	Friday
Introduction to computer vision (HP)	Machine learning 2 (HP)	Meta model 2 (PK)	Deep learning (HP)	Yolo : real time object detection (BF)
Tutorial classification k-NN (HP, AM)	Tutorial machine learning 2 (HP, AM)	Tutorial meta model 2 (PK, AM)	Tutorial deep learning (HP, AM)	Tutorial Yolo (BF)
Machine learning 1 (HP)	Meta Model 1 (PK)	Introduction to neural networks (HP)	Convolutional neural nets (HP)	CNN for Semantic segmentation (JCB)
Tutorial machine learning 1 (HP, AM)	Tutorial meta model 1 (PK, AM)	Tutorial neural networks (HP, AM)	Tutorial CNN (HP, AM)	Written exam



main goal: understand this figure

intro

myself

João Paulo Casagrande Bertoldo

yes, it's a pretty long name... (:

Sep. 2020 - now [Bigméca]

Research x (intern + engineer)

2017 - 2021 MINES ParisTech - PSL University

Executive engineering - Data Science Minor

2019 - 2020 Paris-Dauphine - PSL University

Master IASD: Artificial Intelligence, Systems, Data

2013 - 2020 University of São Paulo (USP)

Mechatronics engineering

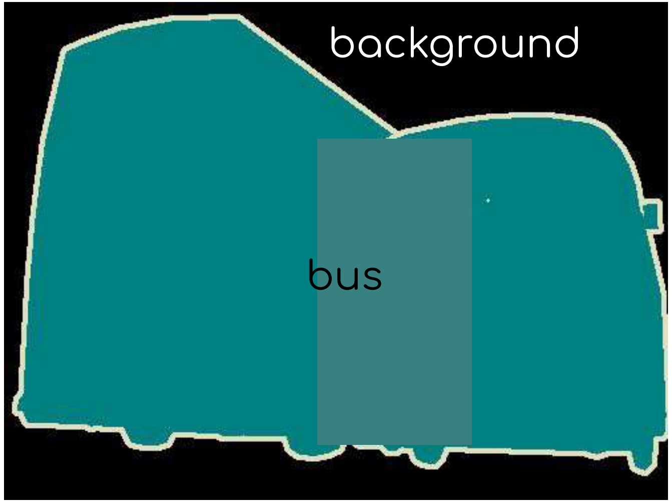
contact

joao.bertoldo@mines-paristech.fr

joaopcbertoldo.github.io

semantic segmentation

$Y = F(X)$



Def.: an (normalized) image X is a 2D array of size p x q, and each of its elements $X_{i,j}$ belongs to **[0, 1]**.

1 pixel = a gray level intensity

"how much light did the sensor capture in that grid cell?"

Def.: a binary segmentation Y is a 2D array of size p x q, and each of its elements $Y_{i,j}$ is 0 or 1.

1 pixel = a "membership"

"does this pixel belong to the object or to the background?"

Image source: http://host.robots.ox.ac.uk/pascal/VOC/

Everingham, M., Van Gool, L., Williams, C.K.I., Winn, J., Zisserman, A., 2010. The Pascal Visual Object Classes (VOC) Challenge. Int J Comput Vis 88, 303–338. https://doi.org/10.1007/s11263-009-0275-4

other image-to-image problems

we are **not** talking about this today

instance segmentation

Everingham, M., Van Gool, L., Williams, C.K.I., Winn, J., Zisserman, A., 2010. The Pascal Visual Object Classes (VOC) Challenge. Int J Comput Vis 88, 303–338.

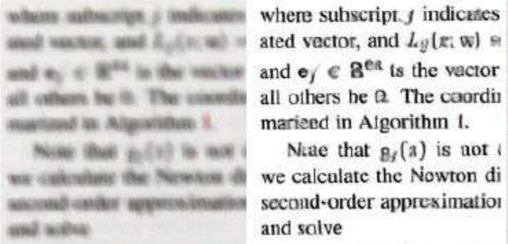


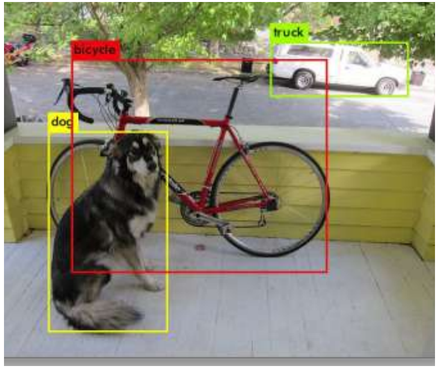
image deblurring

Everingham, M., Van Gool, L., Williams, C.K.I., Winn, J., Zisserman, A., 2010. The Pascal Visual Object Classes (VOC) Challenge. Int J Comput Vis 88, 303–338.

super resolution

Dong, C., Loy, C.C., Tang, X., 2016. Accelerating the Super-Resolution Convolutional Neural Network.

Yang, W., Zhang, X., Tian, Y., Wang, W., Xue, J.-H., 2019. Deep Learning for Single Image Super-Resolution: A Brief Review. IEEE Trans. Multimedia 21, 3106–3121.



object detection

Image source: https://pjreddie.com/darknet/yolo/

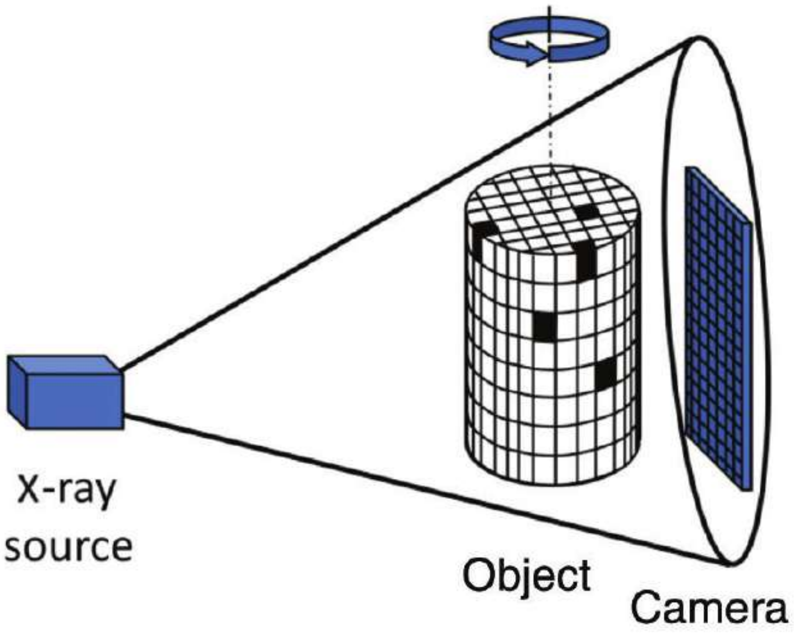
Redmon, J., Divvala, S., Girshick, R., Farhadi, A., 2016. You Only Look Once: Unified, Real-Time Object Detection..

style transfer

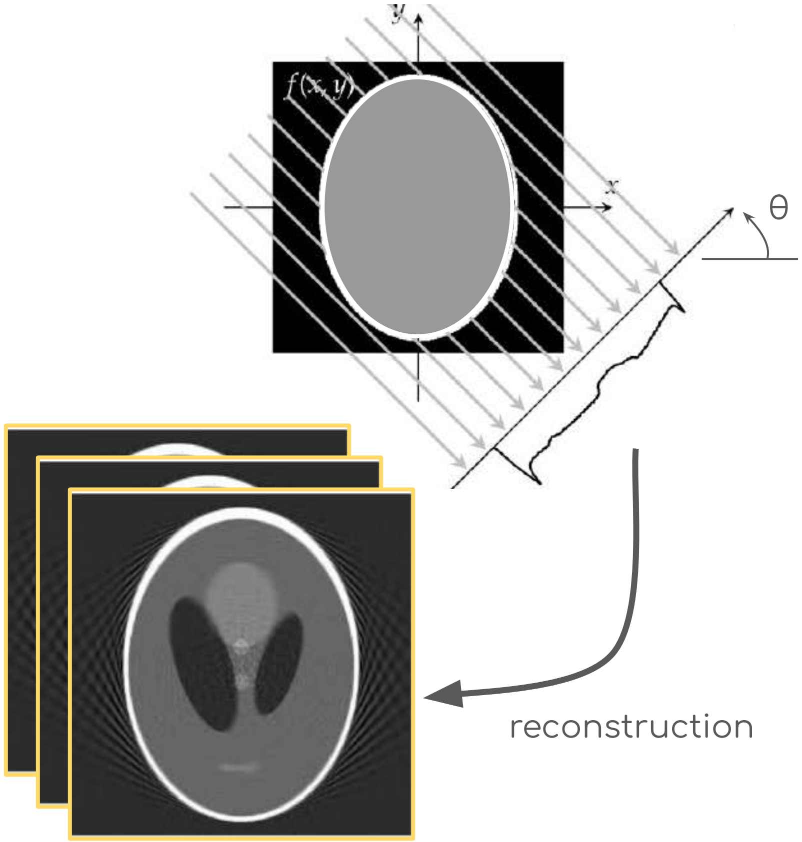
Li, Y., Liu, M.-Y., Li, X., Yang, M.-H., Kautz, J., 2018. A Closed-form Solution to Photorealistic Image Stylization.

materials science
meets
semantic segmentation

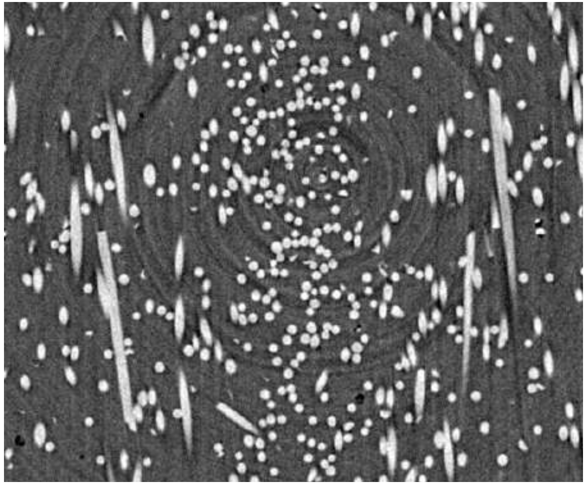
tomography



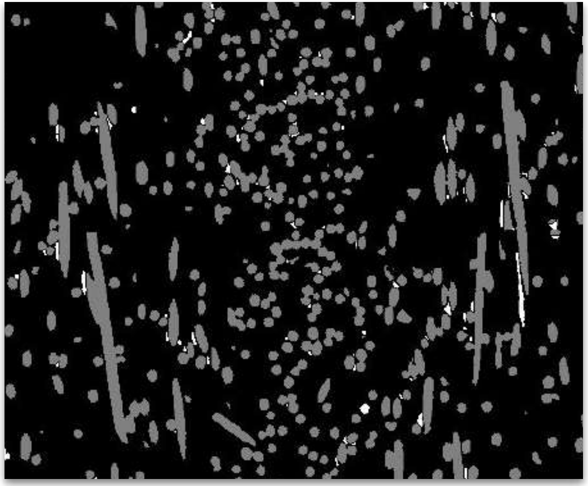
Source: Terzano, Roberto & Denecke, Melissa & Falkenberg, Gerald & Miller, Bradley & Paterson, David & Janssens, Koen. (2019). Recent advances in analysis of trace elements in environmental samples by X-ray based techniques (IUPAC Technical Report). Pure and Applied Chemistry. 91. 10.1515/pac-2018-0605.



phase segmentation



$Y = F(X)$



Def.: an (normalized) image X is a 2D array, and each of its elements $X_{i,j}$ belongs to **[0, 1]**.

1 pixel = a gray level intensity

"how much light did the sensor capture in that grid cell?"

Def.: a segmentation Y is a 2D array, and each of its elements $Y_{i,j}$ has a value in {1, 2, ..., C}, where C is the nb. of **classes**.

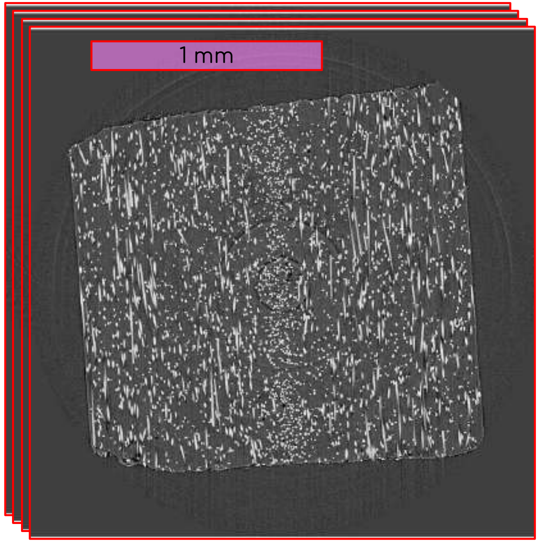
1 pixel = a category = a phase

"to which class (phase) does this area belong to?"

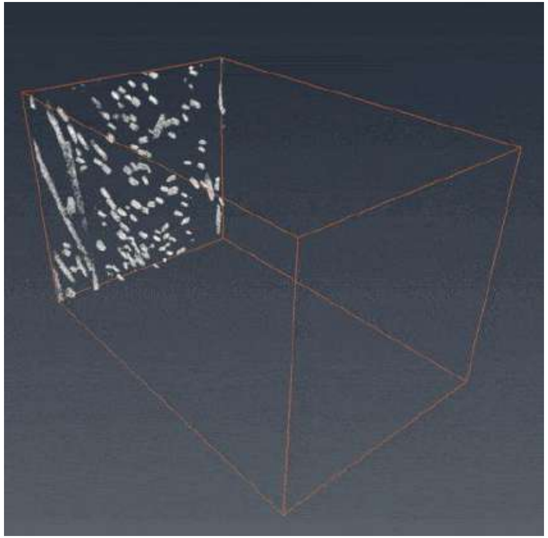
material

we wish to locate separate phases pixel by pixel

- 3 phases
  - PolyAmide 66 matrix
  - Glass fiber reinforcement
  - Porosities (holes) in the matrix
- specimen size: 2mm x 2mm x 6mm
- voxel size: 1.3 μm^3



2048 images of 2048 x 2048



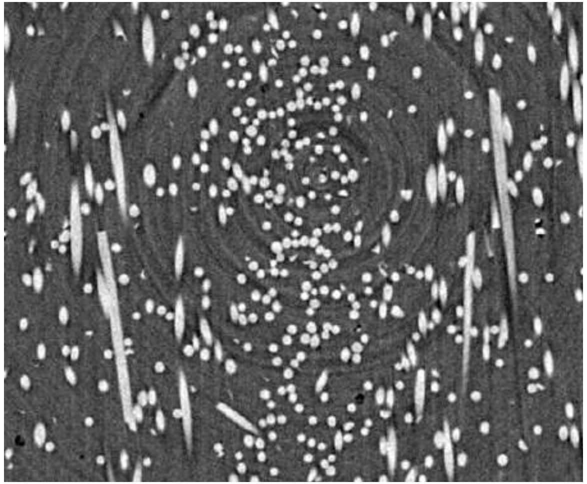
glass fibers (white)

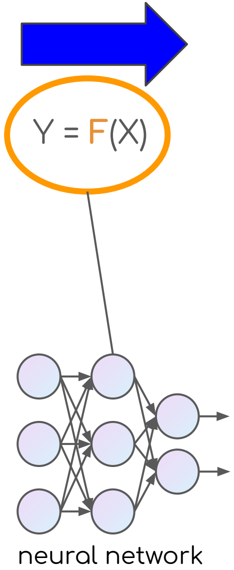
and porosities (red)

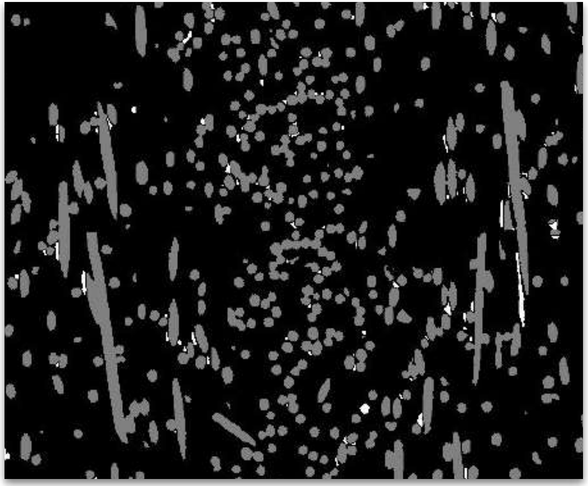
link to the video

semantic segmentation
meets
deep learning

phase segmentation with deep learning



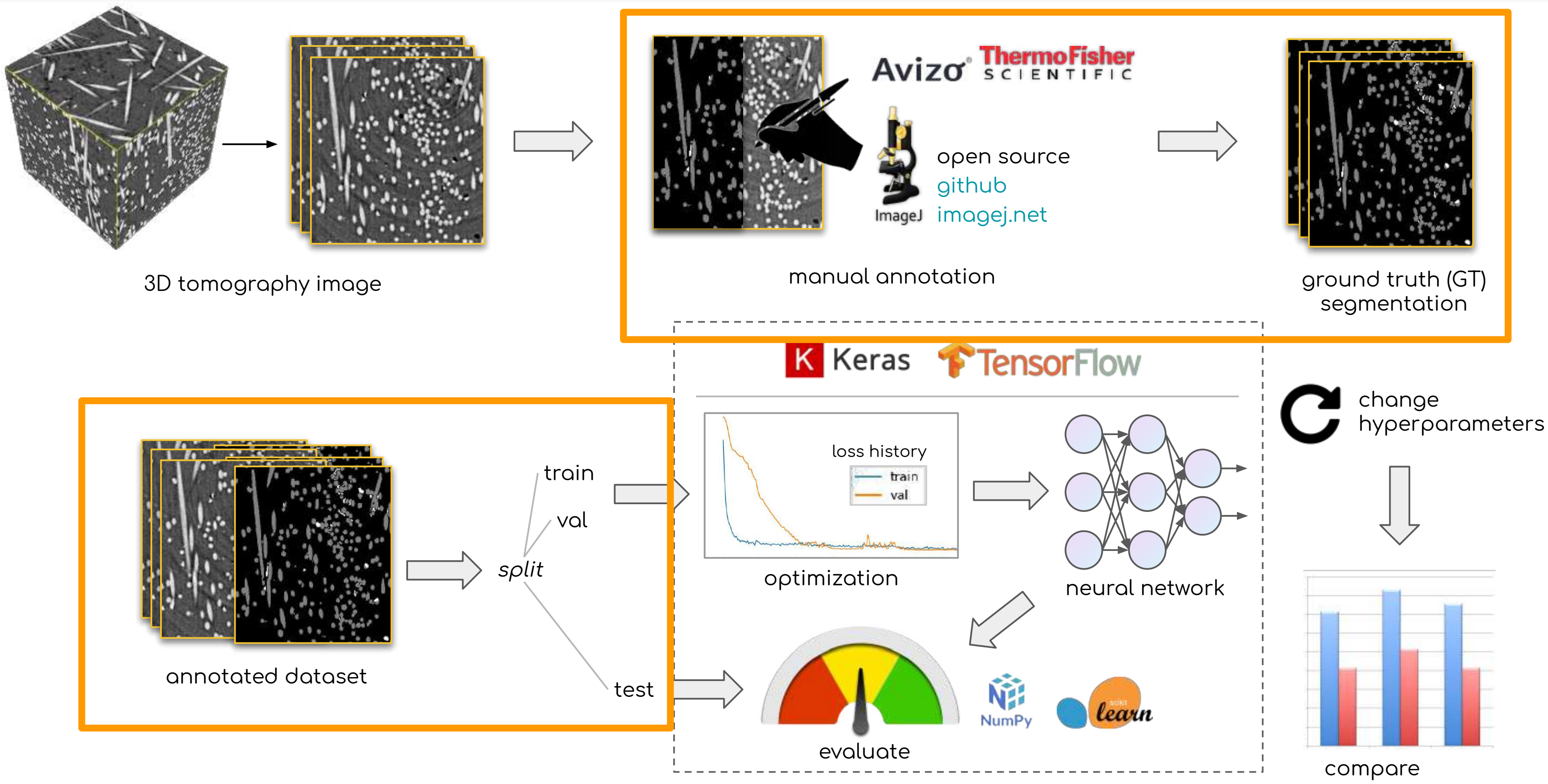




overview

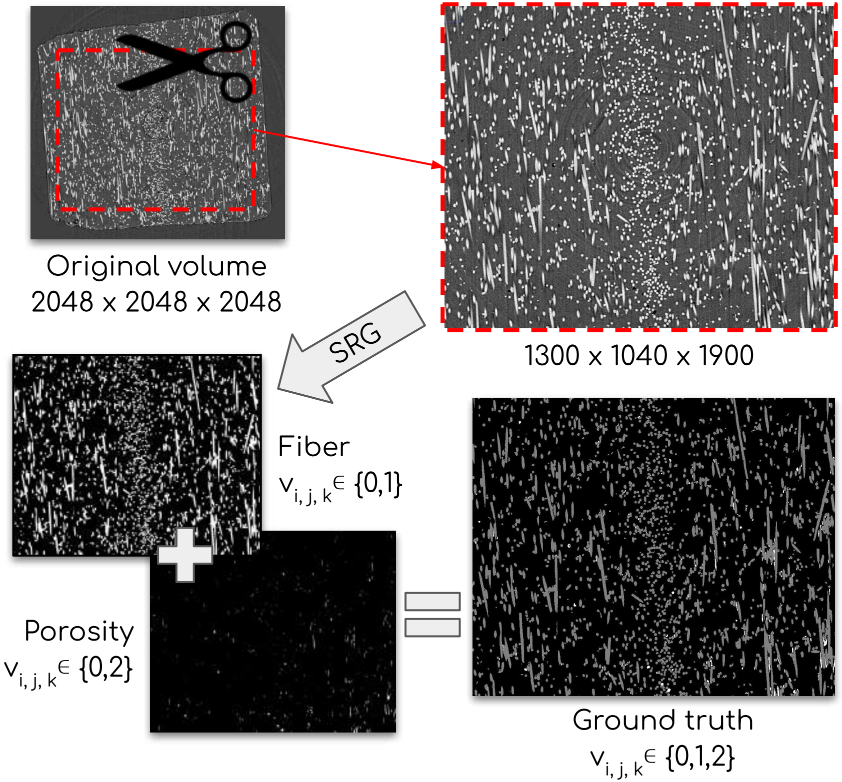
overview

project repository: github.com/joaopcbertoldo/tomo2seg

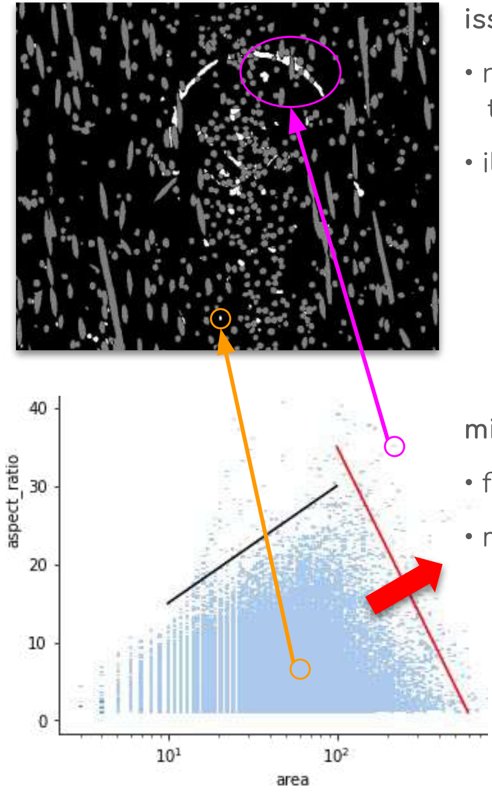


data annotation

phase 1
double Seeded Region Growing (fiji)



phase 2
artifacts correction (avizo)



issue

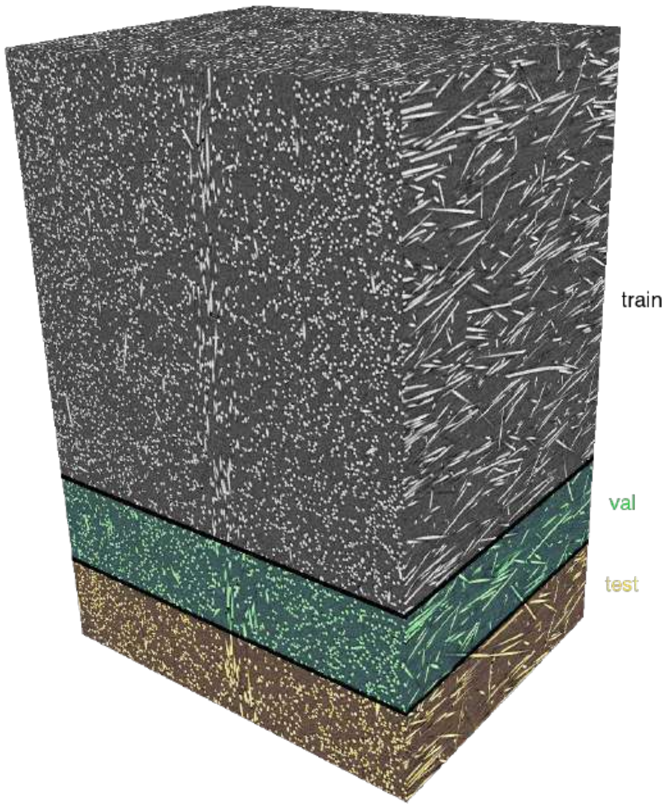
- ring artefacts from the tomography reconstruction
- ill-defined porosity boundary

mitigation strategy

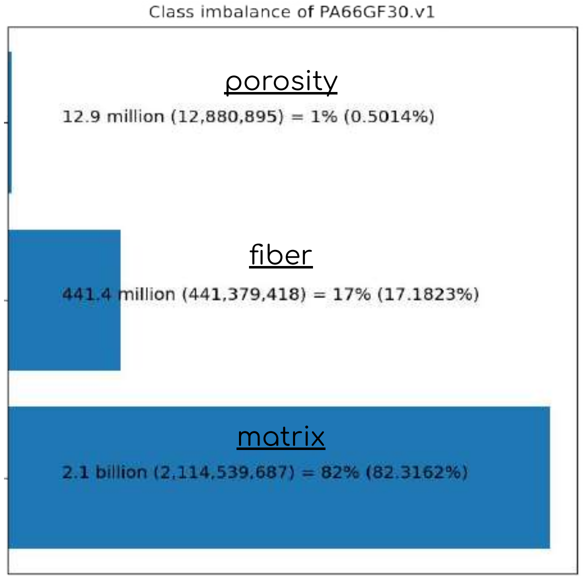
- find large, long 2d blobs*
- manually alter the voxels

** "blob": connected set*

data * (split + imbalance + augmentation)



set	nb. layers (1300x1040)	proportion
train	1300	68%
val	128	9%
test	300	16%



train: gradient descent

val: in-the-loop model selection

test: evaluation

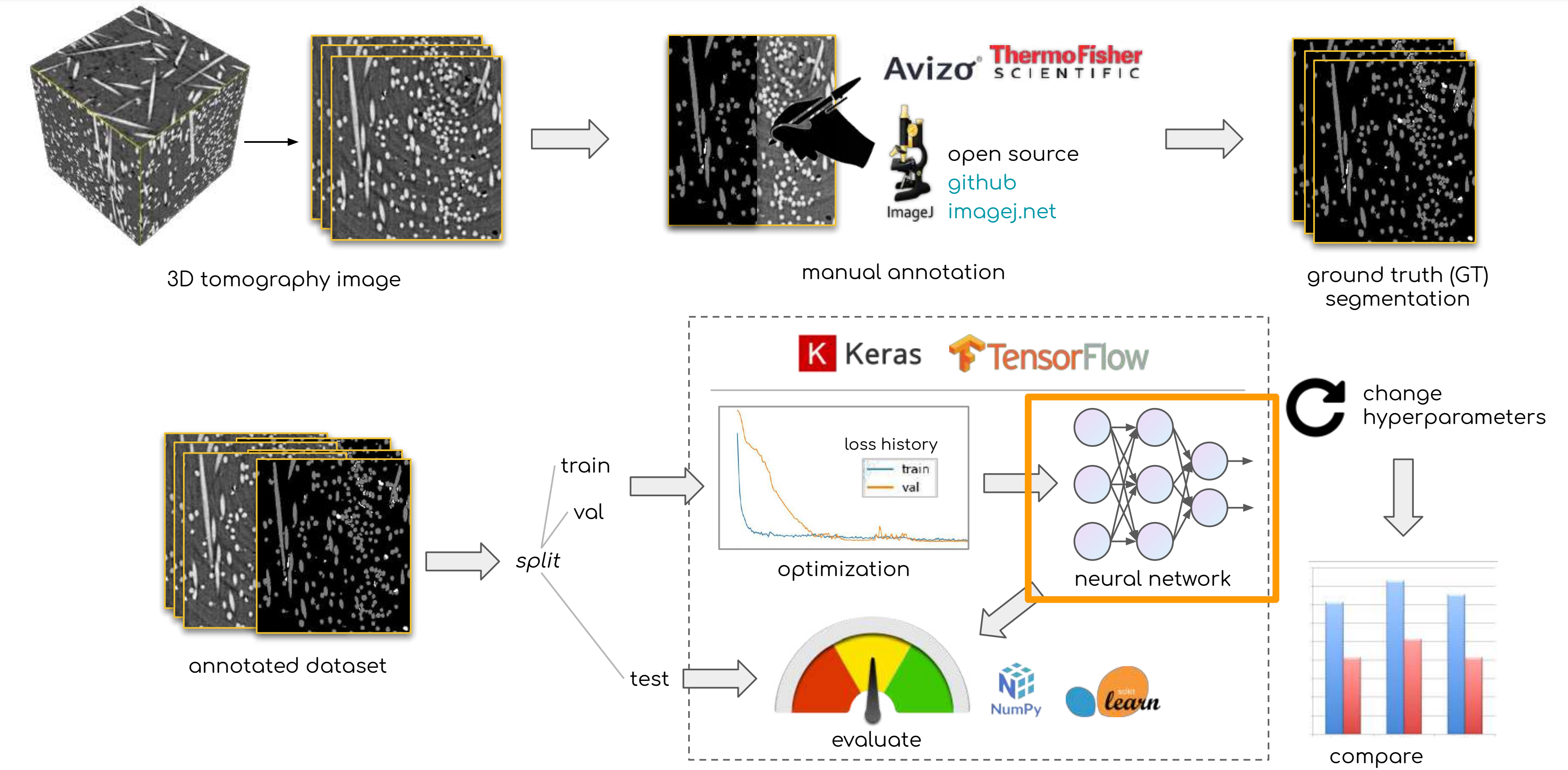
1. 2D/3D crop
2. geometric transformation

both random

neural network

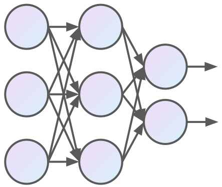
overview

project repository: github.com/joaopcbertoldo/tomo2seg



a classification network

224 x 224 image



neural network

"papillon",
"cat",
"llama",
"ambulance",
...

http://image-net.org

Deng, J., Dong, W., Socher, R., Li, L., Kai Li, Li Fei-Fei, 2009. ImageNet: A large-scale hierarchical image database, in: 2009 IEEE Conference on Computer Vision and Pattern Recognition. Presented at the 2009 IEEE Conference on Computer Vision and Pattern Recognition, pp. 248–255.

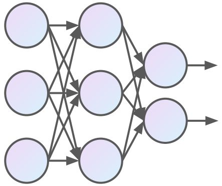
2048 x 2048 image

2048 x 2048 segmentation

per-pixel classification

a classification network

224 x 224 image



neural network

"papillon",
"cat",
"llama",
"ambulance",
...

http://image-net.org

Deng, J., Dong, W., Socher, R., Li, L., Kai Li, Li Fei-Fei, 2009. ImageNet: A large-scale hierarchical image database, in: 2009 IEEE Conference on Computer Vision and Pattern Recognition. Presented at the 2009 IEEE Conference on Computer Vision and Pattern Recognition, pp. 248–255.

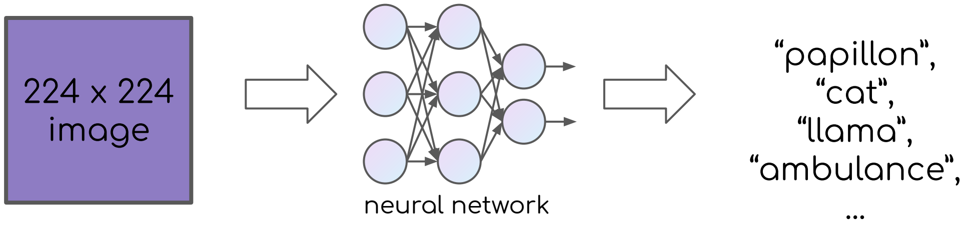
2048 x 2048 image

an easy solution?

2048 x 2048 segmentation

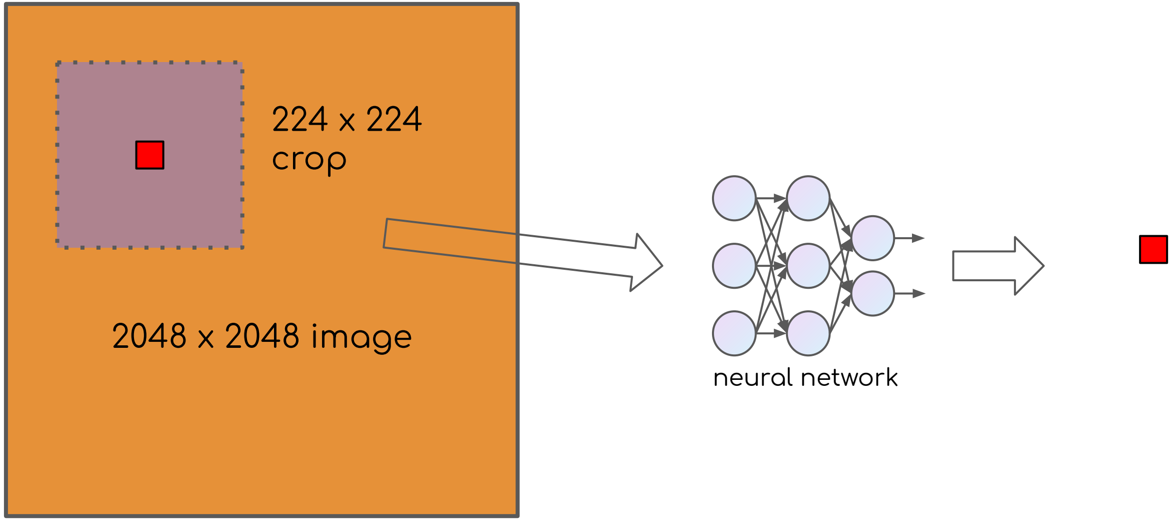
per-pixel classification

a classification network



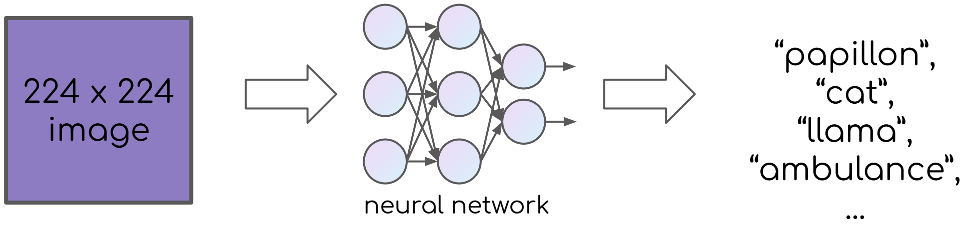
http://image-net.org

Deng, J., Dong, W., Socher, R., Li, L., Kai Li, Li Fei-Fei, 2009. ImageNet: A large-scale hierarchical image database, in: 2009 IEEE Conference on Computer Vision and Pattern Recognition. Presented at the 2009 IEEE Conference on Computer Vision and Pattern Recognition, pp. 248–255.



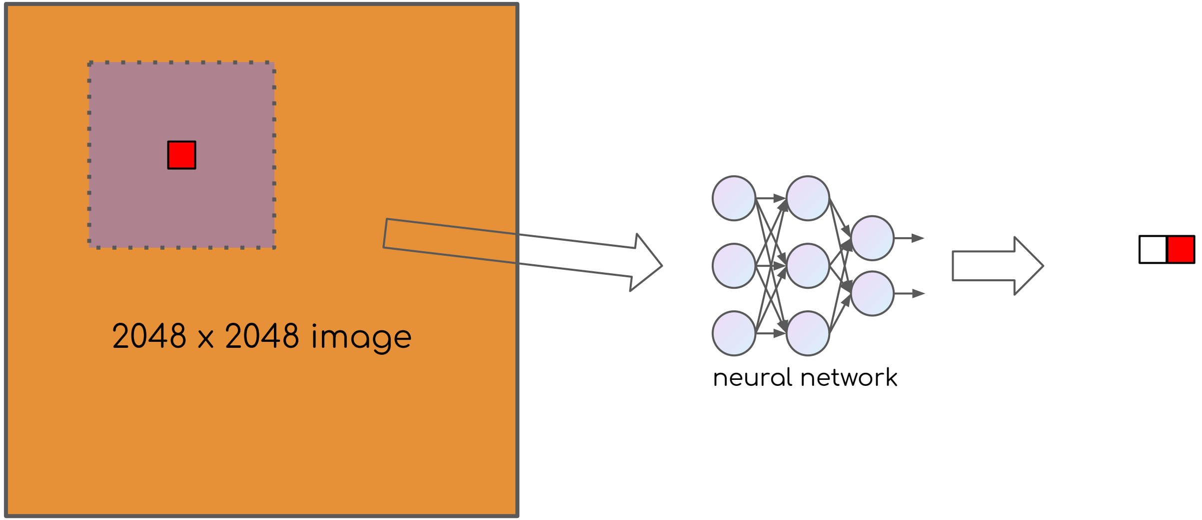
Cireşan, D.C., Giusti, A., Gambardella, L.M., Schmidhuber, J., 2012. Deep neural networks segment neuronal membranes in electron microscopy images, in: Proceedings of the 25th International Conference on Neural Information Processing Systems - Volume 2, NIPS'12. Curran Associates Inc., Red Hook, NY, USA, pp. 2843–2851.

a classification network



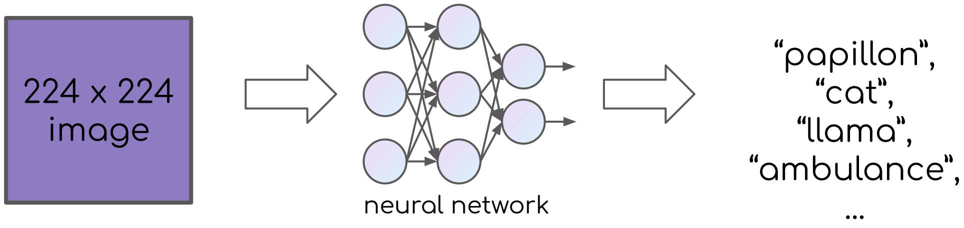
http://image-net.org

Deng, J., Dong, W., Socher, R., Li, L., Kai Li, Li Fei-Fei, 2009. ImageNet: A large-scale hierarchical image database, in: 2009 IEEE Conference on Computer Vision and Pattern Recognition. Presented at the 2009 IEEE Conference on Computer Vision and Pattern Recognition, pp. 248–255.



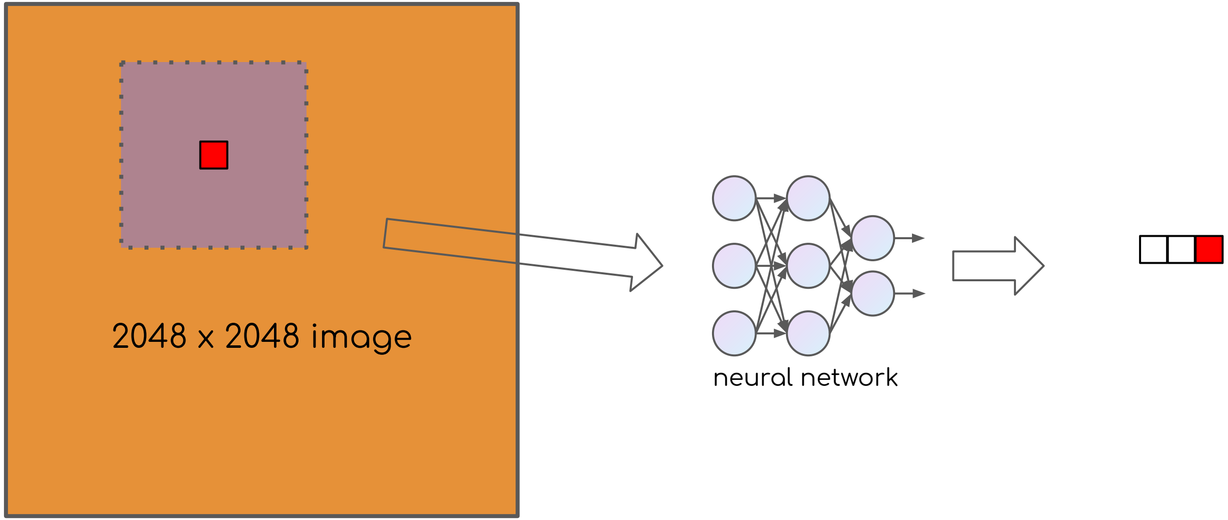
Cireşan, D.C., Giusti, A., Gambardella, L.M., Schmidhuber, J., 2012. Deep neural networks segment neuronal membranes in electron microscopy images, in: Proceedings of the 25th International Conference on Neural Information Processing Systems - Volume 2, NIPS'12. Curran Associates Inc., Red Hook, NY, USA, pp. 2843–2851.

a classification network



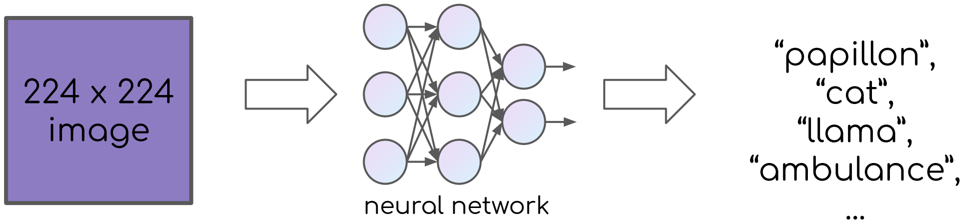
http://image-net.org

Deng, J., Dong, W., Socher, R., Li, L., Kai Li, Li Fei-Fei, 2009. ImageNet: A large-scale hierarchical image database, in: 2009 IEEE Conference on Computer Vision and Pattern Recognition. Presented at the 2009 IEEE Conference on Computer Vision and Pattern Recognition, pp. 248–255.



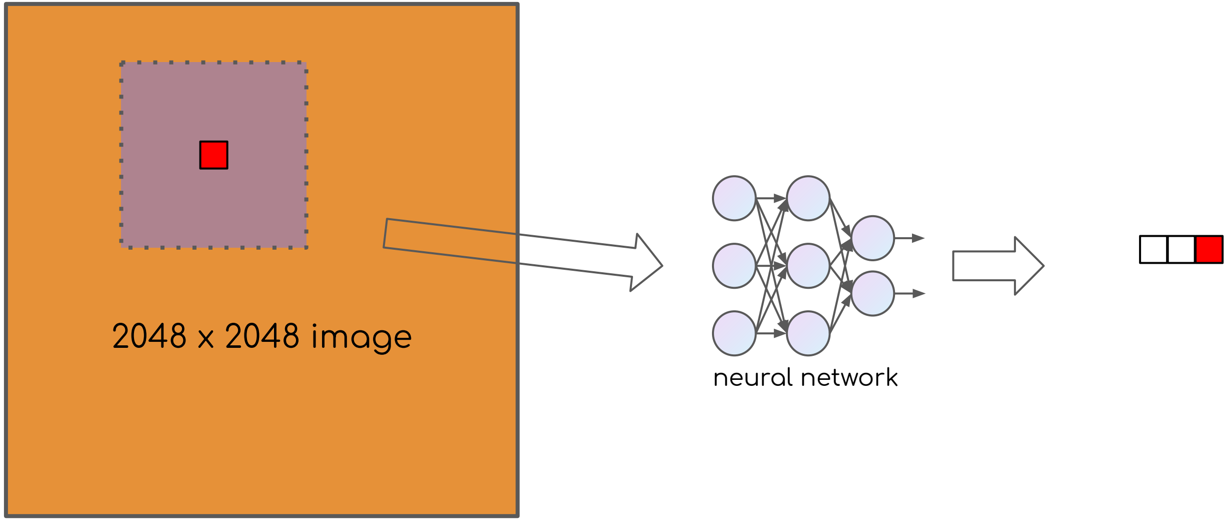
Cireşan, D.C., Giusti, A., Gambardella, L.M., Schmidhuber, J., 2012. Deep neural networks segment neuronal membranes in electron microscopy images, in: Proceedings of the 25th International Conference on Neural Information Processing Systems - Volume 2, NIPS’12. Curran Associates Inc., Red Hook, NY, USA, pp. 2843–2851.

a classification network



http://image-net.org

Deng, J., Dong, W., Socher, R., Li, L., Kai Li, Li Fei-Fei, 2009. ImageNet: A large-scale hierarchical image database, in: 2009 IEEE Conference on Computer Vision and Pattern Recognition. Presented at the 2009 IEEE Conference on Computer Vision and Pattern Recognition, pp. 248–255.

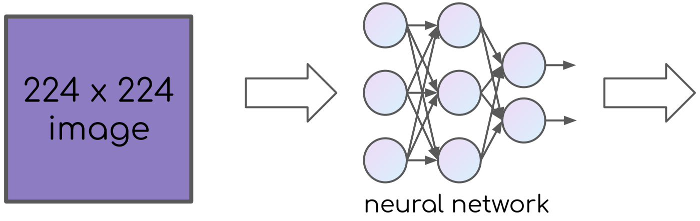


problem

- it’s very inefficient

Cireşan, D.C., Giusti, A., Gambardella, L.M., Schmidhuber, J., 2012. Deep neural networks segment neuronal membranes in electron microscopy images, in: Proceedings of the 25th International Conference on Neural Information Processing Systems - Volume 2, NIPS’12. Curran Associates Inc., Red Hook, NY, USA, pp. 2843–2851.

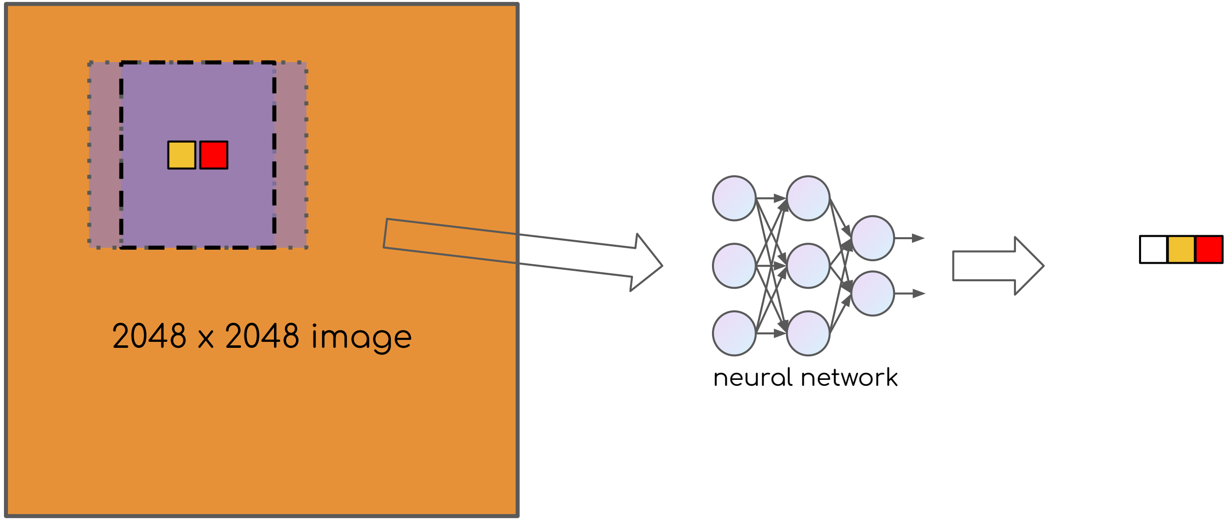
a classification network



"papillon",
"cat",
"llama",
"ambulance",
...

http://image-net.org

Deng, J., Dong, W., Socher, R., Li, L., Kai Li, Li Fei-Fei, 2009. ImageNet: A large-scale hierarchical image database, in: 2009 IEEE Conference on Computer Vision and Pattern Recognition. Presented at the 2009 IEEE Conference on Computer Vision and Pattern Recognition, pp. 248–255.



problem

- it's very inefficient

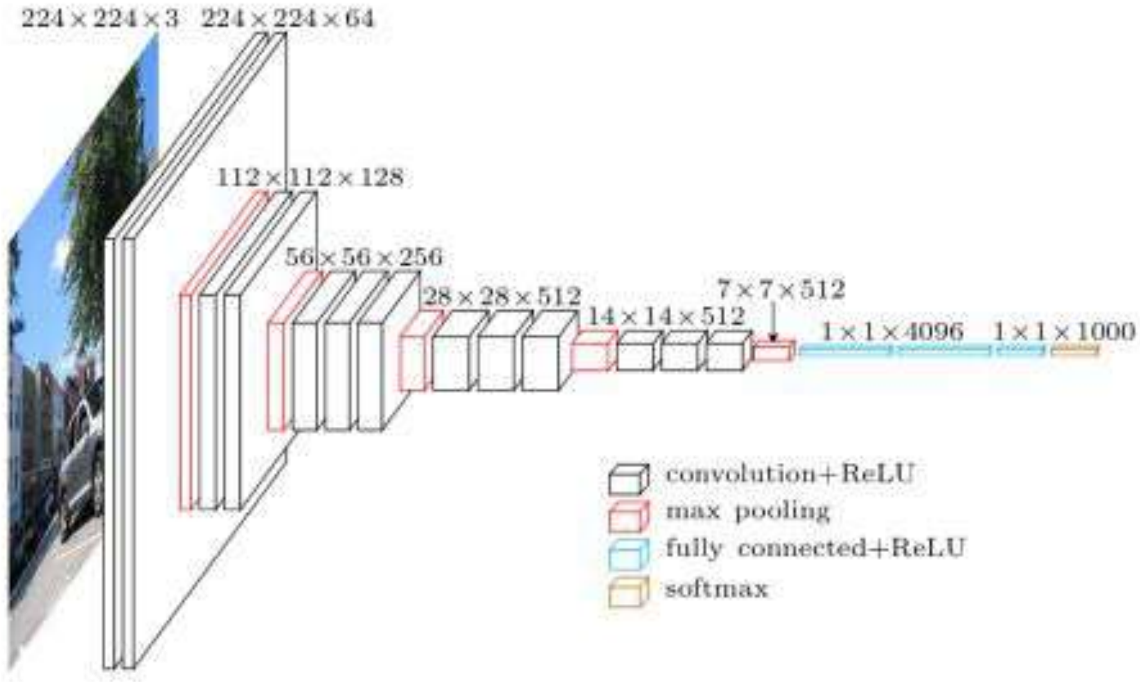
- adjacent pixels share most of the input region

Cireşan, D.C., Giusti, A., Gambardella, L.M., Schmidhuber, J., 2012. Deep neural networks segment neuronal membranes in electron microscopy images, in: Proceedings of the 25th International Conference on Neural Information Processing Systems - Volume 2, NIPS'12. Curran Associates Inc., Red Hook, NY, USA, pp. 2843–2851.

a classification network

input

224 x 224 image



output

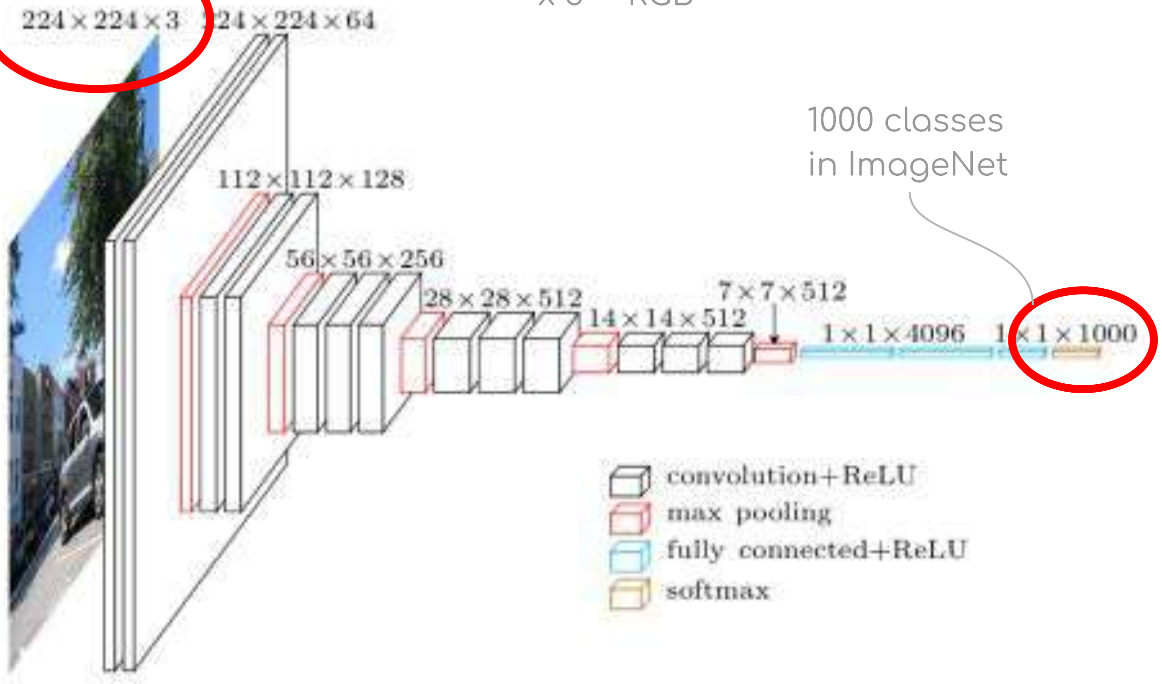
"papillon",
"cat",
"llama",
"ambulance",
...

Credits: David Frossard. Image source:http://www.cs.toronto.edu/~frossard/post/vgg16/

a classification network

input

224 x 224 image



"x 3" = RGB

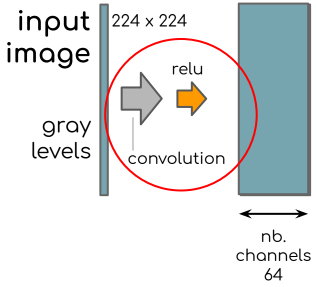
1000 classes in ImageNet

output

"papillon", "cat", "llama", "ambulance", ...

Credits: David Frossard. Image source: http://www.cs.toronto.edu/~frossard/post/vgg16/

a classification network

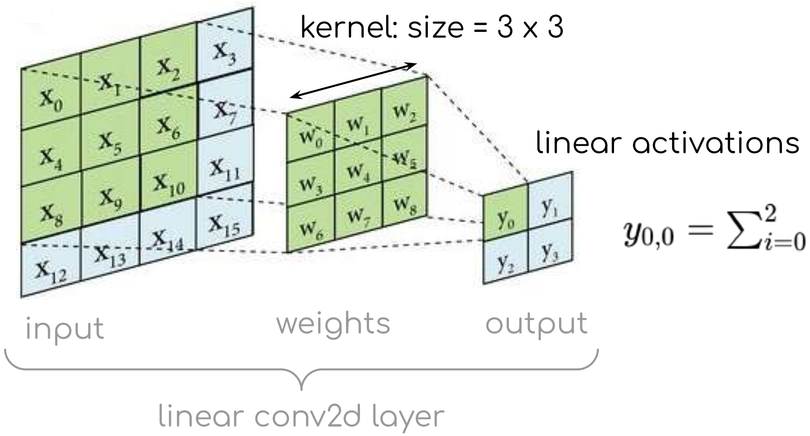


[tf.keras.layers.Conv2D]

[tf.keras.layers.ReLU]

conv2d and relu

Credits: Samrat Sahoo. Source: 2D Convolution using Python & NumPy.



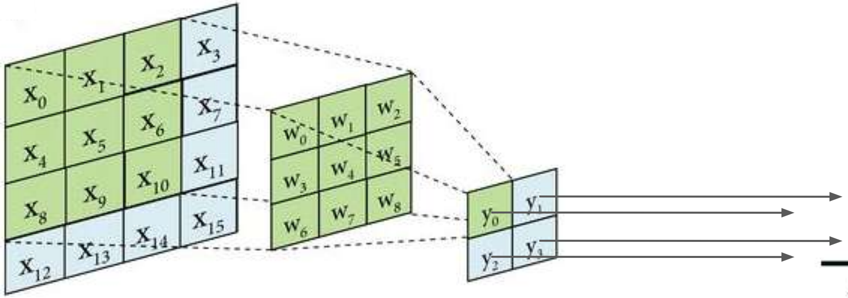
$$y_{0,0} = \sum_{i=0}^{2} \sum_{j=0}^{2} x_{i,j} \times w_{i,j}$$

PS: there is a mismatch between indices of the image and the equation. Be imaginative (:

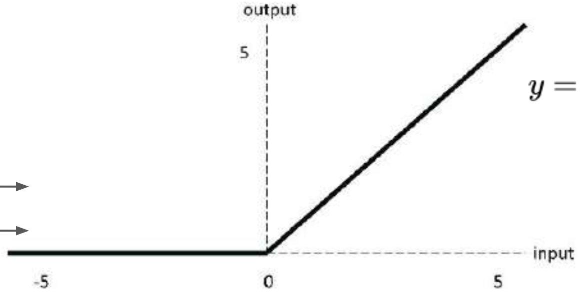
conv2d and relu

Credits: Samrat Sahoo. Source: 2D Convolution using Python & NumPy.

H. Sultan, H., Salem, N., Al-Atabany, W., 2019. Multi-Classification of Brain Tumor Images Using Deep Neural Network. IEEE Access PP, 1–1.



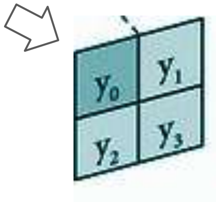
linear conv2d layer



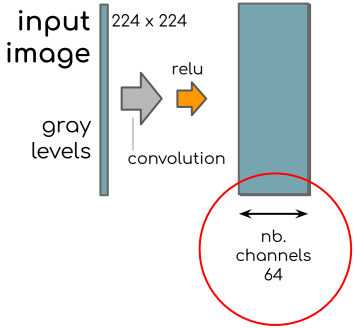
relu

"Rectified Linear Unit"

$$y = \max(0,\, x) = \begin{cases} x & x > 0 \\ 0 & \text{otherwise} \end{cases}$$



a classification network



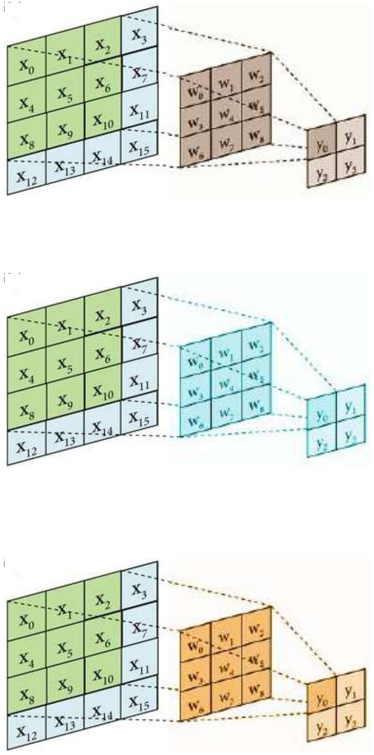
conv2d and relu

many convolution kernels

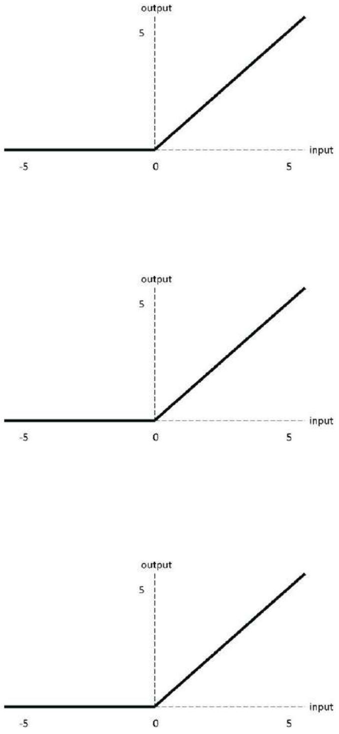
Credits: Samrat Sahoo. Source: [2D Convolution using Python & NumPy](#).

H. Sultan, H., Salem, N., Al-Atabany, W., 2019. Multi-Classification of Brain Tumor Images Using Deep Neural Network. IEEE Access PP, 1–1.

same input



linear conv2d layer



relu

64 kernels (filters)

=

64 times the same operation, with <u>different weights</u> in each

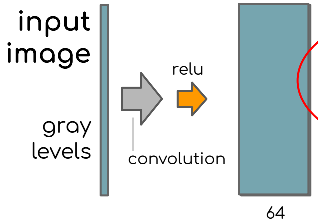
therefore:

64 different activation values/features

64 different spatial patterns

a classification network

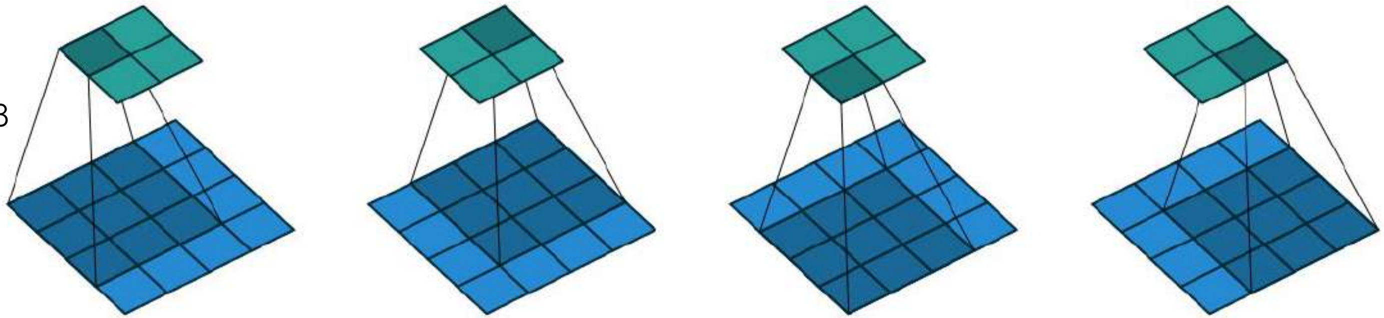
224 x 224



222 x 222 x 64

border loss:

kernel 3x3
(omitted)

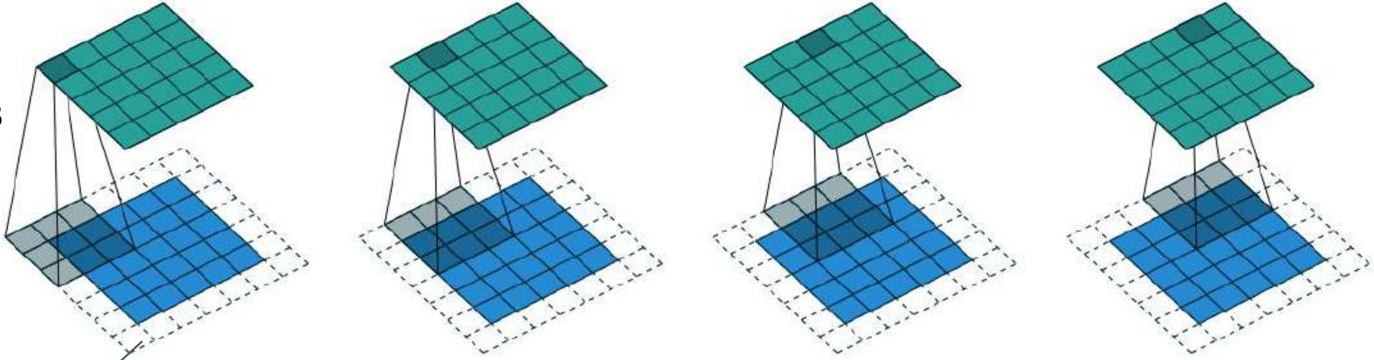


Dumoulin, V., Visin, F., 2018. A guide to convolution arithmetic for deep learning.

conv2d and relu

output 5x5

kernel 3x3
(omitted)



input 5x5

Dumoulin, V., Visin, F., 2018. A guide to convolution arithmetic for deep learning.

value?
constant (0, 1),
mirror (reflection),
...

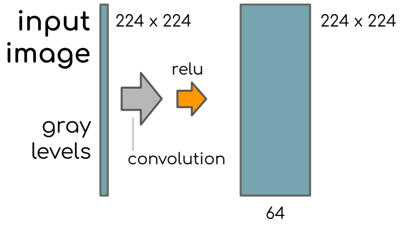
1	5	3	5	1

Credits: Christian Versloot.
Source: [Using Constant Padding, Reflection Padding and Replication Padding with TensorFlow and Keras]

half (same) padding
kernel size = 3
input image = 5 x 5

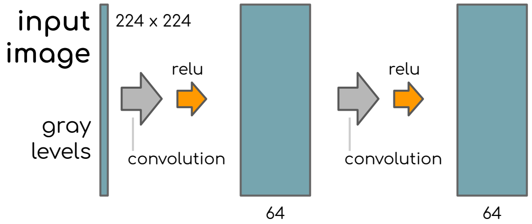
we will suppose half padding from now on!

a classification network



a classification network

VGG16

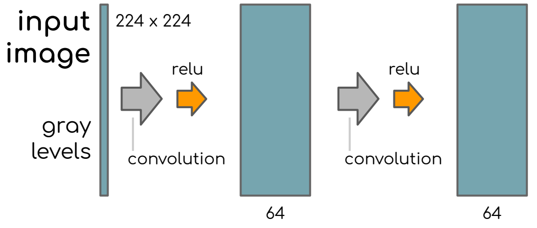


a classification network

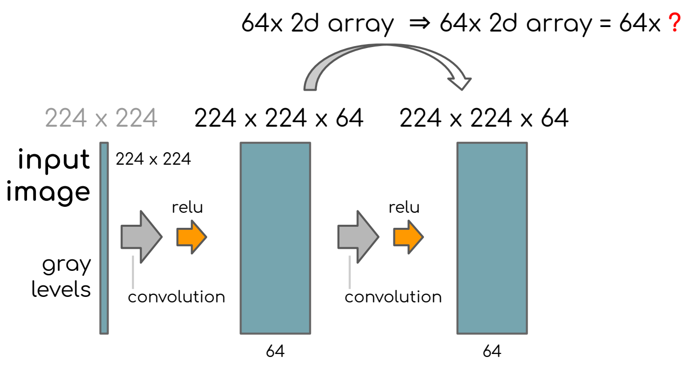
2d array ⇒ 64x 2d array = 64x

$$y_{0,0} = \sum_{i=0}^{2} \sum_{j=0}^{2} x_{i,j} \times w_{i,j}$$

224 x 224 224 x 224 x 64 224 x 224 x 64



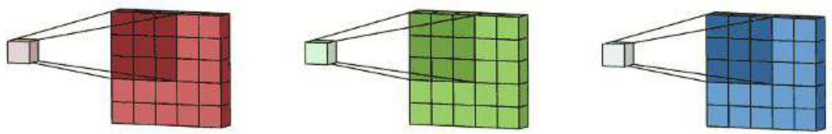
a classification network



conv2d and relu

each input channel (2d “images”)

64x
kernel 3 x 3
(omitted)

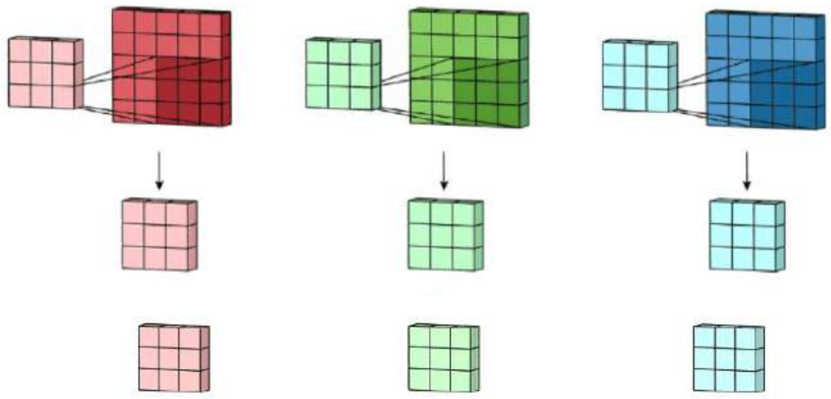


Credits: Kunlun Bai. Source: [A Comprehensive Introduction to Different Types of Convolutions in Deep Learning]

conv2d and relu

each input channel (2d "images")

64x
kernel 3 x 3
(omitted)



Credits: Kunlun Bai. Source: A Comprehensive Introduction to Different Types of Convolutions in Deep Learning

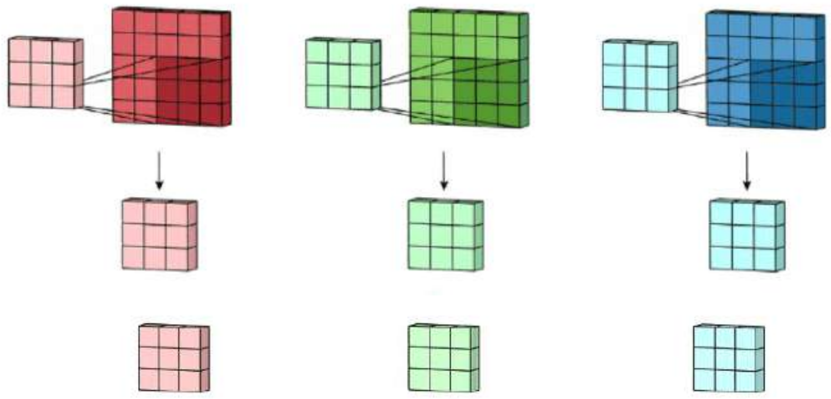
1 output channel

1 (2d) image from 64 (2d) images

convoluted (each) with 1 (2d) kernel

conv2d and relu

each input channel (2d "images")



64 times this operation,
on all the 64 input channels,
with <u>different weights</u>
in each 2d kernel

1 output channel

1 (2d) image from 64 (2d) images

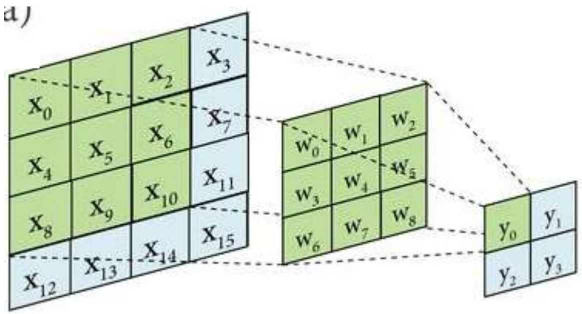
convoluted (each) with 1 (2d) kernel

Credits: Kunlun Bai. Source: [A Comprehensive Introduction to Different Types of Convolutions in Deep Learning]()

conv2d and relu

2d input

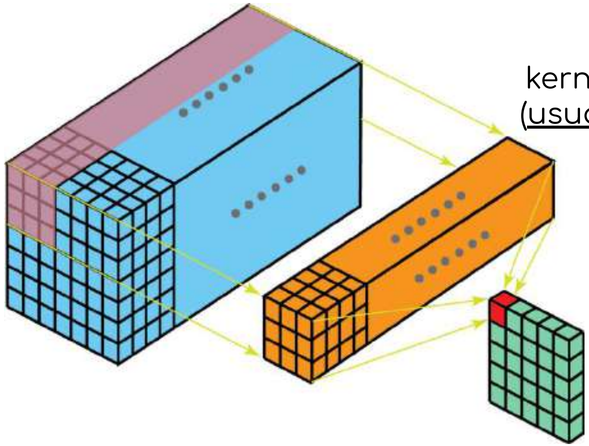
$$y_{0,0} = \sum_{i=0}^{2} \sum_{j=0}^{2} x_{i,j} \times w_{i,j}$$



Credits: Samrat Sahoo. Source: [2D Convolution using Python & NumPy].

3d input
(2d spatial + 1d channels)

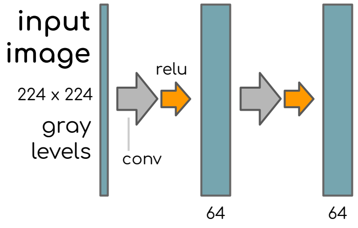
$$y_{0,0} = \sum_{i=0}^{2} \sum_{j=0}^{2} \sum_{c=0}^{64} x_{i,j,c} \times w_{i,j,c}$$

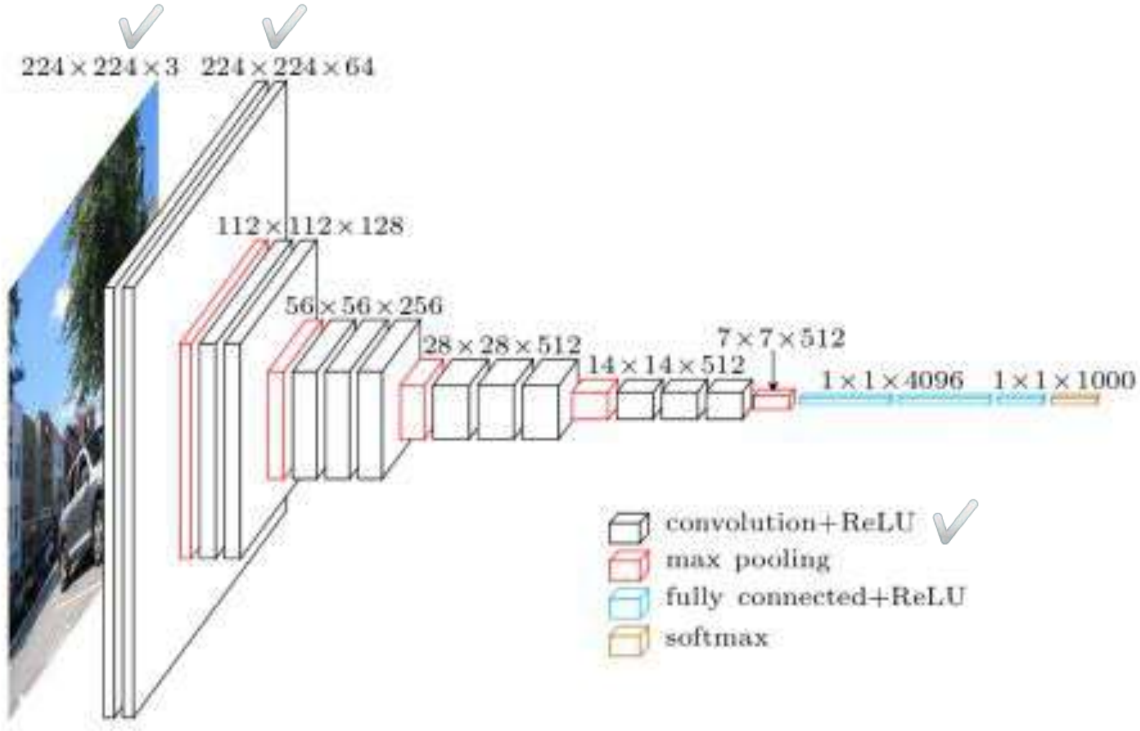


kernel 3 x 3 x 64
(usually omitted)

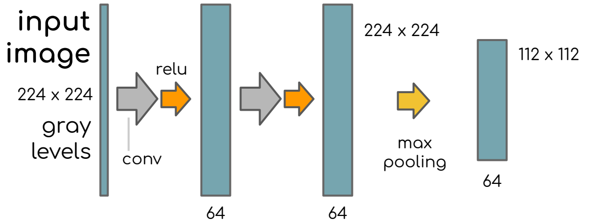
Credits: Kunlun Bai. Source: [A Comprehensive Introduction to Different Types of Convolutions in Deep Learning]

a classification network

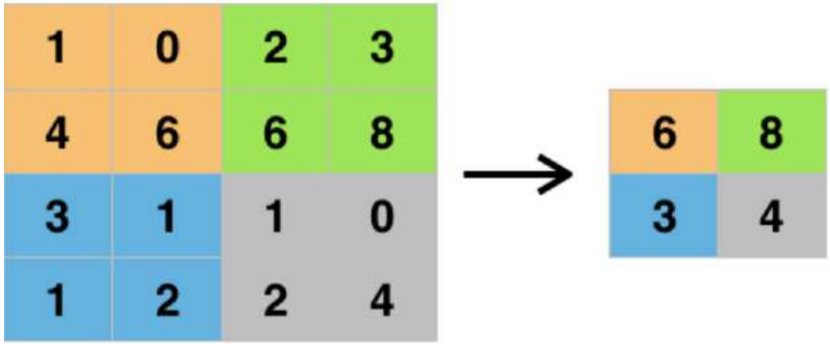




Credits: David Frossard. Image source:http://www.cs.toronto.edu/~frossard/post/vgg16/

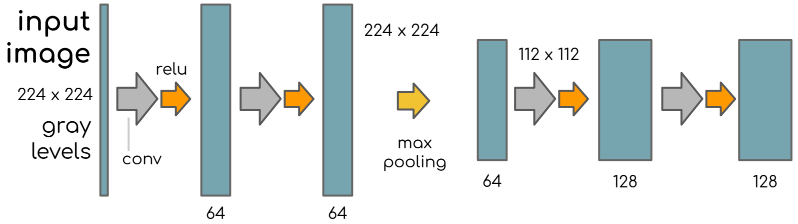


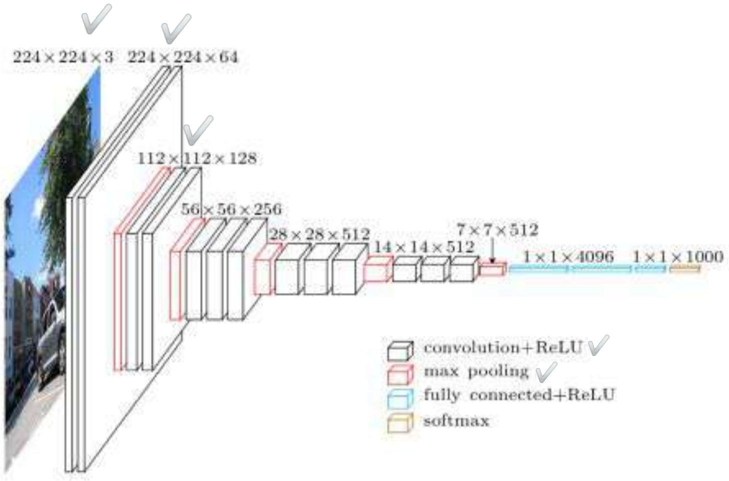
max pooling: tf.keras.layers.MaxPool2D



Credits: DeepAI. Image source: https://deepai.org/machine-learning-glossary-and-terms/max-pooling

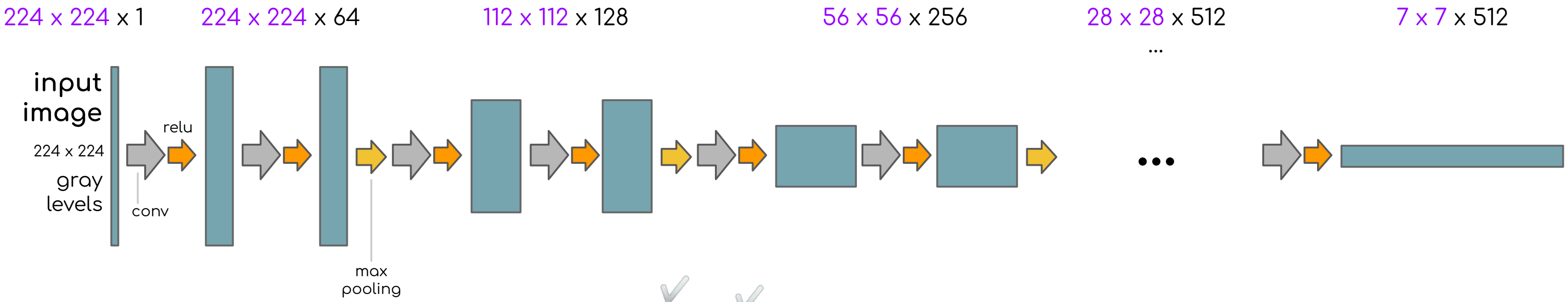
a classification network

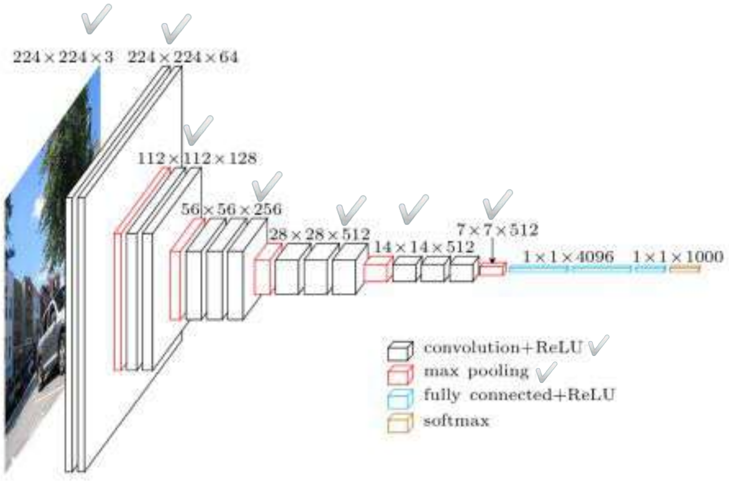




Credits: David Frossard. Image source:http://www.cs.toronto.edu/~frossard/post/vgg16/

a classification network



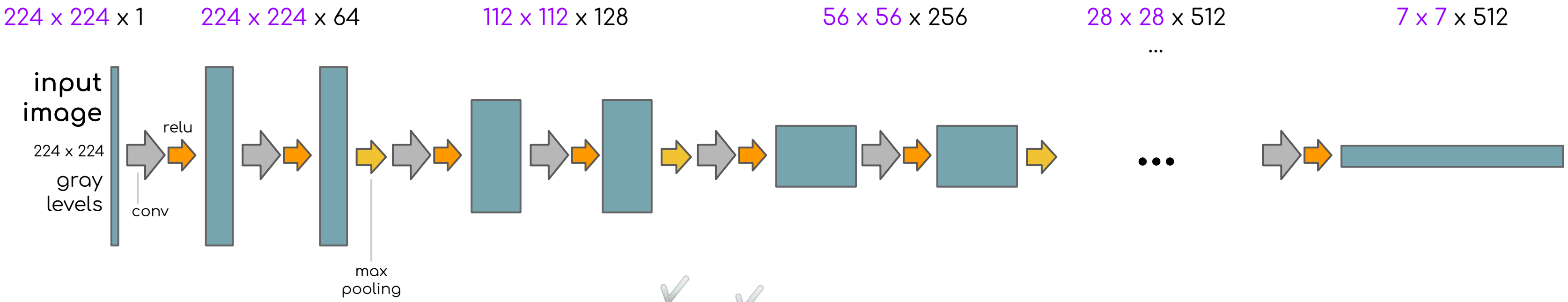


spatial dimensions ↘

number of channels ↗

Credits: David Frossard. Image source:http://www.cs.toronto.edu/~frossard/post/vgg16/

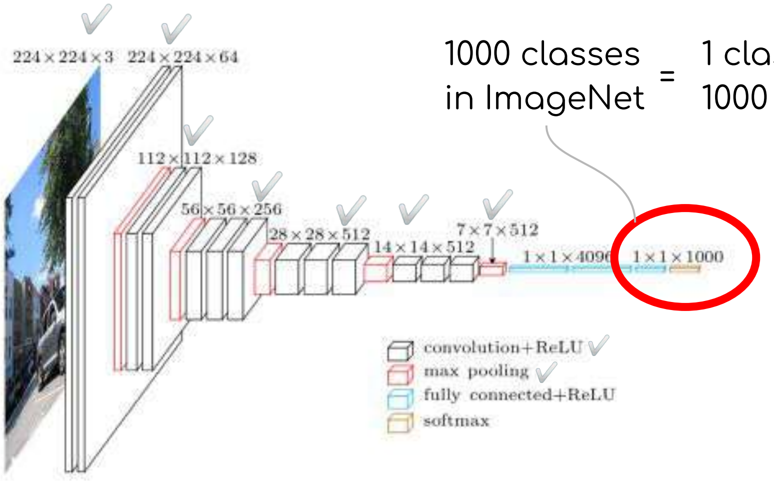
a classification network



1000 classes in ImageNet = 1 classification of 1000 classes = 1 x 1 x 1000

spatial dimensions ↘

number of channels ↗



Credits: David Frossard. Image source:http://www.cs.toronto.edu/~frossard/post/vgg16/

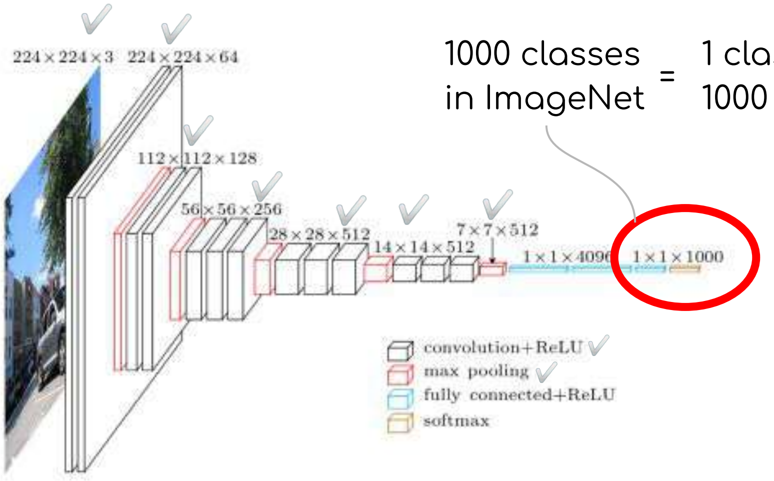
a classification network

2048 x 2048 image ⇨ another easy solution? ⇨ 2048 x 2048 segmentation

2048^2 classif. of 3 classes = 2048 x 2048 x 3

spatial dimensions ↘

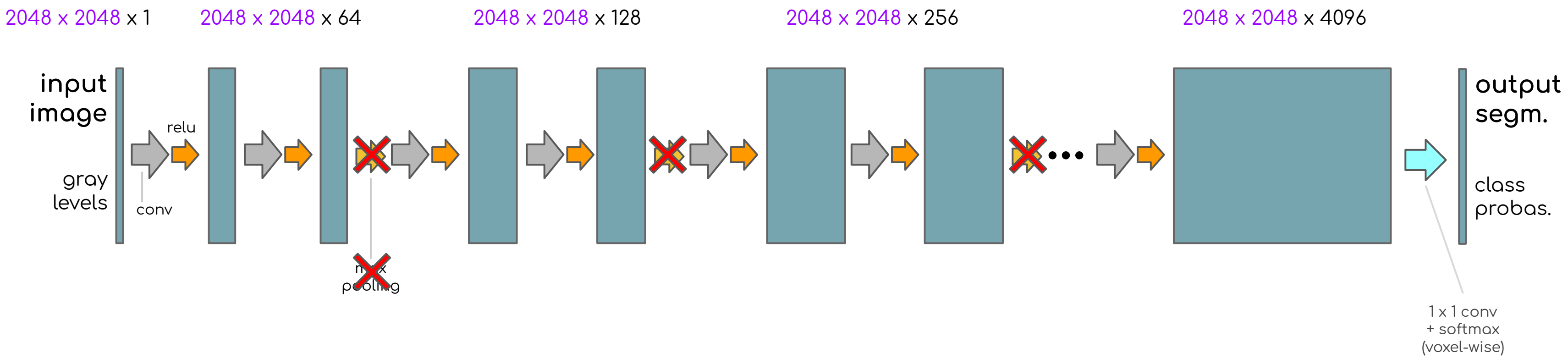
number of channels ↗



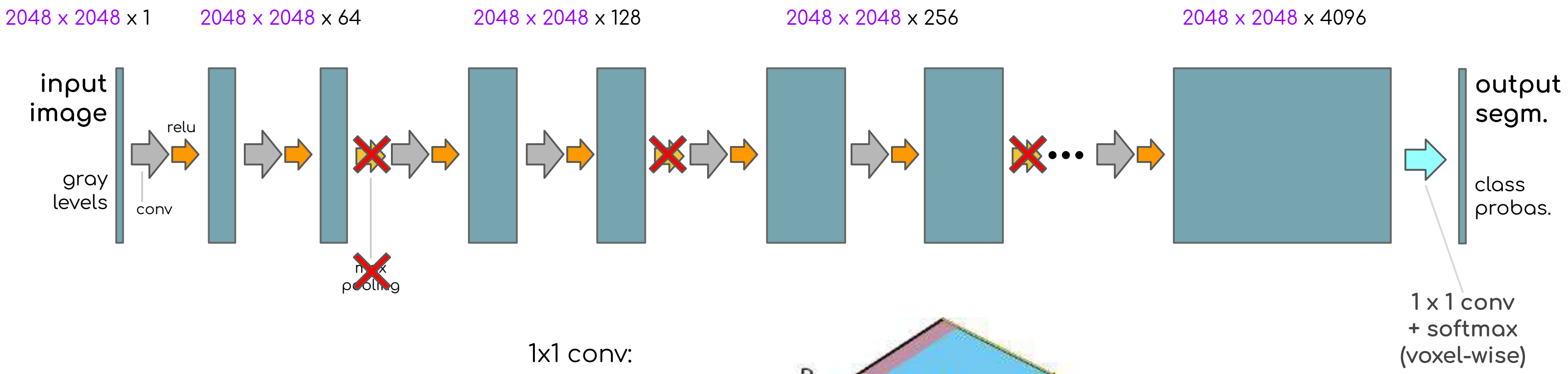
1000 classes in ImageNet = 1 classification of 1000 classes = 1 x 1 x 1000

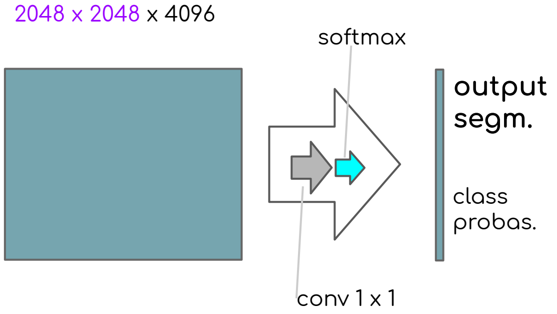
Credits: David Frossard. Image source:http://www.cs.toronto.edu/~frossard/post/vgg16/

an image-to-image network

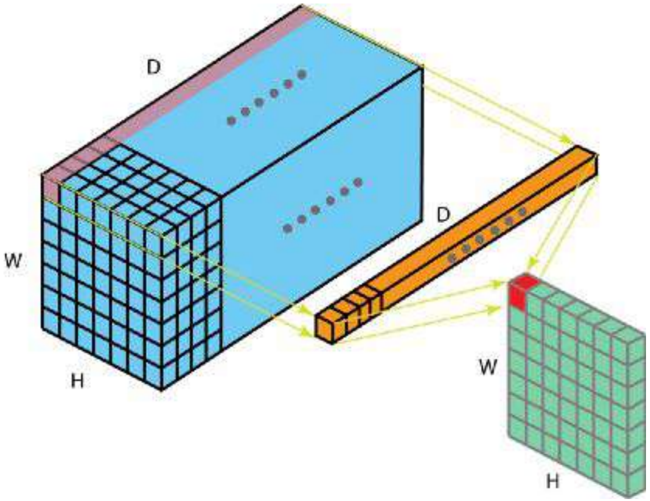


an image-to-image network





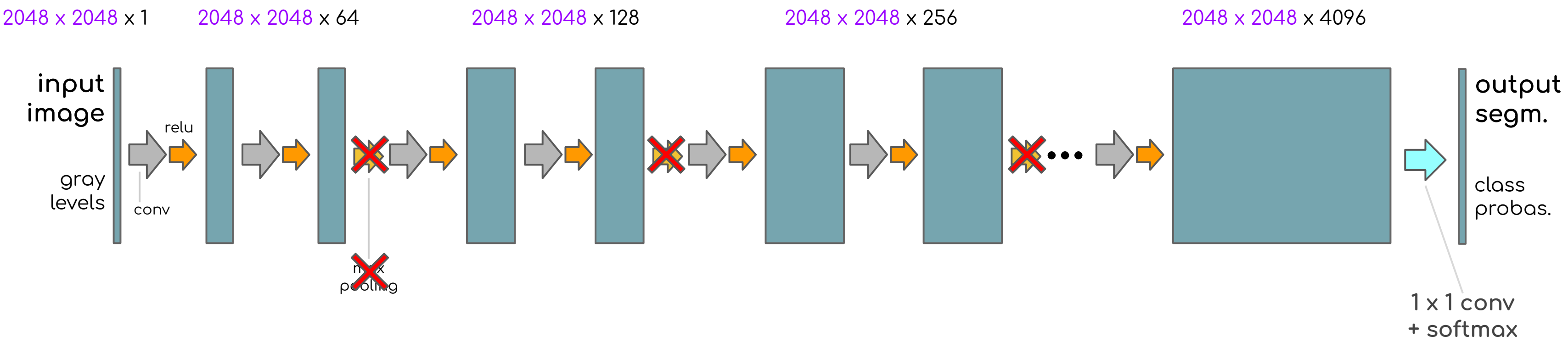
1x1 conv:



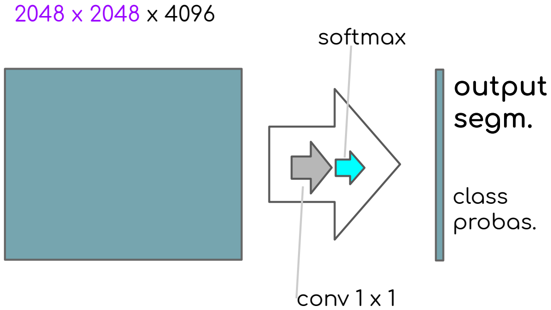
Credits: Kunlun Bai.
Source: [A Comprehensive Introduction to Different Types of Convolutions in Deep Learning]

X nb. of kernel
=
X nb. of classes
=
1 *score* per class per pixel

an image-to-image network



softmax: [tf.keras.layers.Softmax]



scores (z), pixels:

matrix	fiber	porosity
5	4	2
4	2	8
4	4	1

Softmax function →

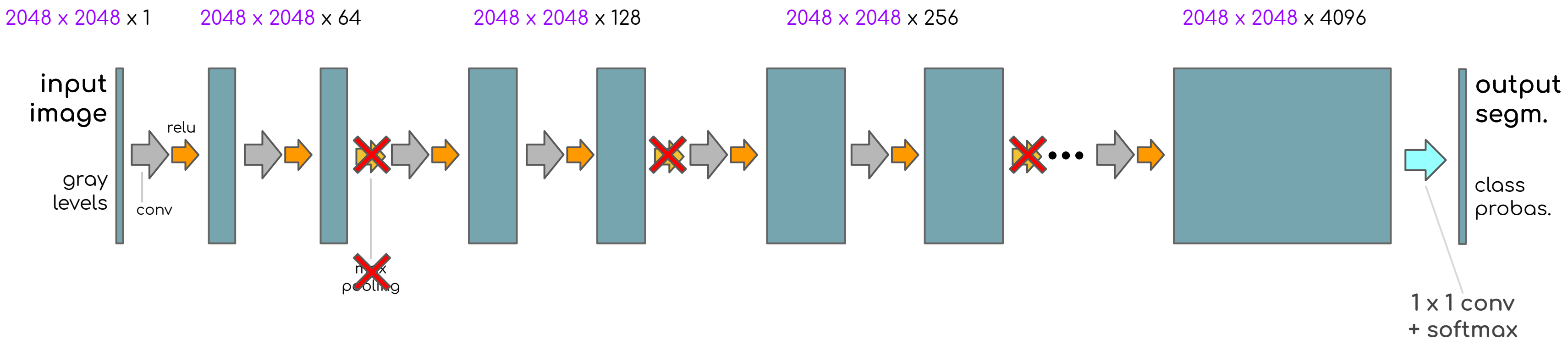
probabilities (ρ):

matrix	fiber	porosity
0.71	0.26	0.04
0.02	0.00	0.98
0.49	0.49	0.02

$$p_k = \frac{e^{z_k}}{\sum_{k=1}^{3} e^{z_k}}$$

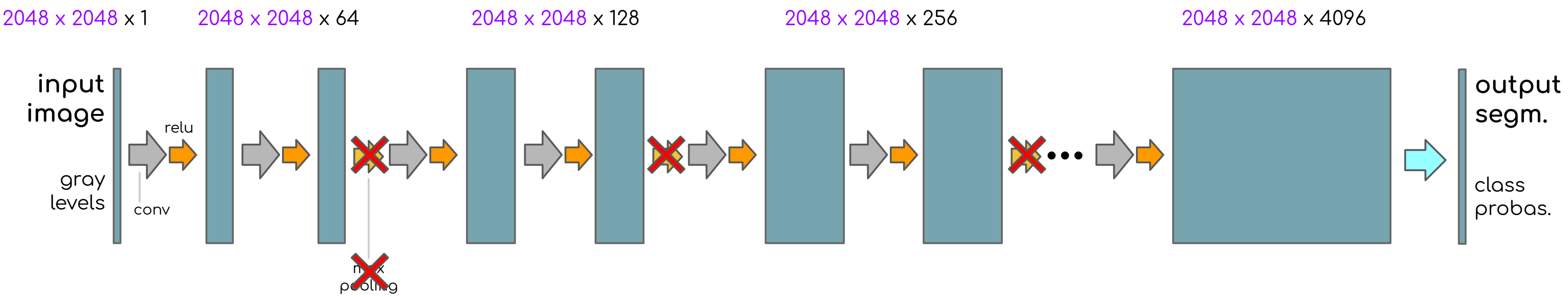
Credits: Lj Miranda. Source: [Understanding softmax and the negative log-likelihood] *break*

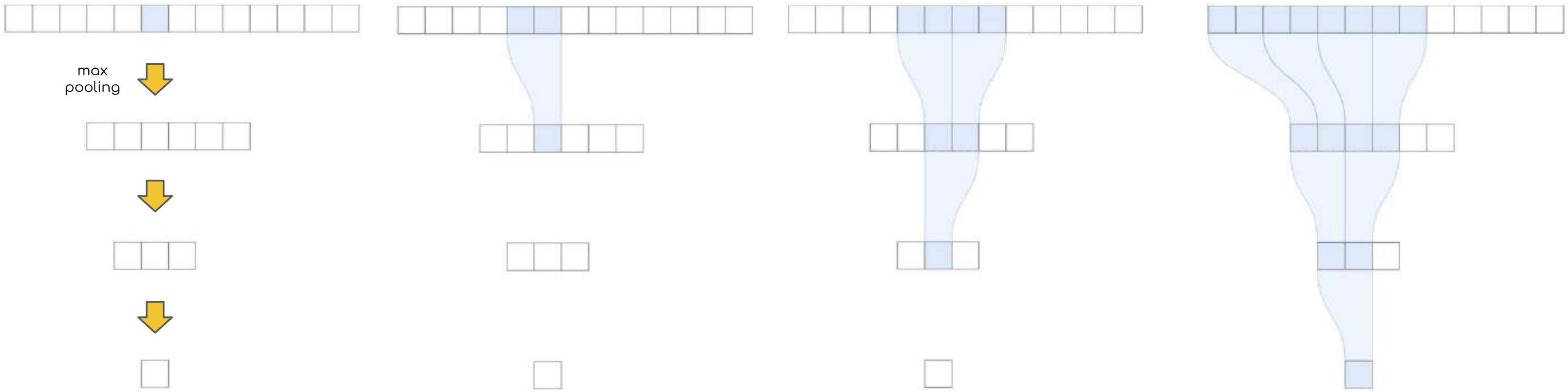
an image-to-image network



an image-to-image network

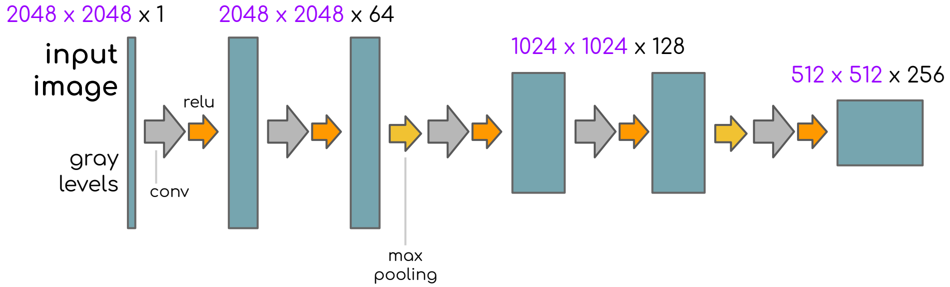
downsampling effect on the receptive field



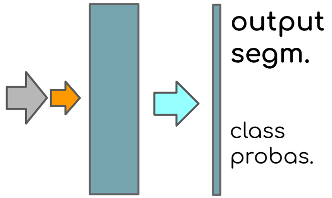


Araujo, et al., "Computing Receptive Fields of Convolutional Neural Networks", Distill, 2019. Images taken from animation in [Computing Receptive Fields of Convolutional Neural Networks](https://distill.pub/2019/computing-receptive-fields/).

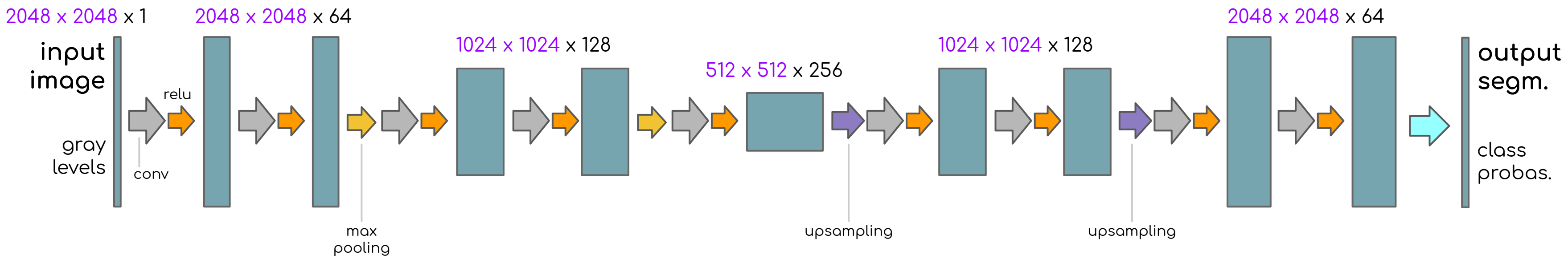
an image-to-image network



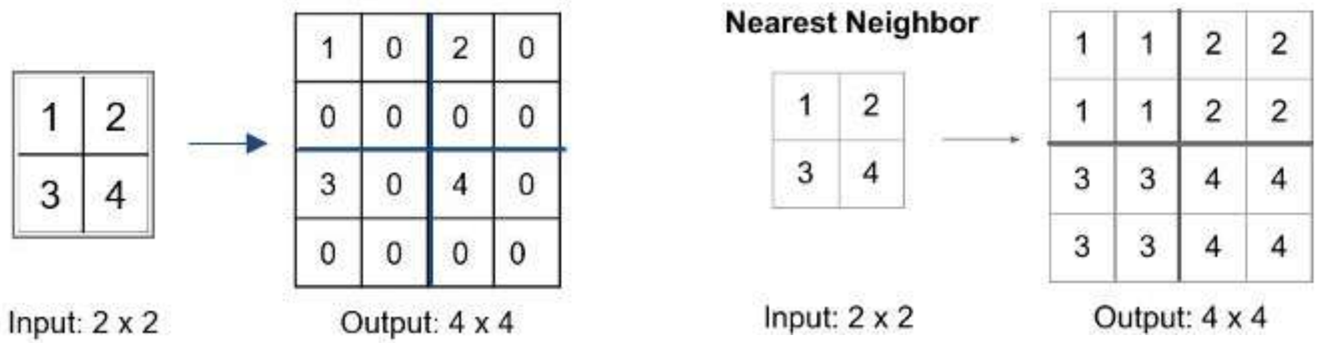
another solution?



an image-to-image network

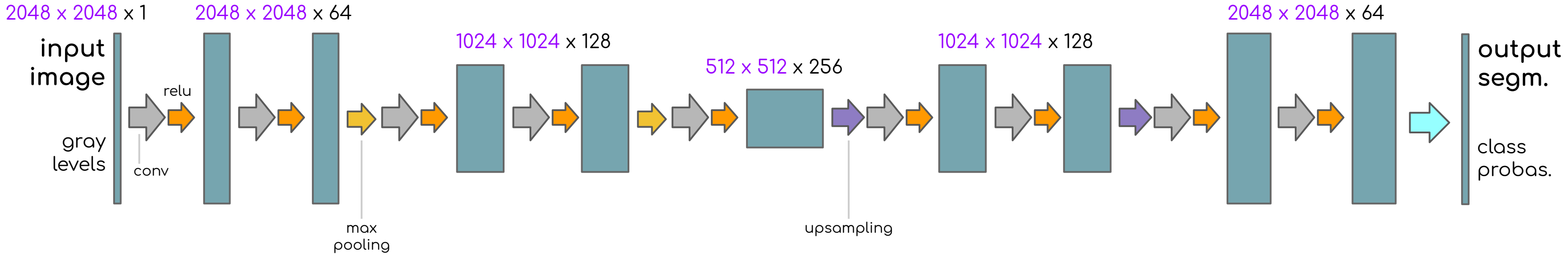


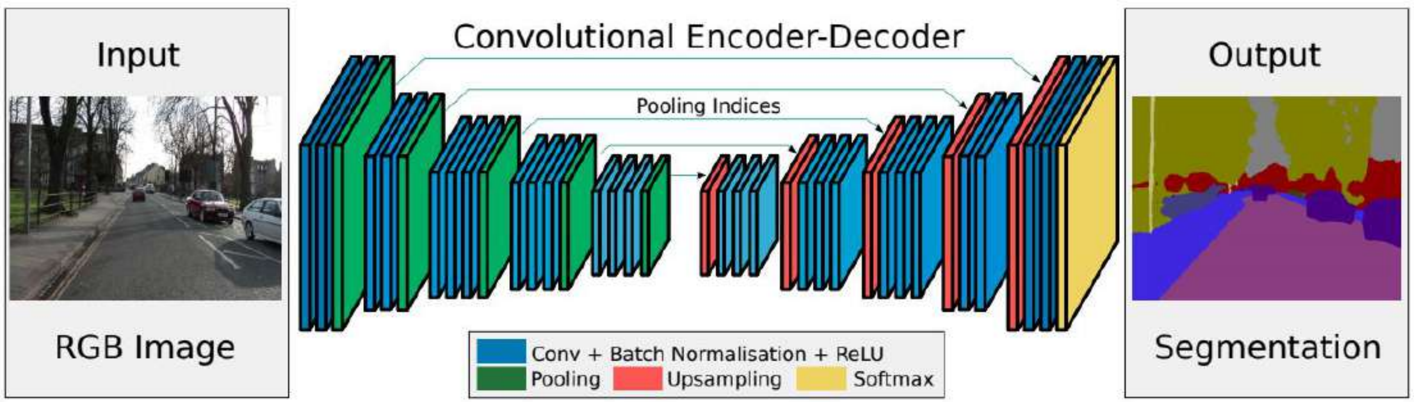
upsampling: tf.keras.layers.UpSampling2D



Credits: Divyanshu Mishra Source: Transposed Convolution Demystified.

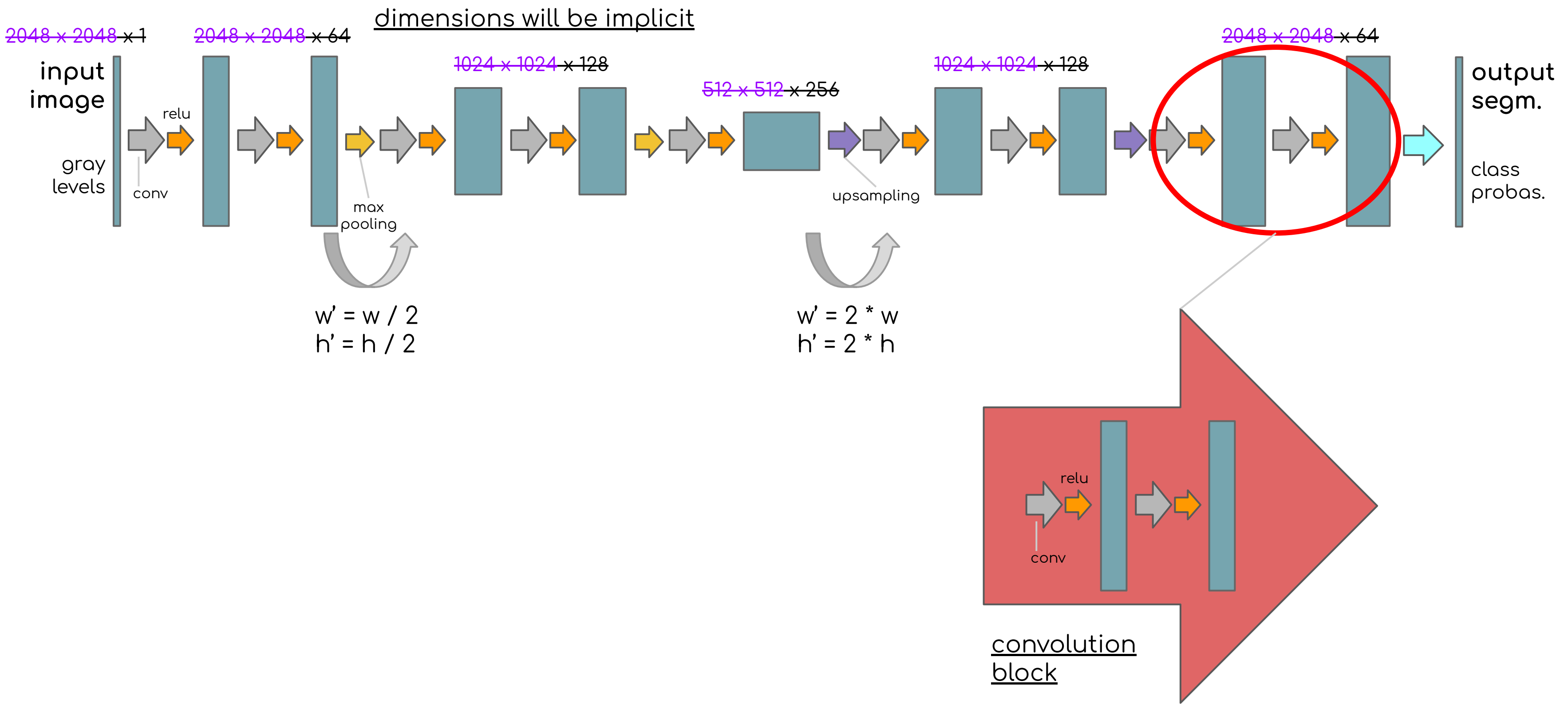
an image-to-image network



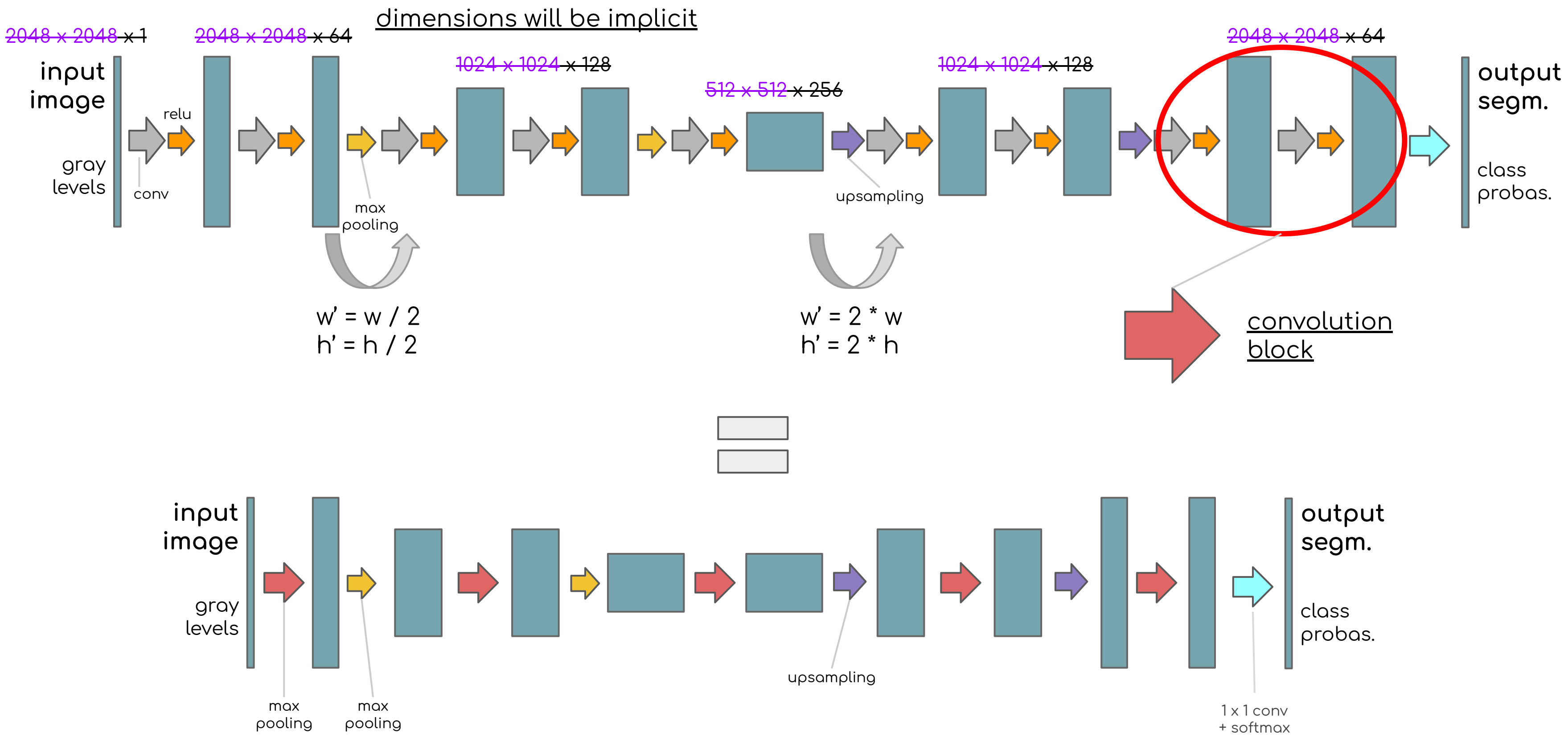


Badrinarayanan, V., Kendall, A., Cipolla, R., 2016. SegNet: A Deep Convolutional Encoder-Decoder Architecture for Image Segmentation.

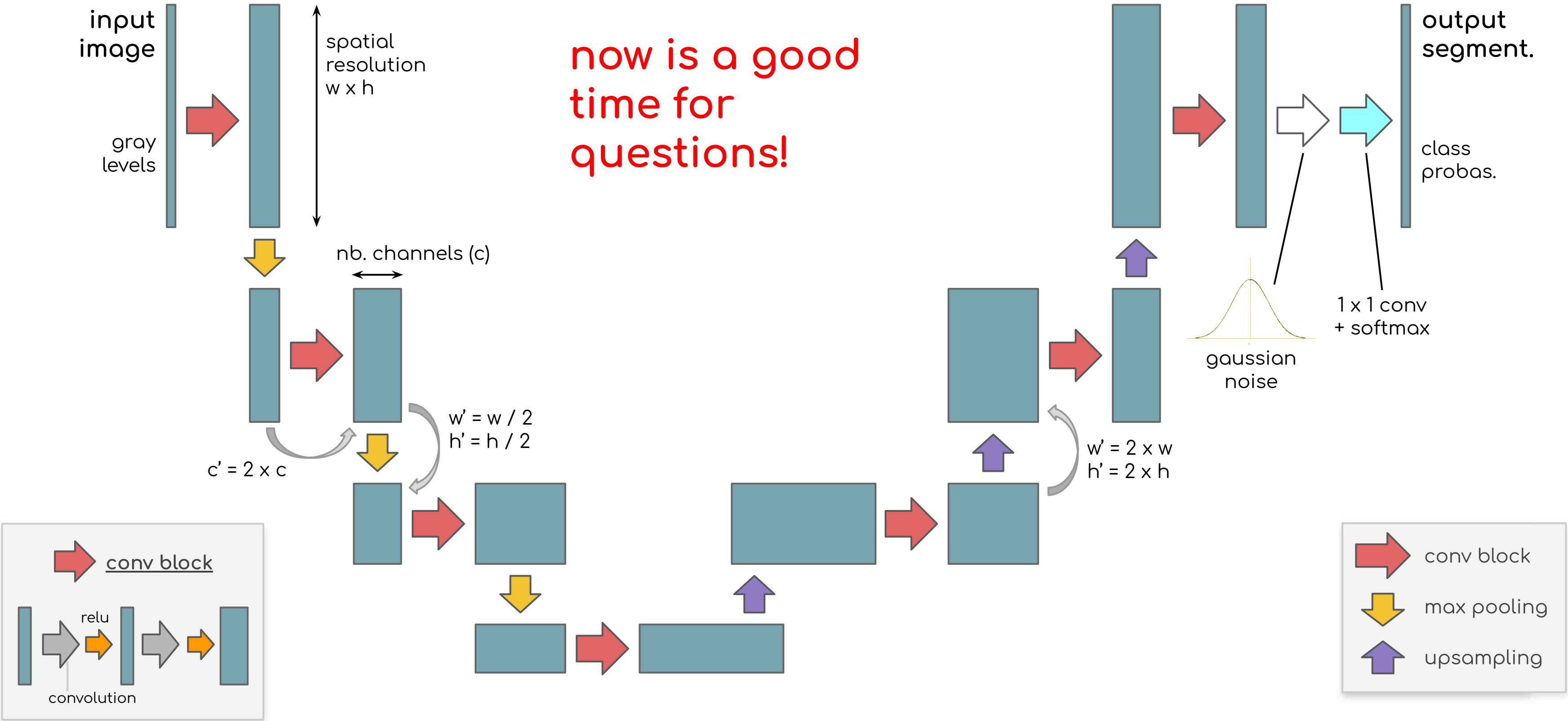
an image-to-image network



an image-to-image network



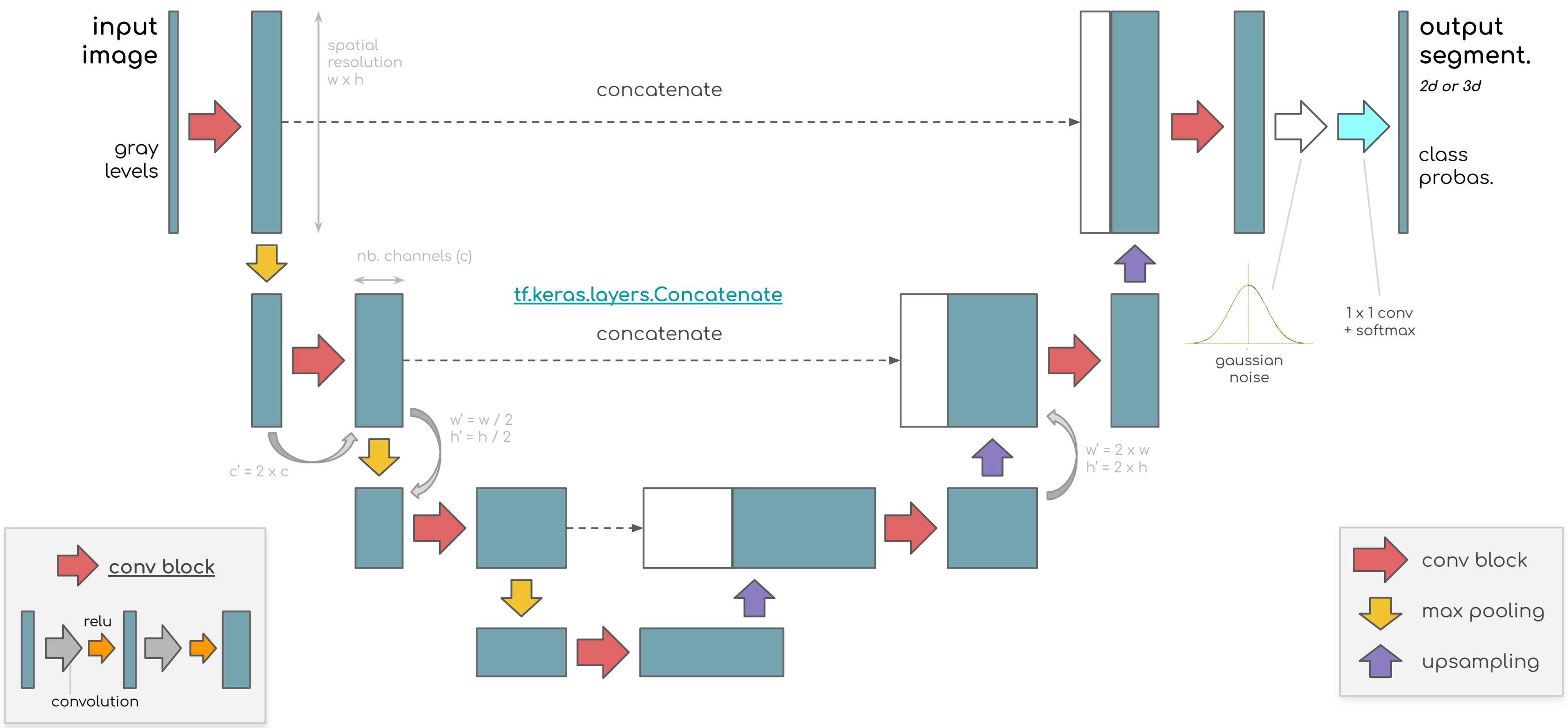
an image-to-image network



now is a good time for questions!

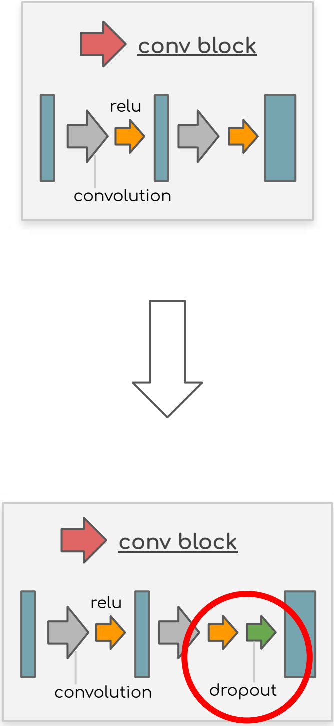
u-net

Ronneberger, O., Fischer, P., Brox, T., 2015. U-Net: Convolutional Networks for Biomedical Image Segmentation.



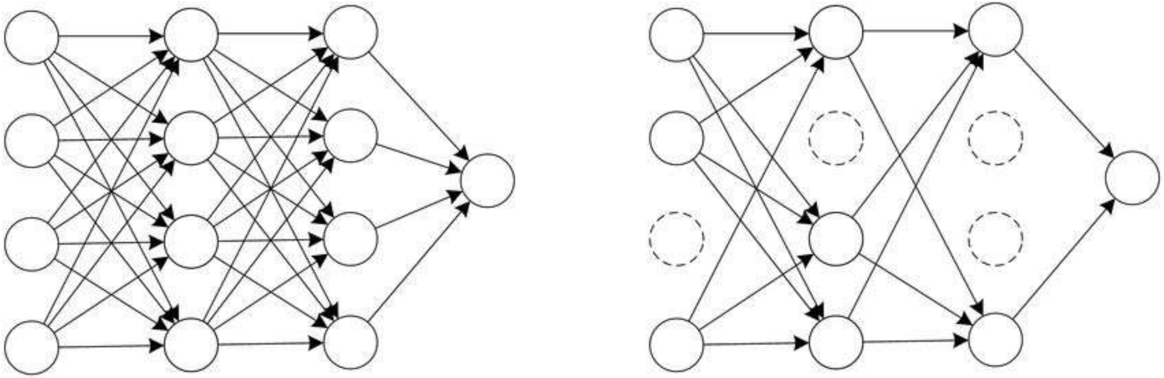
conv block improvements

Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., Salakhutdinov, R., 2014. Dropout: A Simple Way to Prevent Neural Networks from Overfitting. Journal of Machine Learning Research 15, 1929–1958.



tf.keras.layers.Dropout

Image: Khalifa, A.B., Frigui, H., 2016. Multiple Instance Fuzzy Inference Neural Networks.



(a) Standard Neural Network

(b) Network after Dropout

dropout:

randomly forget connections (during the training)

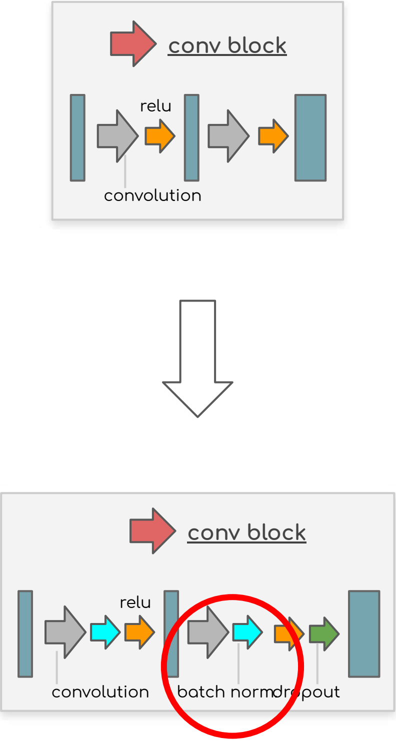
a connection has a probability p of being *dropped out*

why?

it acts as a regularization

conv block improvements

Ioffe, S., Szegedy, C., 2015. Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift.

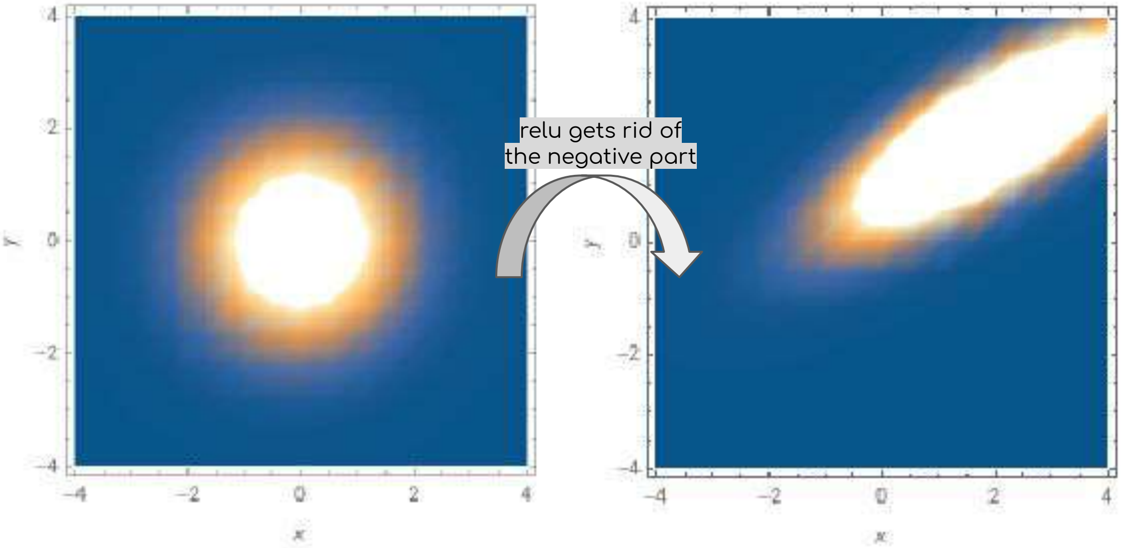


tf.keras.layers.BatchNormalization

distributions of a simplified representation of the images in 2D (only 2 variables)

at the input: centered

after transformations: shifted mean, skewed



Images generated with WOLFRAM Demonstrations Project.

conv block improvements

Ioffe, S., Szegedy, C., 2015. Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift.

batch normalization

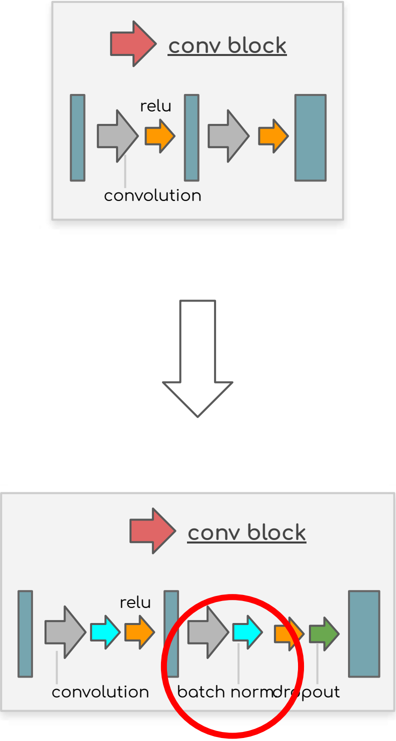
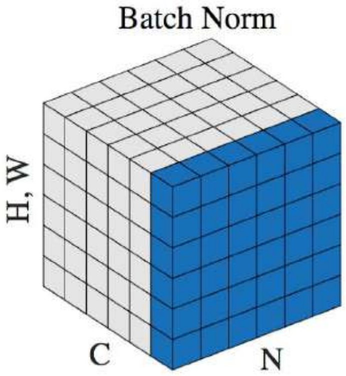


Image source: Wu, Y., He, K., 2018. Group Normalization.



- correct mean shift and scaling
- regularization effect
- smoother landscape

⇒ higher stable *learning rate*

batch mean · batch size · instance in the batch

$$\mu_B = \frac{1}{m}\sum_{i=1}^{m} x_i$$

batch variance

$$\sigma_B^2 = \frac{1}{m}\sum_{i=1}^{m}(x_i - \mu_B)^2$$

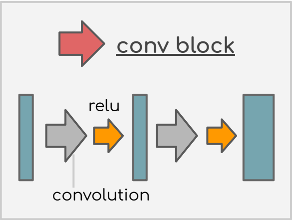
channel

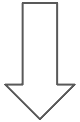
$$\hat{x}_i^{(k)} = \frac{x_i^{(k)} - \mu_B^{(k)}}{\sqrt{\sigma_B^{(k)^2} + \epsilon}}$$

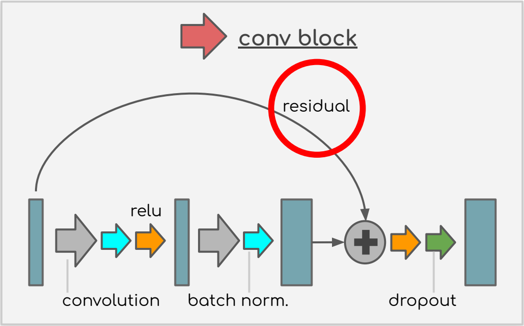
normalized value · instance

conv block improvements

He, K., Zhang, X., Ren, S., Sun, J., 2015.
Deep Residual Learning for Image Recognition.





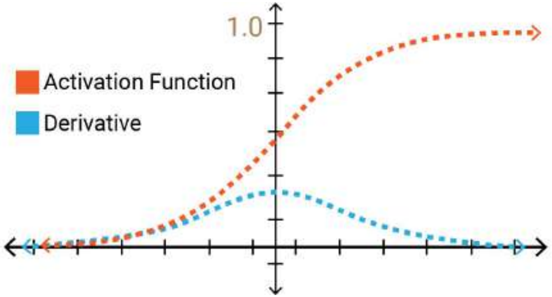


last layer

$$\frac{\partial \mathbf{h}^{(T)}}{\partial \mathbf{h}^{(1)}} = \frac{\partial \mathbf{h}^{(T)}}{\partial \mathbf{h}^{(T-1)}} \cdots \frac{\partial \mathbf{h}^{(2)}}{\partial \mathbf{h}^{(1)}}.$$

first layer

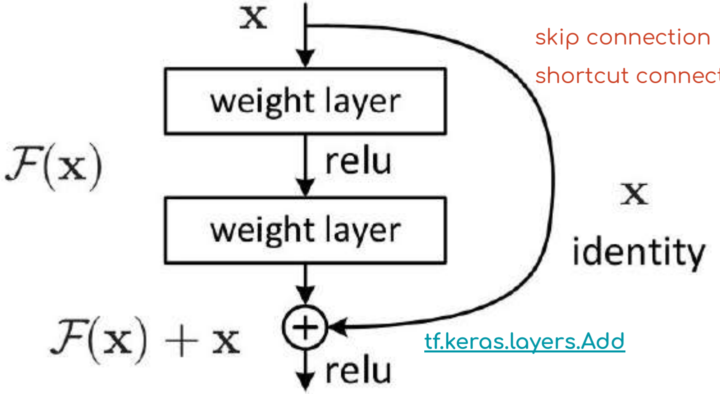
intermediate layers



- speed up the training
- avoid vanishing gradients
- enable deeper networks

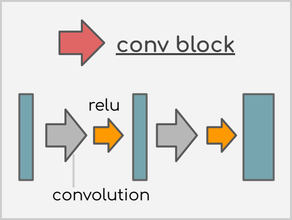
skip connection

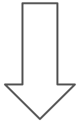
shortcut connection

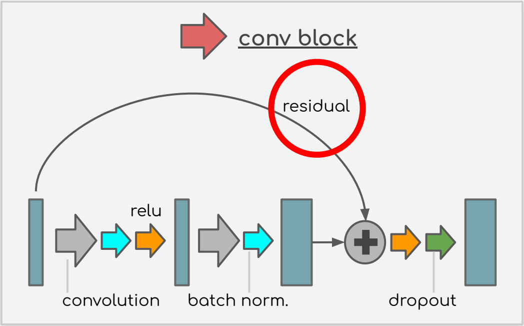


conv block improvements

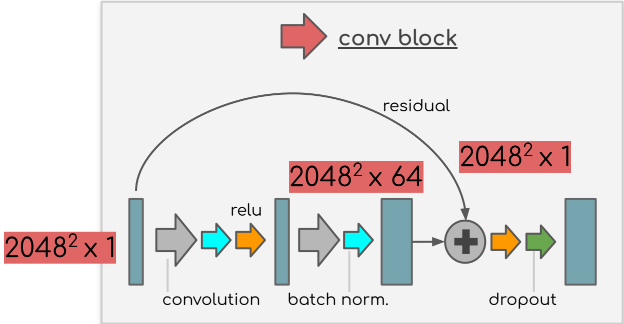
He, K., Zhang, X., Ren, S., Sun, J., 2015.
Deep Residual Learning for Image Recognition.





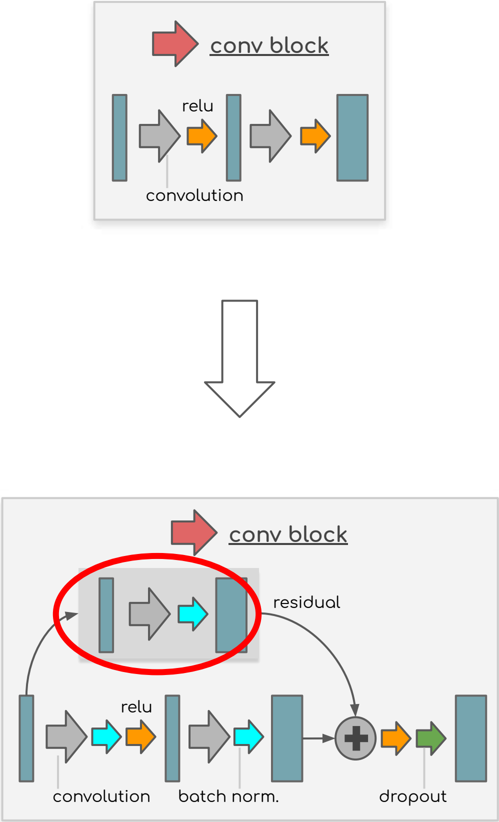


first block:

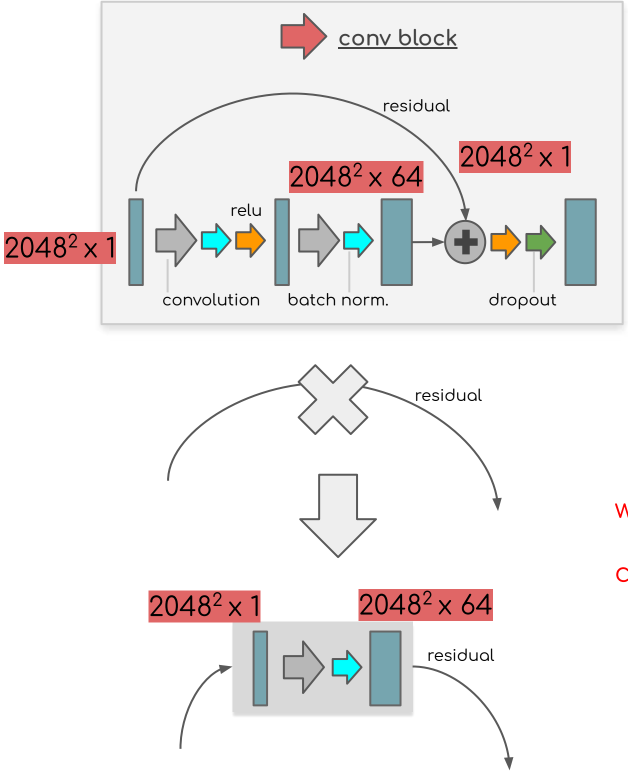


how do you sum tensors of shape

(2048, 2048, 1) and (2048, 2048, 64)?



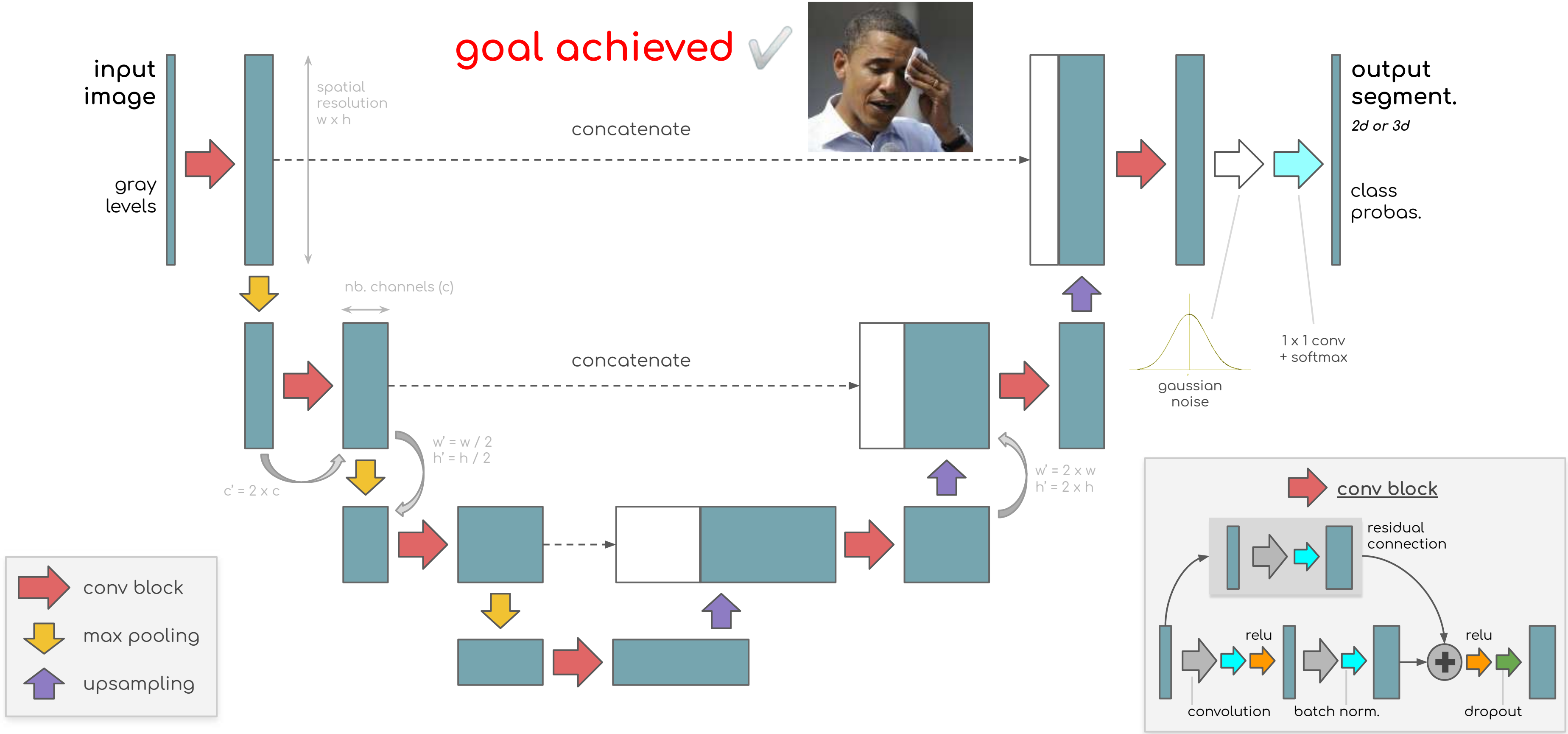
first block:



with another

convolution (:

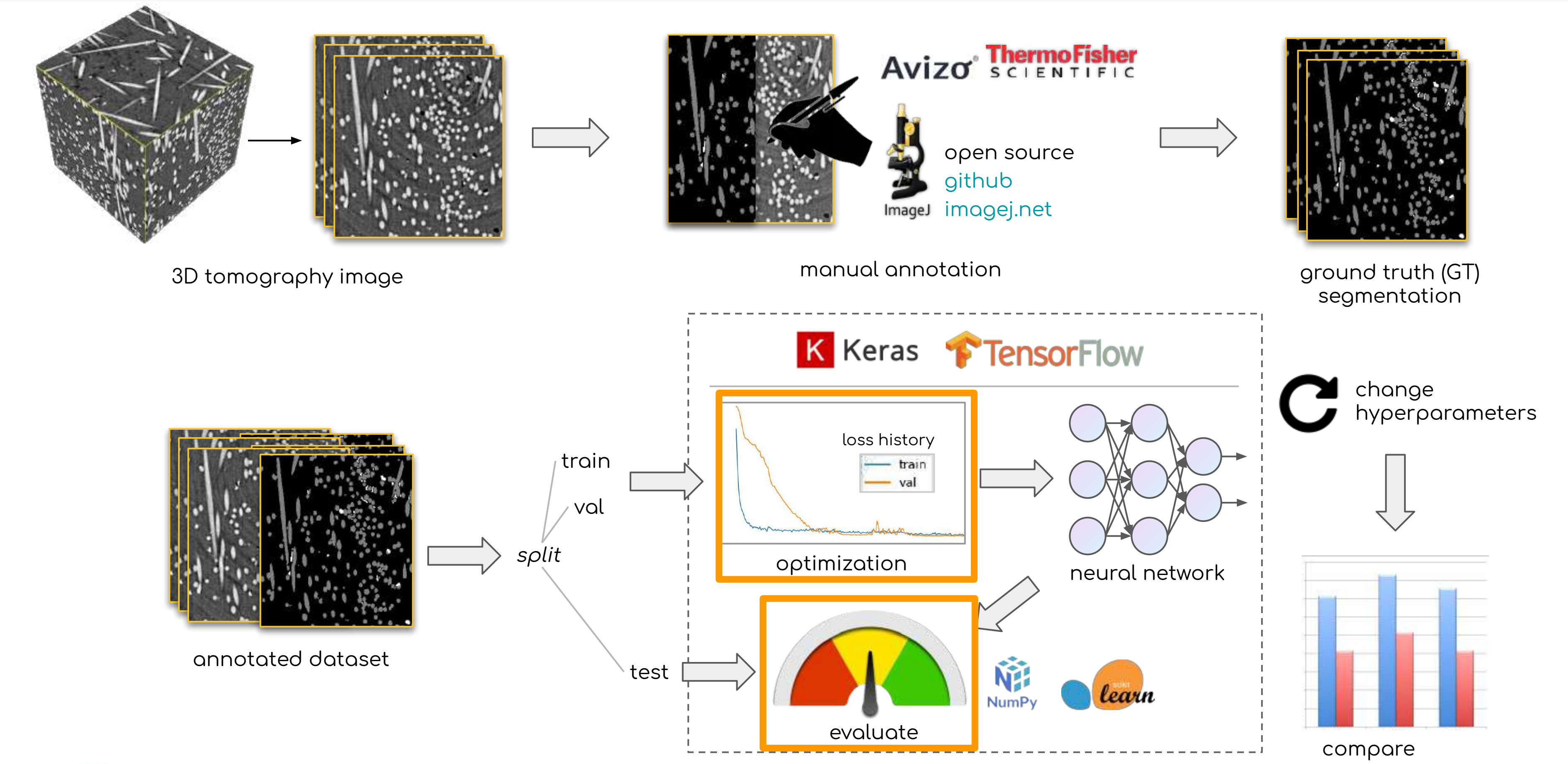
(improved) u-net



evaluation & optimization

overview

project repository: github.com/joaopcbertoldo/tomo2seg



ground truth vs. prediction

making the prediction comparable with the ground truth

$$y \in \{\text{matrix}, \text{fiber}, \text{porosity}\}$$

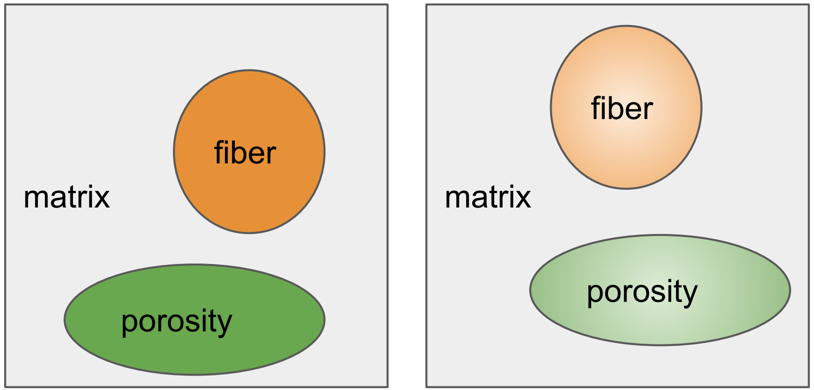
<u>for each voxel:</u>

ground truth (y) $\quad y = \text{porosity}$

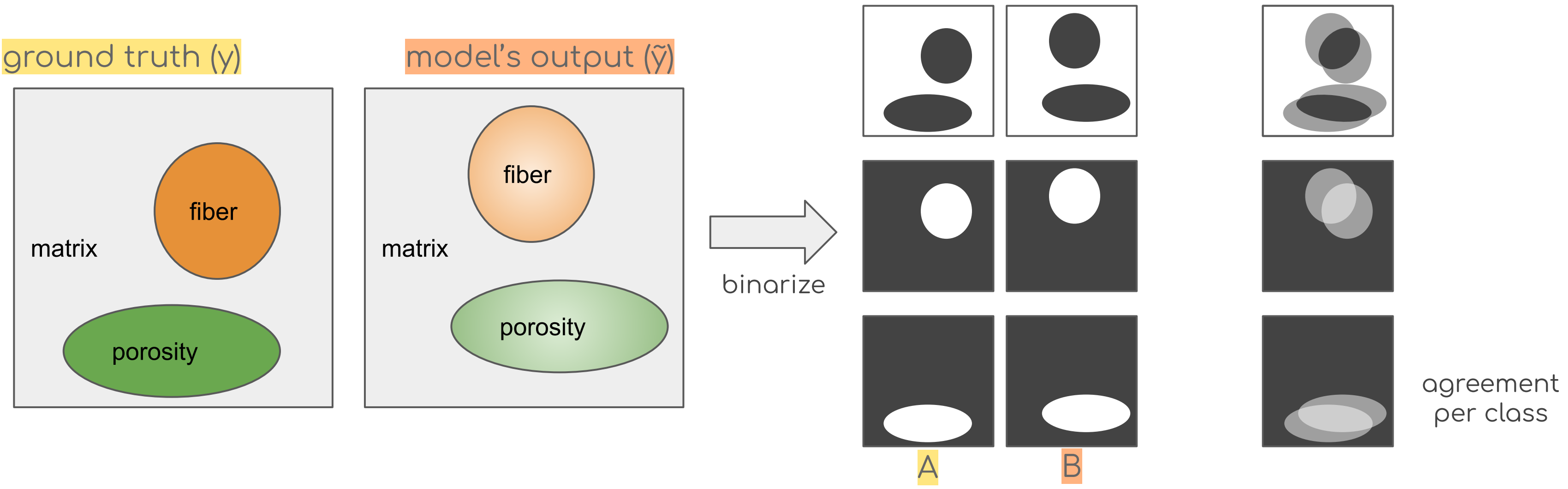
model's output (ỹ) $\quad \tilde{y} = [.10, .10, .80] \xrightarrow{\text{argmax}} \tilde{y} = \text{porosity}$

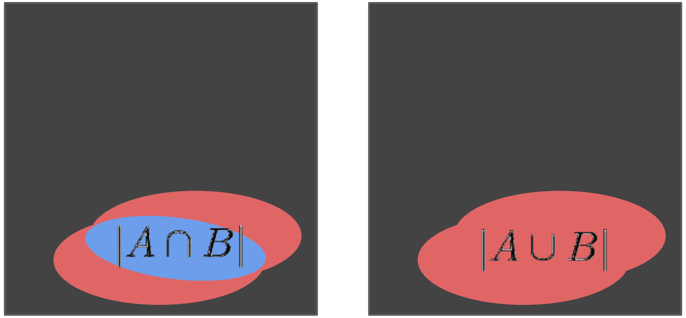
$$\tilde{y} \in [0, 1]^3$$

ground truth (y) | model's output (ỹ)



ground truth vs. prediction



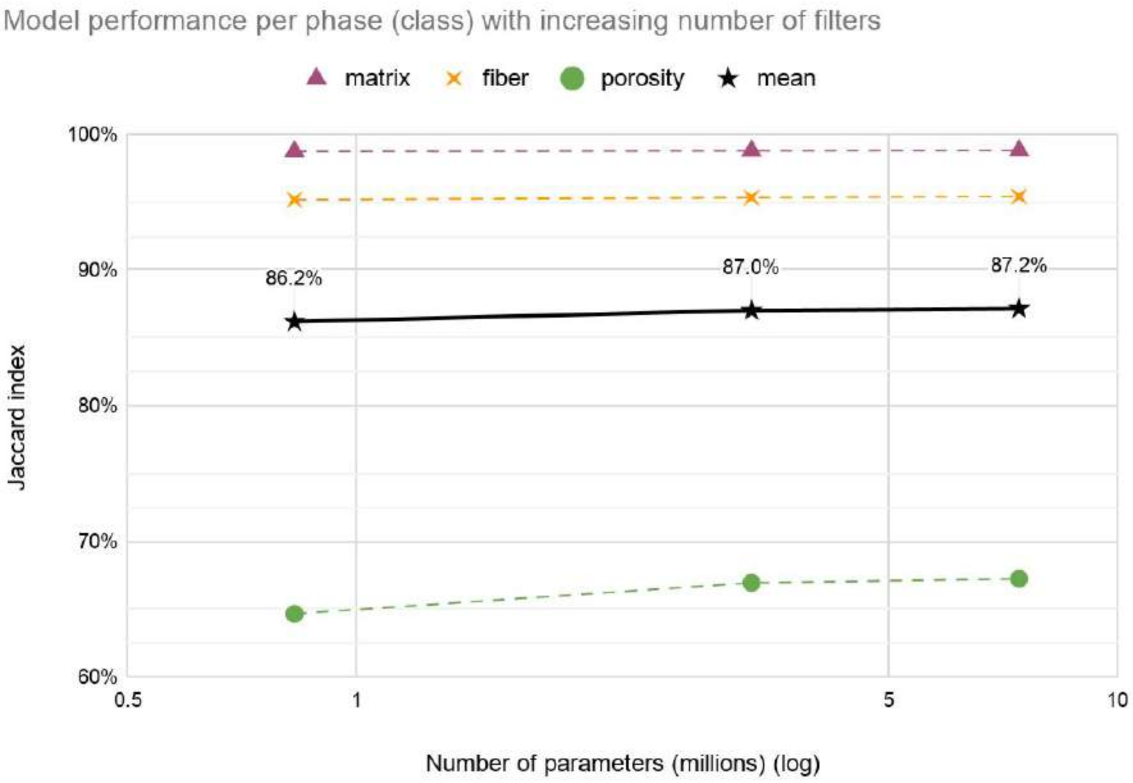


<u>for each phase (class):</u>

$$J = \frac{|A \cap B|}{|A \cup B|} = \frac{|A \cap B|}{|A| + |B| - |A \cap B|}$$

jaccard index

some results



ground truth vs. prediction

making the ground truth comparable with the prediction

<u>for each voxel:</u>

ground truth (y) $y = \text{porosity}$ one hot encoding

	matrix k=1	fiber k=2	porosity k=3
one hot encoding	0	0	1
model's output ($\tilde{y}$) =	.10	.10	.80

model's output ($\tilde{y}$) $\tilde{y} = [.10, .10, .80]$

jaccard[2]: a generalization of jaccard index $J = \dfrac{|A \cap B|}{|A| + |B| - |A \cap B|}$

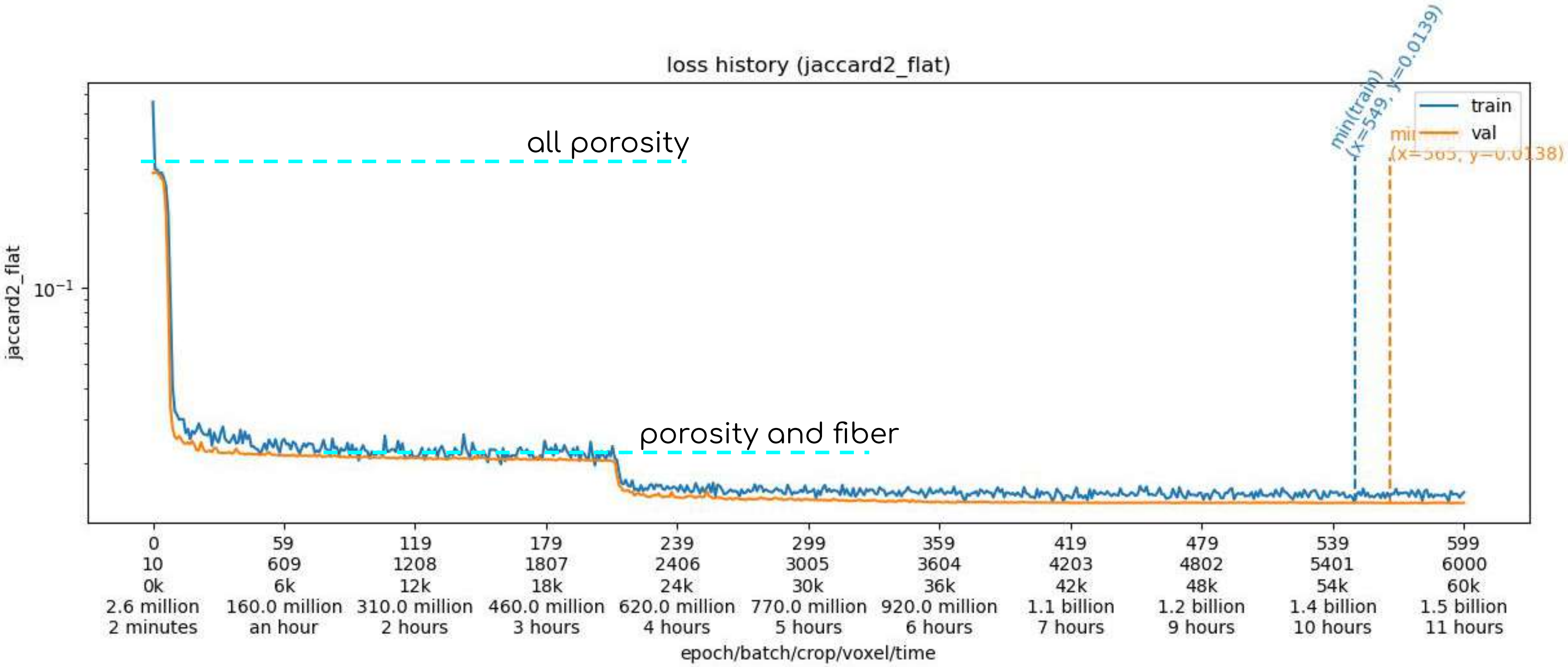
$$I = \sum_{k=1}^{K} y_k \, \tilde{y}_k$$

$$J_2 = \frac{I}{\sum_{k=1}^{K} y_k^2 + \sum_{k=1}^{K} \tilde{y}_k^2 - I}$$

Shouldn't it be a <u>loss</u>?

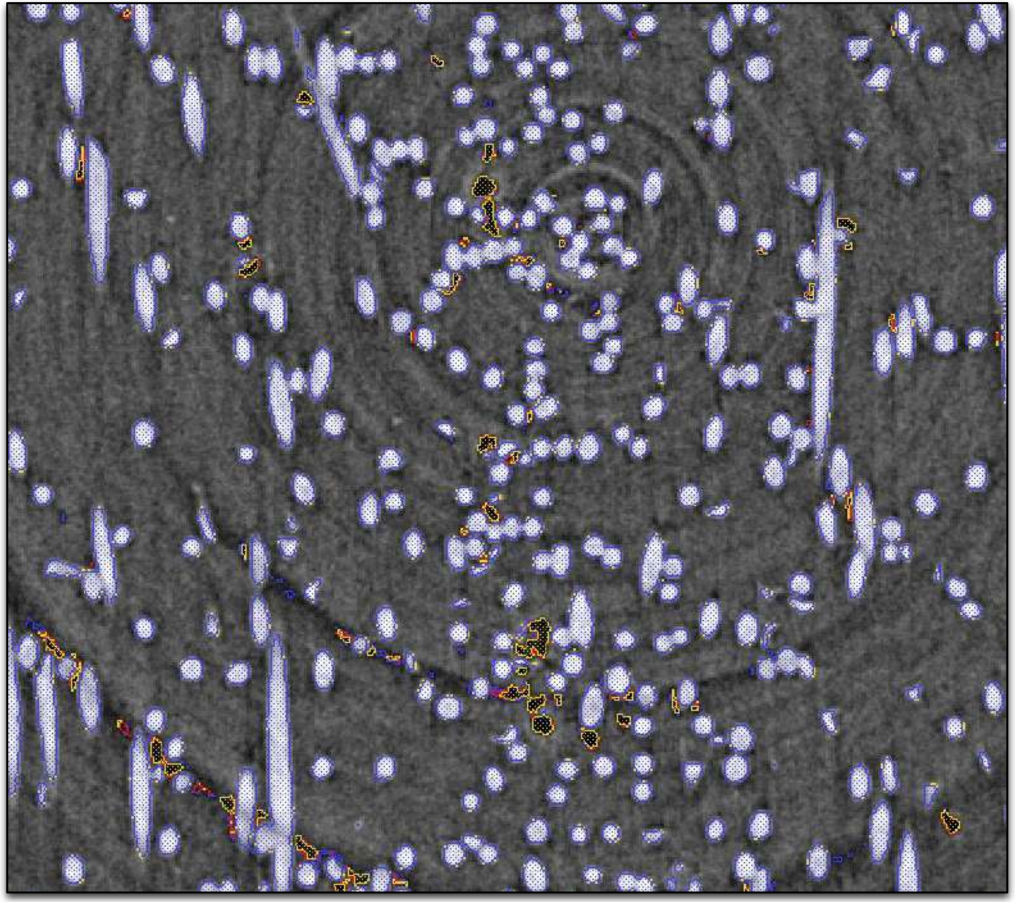
$$\text{loss} = 1 - J_2$$

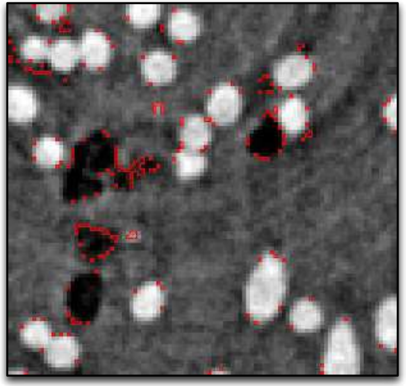
training

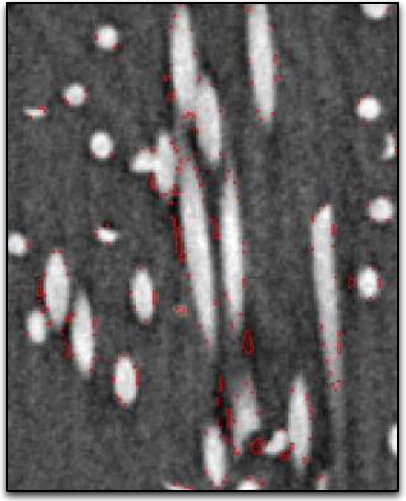


loss history (jaccard2_flat)
jaccard2_flat
10^{-1}
all porosity
porosity and fiber
train
val
min(train)
(x=549, y=0.0139)
(x=565, y=0.0138)
0 10 0k 2.6 million 2 minutes
59 609 6k 160.0 million an hour
119 1208 12k 310.0 million 2 hours
179 1807 18k 460.0 million 3 hours
239 2406 24k 620.0 million 4 hours
299 3005 30k 770.0 million 5 hours
359 3604 36k 920.0 million 6 hours
419 4203 42k 1.1 billion 7 hours
479 4802 48k 1.2 billion 9 hours
539 5401 54k 1.4 billion 10 hours
599 6000 60k 1.5 billion 11 hours
epoch/batch/crop/voxel/time

results



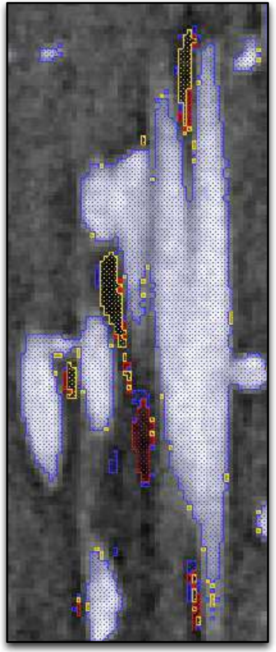




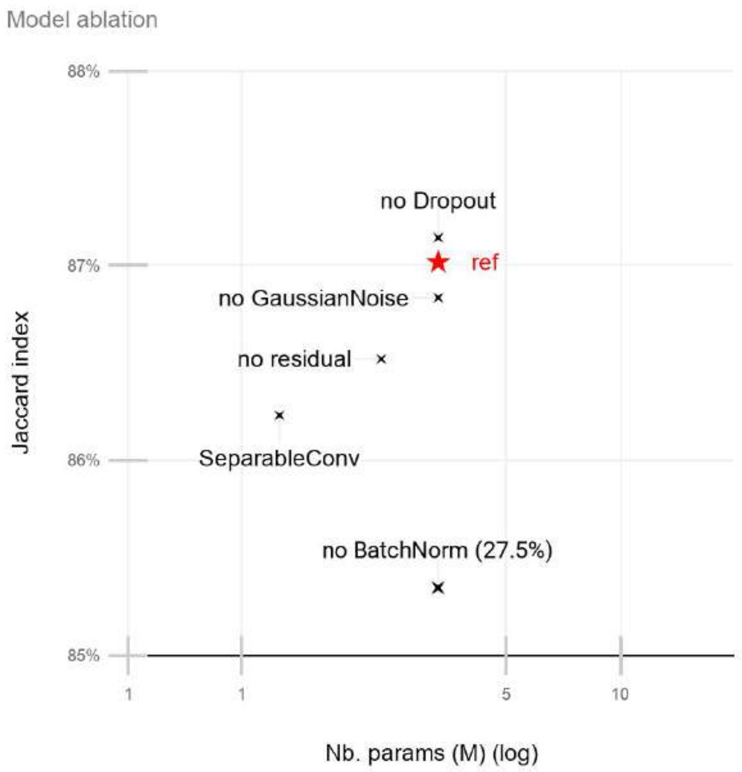
blue: glass fibers

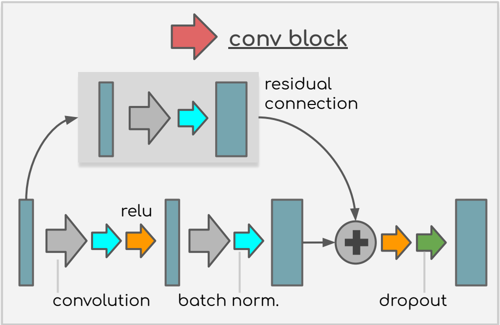
yellow: porosities

red: errors



ablation study





conclusion

this presentation

goals

- put together pieces learned during the week
- showcase an application example

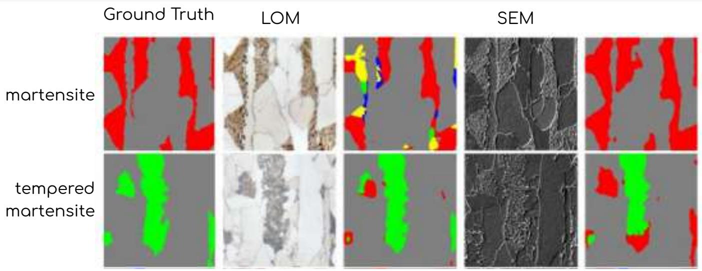
focus

- a neural network architecture and its components

context

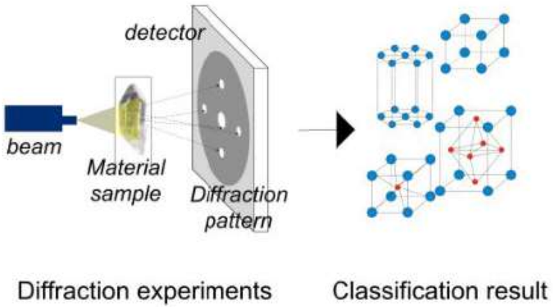
- the (machine) learning problem
- data provenance

other (materials science) ∩ (deep learning)



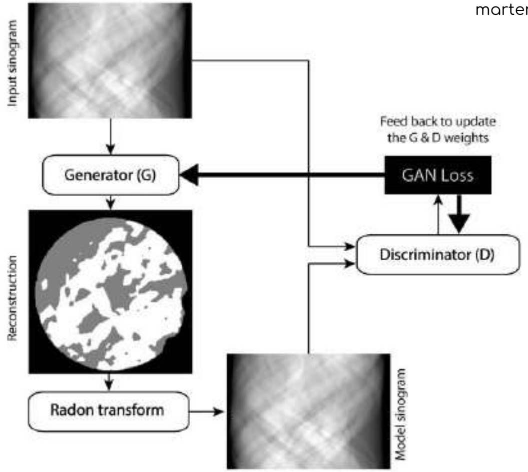
steel microstructural segm.

Azimi, S.M., Britz, D., Engstler, M., Fritz, M., Mücklich, F., 2018. Advanced Steel Microstructural Classification by Deep Learning Methods.



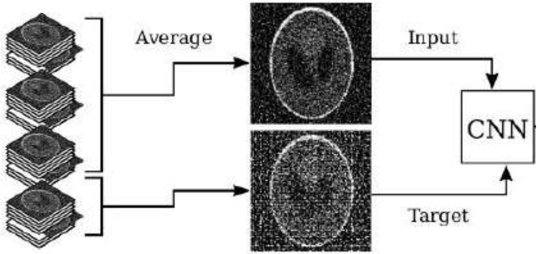
diffraction pattern classif.

Tiong, L.C.O., Kim, J., Han, S.S., Kim, D., 2020. Identification of Crystal Symmetry from Noisy Diffraction Patterns by A Shape Analysis and Deep Learning.



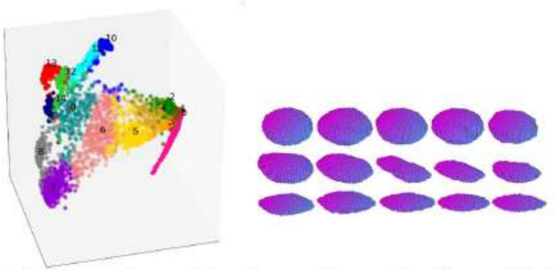
tomography reconstruction

Yang, X., Kahnt, M., Brueckner, D., Schropp, A., Fam, Y., Becher, J., Grunwaldt, J., Sheppard, T.L., Schroer, C., 2020. Tomographic reconstruction with a generative adversarial network. Journal of synchrotron radiation.



tomography denoising

Hendriksen, A.A., Pelt, D.M., Batenburg, K.J., 2020. Noise2Inverse: Self-supervised deep convolutional denoising for tomography. IEEE Trans. Comput. Imaging 6, 1320–1335.



K-medoids clustering after augmentation on Grassmann distances matrix and the representation of the medoids in the same order as on the clustering

pore morphology clustering

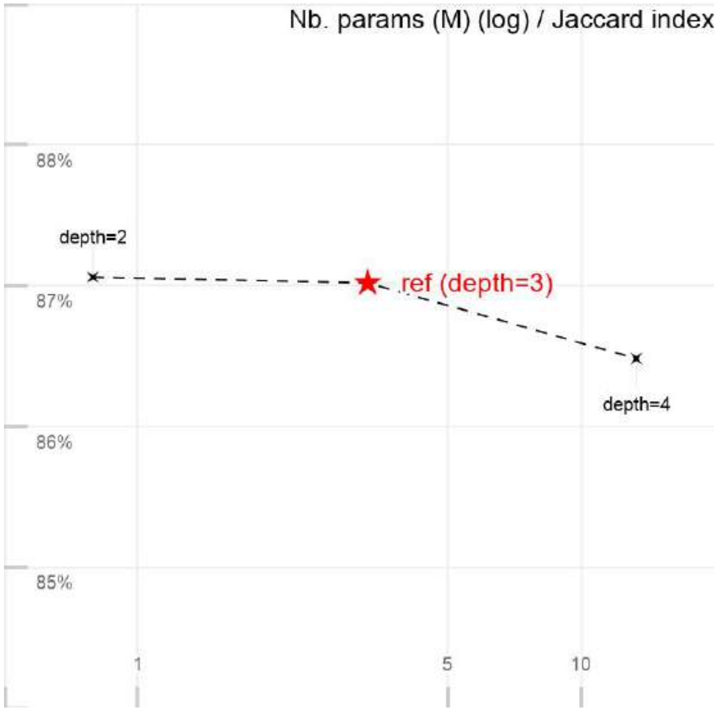
Launay, H.

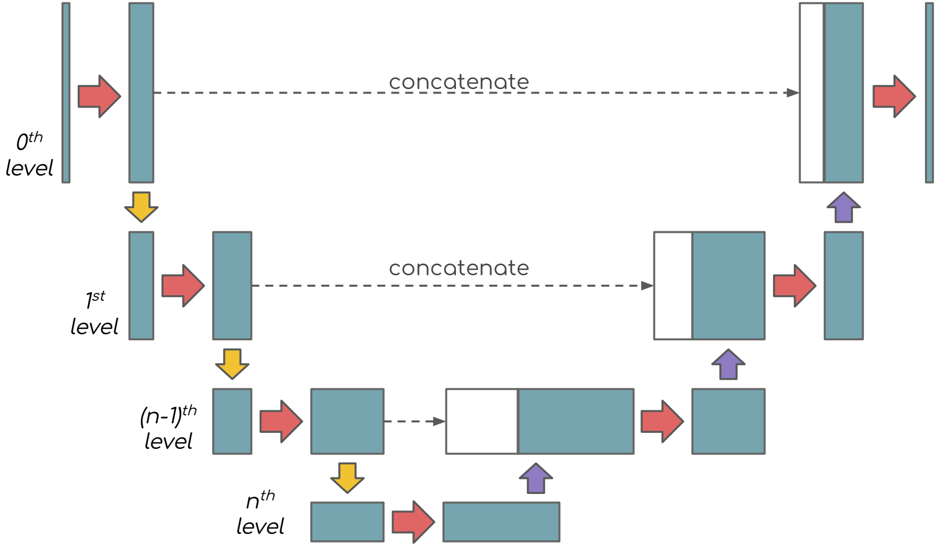
thank you for your attention!

Computer vision and machine learning
for the material scientist (CVML)
28/02/2021

João P C Bertoldo
joao.bertoldo@mines-paristech.fr
joaopcbertoldo@gmail.com

extras





link to the chart

mirror padding

3	5	1
3	6	1
4	7	9

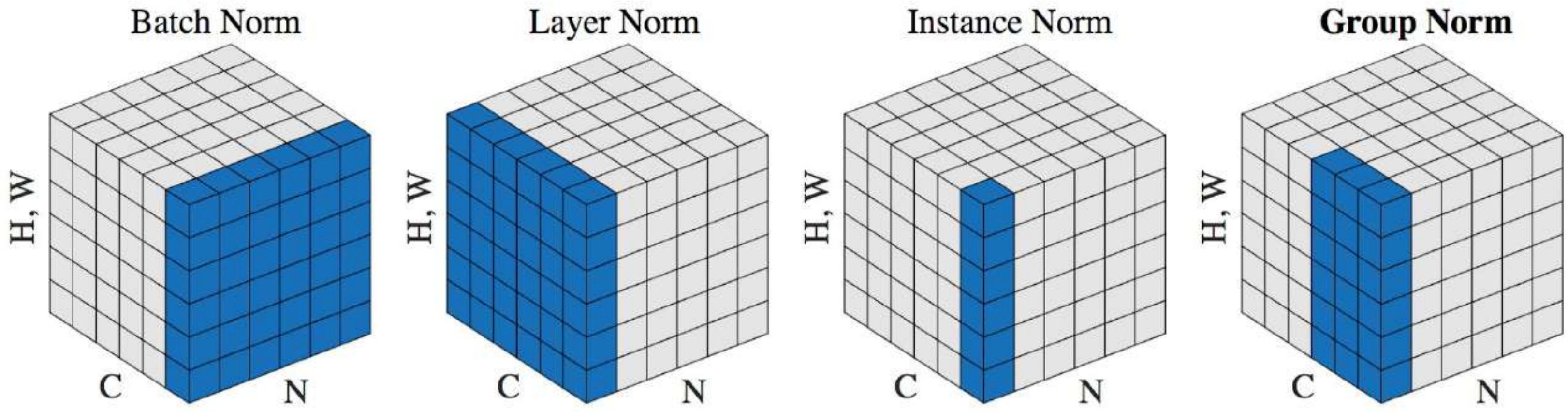
No padding

1	6	3	6	1	6	3
1	5	3	5	1	5	3
1	6	3	6	1	6	3
9	7	4	7	9	7	4
1	6	3	6	1	6	3

(1, 2) reflection padding

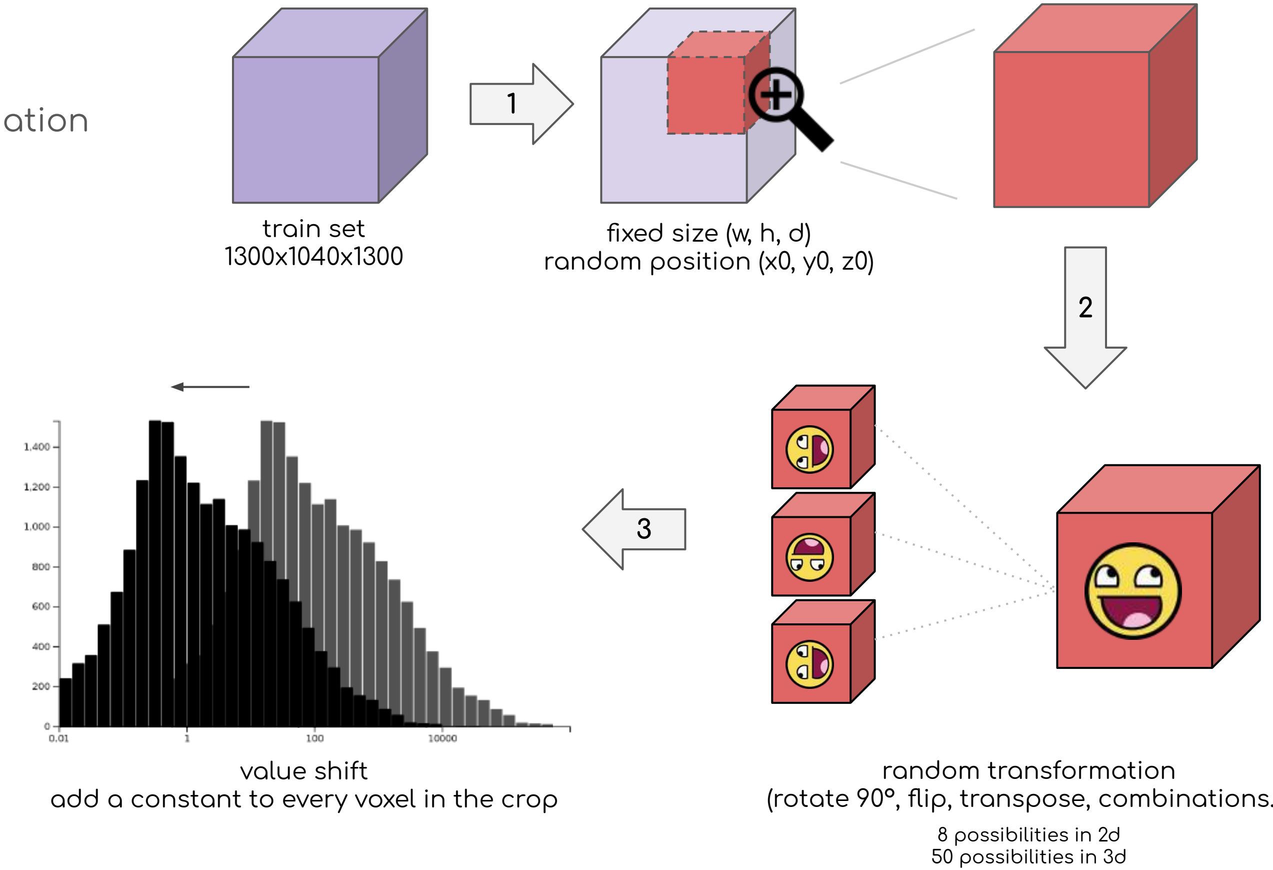
Credits: Christian Versloot.
Source: [Using Constant Padding, Reflection Padding and Replication Padding with TensorFlow and Keras]

other feature normalizations

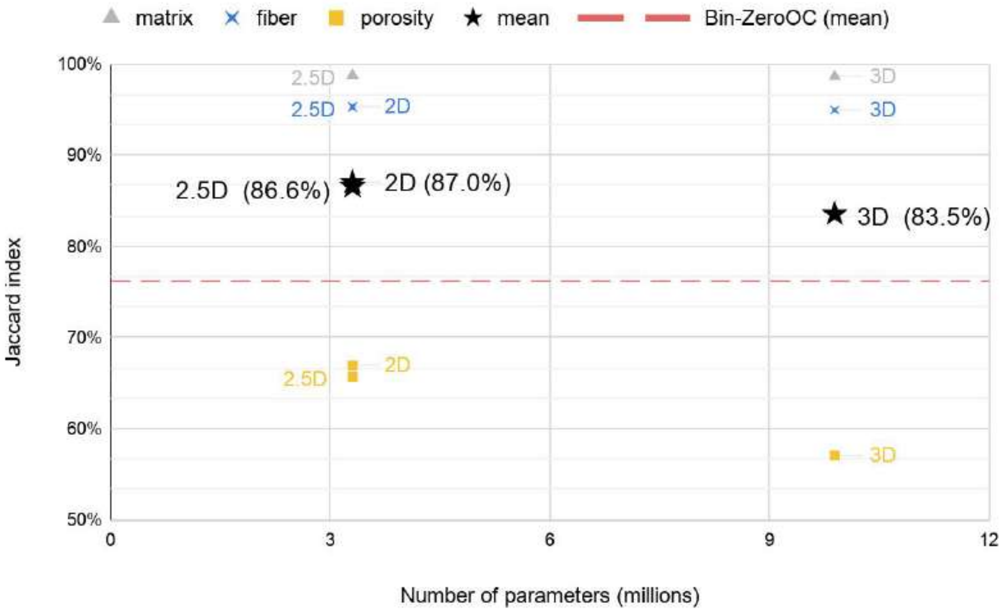


data augmentation

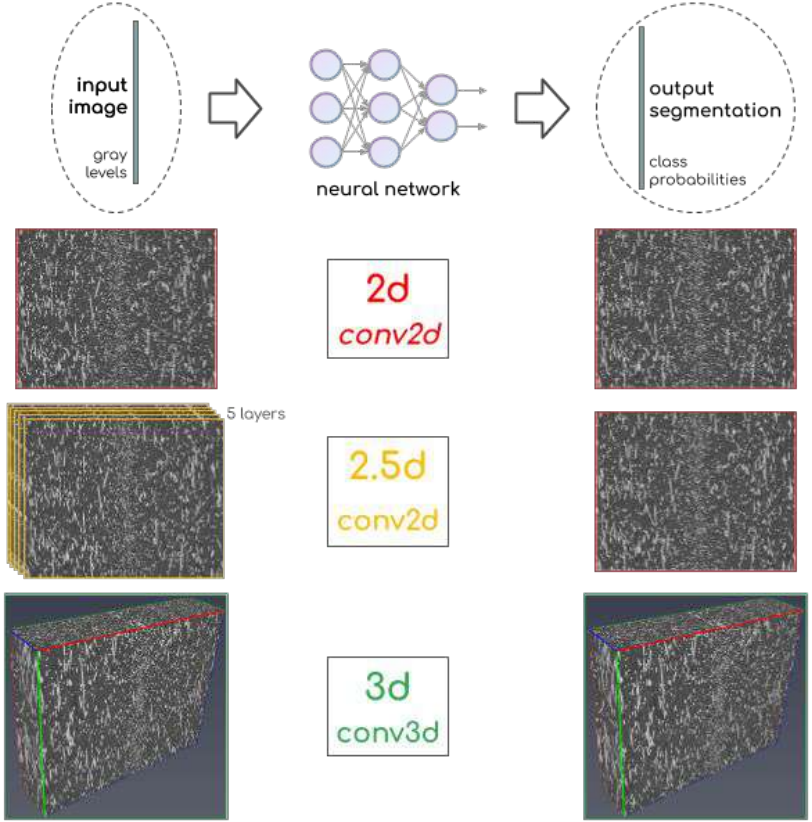
1. random 3D crop
2. geometric transformation
3. value shift



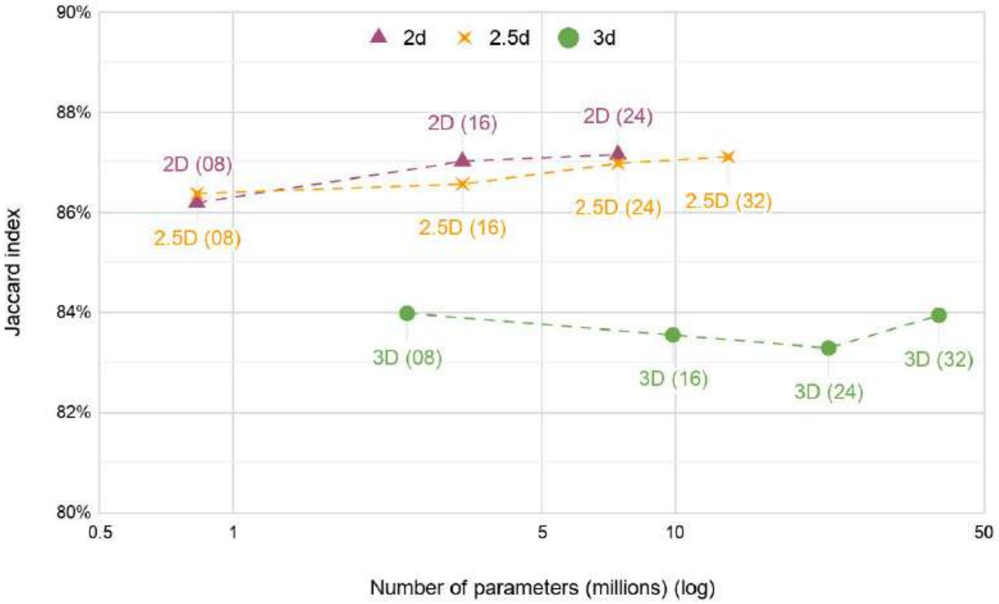
2d vs. 2.5d vs. 3d



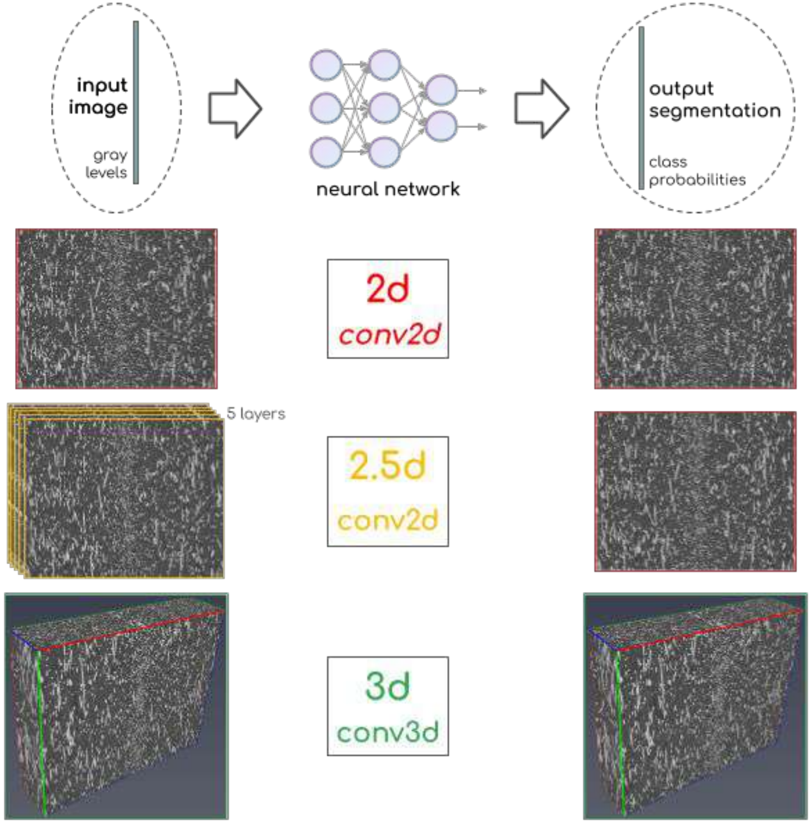
link to the chart



2d vs. 2.5d vs. 3d

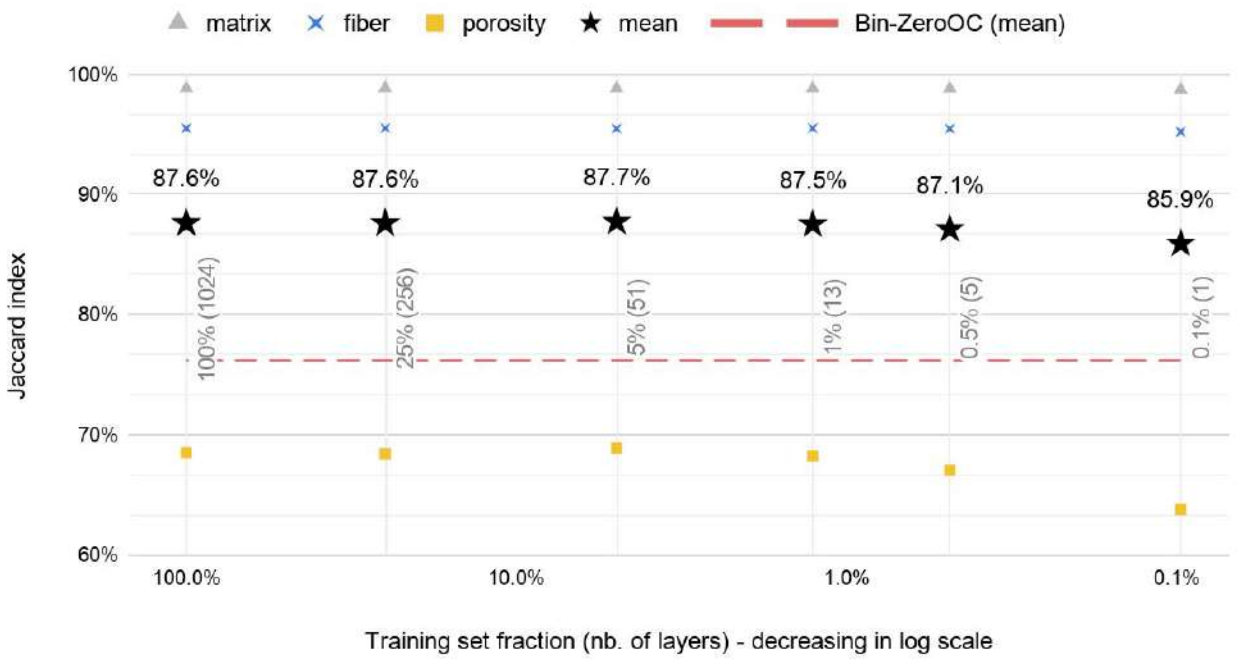


link to the chart



learning curve

link to the chart



"In many biomedical applications, only very few images are required to train a network that generalizes reasonably well."

Çiçek, Ö., Abdulkadir, A., Lienkamp, S., Brox, T., Ronneberger, O., 2016.
3D U-Net: Learning Dense Volumetric Segmentation from Sparse Annotation.

dropout breaks co-adaptations

*"In a standard neural network, **the derivative** received by each parameter **tells it how it should change** so the final loss function is reduced, **given what all other units are doing.***

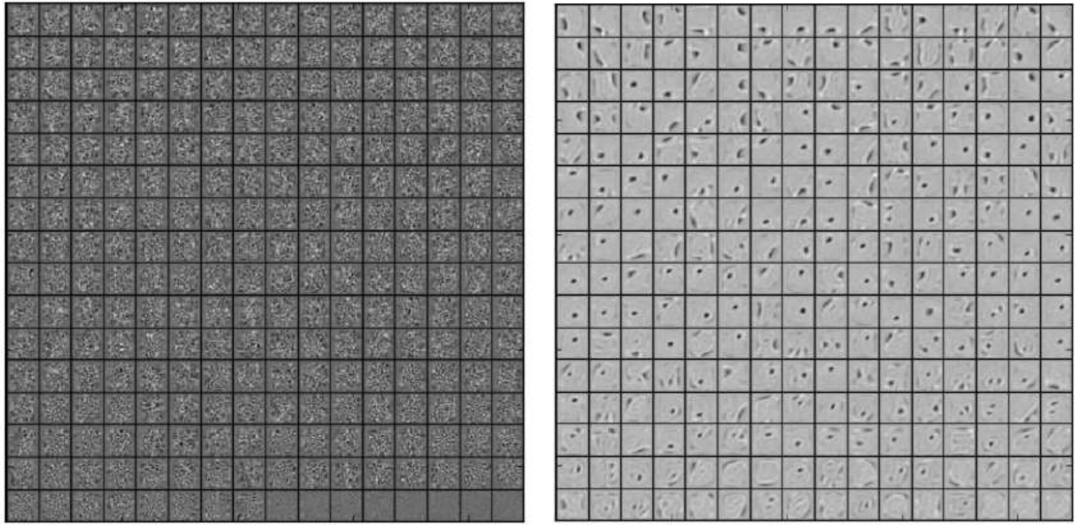
*Therefore, units may **change in a way that they fix up the mistakes of the other units.***

*This may lead to **complex co-adaptations.** This in turn leads to overfitting because these co-adaptations do not generalize to unseen data. We hypothesize that for each hidden unit, **dropout prevents co-adaptation by making the presence of other hidden units unreliable.***

*Therefore, **a hidden unit cannot rely on other specific units** to correct its mistakes. **It must perform well in a wide variety of different contexts** provided by the other hidden units.*

...

*This shows that **dropout does break up co-adaptations,** which is probably the main reason why it leads to lower generalization errors."*



(a) Without dropout

(b) Dropout with $p = 0.5$.

Figure 7: Features learned on MNIST with one hidden layer autoencoders having 256 rectified linear units.

Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., Salakhutdinov, R., 2014. Dropout: A Simple Way to Prevent Neural Networks from Overfitting. Journal of Machine Learning Research 15, 1929–1958.

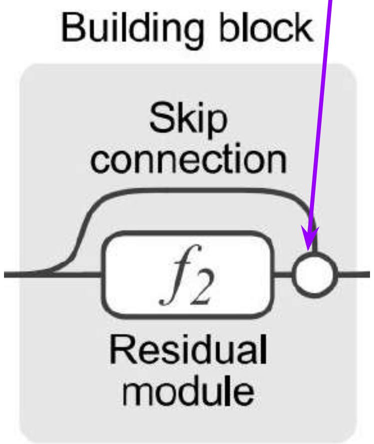
residual networks behave like ensembles

In this work we propose a novel interpretation of ***residual networks*** *showing that they* ***can be seen as a collection of many paths of differing length.*** *Moreover, residual networks seem to enable very deep networks by leveraging only the short paths during training. To support this observation, we rewrite residual networks as an explicit collection of paths. Unlike traditional models, paths through residual networks vary in length. Further,* ***a lesion study reveals that these paths show ensemble-like behavior in the sense that they do not strongly depend on each other****. Finally, and most surprising, most paths are shorter than one might expect, and only the short paths are needed during training, as longer paths do not contribute any gradient. For example, most of the gradient in a residual network with 110 layers comes from paths that are only 10-34 layers deep. Our results reveal one of the key characteristics that seem to enable the training of very deep networks: Residual networks avoid the vanishing gradient problem by introducing short paths which can carry gradient throughout the extent of very deep networks.*

Veit, A., Wilber, M., Belongie, S., 2016. Residual networks behave like ensembles of relatively shallow networks, in: Proceedings of the 30th International Conference on Neural Information Processing Systems, NIPS'16. Curran Associates Inc., Red Hook, NY, USA, pp. 550–558.

current tensor current layer previous tensor

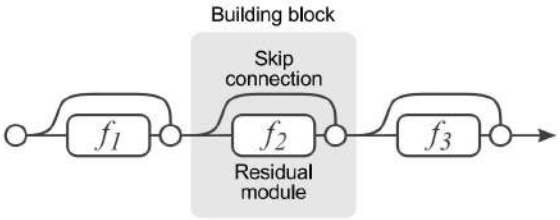
$$y_i \equiv f_i(y_{i-1}) + y_{i-1}$$



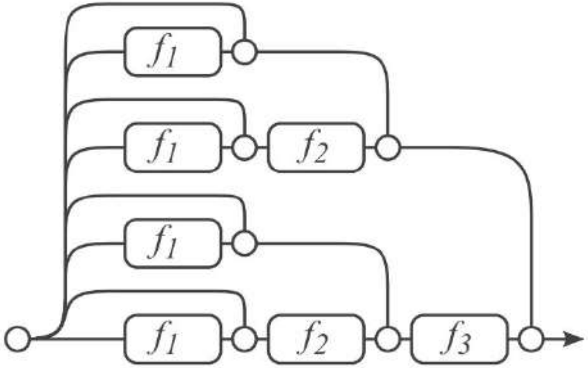
residual networks behave like ensembles

*In this work we propose a novel interpretation of **residual networks** showing that they **can be seen as a collection of many paths of differing length.** Moreover, residual networks seem to enable very deep networks by leveraging only the short paths during training. To support this observation, we rewrite residual networks as an explicit collection of paths. Unlike traditional models, paths through residual networks vary in length. Further, **a lesion study reveals that these paths show ensemble-like behavior in the sense that they do not strongly depend on each other**. Finally, and most surprising, most paths are shorter than one might expect, and only the short paths are needed during training, as longer paths do not contribute any gradient. For example, most of the gradient in a residual network with 110 layers comes from paths that are only 10-34 layers deep. Our results reveal one of the key characteristics that seem to enable the training of very deep networks: Residual networks avoid the vanishing gradient problem by introducing short paths which can carry gradient throughout the extent of very deep networks.*

Veit, A., Wilber, M., Belongie, S., 2016. Residual networks behave like ensembles of relatively shallow networks, in: Proceedings of the 30th International Conference on Neural Information Processing Systems, NIPS'16. Curran Associates Inc., Red Hook, NY, USA, pp. 550–558.



$$
\begin{aligned}
y_3 &= y_2 + f_3(y_2) \\
&= [y_1 + f_2(y_1)] \;+\; f_3(y_1 + f_2(y_1)) \\
&= \big[y_0 + f_1(y_0) + f_2(y_0 + f_1(y_0))\big] \;+\; f_3\big(y_0 + f_1(y_0) + f_2(y_0 + f_1(y_0))\big)
\end{aligned}
$$

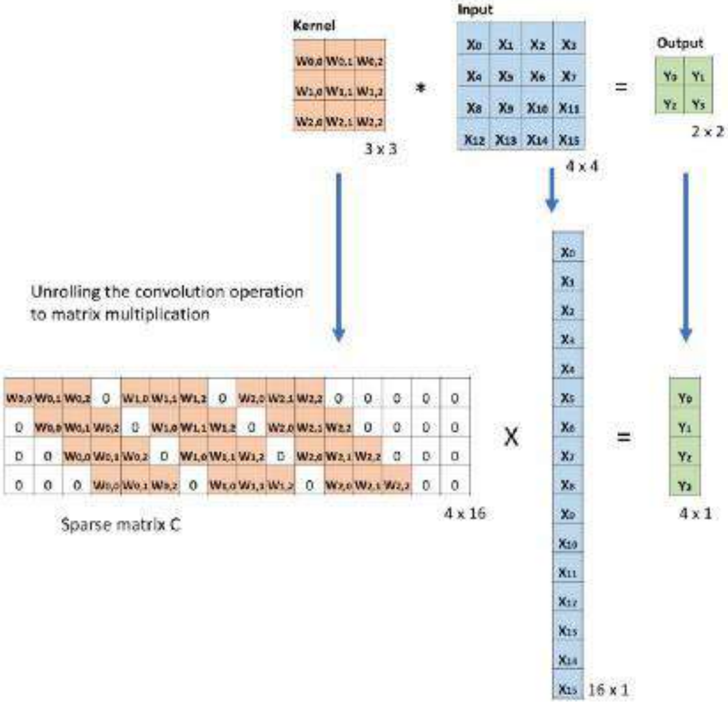


(b) Unraveled view of (a)

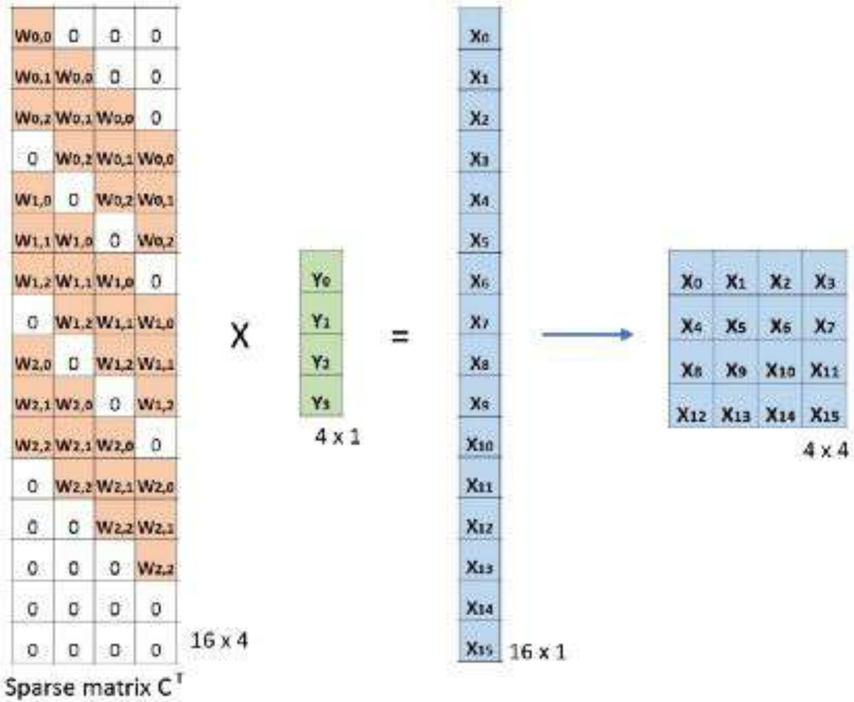
transposed conv2d

Figures ref: [A Comprehensive Introduction to Different Types of Convolutions in Deep Learning](), [Kunlun Bai]()
Tnx, Kunlun!

conv 2d



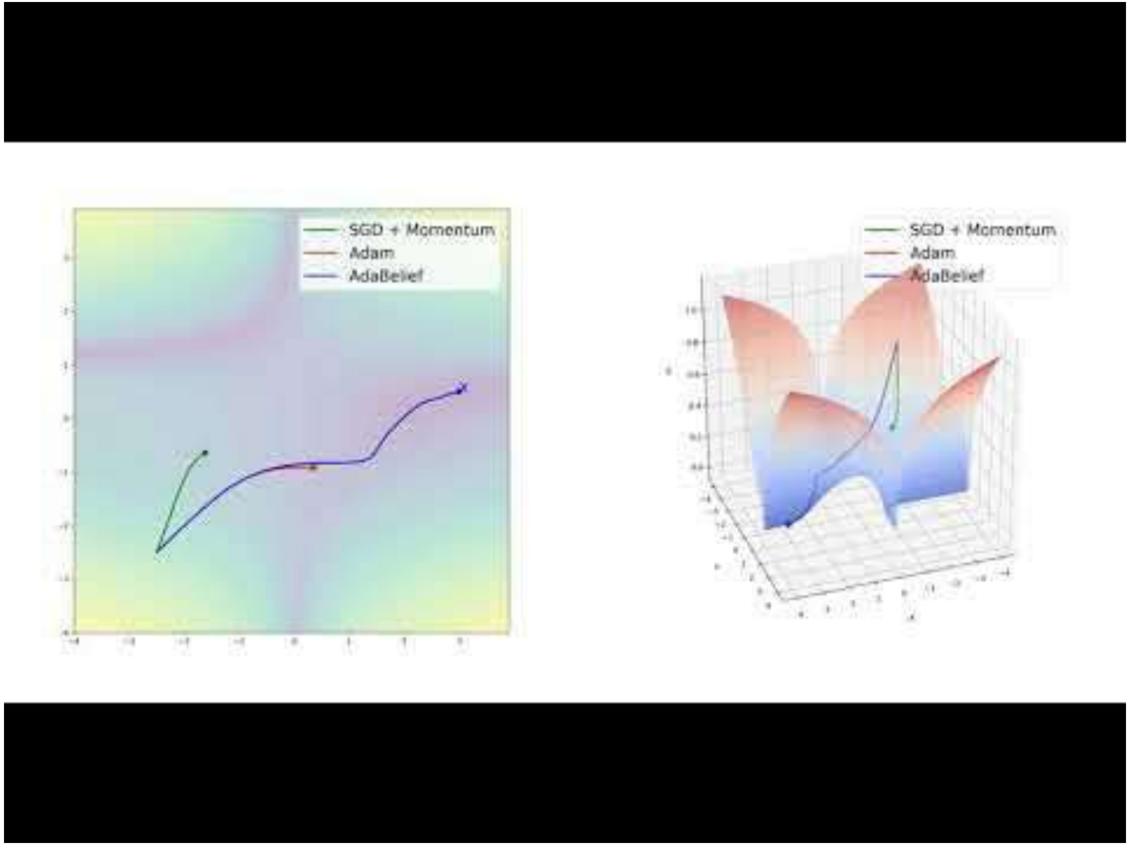
transposed conv 2d



adabelief

link to the video

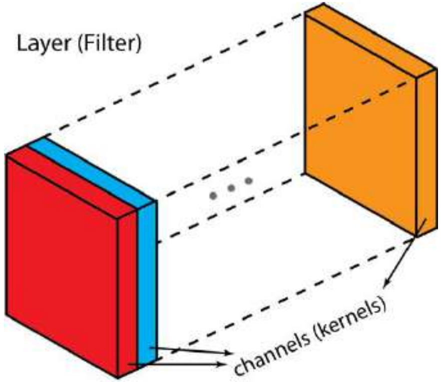
credits: Juntang Zhuang



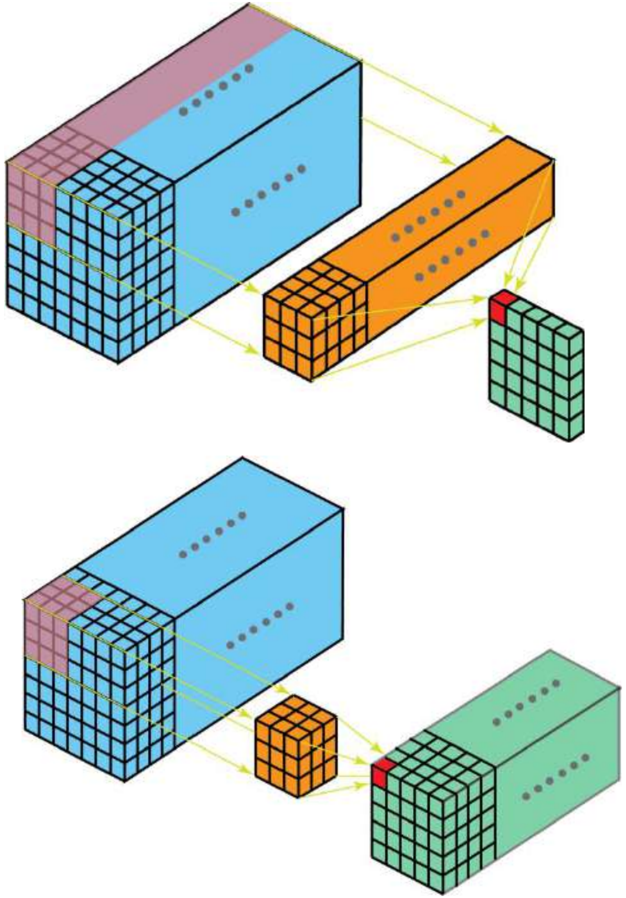
> "The intuition for AdaBelief is to adapt the stepsize according to the "belief" in the current gradient direction."
>
> *J. Zhuang et al., "AdaBelief Optimizer: Adapting Stepsizes by the Belief in Observed Gradients," arXiv:2010.07468 [cs, stat], Oct. 2020, Accessed: Oct. 27, 2020. [Online]. Available: http://arxiv.org/abs/2010.07468.*

conv3d

Figures ref: [A Comprehensive Introduction to Different Types of Convolutions in Deep Learning](), [Kunlun Bai]()
Tnx, Kunlun!

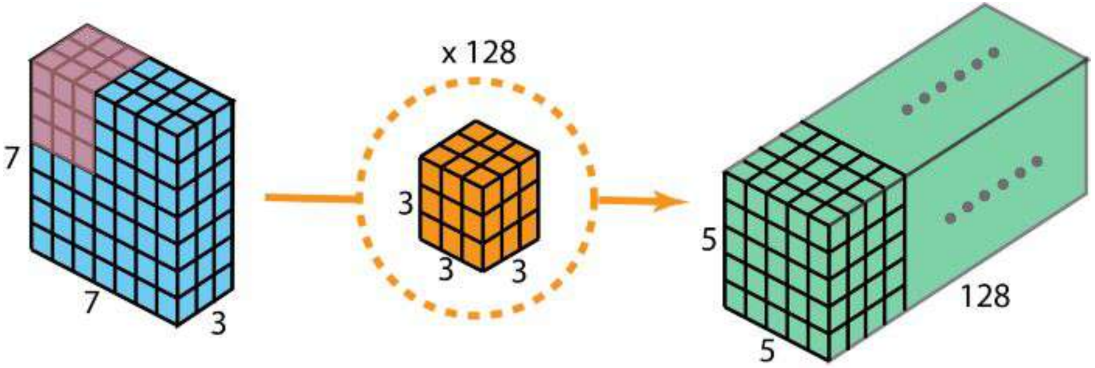


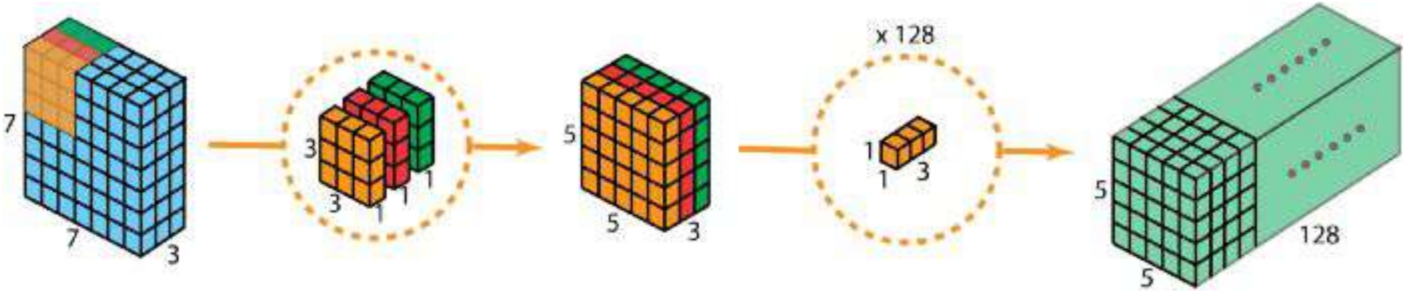
the 4th dimension now!

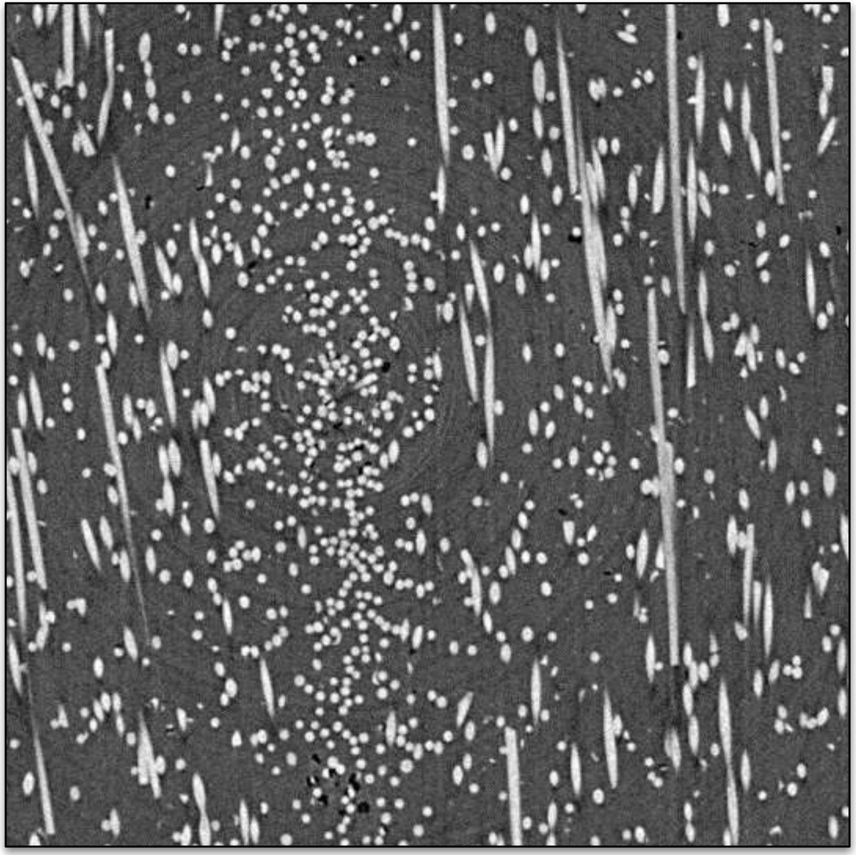


separable conv2d

Figures ref: [A Comprehensive Introduction to Different Types of Convolutions in Deep Learning](), [Kunlun Bai]()
Tnx, Kunlun!







PA66GF30
PolyAmide 66
reinforced with glass fibers
(zoom)

link to the video

A few things to notice:

- *ring artifacts*
- *fibers in different directions*
- *blurred fiber-matrix interfaces*
- *porosities close to fibers*