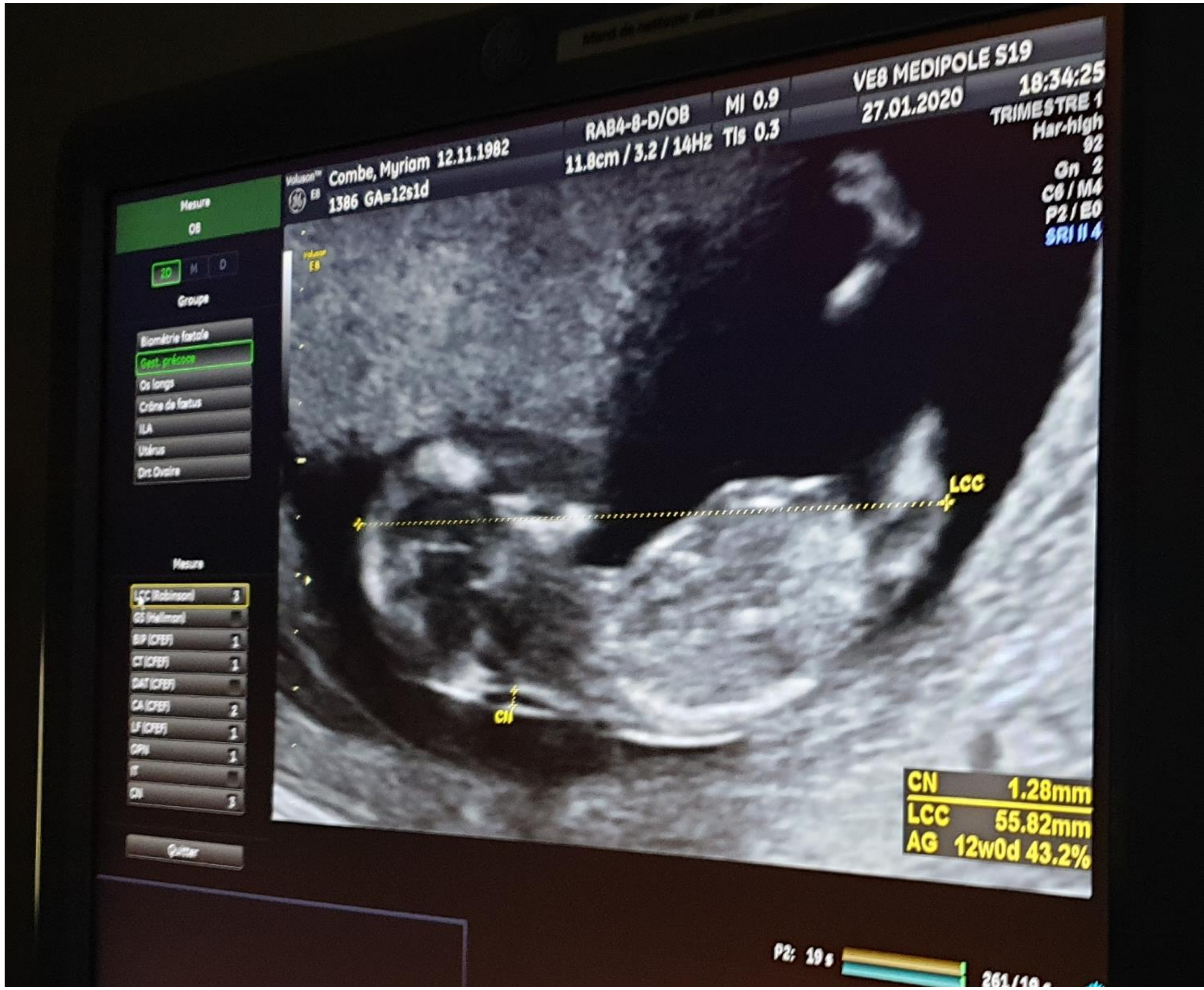
Introduction to Unet, application to **image** segmentation

Thomas Grenier

CREATIS; CNRS (UMR 5220); INSERM (U1206); INSA Lyon; Université de Lyon, France

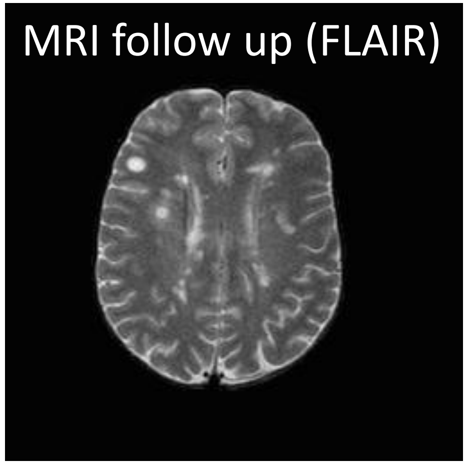
thomas.grenier@creatis.insa-lyon.fr

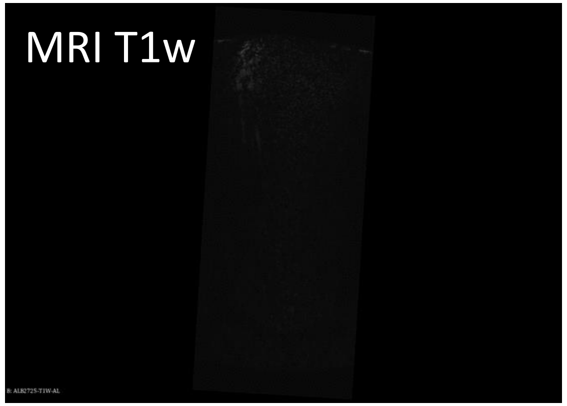

VE8 MEDIPOLE S19
18:34:25
27.01.2020
RAB4-8-D/OB
MI 0.9
11.8cm / 3.2 / 14Hz
TIs 0.3
Combe, Myriam 12.11.1982
1386 GA=12s1d
TRIMESTRE 1
Har-high
92
Gn 2
C6 / M4
P2 / E0
SRI II 4
Mesure
OB
2D M D
Groupe
Biométrie fœtale
Gest. précoce
Os longs
Crâne de fœtus
ILA
Utérus
Mesure
LCC (Robinson)
GS (Hellman)
BIP (CFEF)
CT (CFEF)
DAT (CFEF)
CA (CFEF)
LF (CFEF)
OPN
IT
CN
Quitter
LCC
CN
CN 1.28mm
LCC 55.82mm
AG 12w0d 43.2%
P2: 19 s

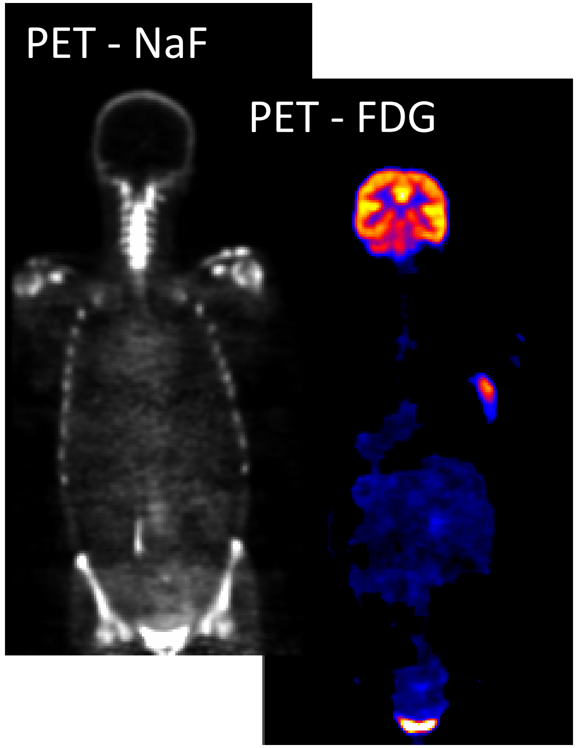
IBM: « Images represent 90% of medical data »

Medical Images, many modalities

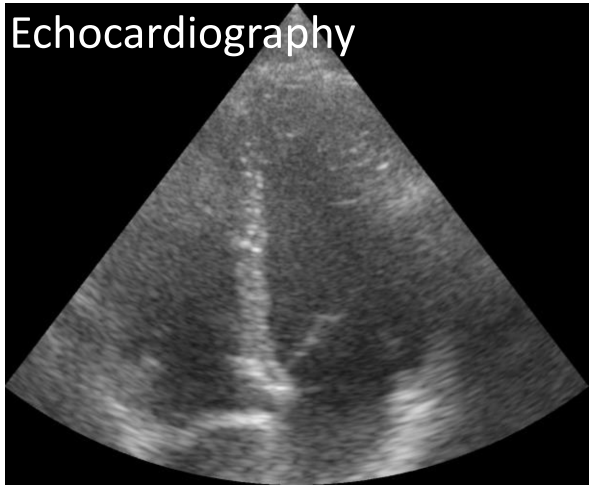
- 3D/2D (+t), different *physics*





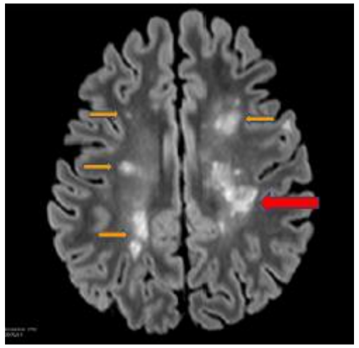


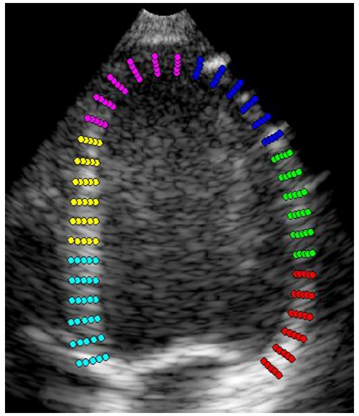
Positron Emission Tomography



Fundus examination

Medical Image Analysis and Diagnostic





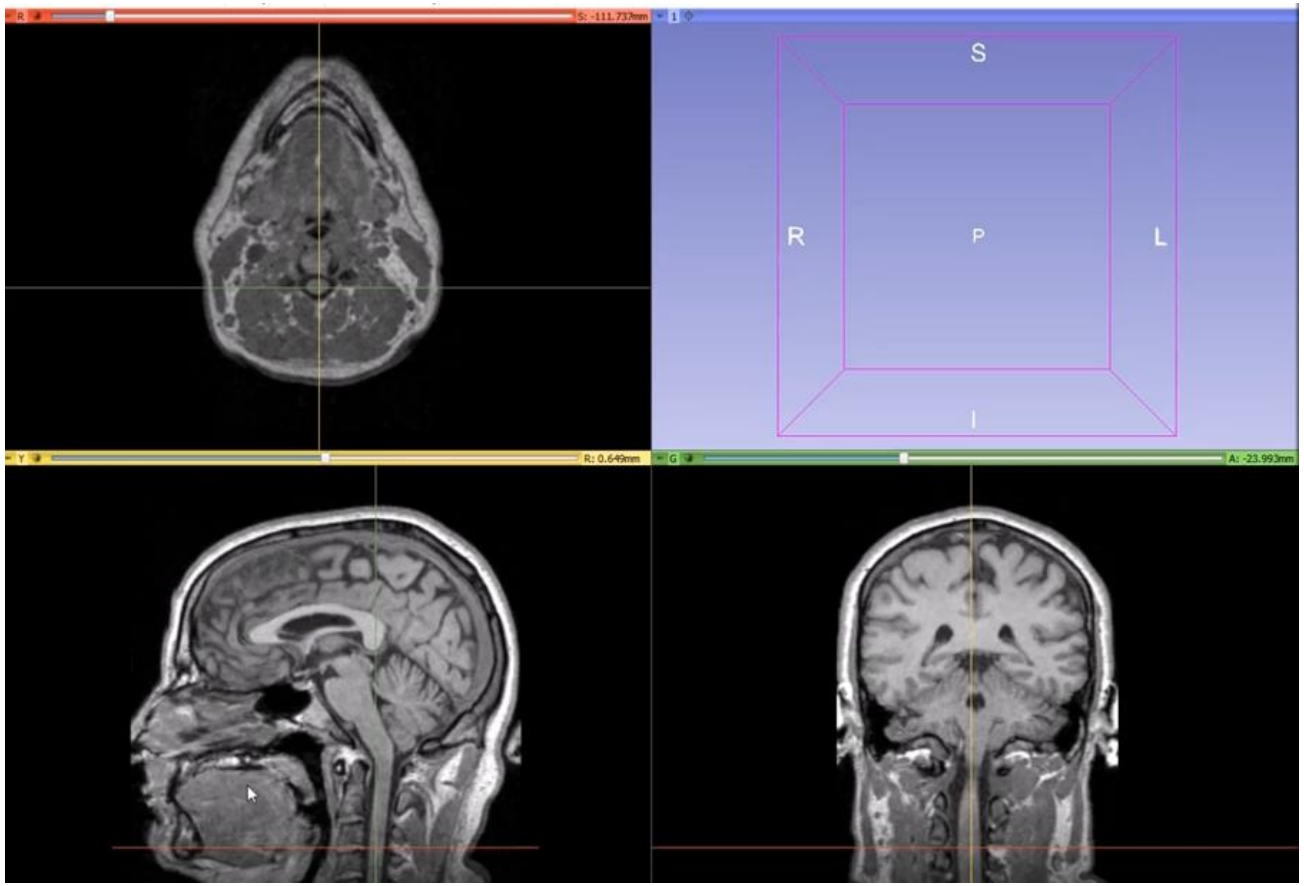
- Computer Aided Diagnosis (CAD)
  - Detection of pathologies (i.e. presence or not, malignant or not)
  - Measures : size, area/volume, shape…*

- Image or data processing involved in Computer Aided Diagnotic
  - Reconstruction (tomographic) and simulation
  - Filtering (denoising, deblurring)
  - Registration (intra or inter patient, intra or inter modalities)
  - Segmentation (delineation of organs, lesions, …)
  - Feature extraction (morphologic, normalized values, radiomics,…)

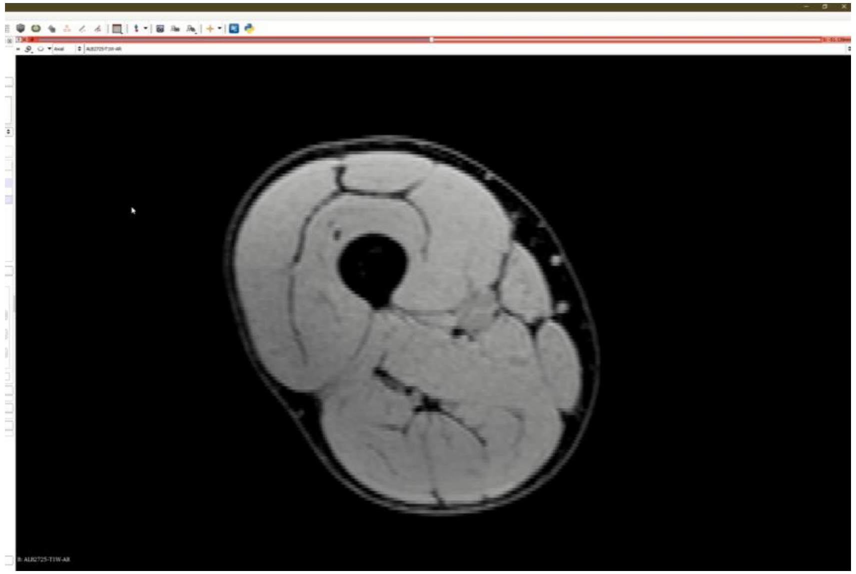
  → Analysis (statistics, classification, clustering, …)

➔ Many tasks need or derive to a **segmentation** problem, which is hard

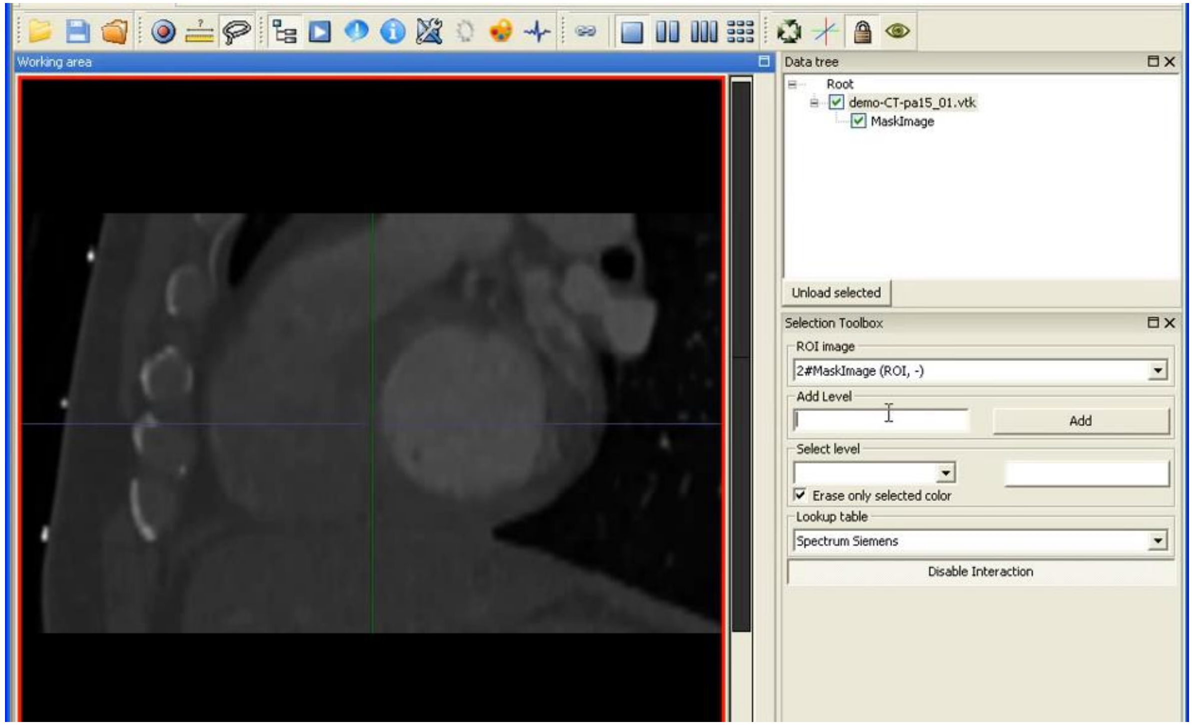
Manual image segmentation … a boring task!



3D Slicer – Quick Manual Segmentation with interpolation
https://www.youtube.com/watch?v=u93kI1MG6Ic

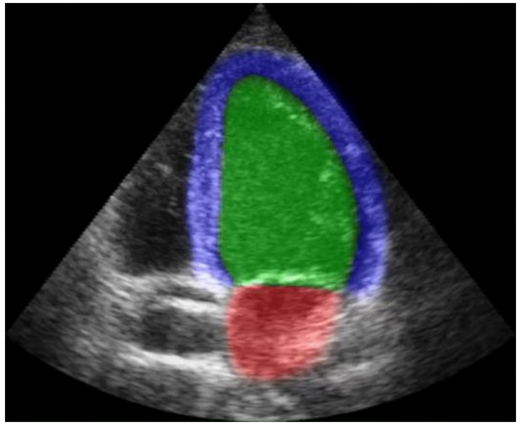


3D Slicer
Hoai-Thu Nguyen



GIMIAS Manual Segmentation
https://www.youtube.com/watch?v=rAHA1OZC8h8

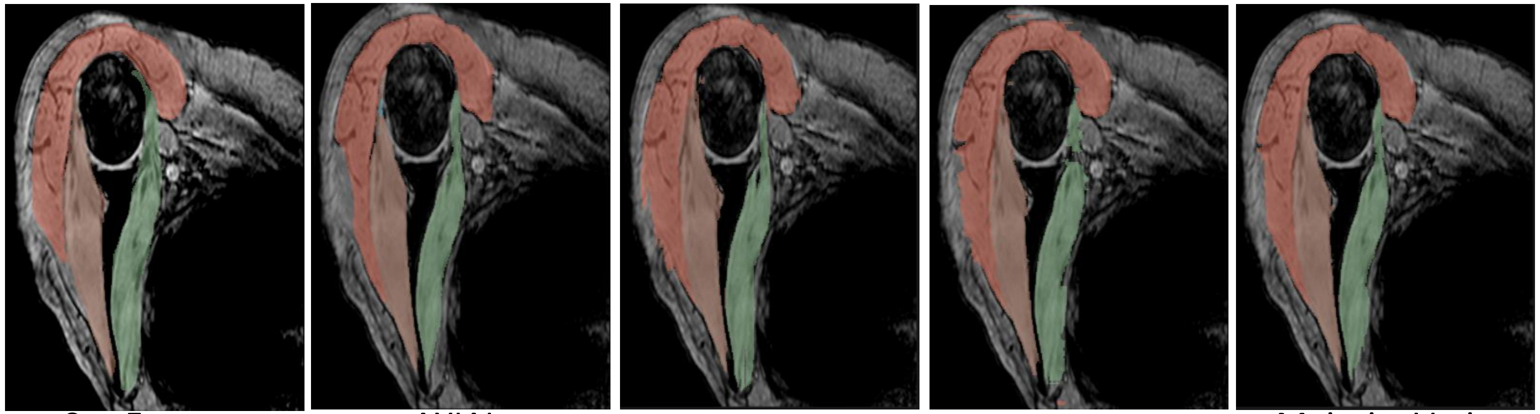
UNET : Automatic Image semantic segmentation



- Proposed by Ronneberger in 2015

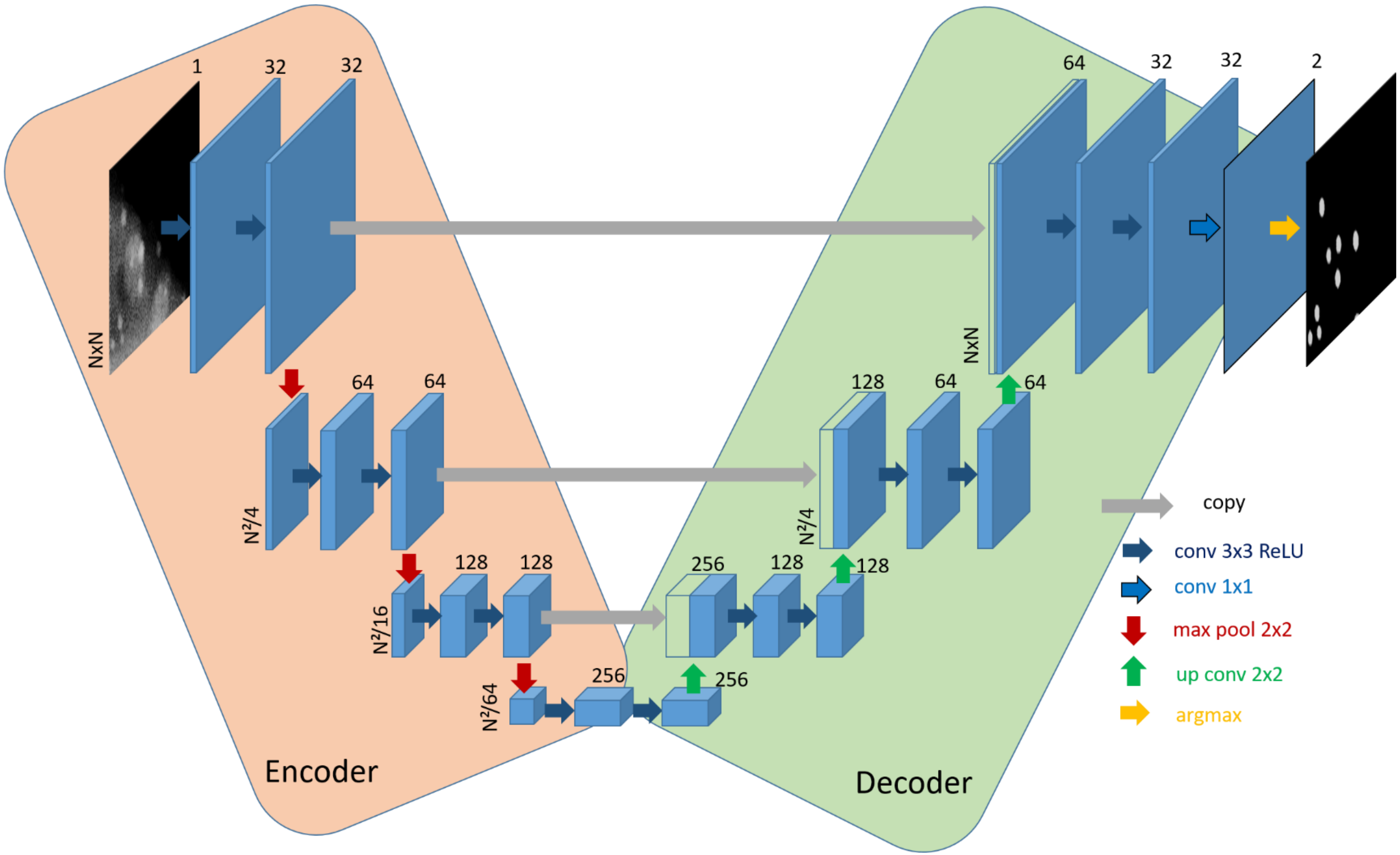
  a revolution : IOU 46% → 77%

- Work in 2D and extended to 3D
- Now, many “child”
  a. 3D UNet, V-NET, …
  b. UNet++, Unet 3+,
  c. ResUnet, …



Seg Expert AXIAL CORONAL SAGITTAL Majority Voting

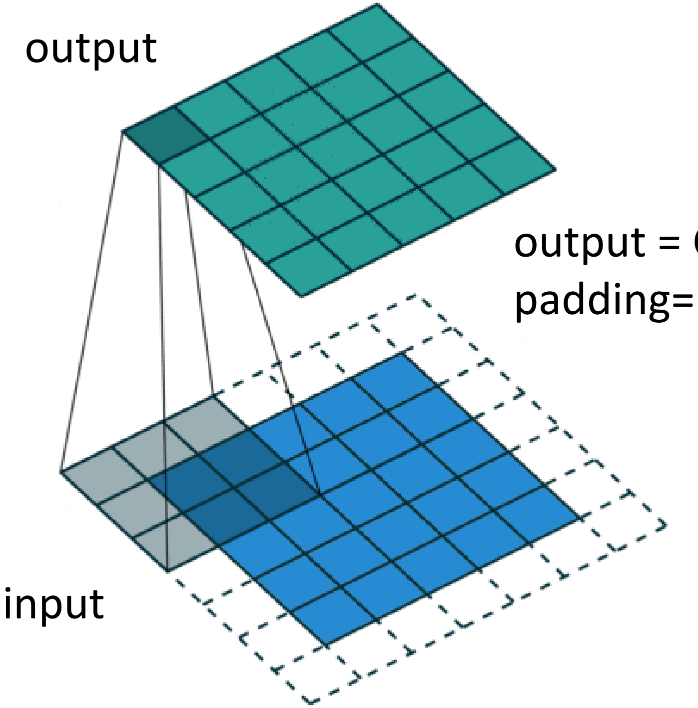
Binary UNet



1
32
32
NxN
64
64
N²/4
128
128
N²/16
256
N²/64
Encoder
64
32
32
2
NxN
128
64
64
N²/4
256
128
128
256
Decoder
copy
conv 3x3 ReLU
conv 1x1
max pool 2x2
up conv 2x2
argmax

Convolutions 2D and max pooling

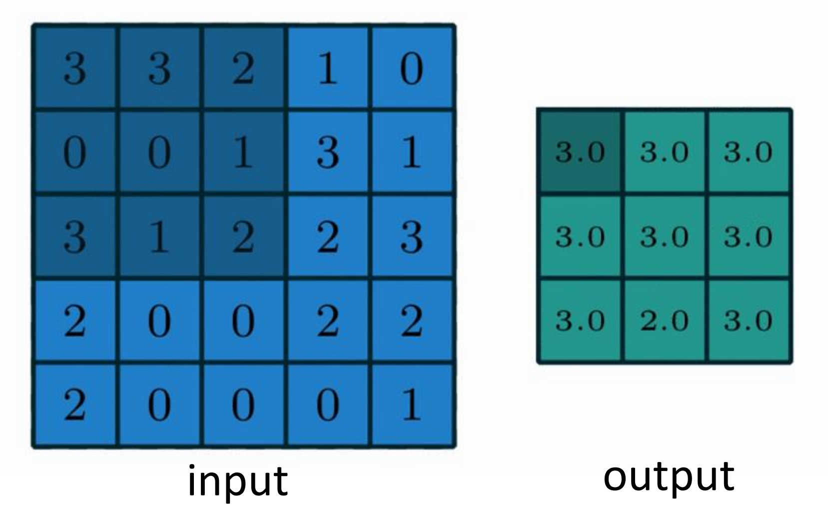
2D convolution using a kernel size of 3, stride of 1 and padding



output = Conv2D(1 , (3,3), strides=(1,1), padding='same') (input)

Number of filters

Max pooling kernel size of 3, stride of 1, no padding



output = MaxPooling2D((3, 3), strides=(1,1), padding='valid') (input)

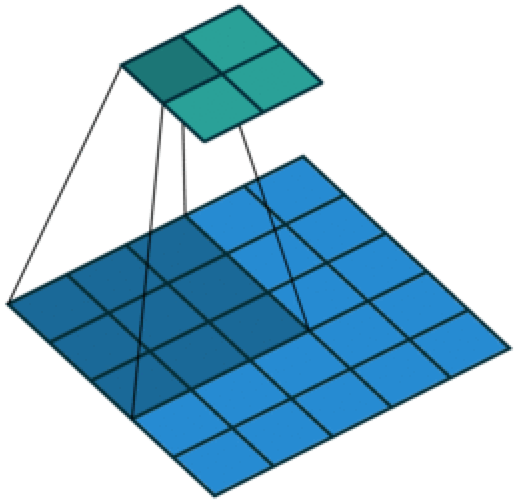
https://towardsdatascience.com/types-of-convolutions-in-deep-learning-717013397f4d

Upsampling: interpolation or **Up convolution**

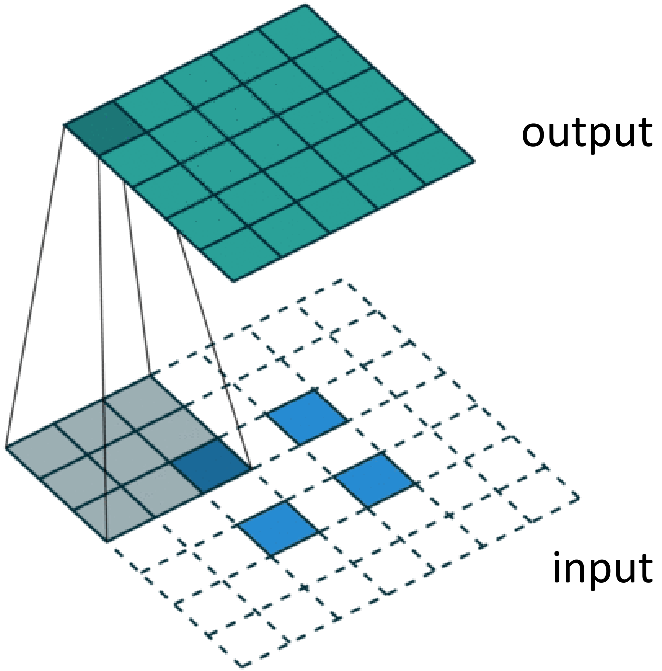
output = UpSampling2D((2,2), interpolation='nearest') (input)

or bilinear

2D convolution with no padding, stride of 2 and kernel of 3



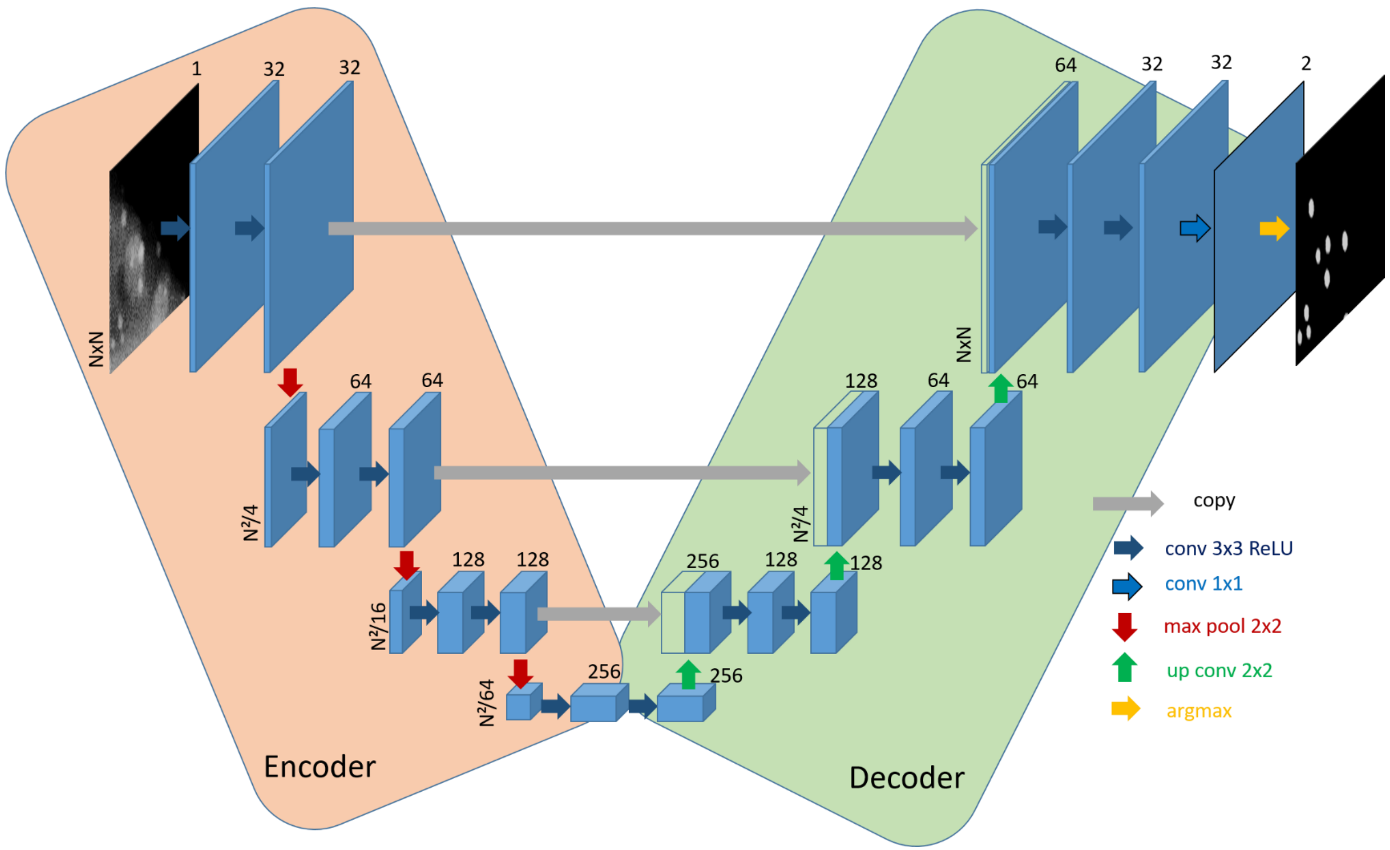
Transposed 2D convolution with padding, stride of 2 and kernel of 3

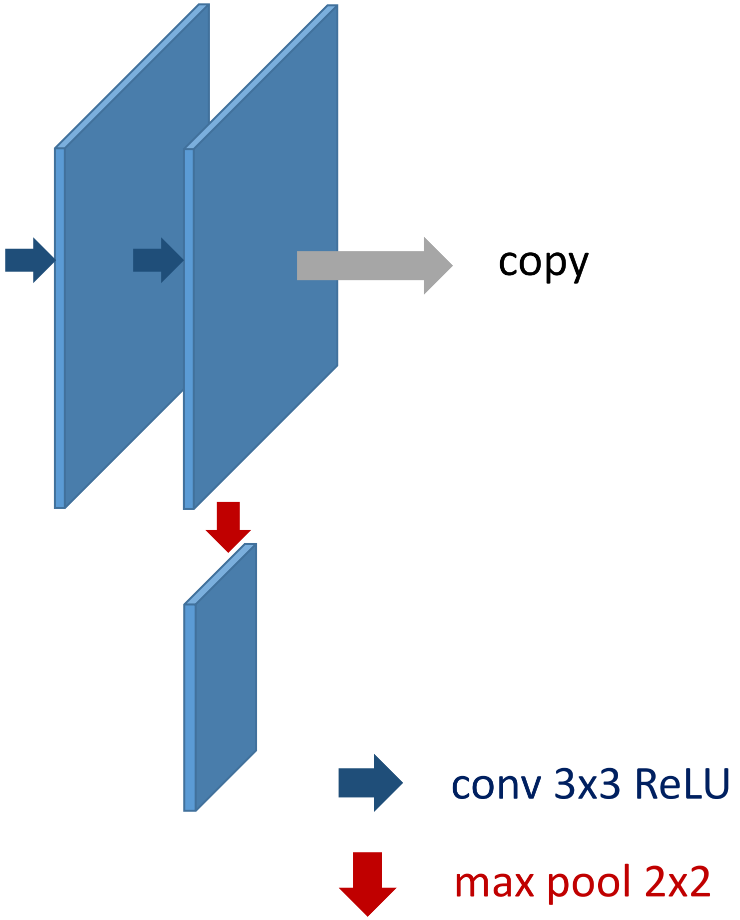


output = Conv2DTranspose(1, (3,3) , strides= (2,2), padding='same') (input)

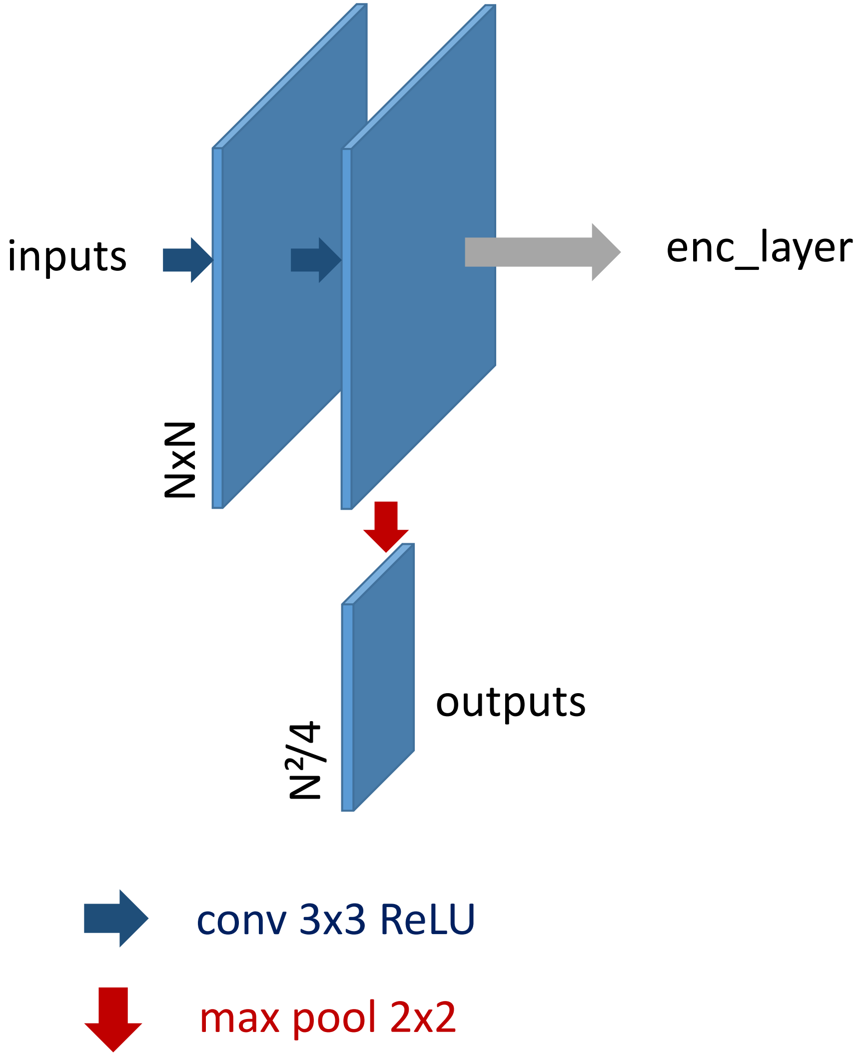
https://towardsdatascience.com/types-of-convolutions-in-deep-learning-717013397f4d

UNet code, encoder





UNet encoder part



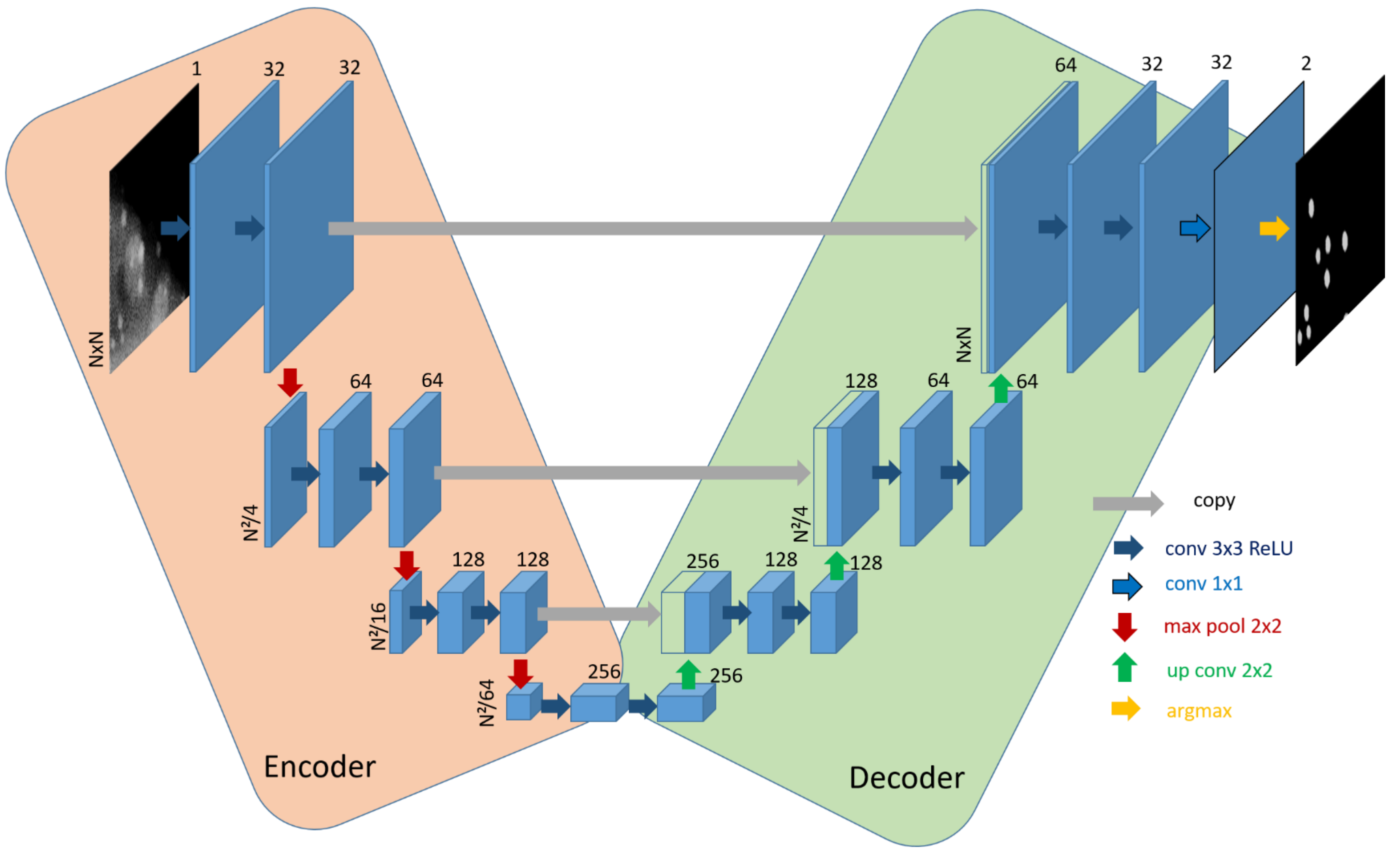
c = Conv2D(filters, (3,3), activation='relu',
kernel_initializer=kernel_initializer, padding='same') (inputs)

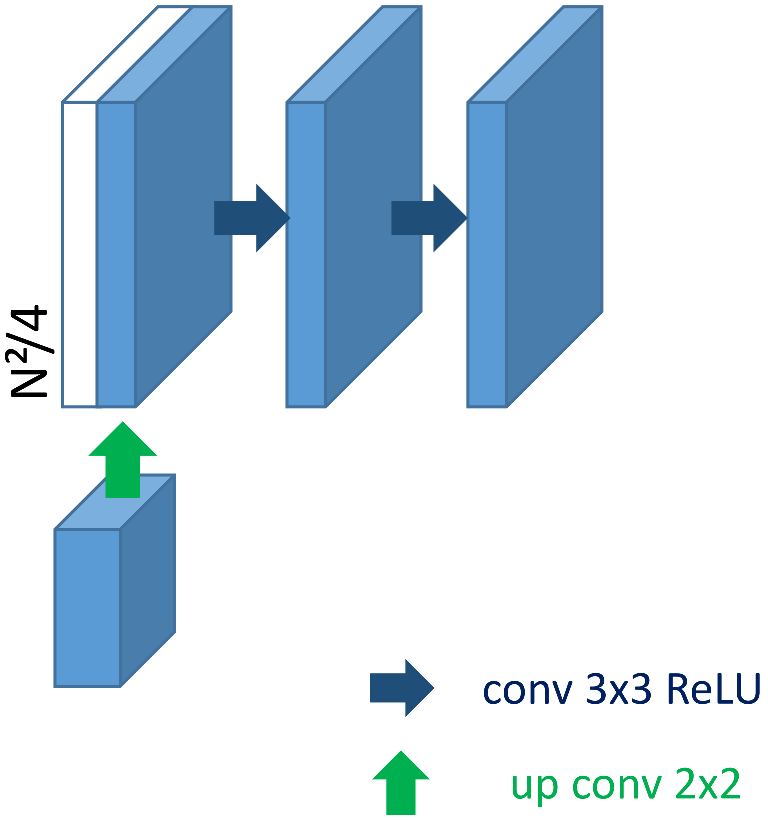
c = Conv2D(filters, (3,3), activation='relu',
kernel_initializer=kernel_initializer, padding='same') (c)

enc_layer = c

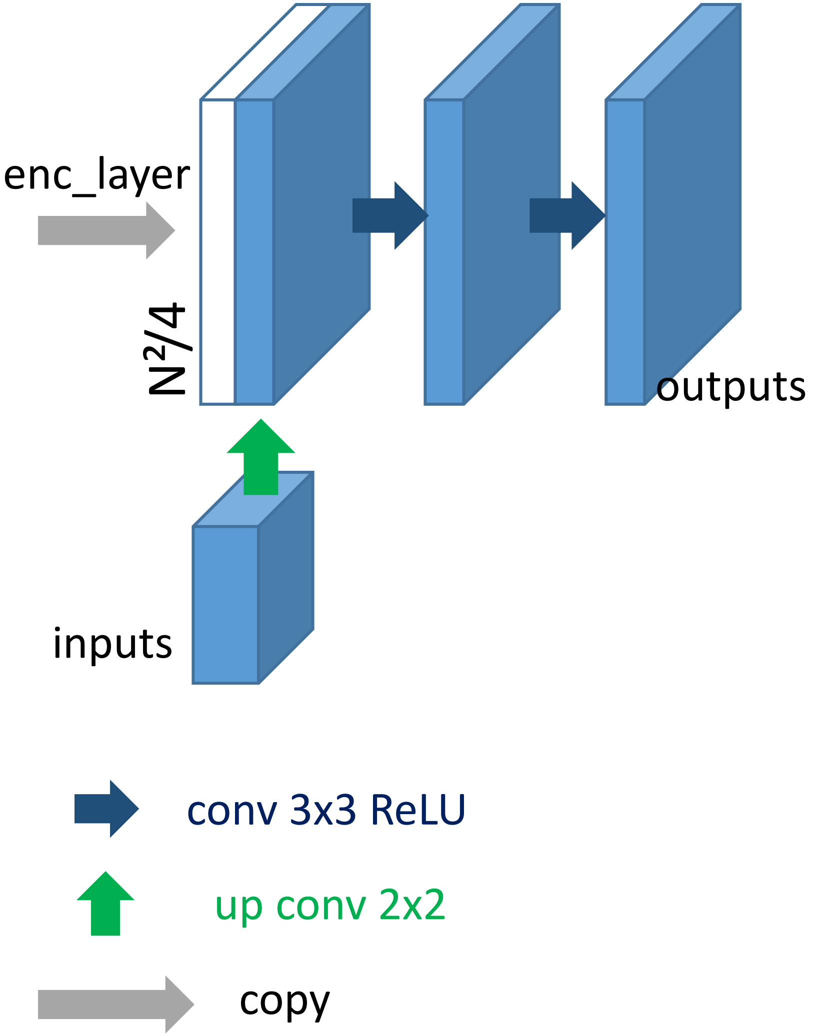
outputs= MaxPooling2D((2, 2)) (c)

UNet code, decoder





Unet Decoder part



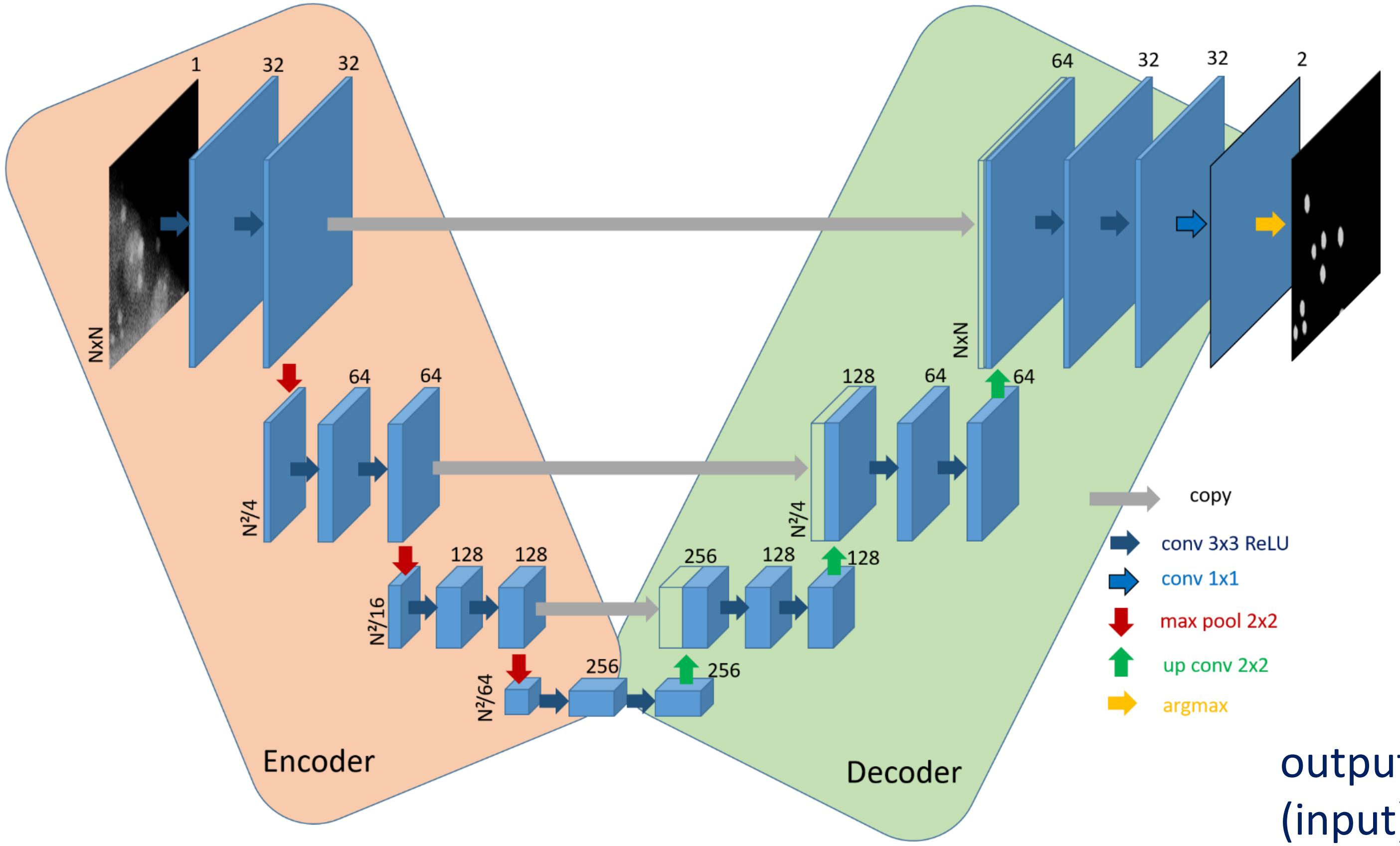
c = Conv2DTranspose(filters, (2, 2), strides=(2, 2), padding='same') (input)

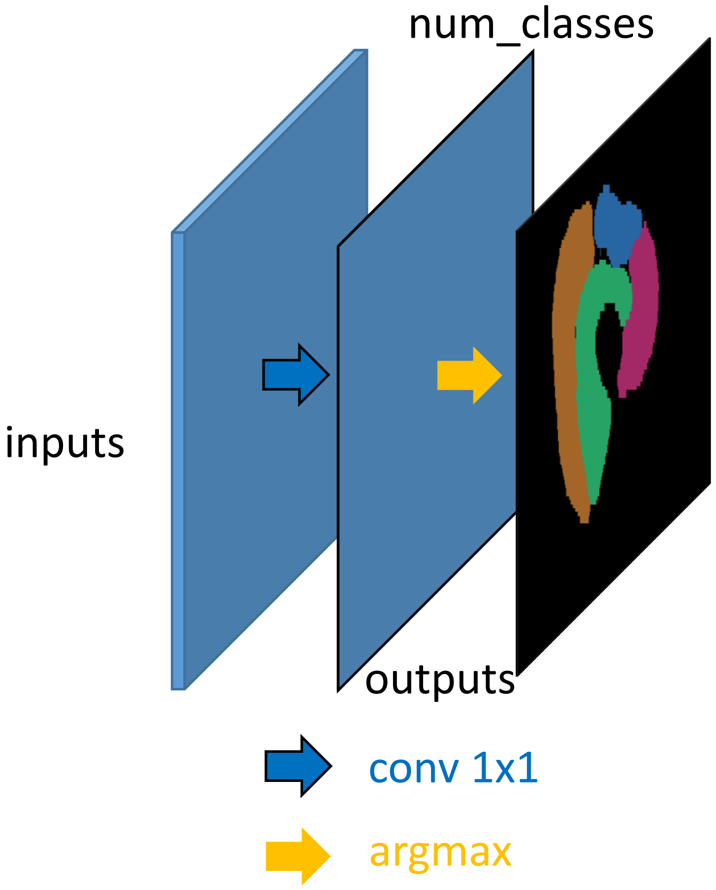
c = Concatenate()([c, enc_layer])

c = Conv2D(filters, (3,3), activation='relu', kernel_initializer=kernel_initializer, padding='same') (inputs)

outputs = Conv2D(filters, (3,3), activation='relu', kernel_initializer=kernel_initializer, padding='same') (c)

UNet code, output maps





outputs = Conv2D(num_classes, (1, 1), activation='sigmoid')(input)

The argmax is done outside the network (loss or metric, display, …)

Test it!

On FloydHub

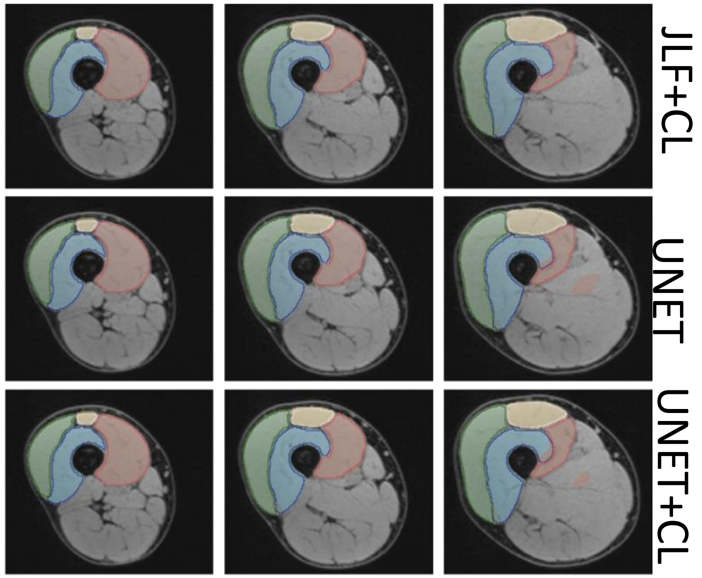
1- create Project

2- create Workspace

3- start the Hands-on

Regularizations

- Batch normalization : can produce noisy learning …
  →consider to standardize your medical data.
- (batch size : not under 4 with batch norm)
- Drop out : efficiency improvement observed
- L1, L2 or ElasticNet regularization : seems to help learning and producing accurate filtering…
- **Post processings** : often needed to remove extra regions and to fill holes in regions



[Nguyen 2020]

In fact : this is only true for some studies !

Loss and metrics

- Categorcial cross entropy (CCE)
- (originally: binary cross entropy)

- Dice Loss and multi-class DICE Loss (DL) (or F1 score)
- Intersection over Union (IoU) or Jaccard

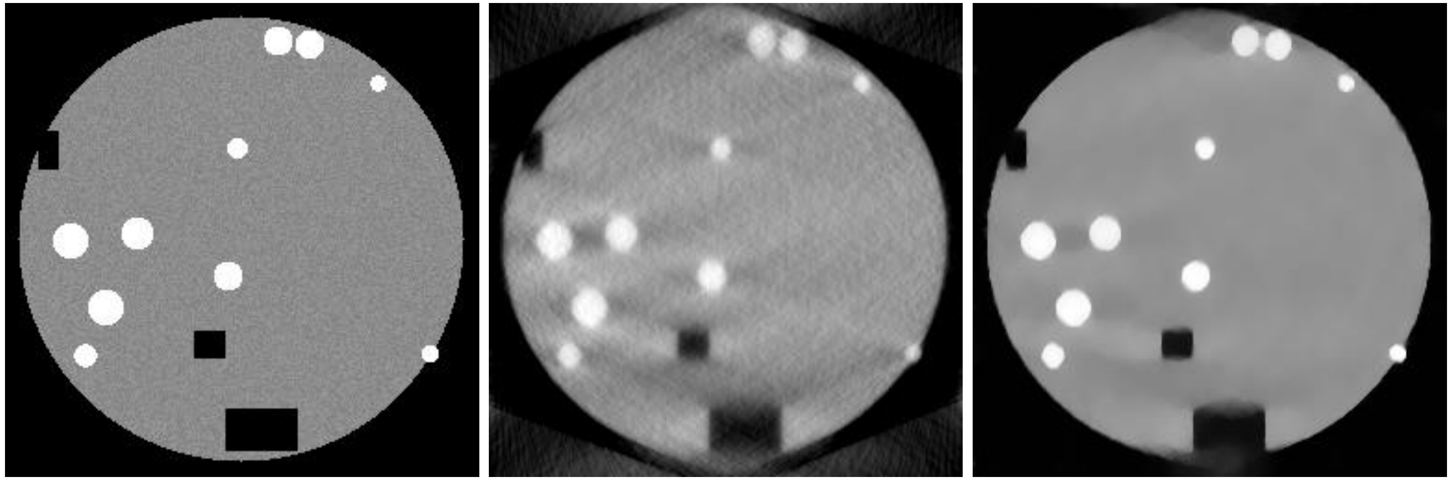
$$CCE(p, gt) = -\frac{1}{N \times (|C|)} \times \sum_{c=0}^{|Y|-1} \sum_{i=1}^{N} gt(i = c) \log(p(i, c))$$

$$DL(p, gt) = 1 - \frac{1}{|C| \times N} \times \sum_{c=0}^{|Y|-1} \sum_{i=1}^{N} \frac{2 \times p(i, c) \times gt(i = c)}{p(i, c) + gt(i = c)}$$

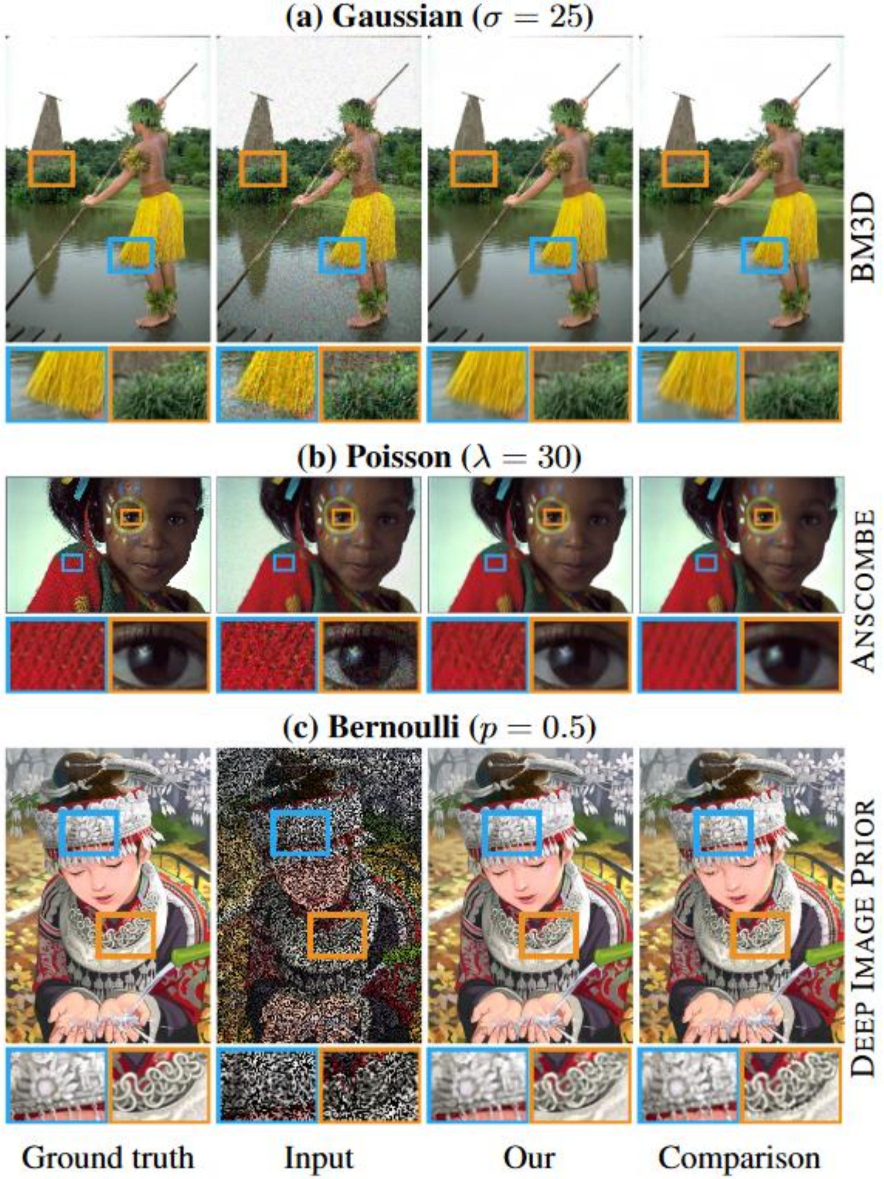
with $p(i, c)$ the probability for the sample i to belong to class $c \in Y$, and ϵ a small number added to avoid divisions by zero. The values $gt(i = c)$ are constants and binary: $gt(i = c) = 1$ when $i = c$, 0 otherwise.

Other usages of Unet: image filtering and restoration

1. Used to correct reconstruction artefacts
2. Used to learn filter (TV, noises,...)



Original and SIRT, SIRT FISTA-TV-NET of noisy and miising angle recontruction. [Banjack 2018]



Noise2Noise: Learning Image Restoration without Clean Data, Lehtinen 2018