Applications of Landmark Localization on Medical Images

Ahmet Emin Yetkin

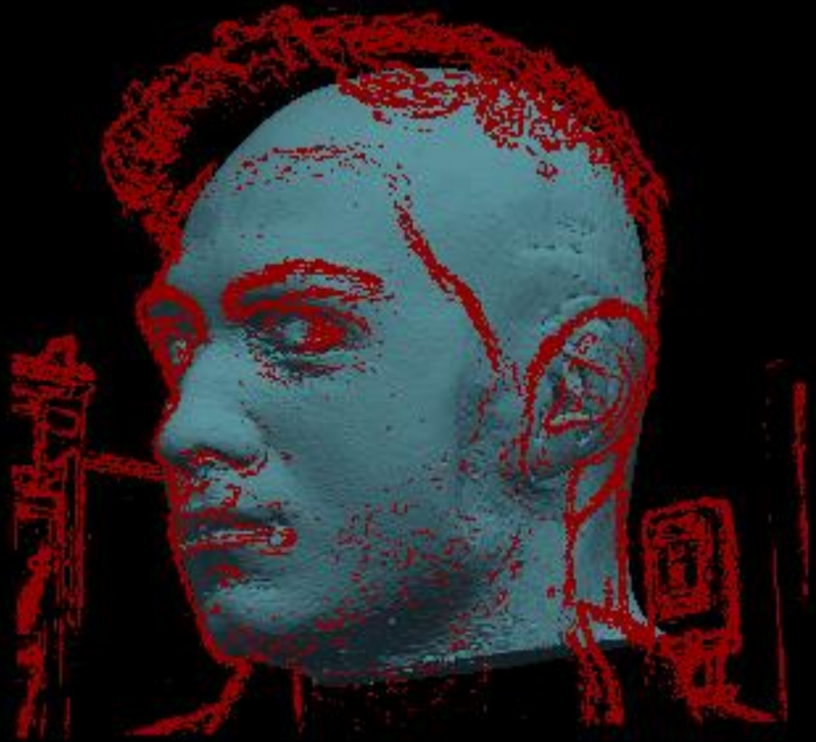
Research Assistant

Yeditepe University, Biomedical Engineering

Motivation : Why to locate keypoints?

- Pose estimation
- Object detection & localization
- Object recognition
- 3D Reconstruction
- Testing sufficiency model & dataset

LEFT : Contoured Image of a person



RIGHT : Registered contour image to 3D model created from same person's MRI surface

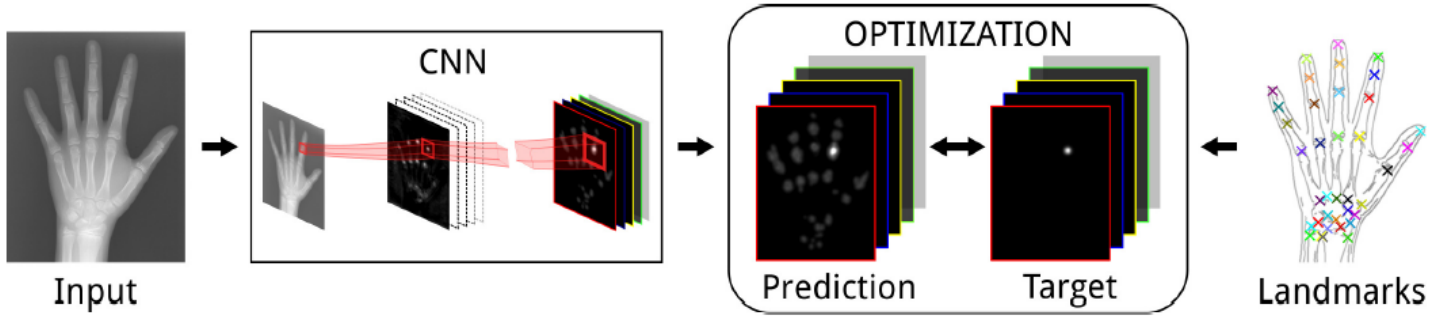
Source : Yetkin, Hamamcı, 2017

Heatmap Regression with Fully Convolutional Network

Regressing Heatmaps for Multiple Landmark Localization using CNNs

Christian Payer[1,*], Darko Štern[2], Horst Bischof[1], and Martin Urschler[2,3]

[1]Institute for Computer Graphics and Vision, Graz University of Technology, Austria
[2]Ludwig Boltzmann Institute for Clinical Forensic Imaging, Graz, Austria
[3]BioTechMed-Graz, Graz, Austria



Regression with fully Convolutional Neural Netwok

- It’s pretty easy and popular, actually.
- Reduces computational loss.
- Normally, output of last convolution layer is flattened and sent to fully connected neural network layers. Loss is applied to last layer of network.
- But now, instead of fully connected layers, there are convolutional layers with 1x1 kernel size. What they do is producing a single value among feature maps.
- This conv. layers with 1x1 kernel size may have different activation function(sometimes none), and may be tied to different normalization functions.(sometimes none)

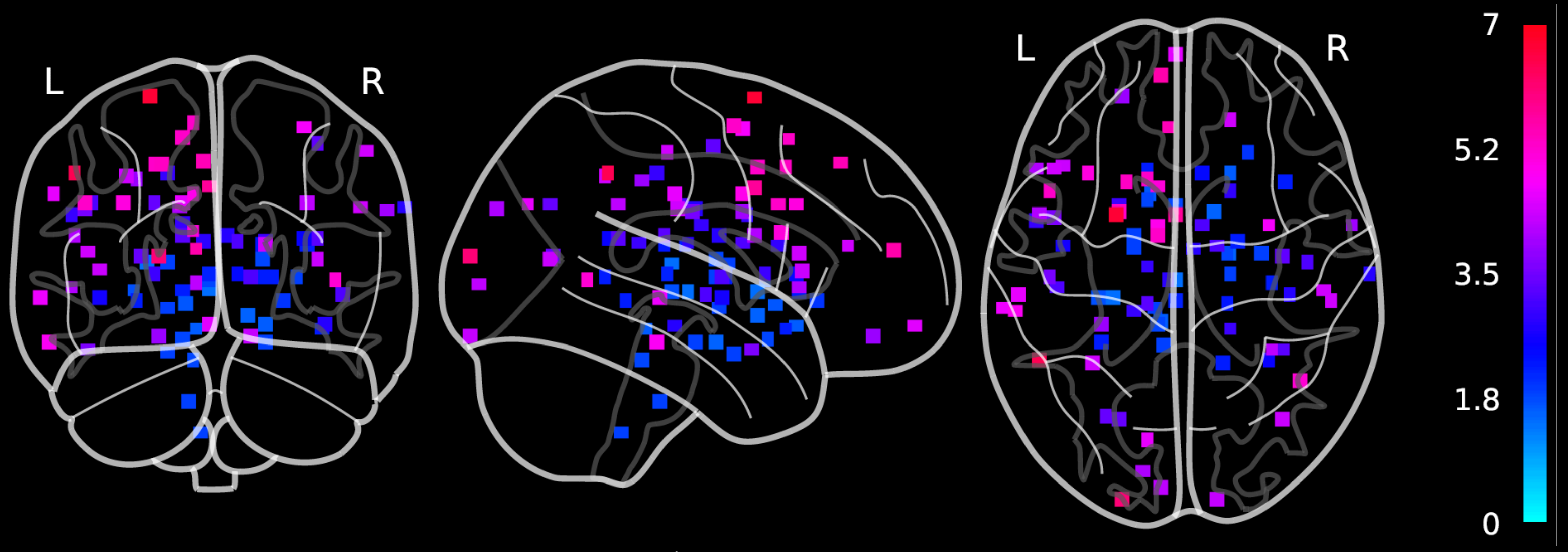
Useful Python Libraries

- SimpleITK : simplified layer built on top of ITK,
- nipy, analysis of structural and functional neuroimaging data.
  - nilearn : *fast and easy statistical learning on Neuro-Imaging data,*
  - nibabel : Read / write common neuroimaging file formats,
  - ... it goes on and on

Besides, **3D Slicer** is an open source software platform for medical image informatics, image processing, and three-dimensional visualization.

- It has Python wrapper ! You can move your own work to Slicer environment as an extension easily.
- The advantage you get by doing this, you could use Slicer's utilities.

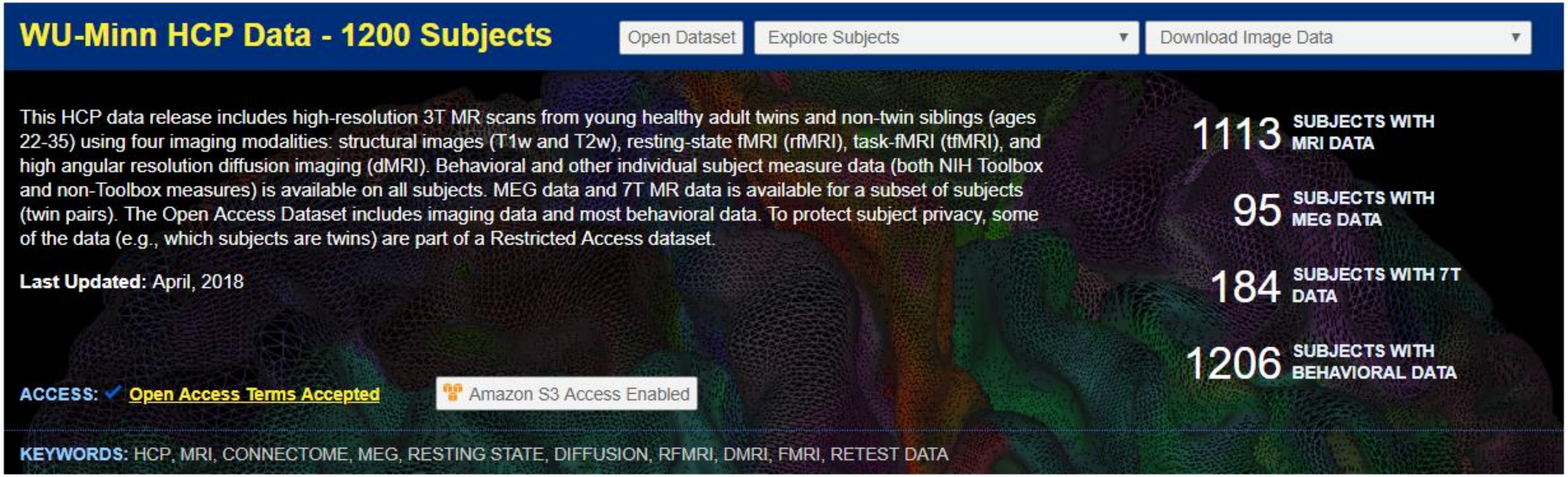
```
import nilearn.plotting as nip
result =  nibabel.Nifti1Image(array,affine)
nip. plot_glass_brain(result,other_parameters)
```



Source : Yetkin, Hamamcı, 2018

Dataset : Human Connectome Project

db.humanconnectome.org



Network Structure

- Hyperparameters (*kernel size, depth of layers, # of layer, lr, optimizer,loss func.*) play key role and are problem specific.
- Write your own loss function? Why not ?

```
def my_loss_func_1(y_true, y_pred):
    lmbda = 0.05  ## penalizing factor
    out = K.square(y_true-y_pred)
    return K.mean(out, axis=1) * lmbda
```

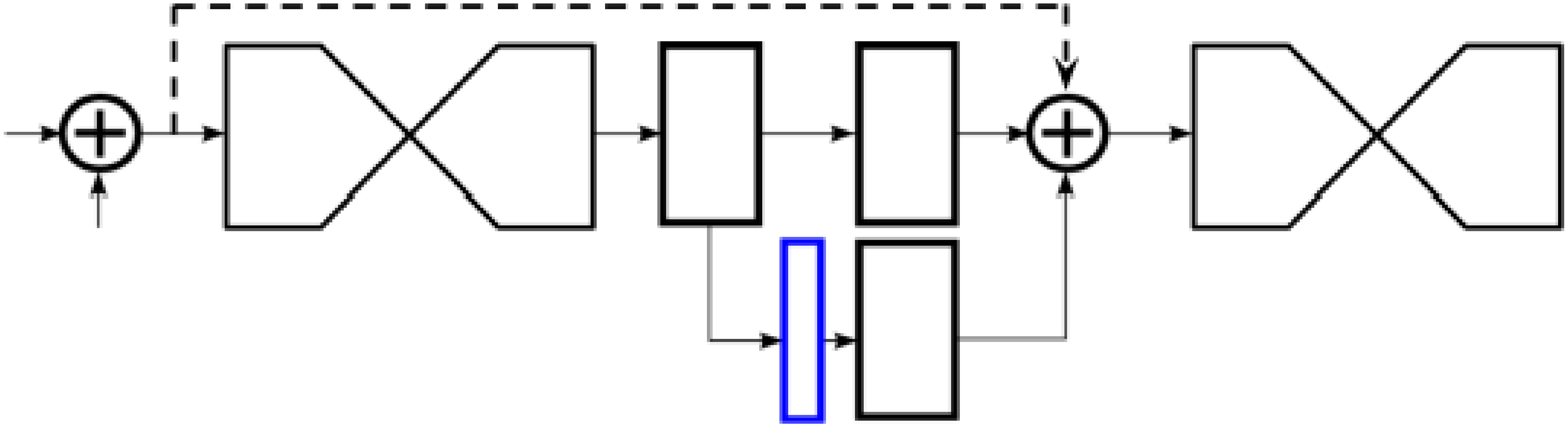
- Activation function is also important
- Mini batch if data is too big

- ROI is very related to problem and dataset. It affects receptive field, spatial subsampling.

Need to make it deeper ? How to handle?

- Batch normalization [Ioffe and Szegedy, 2014]
- Activation function (ReLU, ELU, PRELU...) can solve vanishing gradients problem
- Spatial Dropout (Especially when your model deals with big size data eg: input size of model is : 320x320x3)
- Residual Layers [He and Zhang, 2015]
- Intermediate Supervision [Newell and Yang, 2016]

Intermediate Supervision



Source : Newell, Yang, 2016

```
Merged_Model = Model(inp,[int_out_1,int_out_2,final_out])
Mrged_Model.compile(optimizer = RMSProp, loss=['mean_squared_error', 'my_loss_func_1', 'mean_squared_error'])
```

Stacked Hourglass Networks for Human Pose Estimation

Alejandro Newell, Kaiyu Yang, and Jia Deng

University of Michigan, Ann Arbor
{alnewell,yangky,jiadeng}@umich.edu

Large Pose 3D Face Reconstruction from a Single Image via Direct Volumetric CNN Regression

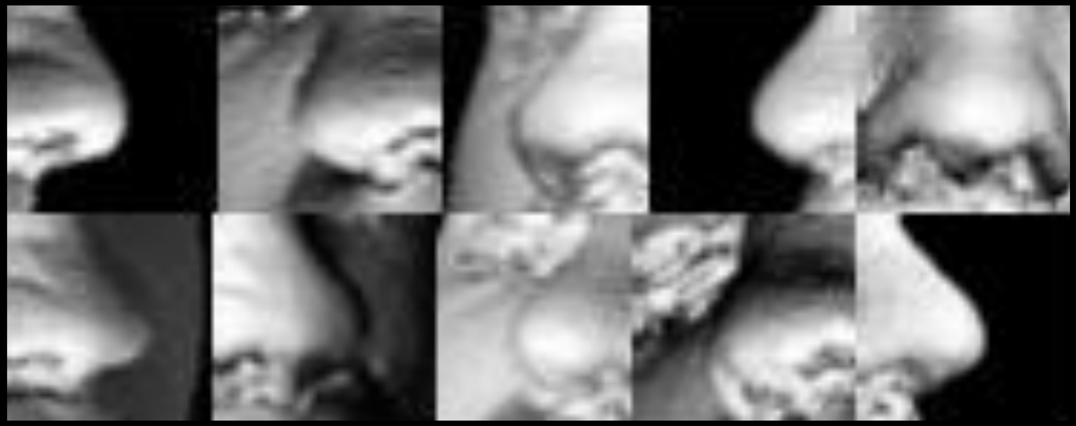
Aaron S. Jackson[1] Adrian Bulat[1] Vasileios Argyriou[2] Georgios Tzimiropoulos[1]

[1] The University of Nottingham, UK [2] Kingston University, UK

[1]{aaron.jackson, adrian.bulat, yorgos.tzimiropoulos}@nottingham.ac.uk
[2] vasileios.argyriou@kingston.ac.uk

Dataset

DeepCon'18



Cropped nose images from head model

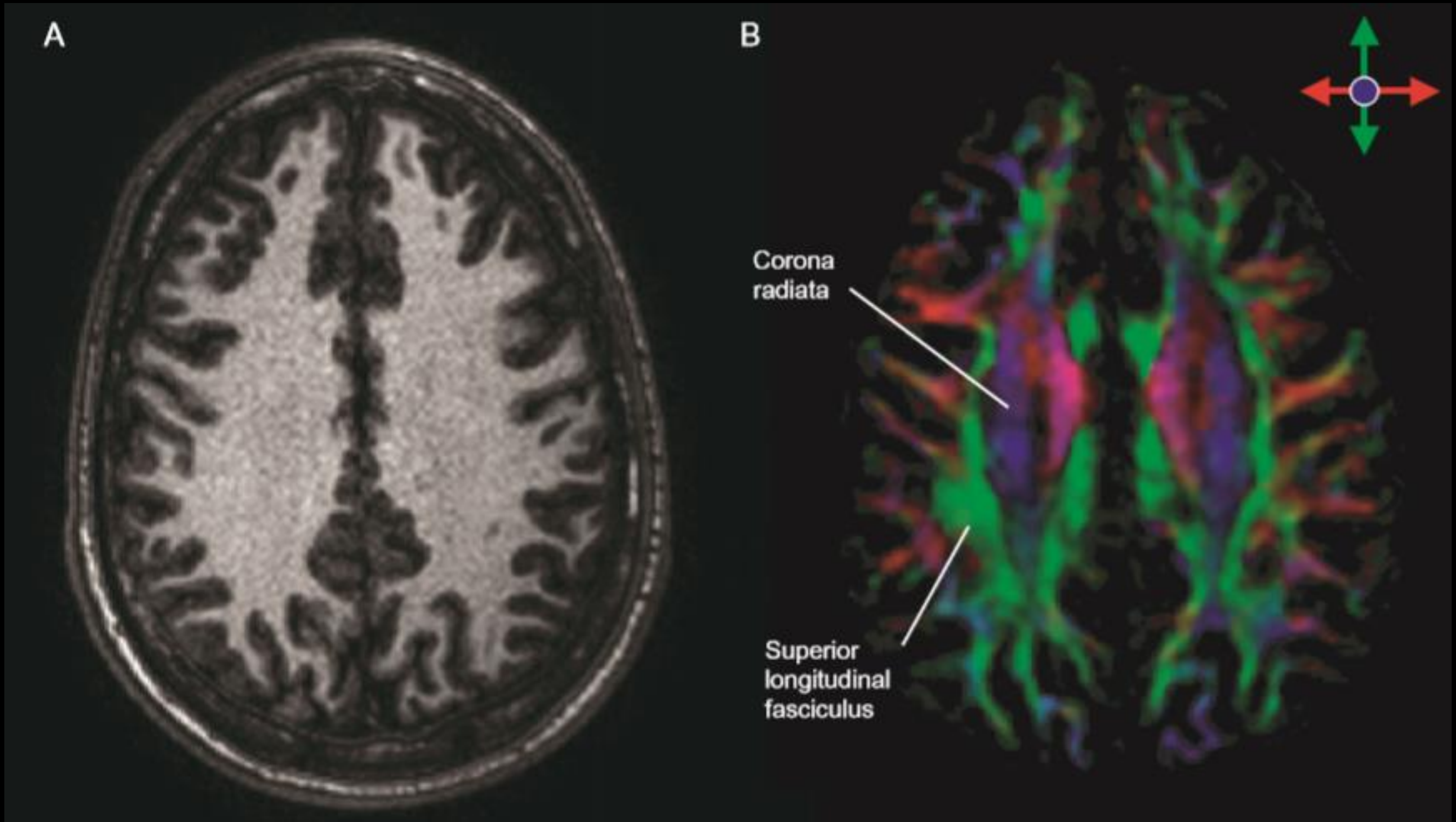
Source : Yetkin, Hamamcı, 2017

What is Diffusion Tensor Imaging ?

- Measures molecular motion along an arbitrary, predetermined axis; we can measure water diffusion along right-left, fore-aft, up-down, or any oblique angle.
- If we are measuring freely diffusing water, this unique capability does not mean much, because measurements along any orientation give the same result. This is what we call isotropic diffusion.
- However, the situation changes when we study biological tissues such as muscle and brain, which consist of fibers with coherent orientations. In such systems, water tends to diffuse along the fiber, and diffusion becomes anisotropic. This means that the results of diffusion measurements are not the same if they are measured along different orientations.

T1/T2 MRI vs DTI

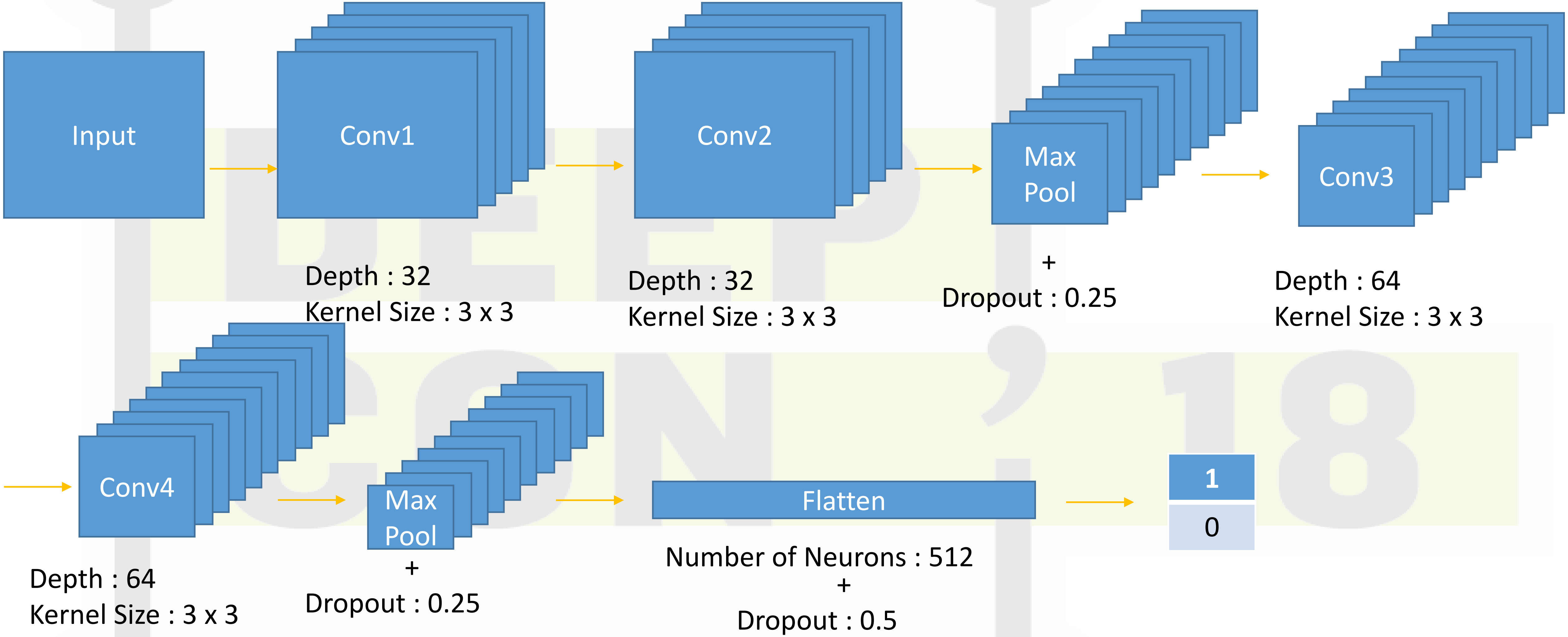
- Conventional MRI based on relaxation time relies on differences in chemical composition for their contrasts. For T1- and T2-weighted images, the amount of myelin plays a major role in differentiating the gray and white matter.
- However, the white matter looks quite homogeneous because it is homogeneous in terms of the chemical composition.
- In contrast, DTI can generate contrasts that are sensitive to fiber orientations



Source : Mori; 2007

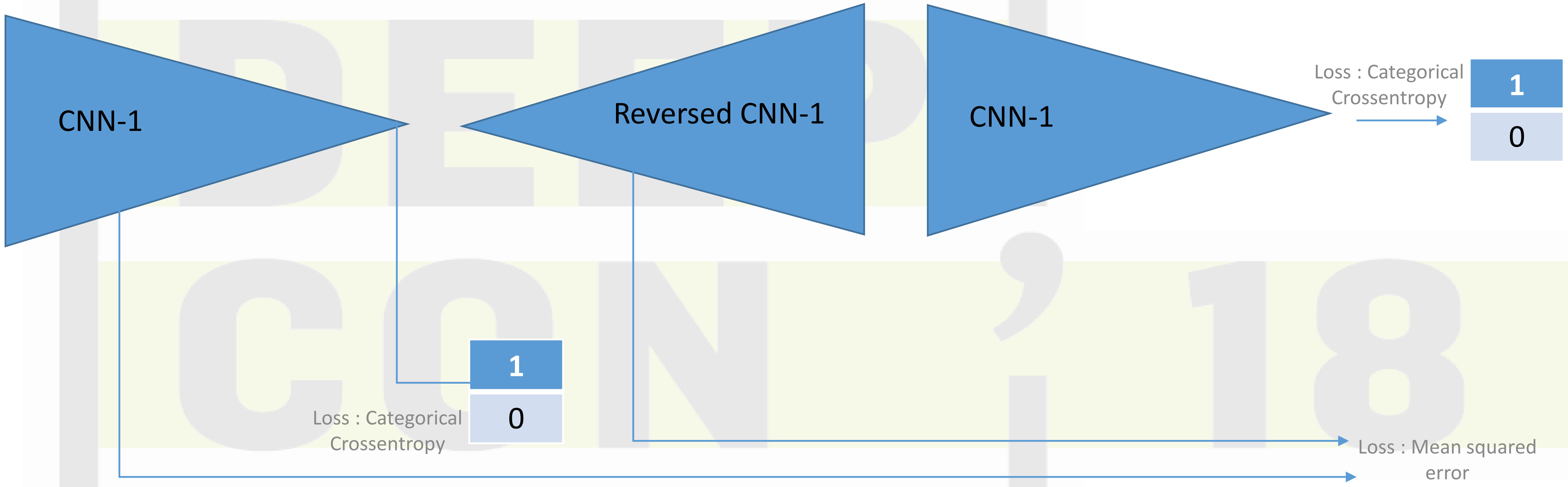
Comparison between a conventional MRI (T1-weighted image) and a DTI-based map (color map). In the color map, color represents fiber orientations; red, green, and blue represent fibers running along the right-left, anterior-posterior, and superior-inferior orientations.

CNN - 1

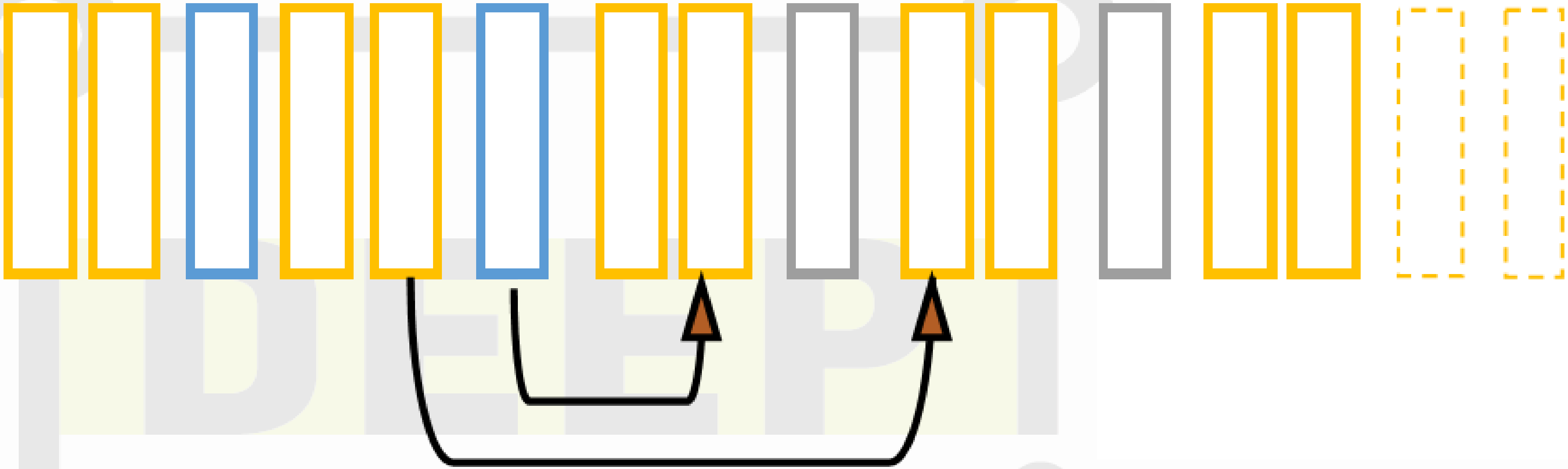


Source : Yetkin, Hamamcı, 2017

CNN - 2



Source : Yetkin, Hamamcı, 2017



Convolution by 3x3

Convolution by 1x1

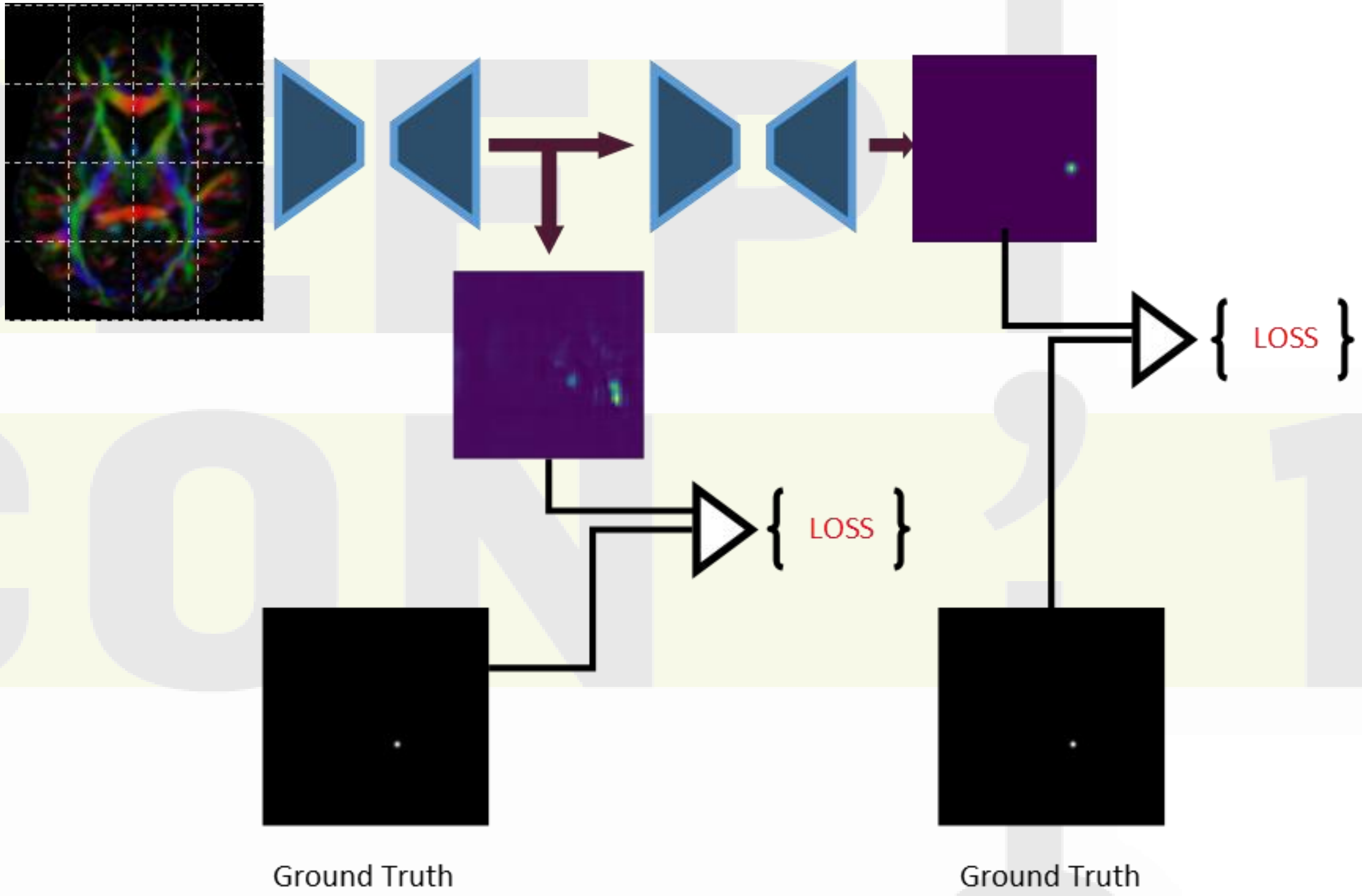
Max Pooling Operation

Up Sampling Operation

Source : Yetkin, Hamamcı, 2018

Source : Yetkin, Hamamcı, 2017

Intermediate Supervision Implementation



Source : Yetkin, Hamamcı, 2018

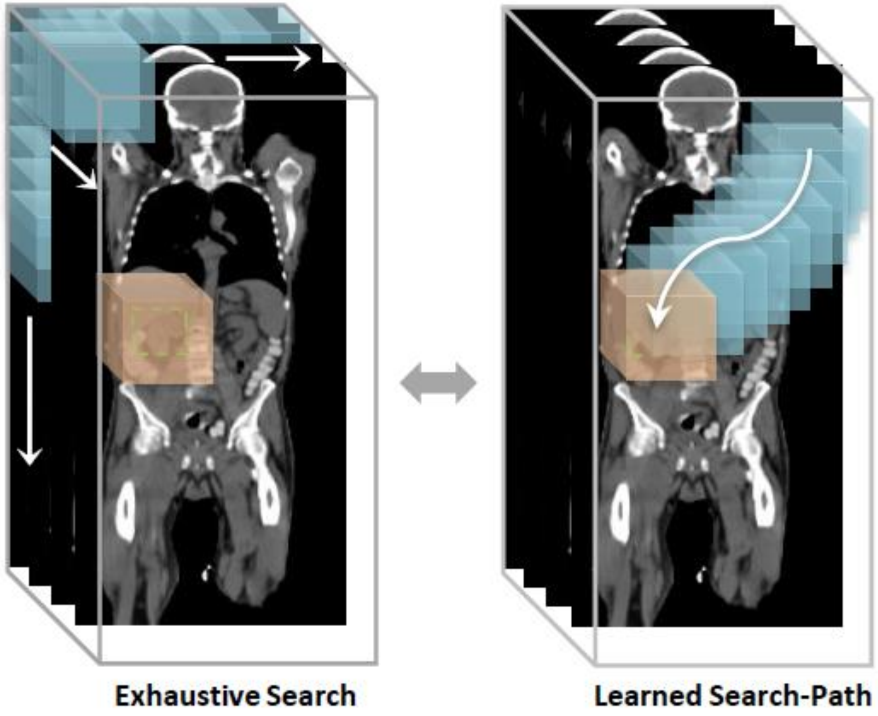
Mini Suggestion about debugging the Net

- Keep log about everything you can.
- Visualize kernels.
- Test the model during training.
- (ex: every 500 iteration, save results, compare results)
- It’s an easy way to know that model converges. Saves time.

Read Suggestion

Multi-Scale Deep Reinforcement Learning for Real-Time 3D-Landmark Detection in CT Scans

Florin C. Ghesu, Bogdan Georgescu, *Member, IEEE,* Yefeng Zheng, *Senior Member, IEEE,* Sasa Grbic, Andreas Maier, Joachim Hornegger, and Dorin Comaniciu, *Fellow, IEEE*



Thank you for listening.
Any questions?