AI for PET Image Reconstruction

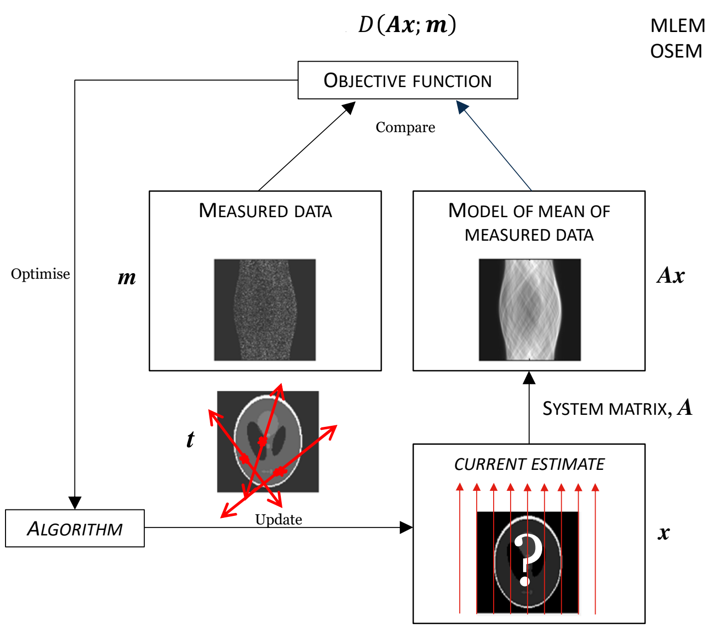
Andrew J. Reader

School of Biomedical Engineering and Imaging Sciences
King's College London
UK

andrew.reader@kcl.ac.uk

1

Conventional image reconstruction *(using models only, no training data)*



MLEM or OSEM

$$\boldsymbol{x}^{n+1} = \frac{\boldsymbol{x}^n}{\boldsymbol{A}^T\boldsymbol{1}}\boldsymbol{A}^T\left(\frac{\boldsymbol{m}}{\boldsymbol{Ax}^n}\right)$$

Poisson (1781–1840)

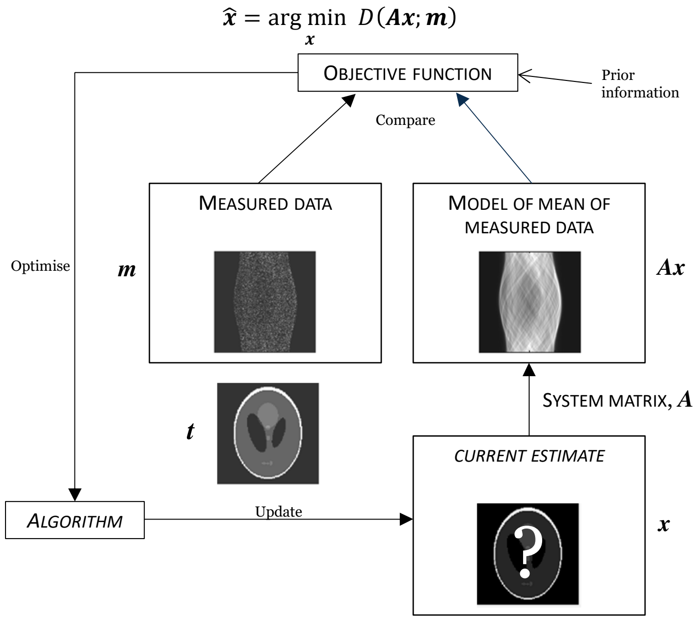
$$\boldsymbol{x}^{LS} = (\boldsymbol{A}^T\boldsymbol{A})^{-1}\boldsymbol{A}^T\boldsymbol{m}$$

Gauss (1777–1855)

Radon (1887-1956)

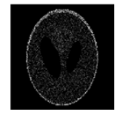
2

Conventional image reconstruction *(using models only, no training data)*

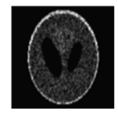


Possible problems:

1. Fits $\boldsymbol{x}$ to **noisy** data $\boldsymbol{m}$!



2. Regularisation (MAPEM)
Which prior / penalty?
Hyperparameter β = ?

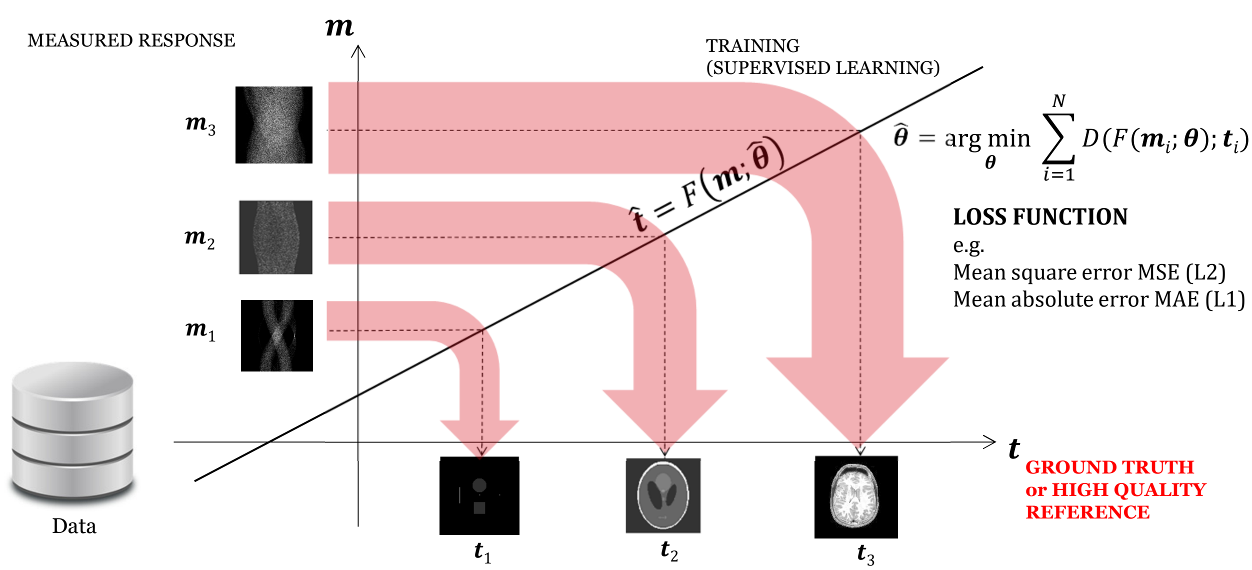


$$R(\boldsymbol{x}) = \|\boldsymbol{Tx}\|_2^2$$

3. Assumes system matrix is correct, assumes noise model is correct

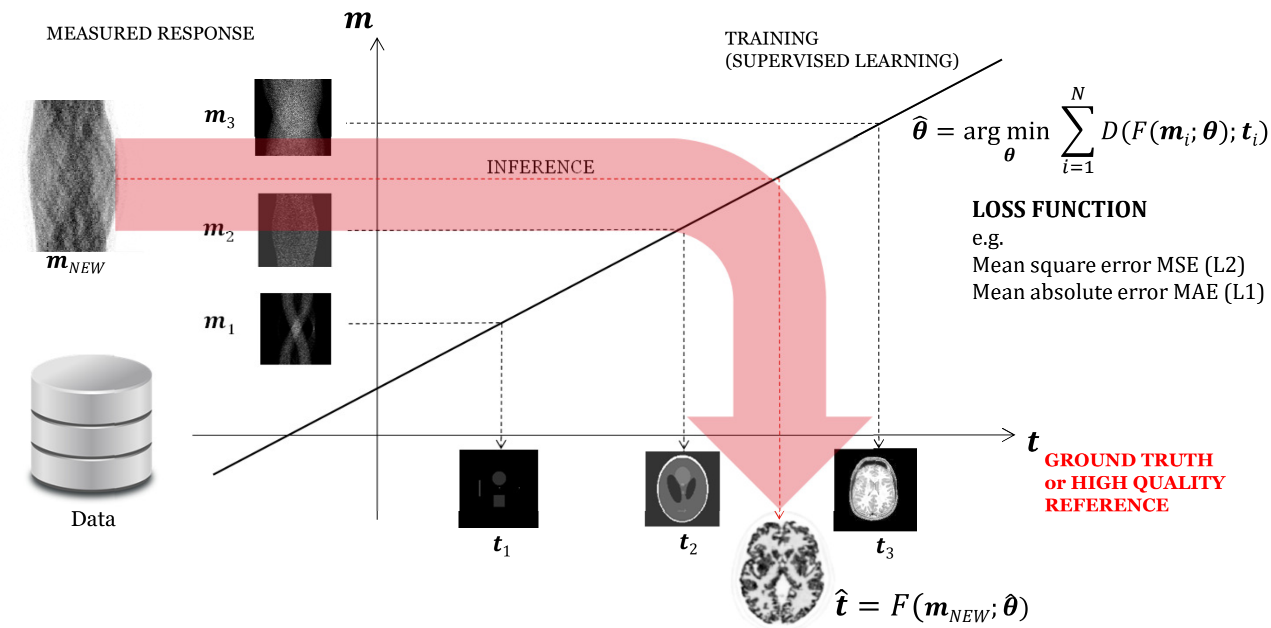
3

Deep learning, when relying mainly on training data...



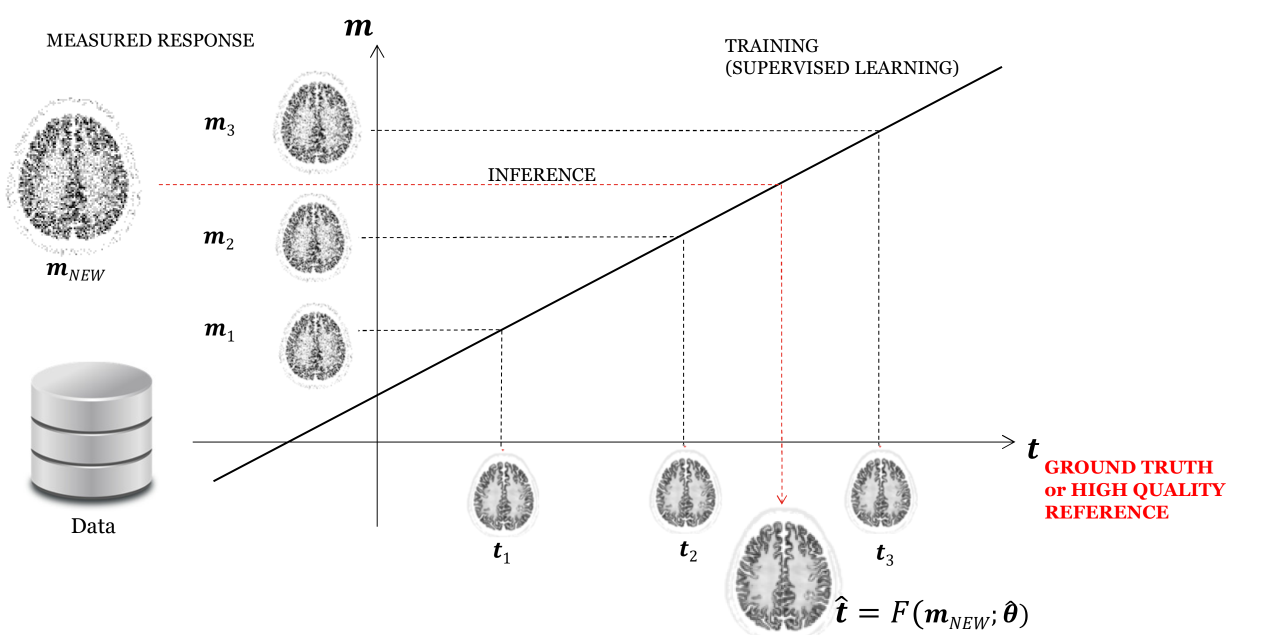
4

Deep learning, when relying mainly on training data…



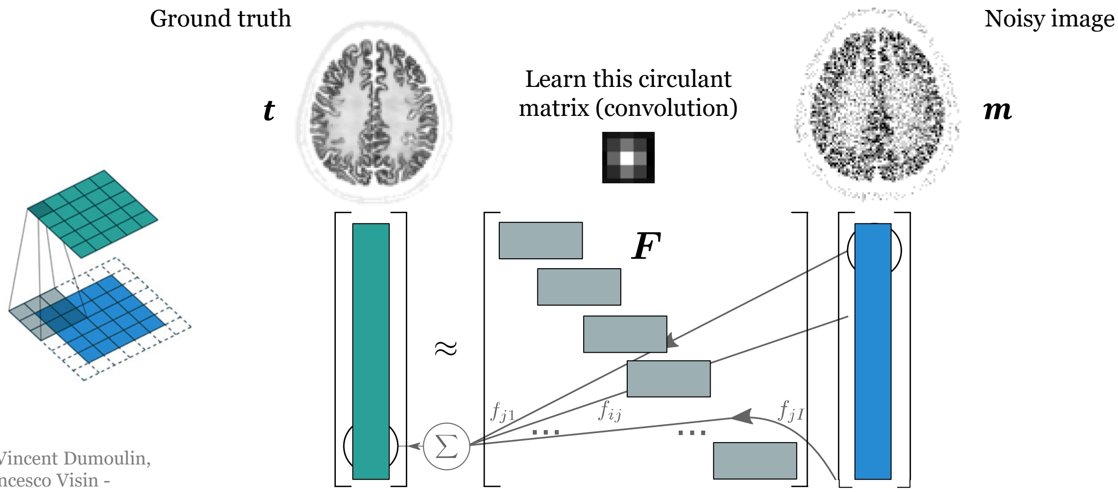
5

First simplify to an image to image denoising mapping



6

Image to image mapping with few parameters: convolution

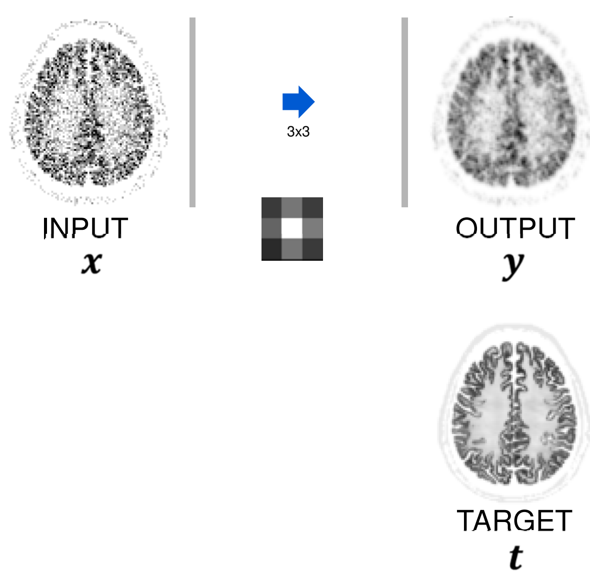


By Vincent Dumoulin, Francesco Visin - https://github.com/vdumoulin/conv_arithmetic, MIT, https://commons.wikimedia.org/w/index.php?curid=78003456

Convolution with a 3x3 kernel:
just 9 parameters for a whole image to image mapping!

7

Learning 1 convolution kernel: denoising



Architecture:
- One 3×3 convolution kernel
- 9 parameters to learn

Training data:
- Input x: **noisy** image
- Target t: ground truth

Loss function:
- Mean square error (MSE)

$$\frac{1}{V}\sum_{v=1}^{V}(y_v - t_v)^2$$

Optimiser:
- Gradient descent (GD)
  - update parameters by subtracting gradient of loss function
- Stochastic GD (SGD)

8

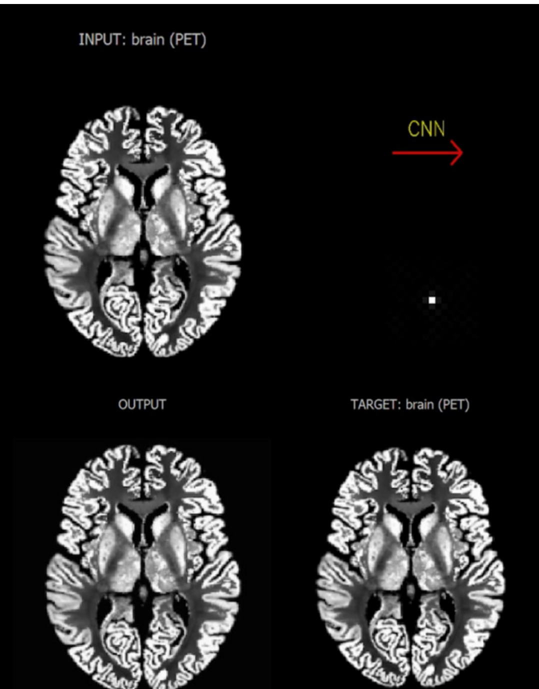
Learning 1 convolution kernel

https://youtu.be/JvJgvjm1hco

9

Learning 1 convolution kernel

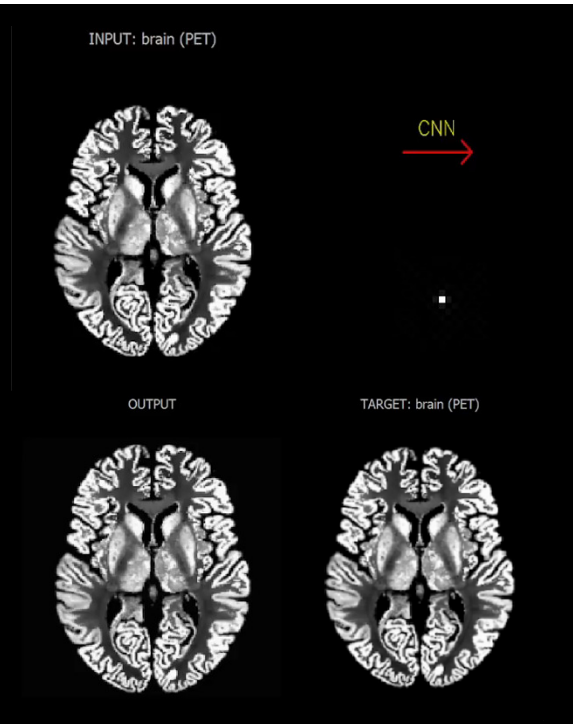
https://youtu.be/JvJgvjm1hco



10

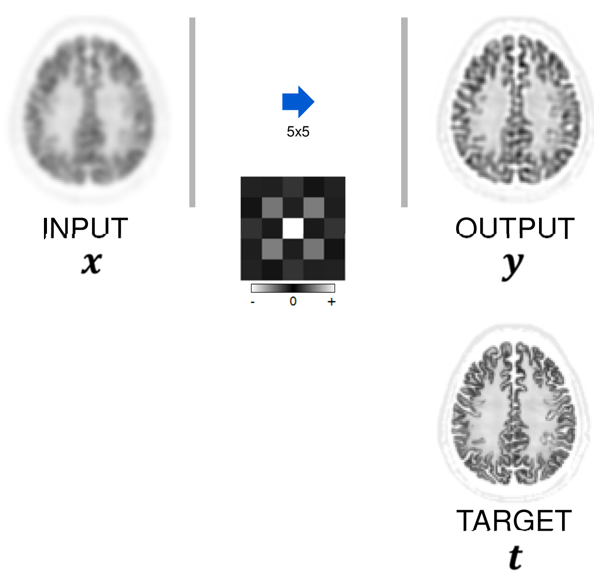
Learning 1 convolution kernel

https://youtu.be/JvJgvjm1hco



11

Learning 1 convolution kernel : sharpening



Architecture:
- One 5×5 convolution kernel
- 25 parameters to learn

Training data:
- Input $\boldsymbol{x}$: **blurred** image
- Target $\boldsymbol{t}$: ground truth

Loss function:
- Mean square error (MSE)

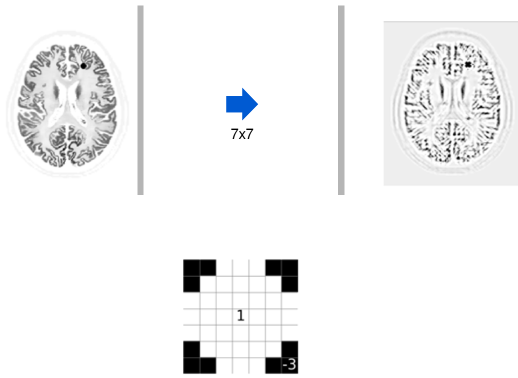
$$\frac{1}{V}\sum_{v=1}^{V}(y_v - t_v)^2$$

Optimiser:
- Gradient descent (GD)
- Stochastic GD (SGD)

12

Convolution with bias & activation: feature detection

Architecture:
One 7 × 7 kernel

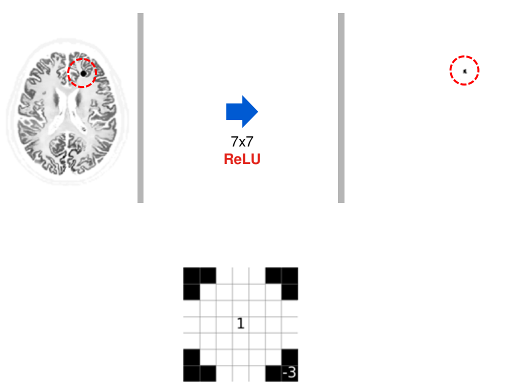


13

Convolution with bias & activation: feature detection

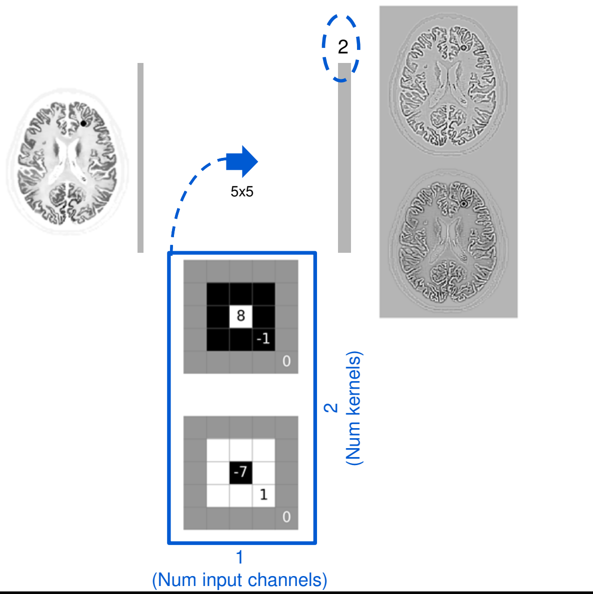
ReLU = rectified linear unit (just sets any negative values to zero!)

Architecture:
One 7 × 7 kernel +bias



14

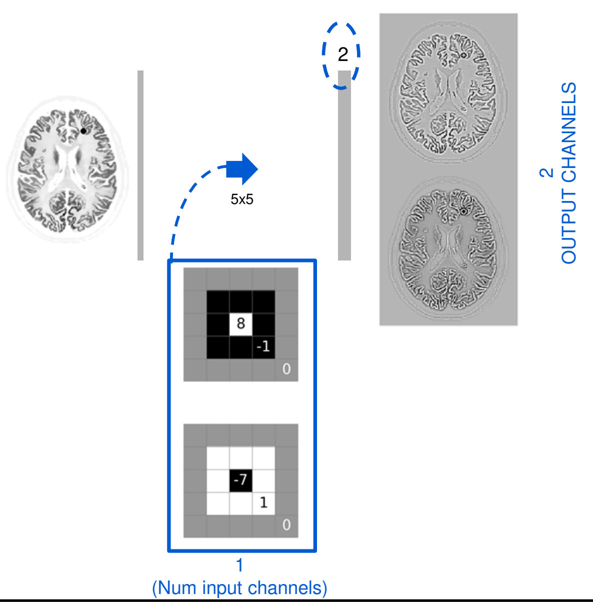
Convolutional neural networks (CNNs): multiple kernels



Architecture:
Two, 1 channel, 5 × 5 kernels (1st layer)

15

Convolutional neural networks (CNNs): multiple kernels



Architecture:
Two, 1 channel, 5 × 5 kernels (1st layer)

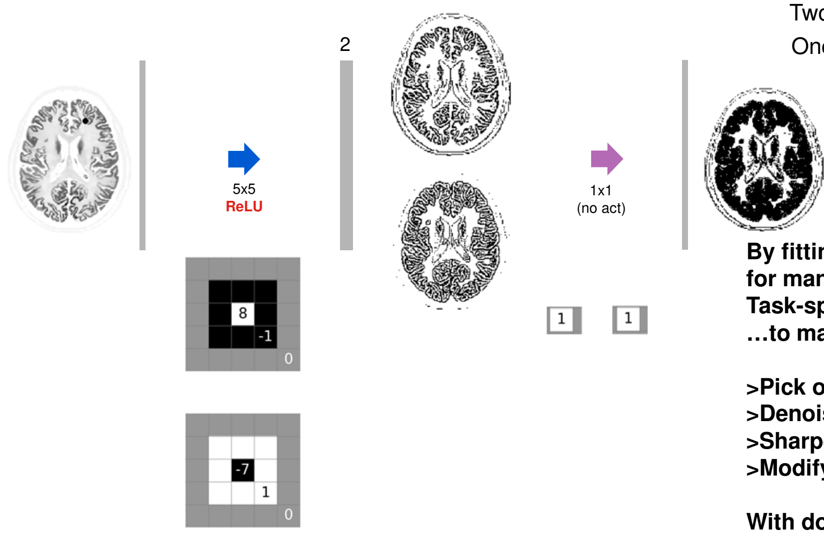
16

Convolutional neural networks (CNNs): multiple kernels

Architecture:

Two, 1 channel, 5 × 5 kernels (1st layer)

One, 2 channel, 1 × 1 kernel (2nd layer)



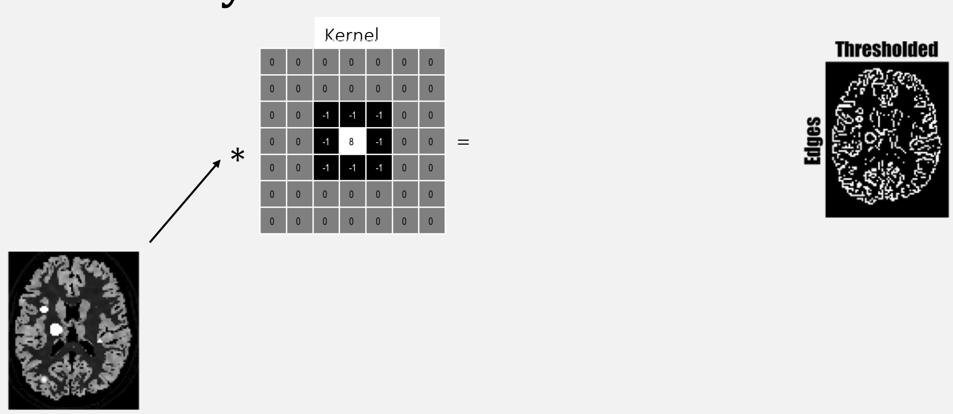
By fitting kernel parameters & biases…
for many kernels and many layers…
Task-specific processing can be learned
…to map inputs to desired outputs

>Pick out features
>Denoise
>Sharpen
>Modify!

With downsampling:
Can learn compositions of features, feature hierarchies, for increasing abstraction

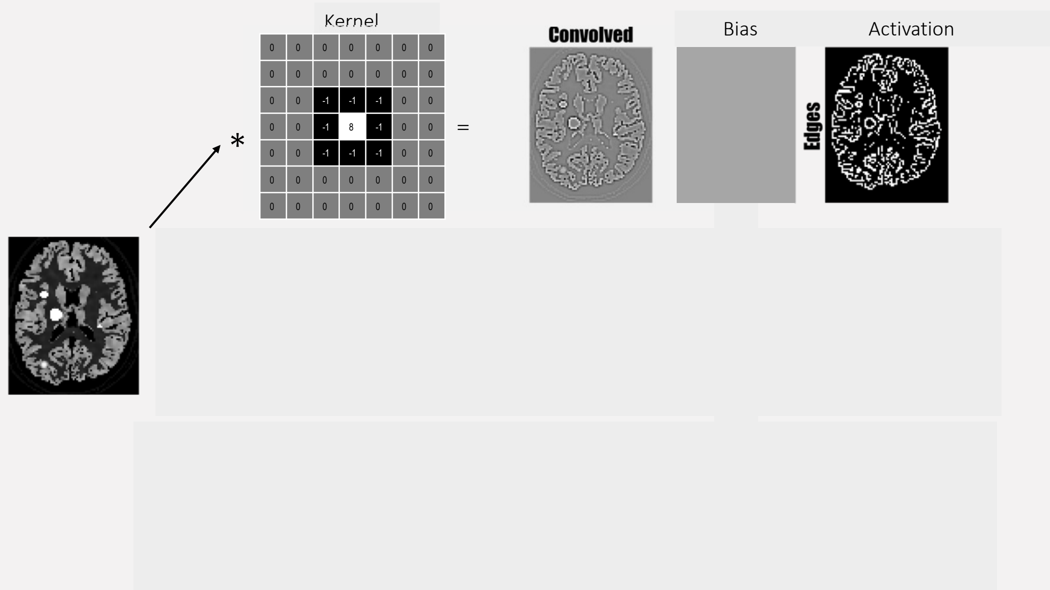
17

Convolutional layer with 3 kernels



18

Convolutional layer with 3 kernels

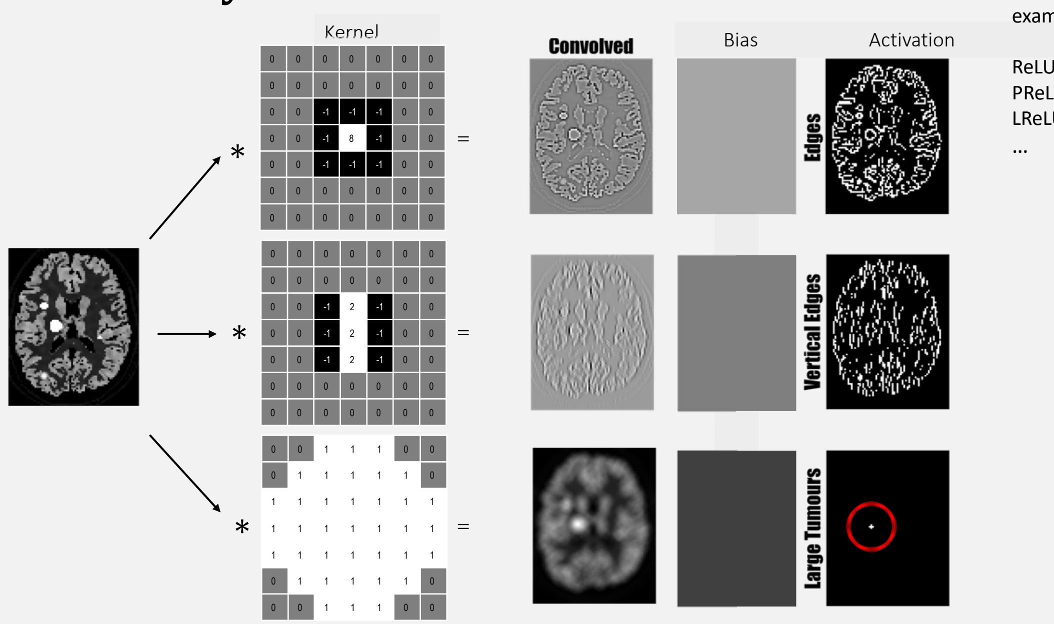


This is called a "Convolutional Layer"

3 kernels → 3 output channels

19

Convolutional layer with 3 kernels



Activation function examples:

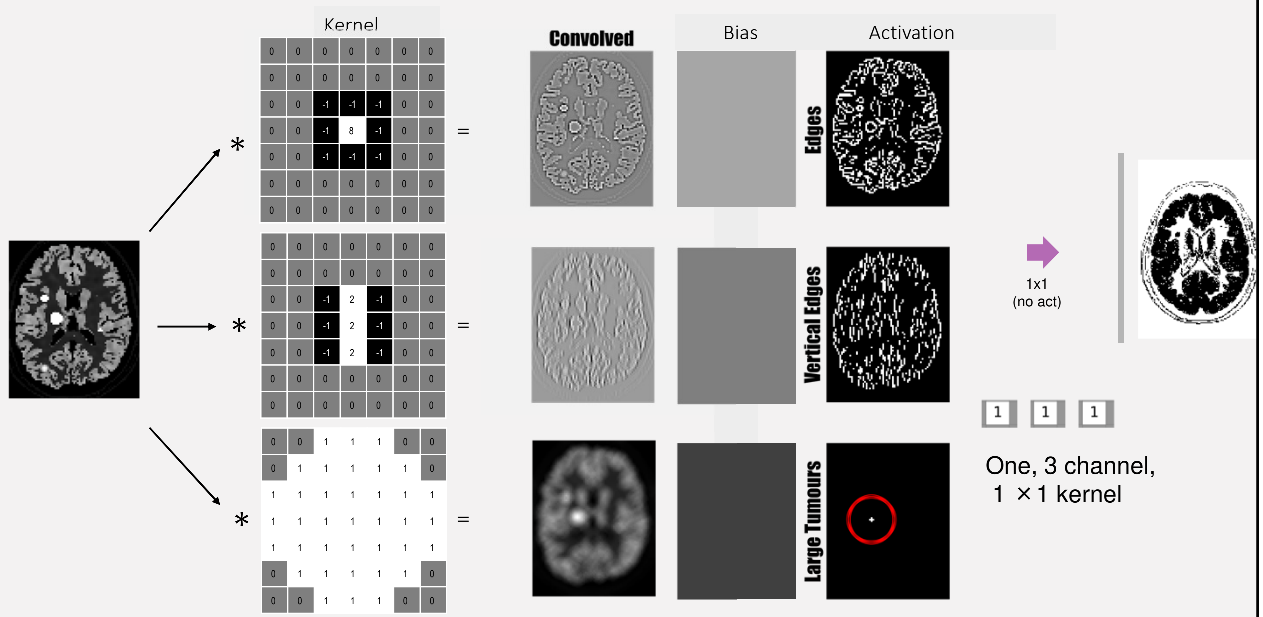
ReLU
PReLU
LReLU
...

This is called a "Convolutional Layer"

3 kernels → 3 output channels

20

Convolutional layer with 3 kernels



This is called a "Convolutional Layer"

3 kernels → 3 output channels

21

Deep learning components

1. **Training data**
   From no training data….
   …to tens of examples pairs… to thousands

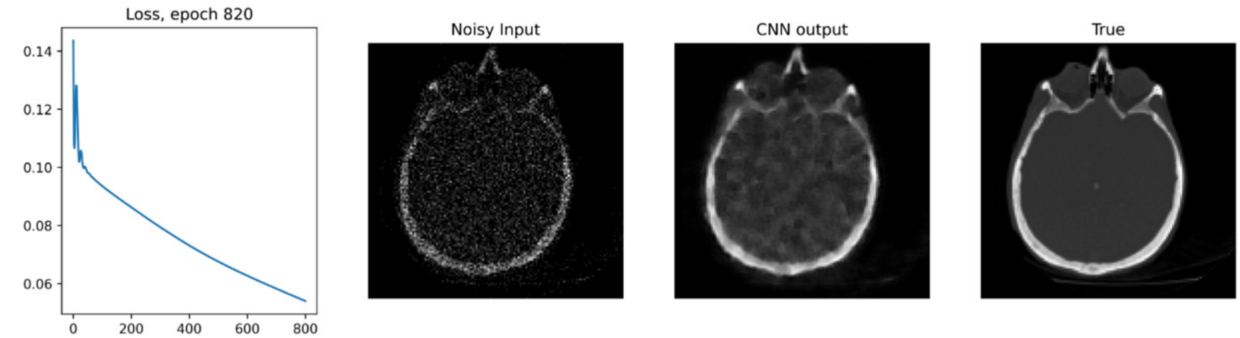
2. **Architecture / inductive prior for the mapping from input to output**
   Trainable parameters for a code structure
   E.g. fully-connected (linear) layers, convolutional neural networks (CNNs), transformers

3. **Loss functions to decide how well a mapping is doing its job**
   Mean squared error (MSE) or L2 norm
   Mean absolute error (MAE) or L1 norm
   Perceptual loss
   Adversarial loss

4. **Optimisers**
   Stochastic gradient descent (SGD)
   Adam
   …and many more

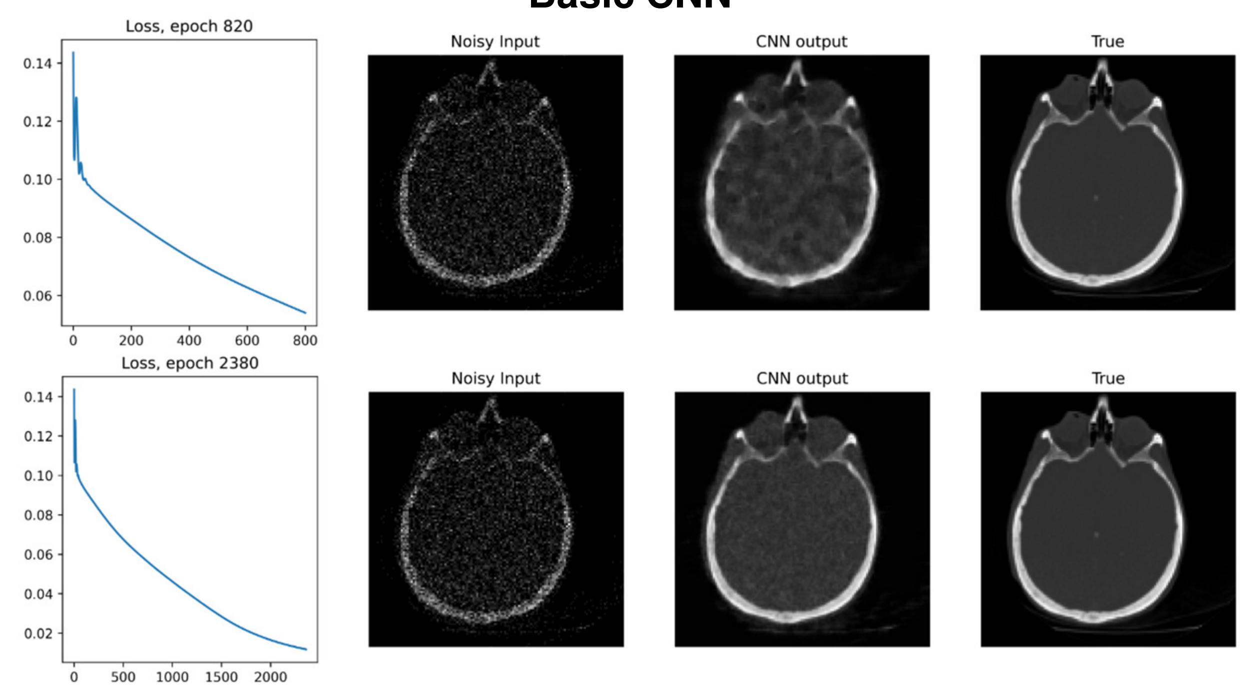
22

Basic CNN



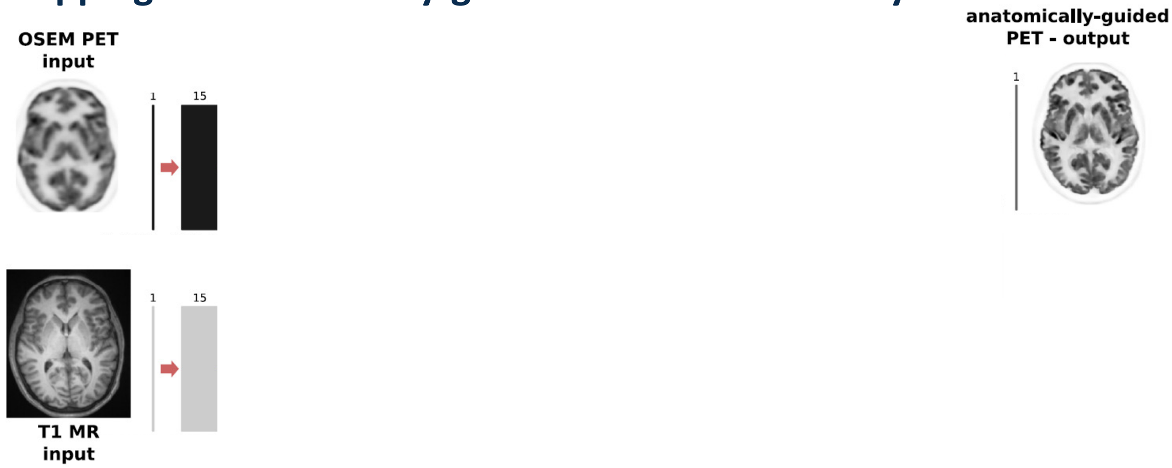
23

Basic CNN



24

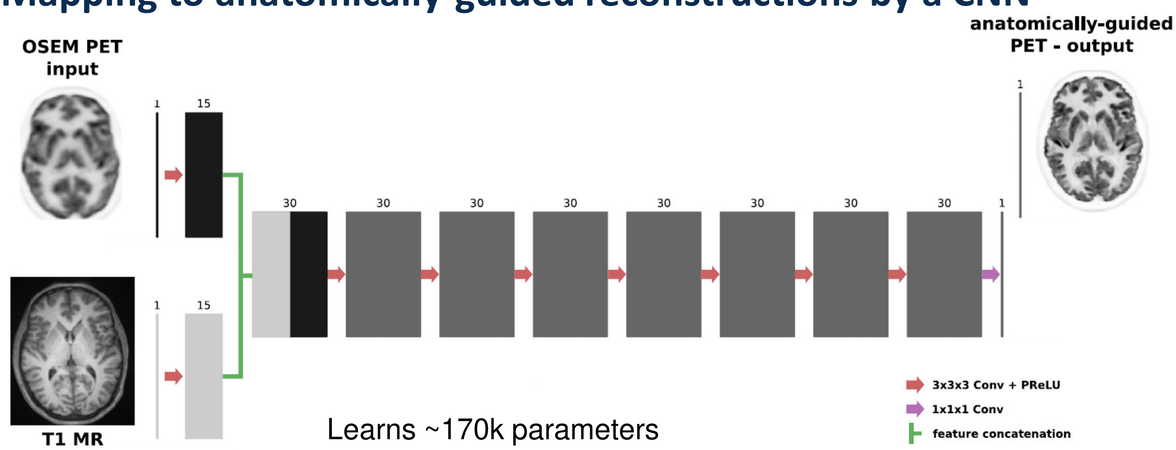
Mapping to anatomically-guided reconstructions by a CNN



Schramm *et al*, Neuroimage 2021 (https://github.com/gschramm/pyapetnet)

25

Mapping to anatomically-guided reconstructions by a CNN

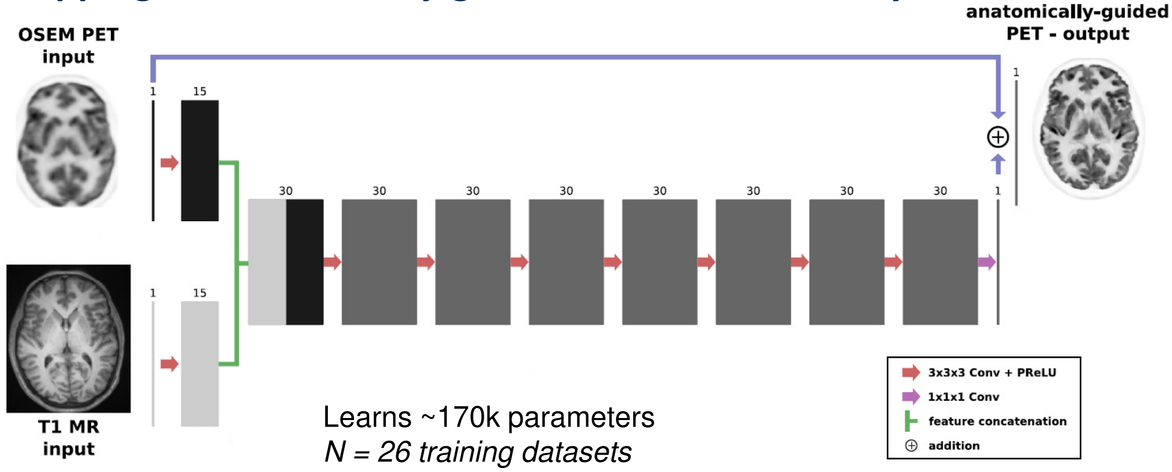


Learns ~170k parameters
N = 26 training datasets
(2 tracers, 2 scanners)
MSE loss

Schramm *et al*, Neuroimage 2021 (https://github.com/gschramm/pyapetnet)

26

Mapping to anatomically-guided reconstructions by a CNN



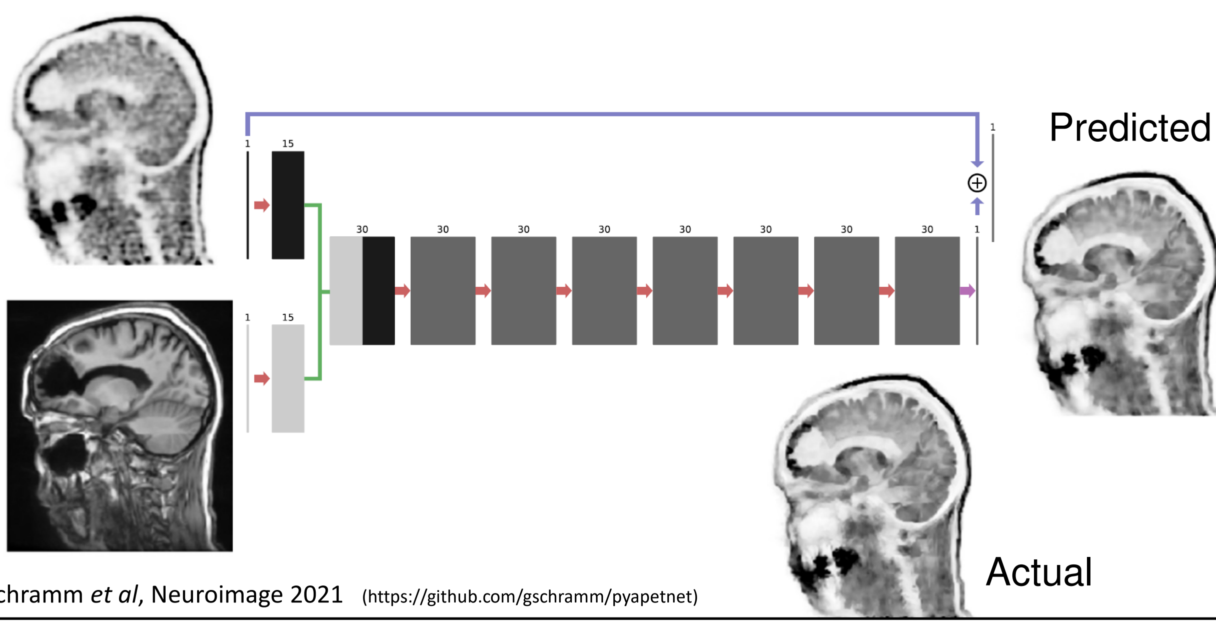
Learns ~170k parameters
N = 26 training datasets
(2 tracers, 2 scanners)
MSE loss

Order(s) of magnitude faster:
CNN: ~ 1min (CPU), ~1s (GPU)
MAP-EM: 10 – 100+ mins (CPU)

Schramm *et al*, Neuroimage 2021 (https://github.com/gschramm/pyapetnet)

27

Testing on a different radiotracer [^{18}F]FET



Schramm *et al*, Neuroimage 2021 (https://github.com/gschramm/pyapetnet)

28

So far:
No downsampling / upsampling: shift-equivariant mappings

Suitable for image to image mappings

29

CNNs can do more

Using downsampling (or max pooling) / upsampling

Non-linear, shift-variant mappings

Suitable for sinogram to image mappings

30

Hierarchy of features – increasing levels of abstraction

"Atoms" …. *"Molecules"* …. *"Materials"* … *"Objects"* ….

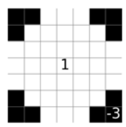
Feature map shows locations of features

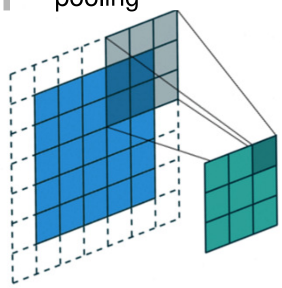
Feature map shows location of cluster of features

7x7
ReLU

Can downsample by stride, or max pooling

… → …





By Vincent Dumoulin, Francesco Visin - https://github.com/vdumoulin/conv_arithmetic, MIT, https://commons.wikimedia.org/w/index.php?curid=78003456

31

Hierarchy of features – increasing levels of abstraction

"Atoms" …. *"Molecules"* …. *"Materials"* … *"Objects"* ….

Feature map shows locations of features

Feature map shows location of cluster of features

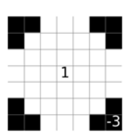
7x7
ReLU

Can downsample by stride, or max pooling

… → …

Feature maps / latent space used by object detection networks

Or decoded back to images



Many "atoms"

Even more "molecules"

Still more "materials"

High spatial sampling

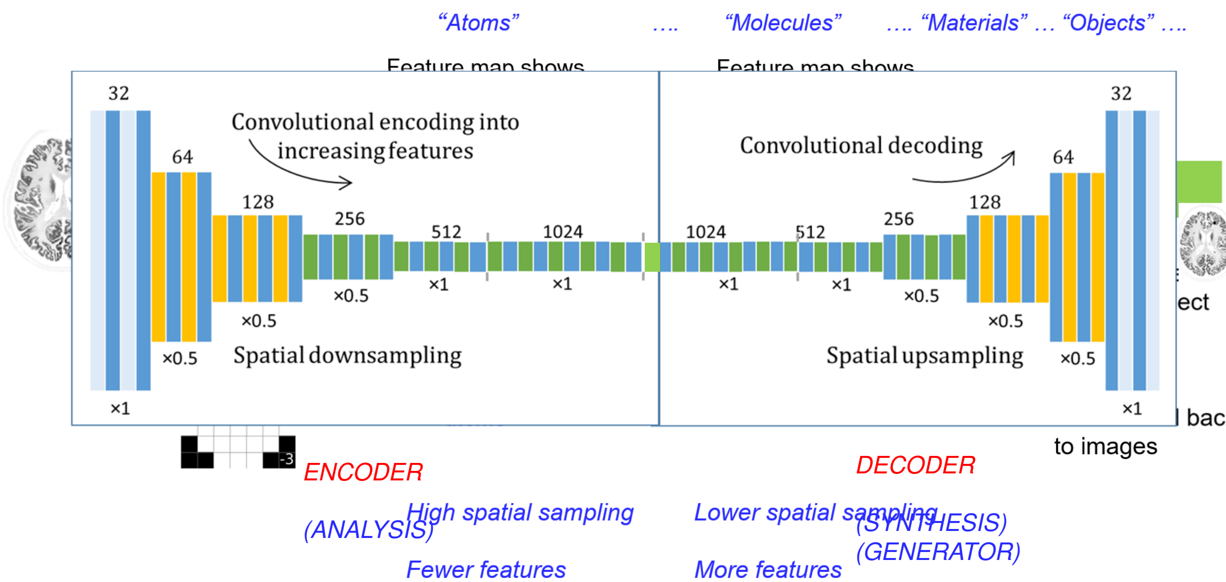
Fewer features

Lower spatial sampling

More features

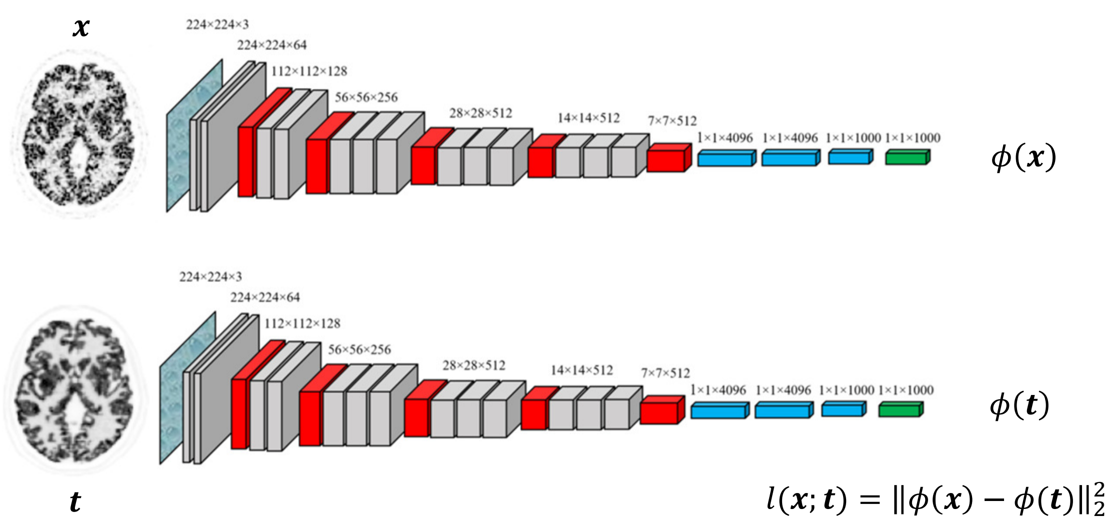
32

Hierarchy of features – increasing levels of abstraction



33

Previously had mentioned MSE, MAE loss… but can also have Perceptual loss (use latent space feature maps for comparison of estimate with desired target)



$$l(x;t) = \|\phi(x) - \phi(t)\|_2^2$$

VGG network: Simonyan and Zisserman 2015

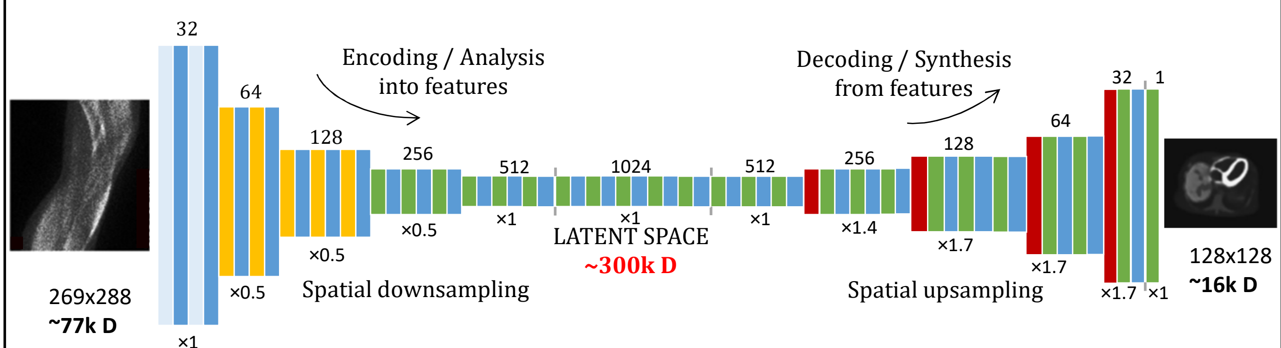
34

Sinograms to image

Direct reconstruction

35

Direct DL reconstruction from sinograms: DeepPET



Conv 7×7 | Conv 5×5 | Conv 3×3 | BN + ReLU | Upsampling

Add on adversarial network: DPIR-Net Hu et al, TRPMS 2020

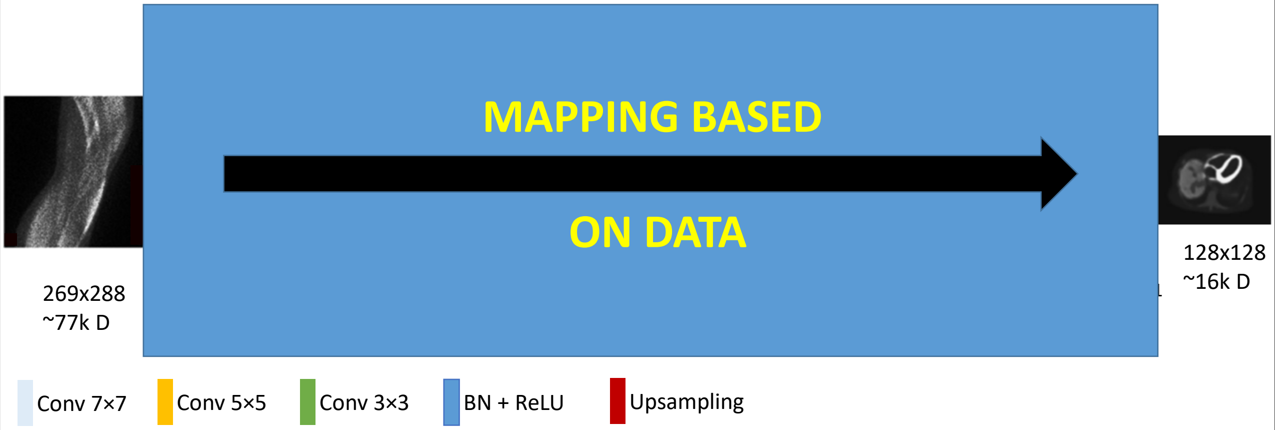
Convolutional encoder decoder:

- **200,000 simulated training data pairs** to learn **~60 million parameters** via MSE loss
- (Simple linear mapping would need > 1 billion parameters)

Figure represents method of Häggström *et al*, MIA 2019

36

Direct DL reconstruction from sinograms: DeepPET



Convolutional encoder decoder:

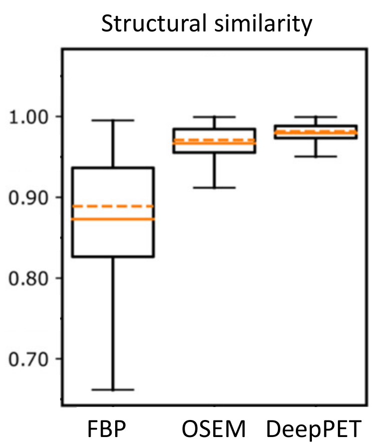
- **200,000 simulated training data pairs** to learn **~60 million parameters** via MSE
- Note a simple linear mapping would need > 1 billion parameters

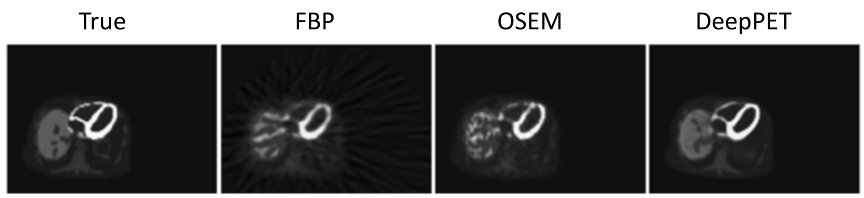
Figure represents method of Häggström *et al*, MIA 2019

37

DeepPET: results

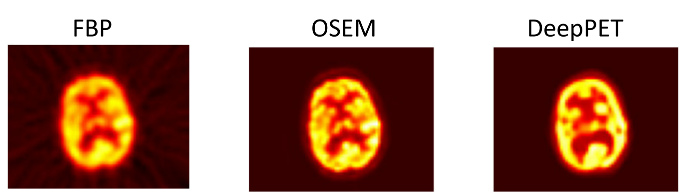
Simulated data





✓ **Within distribution**

Real data



- **Outside distribution**

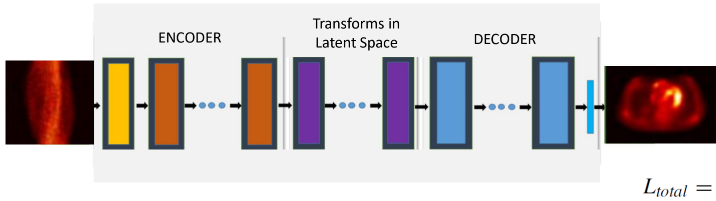
~10% lower noise (RMSE) than OSEM conventional reconstruction
~100x faster than OSEM

Häggström *et al*, MIA 2019

+discriminator, perceptual loss, DPIR-Net: Zhanli Hu et al, IEEE TRPMS (5) Jan 2021

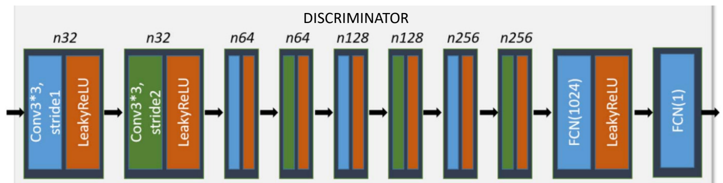
38

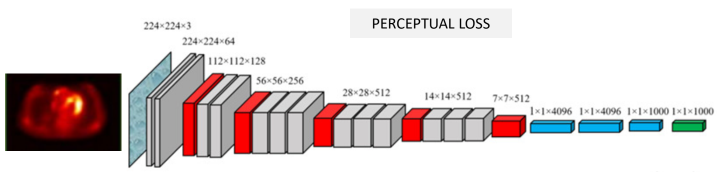
Adding perceptual loss, and a discriminator: DPIR-Net



Karras et al ICLR 2018

$$L_{total} = \min_G \max_D L_{WGAN}(D, G) + \lambda_1 * L_{vgg} + \lambda_2 * L_{mse}$$



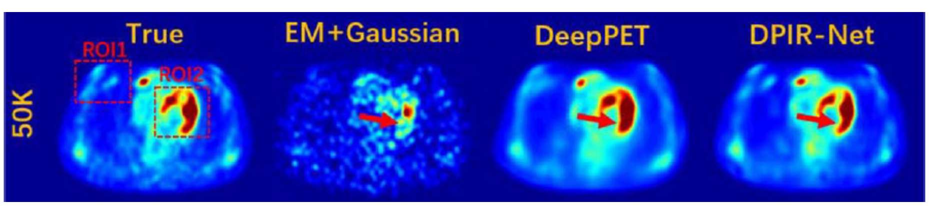


Nshafiei Creative Commons Attribution-ShareAlike 4.0

Zhanli Hu et al, IEEE TRPMS (5) Jan 2021

39

Adding a discriminator and perceptual loss - DPIR-Net

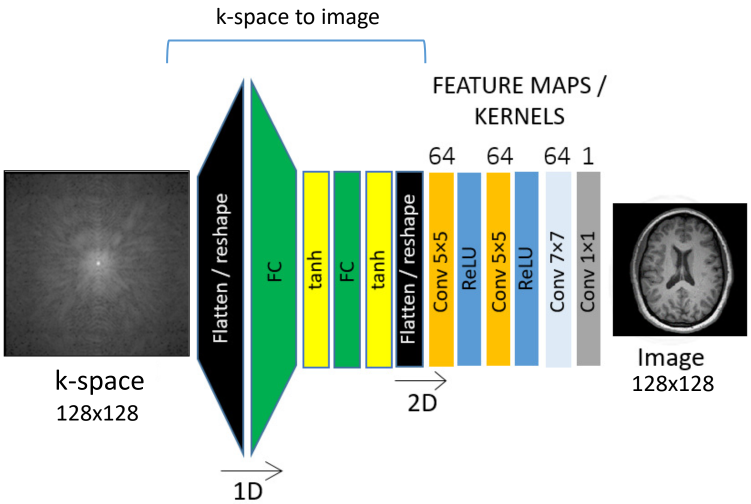


Method	50K		500K		5M	
	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM
EM+ Gaussian	18.029	0.305	18.649	0.723	21.245	0.736
DeepPET	31.984	0.916	34.510	0.935	36.077	0.943
DPIR-Net	**40.437**	**0.917**	**43.556**	**0.954**	**43.793**	**0.963**

Zhanli Hu et al, IEEE TRPMS (5) Jan 2021

40

MRI reconstruction from k-space: AUTOMAP



Training on MR brain data only yielded best results (compared to natural images, or random noise as ground truth examples)

Reduced noise and artefacts compared to conventional reconstruction (RMSE nearly halved)

Faster reconstruction!

Demonstrated for PET data also.

Uses ~ **50,000 training data pairs** to learn **~800 million parameters**

Figure represents method of Zhu *et al*, Nature 2018

41

COMPARISON OF DIRECT RECONSTRUCTION METHODS

	NAME	ARCHITECTURE	LOSS FUNCTION	DATA SIZES	NUMBER OF TRAINING PAIRS
PET	**DeepPET** Häggström et al. MIA 2019	CNN (CED) [>60 million parameters]	MSE	269x288 to 128x128 (2D)	~200,000
	DPIR-Net Hu et al. TRPMS 2020	As above + discriminator [>60 million parameters]	MSE, perceptual loss, discriminator	269x288 to 128x128 (2D)	~40,000
MR	**AUTOMAP** Zhu et al. Nature 2018	FC layers, CNN [>800 million parameters]	MSE with L1 penalty	128x128 to 128x128 (2D)	~50,000
CT	**iCT-Net** Li et al, IEEE TMI 2019	CNN+FC [~ <1 million parameters]	MSE	512x512 (2D)	58 real scans [millions of simulated samples for pre-training]
	DirectPET Whiteley et al. MIC 2019 J. Med. Imag. 2020	FC layers, CNN [~350 million parameters]	MAE and perceptual loss and MS-SSIM	400x168x16 to 400x400x16	~2,000

42

Direct DL reconstruction summary

Examples: DeepPET, DPIR-Net, AUTOMAP, …

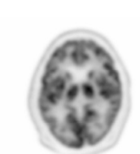
Häggström *et al*, MIA 2019, Zhanli Hu *et al*, IEEE TRPMS (5) Jan 2021, Zhu *et al* Nature 2018

- ✓ Few model assumptions (less modeling error)
- ✓ Data driven, just the network's inductive prior
- ✓ Fast reconstructions

- Slow training
- **Huge data needs** (>>10k data pairs)
- Relearns physics & statistics
- Huge network (10-100 million parameters, just for 2D)
- Applied mainly for **2D reconstruction**, not yet really for fully 3D
- Generalisation / outside training distribution?
- Stability? Antun *et al*, PNAS 2020

Fast PET (for TOF histo images)

Simply a U-Net type architecture, only 20 million parameters
Works for full 3D volumes
67x faster than OSEM
Noise reductions



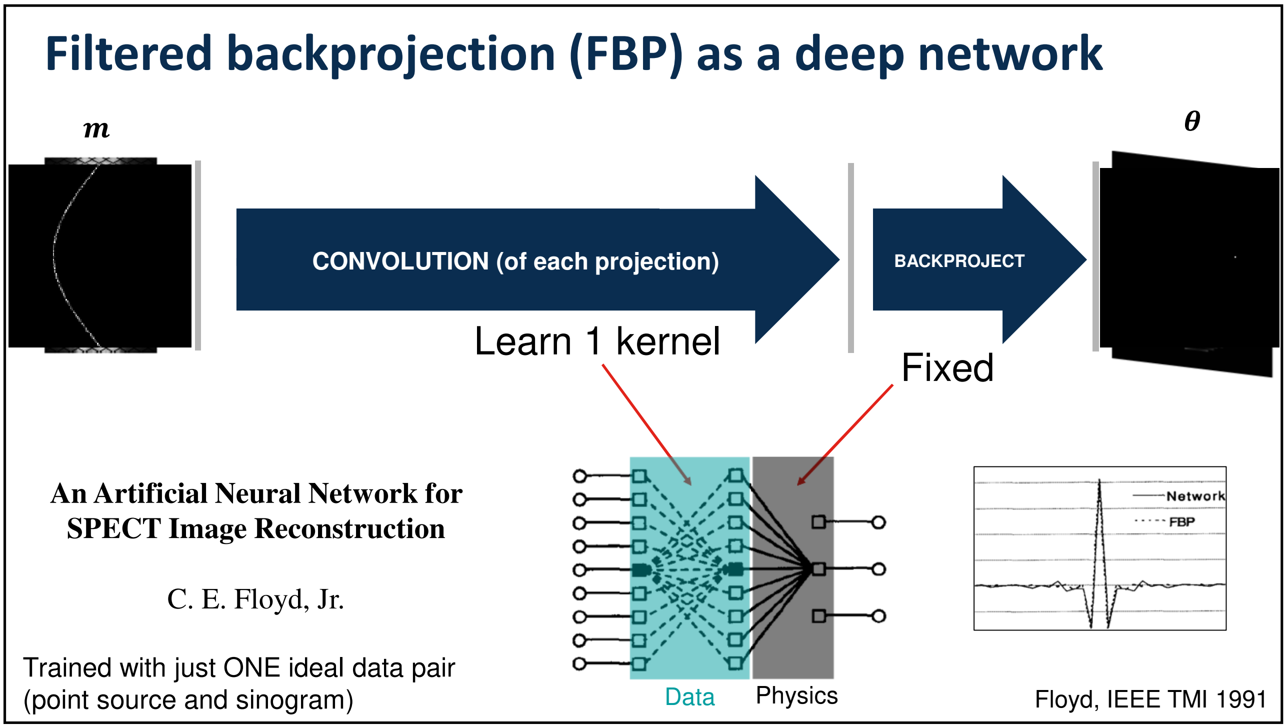
Target

Whiteley et al. IEEE TRPMS 2021

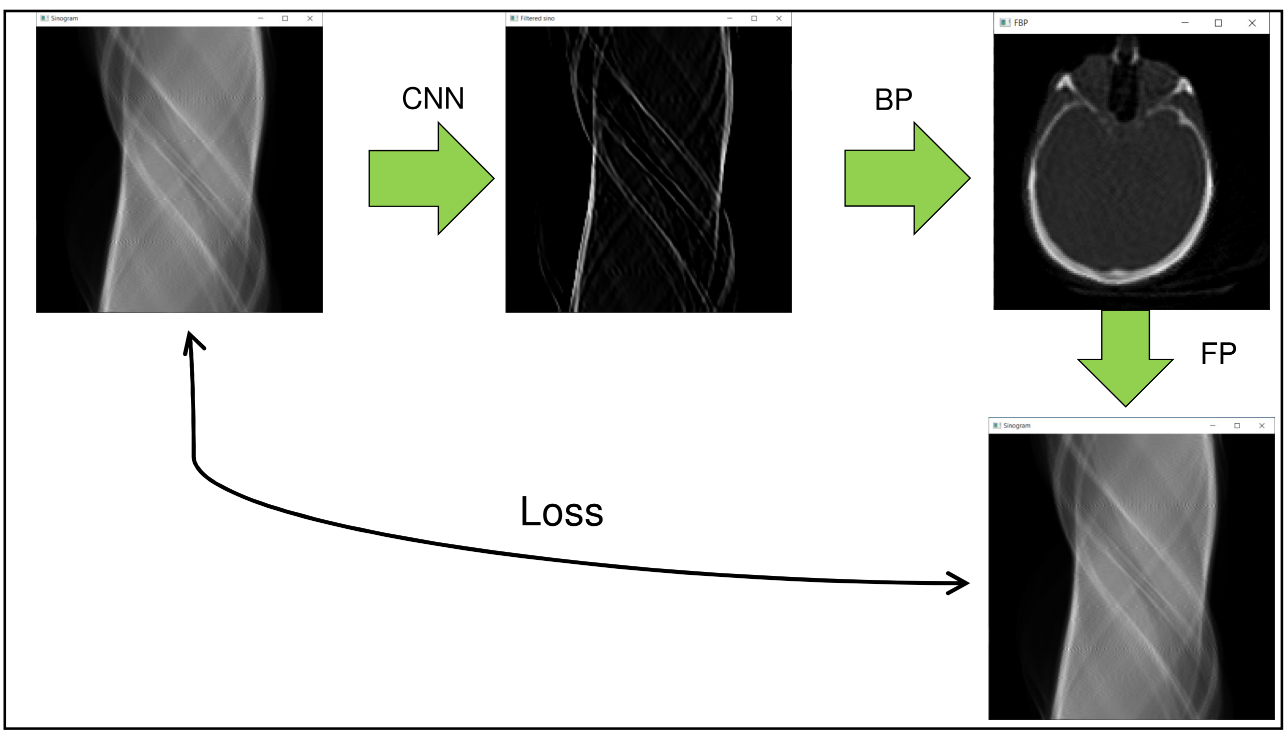
Learned FBP

Filtered backprojection (FBP)

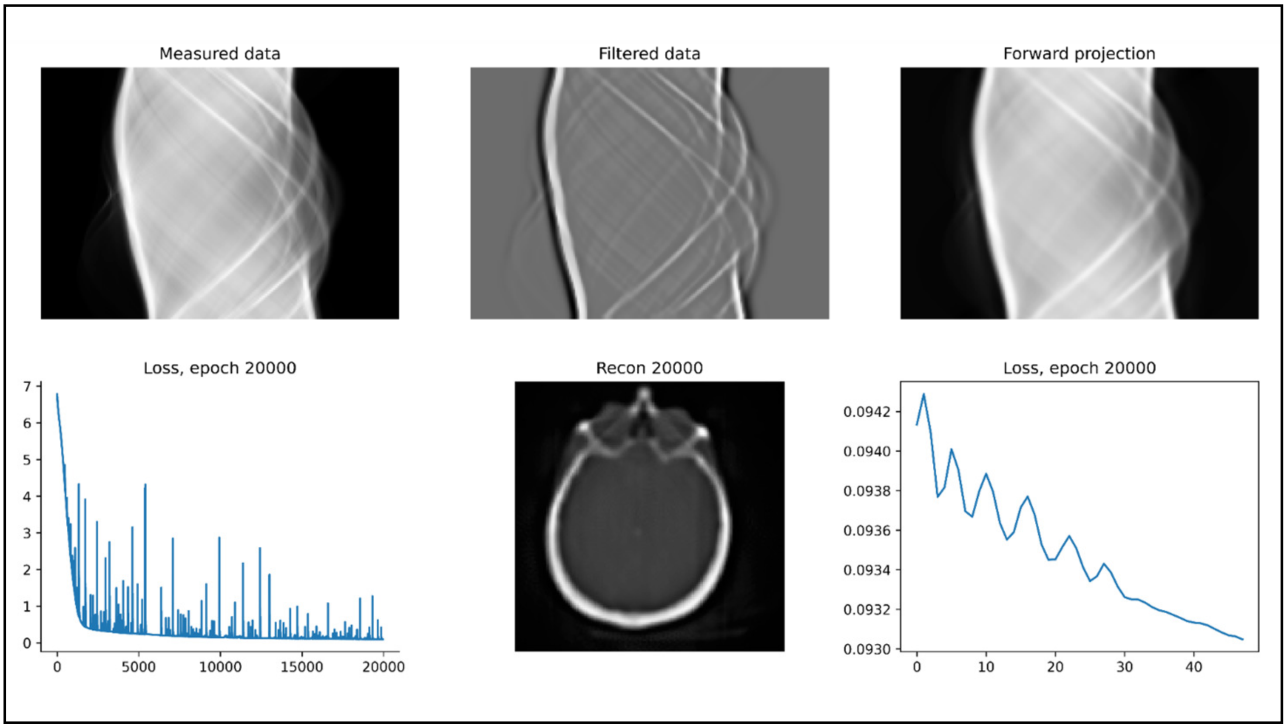

```
sinogram = radon(test_object, azi_angles);

ft_sinogram = zeros(size(sinogram));
ramp_ft_sino = zeros(size(sinogram));
filtered_sino = zeros(size(sinogram));  rd = size(sinogram,1);

ramp_filter = [0:floor(rd/2) floor(rd/2)-1:-1:0];
for angle=1:num_angles
    pp = sinogram(:,angle);
    pp_ft = fft(pp);
    ft_sinogram(:,angle)=pp_ft;
    pp = ft_sinogram(:,angle);
    ramp_ft_sino(:,angle) = ft_sinogram(:,angle) .* ramp_filter(:);
    filtered_sino(:,angle)=real(ifft(ramp_ft_sino(:,angle)));
end

bp_rec = iradon(sinogram, azi_angles,'none', xd);
fbp_rec = iradon(filtered_sino, azi_angles,'none', xd);
```


Filtered backprojection (FBP) as a deep network
m
θ
CONVOLUTION (of each projection)
BACKPROJECT
Learn 1 kernel
Fixed
An Artificial Neural Network for SPECT Image Reconstruction
C. E. Floyd, Jr.
Network
FBP
Data
Physics
Trained with just ONE ideal data pair (point source and sinogram)
Floyd, IEEE TMI 1991

47


CNN
BP
FP
Loss

48



49

Deep Image Prior

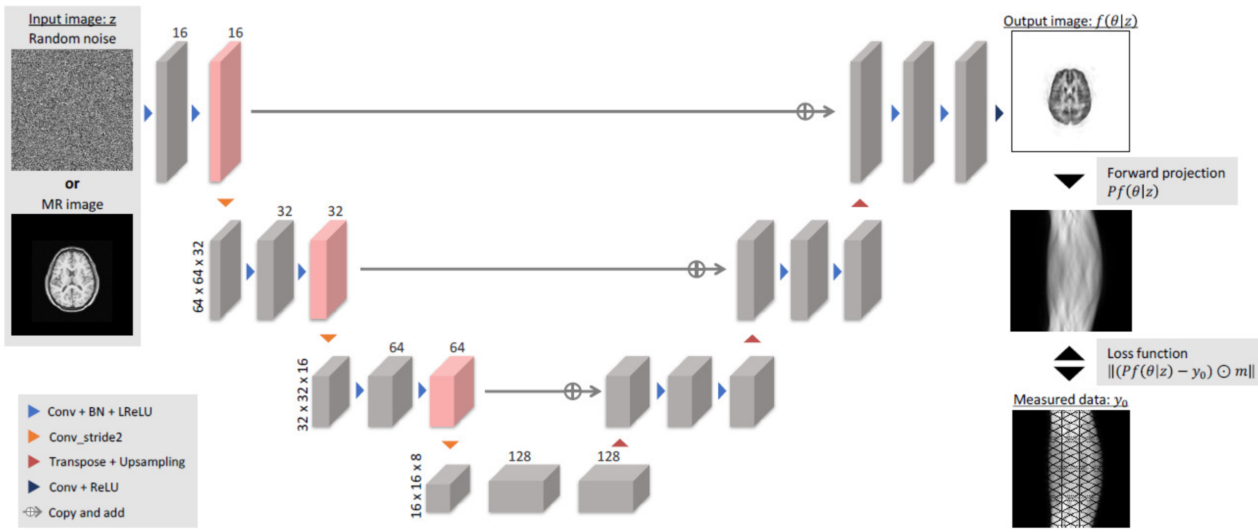
50

Deep image prior
with system model

Deep Image Prior

2017

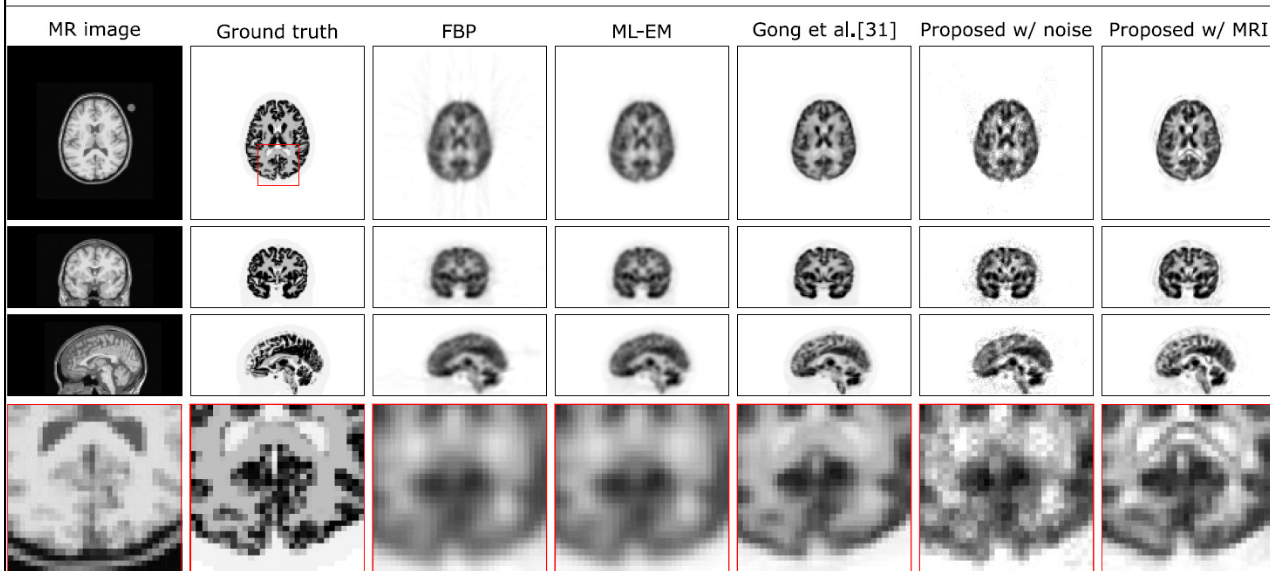
Dmitry Ulyanov · Andrea Vedaldi · Victor Lempitsky



Hashimoto et al IEEE TRPMS 2022

51

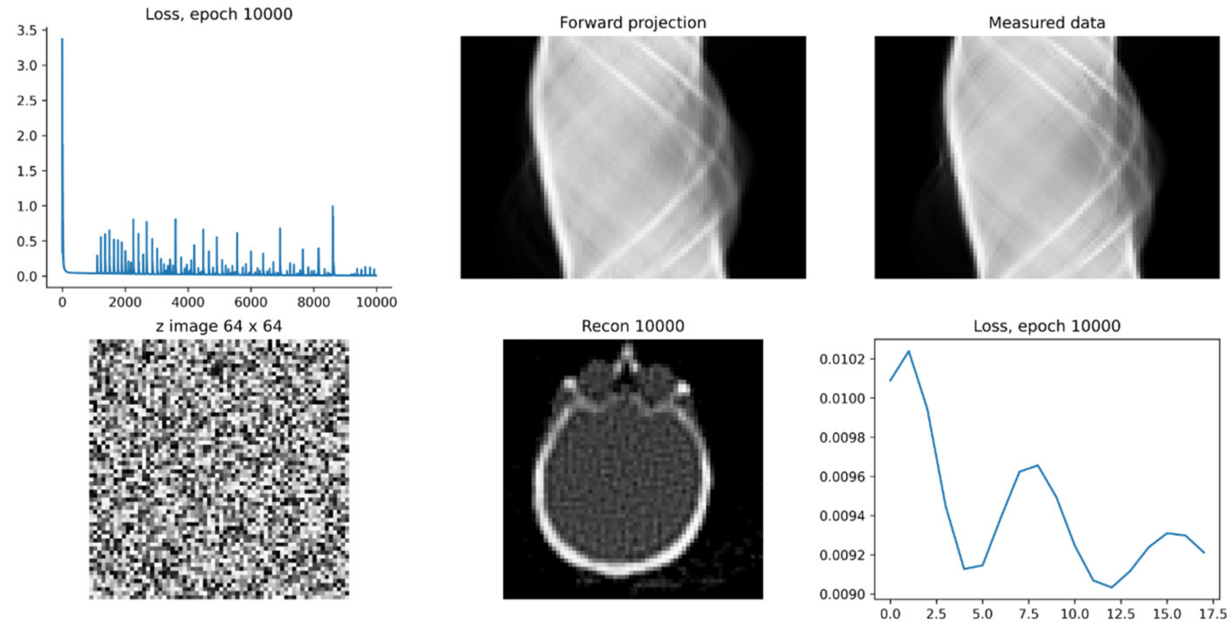
Deep image prior with system model



Hashimoto et al IEEE TRPMS 2022

52

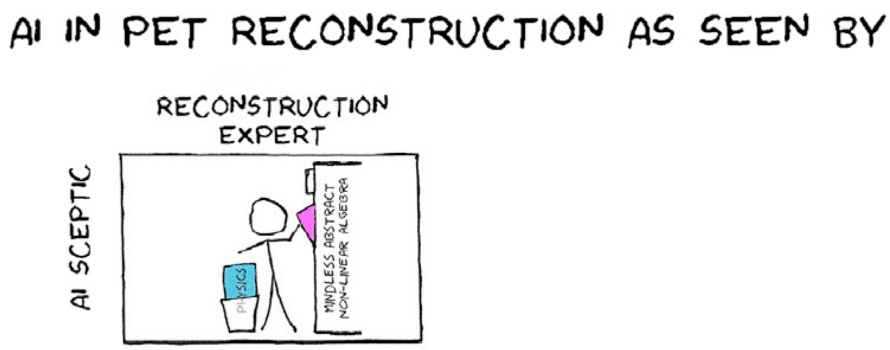
DEEP IMAGE PRIOR



53

Iterative reconstruction with DL

54



Journal of Nuclear Medicine October 2021, 62 (10) 1330-1333; DOI: https://doi.org/10.2967/jnumed.121.262303

HOT TOPICS

Artificial Intelligence for PET Image Reconstruction

Andrew J. Reader[1] and Georg Schramm[2]

[1]School of Biomedical Engineering and Imaging Sciences, King's College, London, United Kingdom; and [2]Division of Nuclear Medicine, Department of Imaging and Pathology, KU/UZ, Leuven, Belgium

55

Embedding deep learning into iterative reconstruction

Unrolled iterative methods:

✓ Iterative reconstruction uses physics and statistics modelling and theoretically convergent algorithms

✓ use DL for the regularisation (the prior, defined by the image manifold of the training data)

Compared to direct DL

- ✓ **Practical for 3D**
- ✓ **Reduced** training **data** needs (~*tens* of 3D images)
- ✓ Expect improved generalisation outside the training distribution

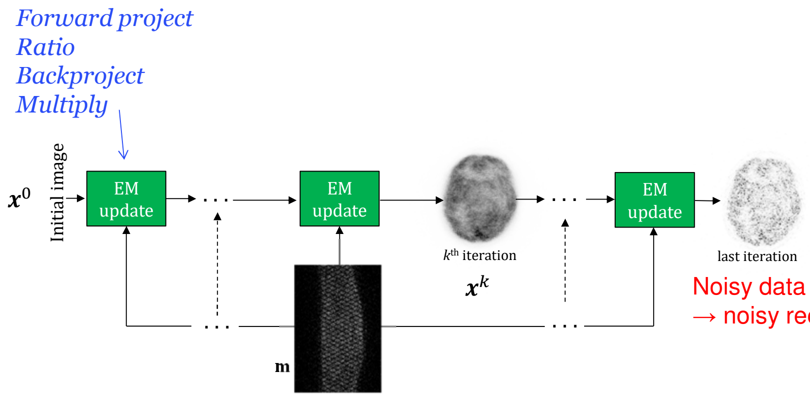
- **Examples**
  - Lim *et al* 2018 (BCD-Net for low count PET), TMI 2020 (Iterative NN)
  - Gong *et al* 2019 (MAPEM-Net)
  - Mehranian and Reader 2020 (FBSEM-Net)
  - Rui Hu, Huafeng Liu 2022 (TransEM)

56

Conventional ML-EM (and OSEM)

$$x^{n+1} = \frac{x^n}{A^T \mathbf{1}} A^T \left(\frac{m}{Ax^n} \right)$$

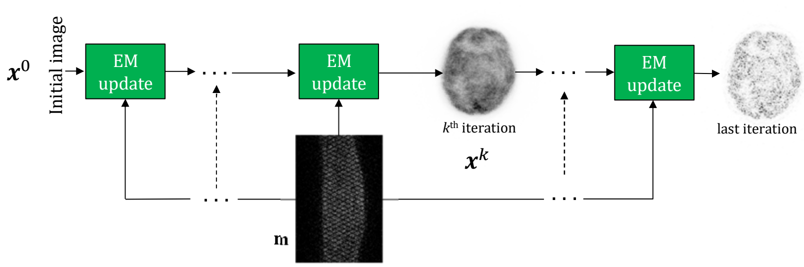
Iterative reconstruction unrolled into a **deep network**:



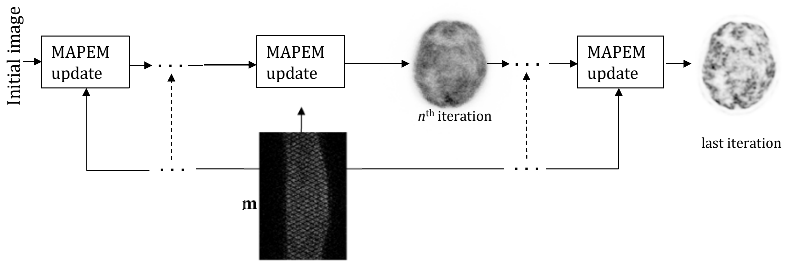
Noisy data (time, dose, scanner)
→ noisy reconstructions

Unrolling iterative recon: Gregor & LeCun 2010

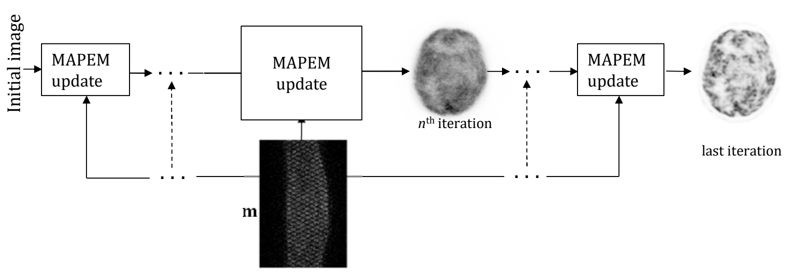
57



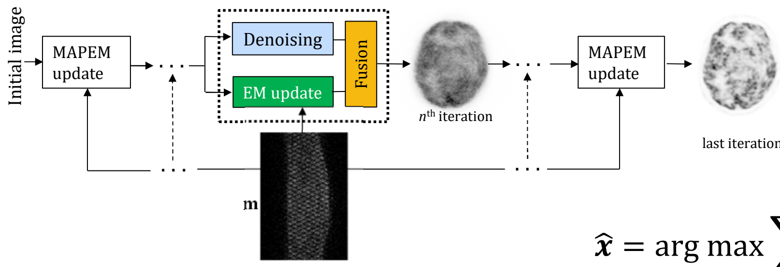
58


Initial image
MAPEM update
MAPEM update
n^{th} iteration
MAPEM update
last iteration
m

59


Initial image
MAPEM update
MAPEM update
n^{th} iteration
MAPEM update
last iteration
m

60



$$\hat{\boldsymbol{x}} = \arg\max_{\boldsymbol{x}} \sum_{i=1}^{I} \left(m_i \ln(\boldsymbol{A}\boldsymbol{x})_i - (\boldsymbol{A}\boldsymbol{x})_i\right) - \beta U(\boldsymbol{x})$$

$$x_j^{EM} = \frac{x_j^k}{\sum_{i=1}^{I} a_{ij}} \sum_{i=1}^{I} a_{ij} \frac{m_i}{(\boldsymbol{A}\boldsymbol{x}^k)_i}$$

$$x_j^{SM} = \frac{1}{2\sum_{l=1}^{J} w_{jl}} \sum_{l=1}^{J} w_{jl}\left(x_l^k + x_j^k\right)$$

De Pierro TMI 1995

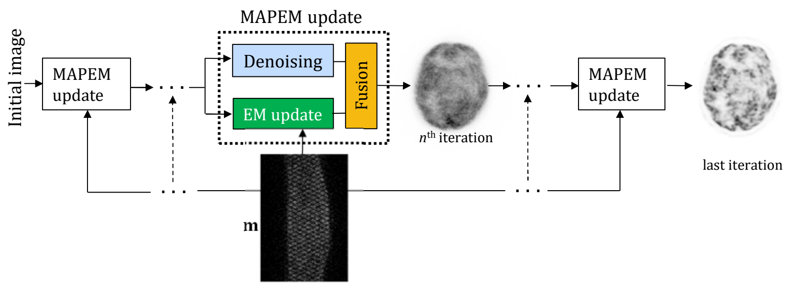
$$x_j^{k+1} = \frac{2x_j^{EM}}{\left(1 - \beta v_j x_j^{SM}\right) + \sqrt{\left(1 - \beta v_j x_j^{SM}\right)^2 + 4\beta v_j x_j^{EM}}}$$

$$v_j = \frac{\sum_{l=1}^{J} w_{jl}}{s_j}$$

$$s_j = (\boldsymbol{A}^{\mathrm{T}}\boldsymbol{1})_j$$

61

Example of PET regularisation



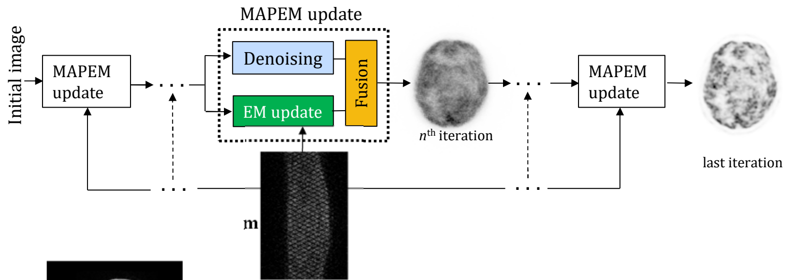
Mathematically convenient
(e.g. quadratic prior)

Unlikely to be optimal

De Pierro TMI 1995

62

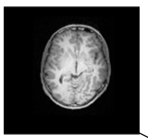
Example of PET regularisation



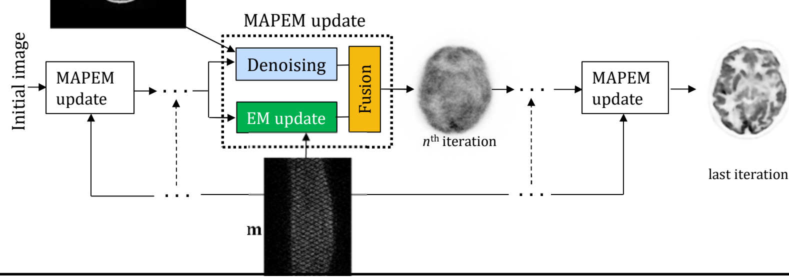
Mathematically convenient (e.g. quadratic prior)

Unlikely to be optimal

De Pierro TMI 1995



Can use MR to guide the denoising



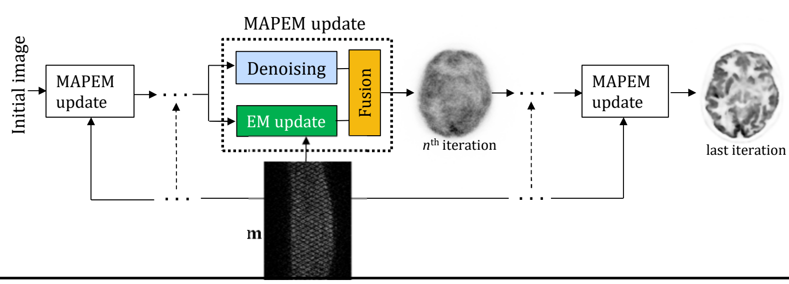
How much MR guidance?

Unlikely to be optimal

…use deep learning

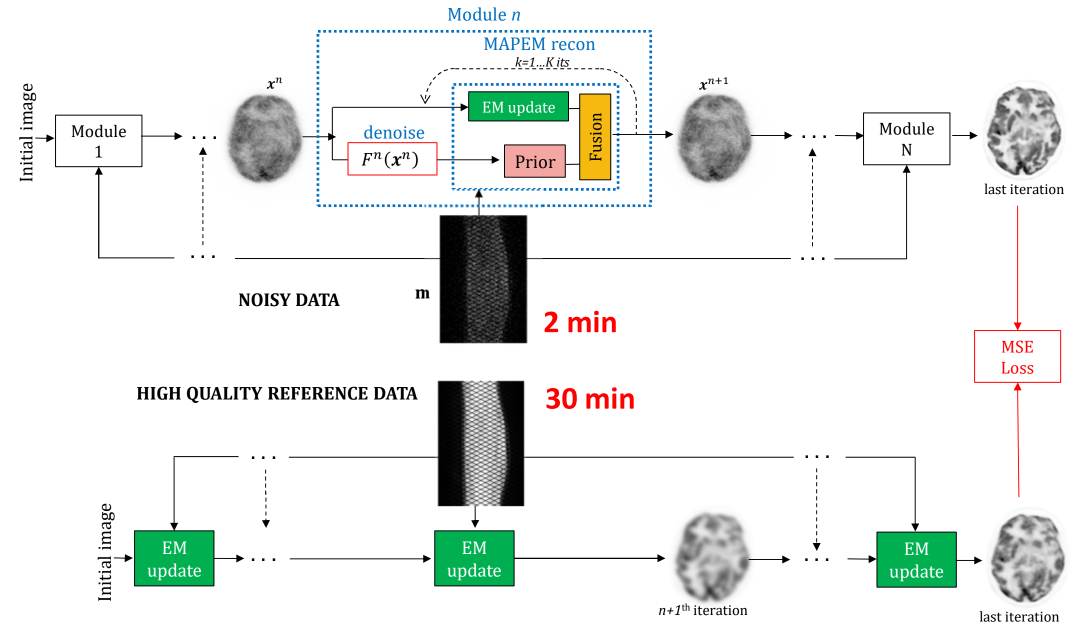
Bowsher IEEE NSS MIC 2004

63



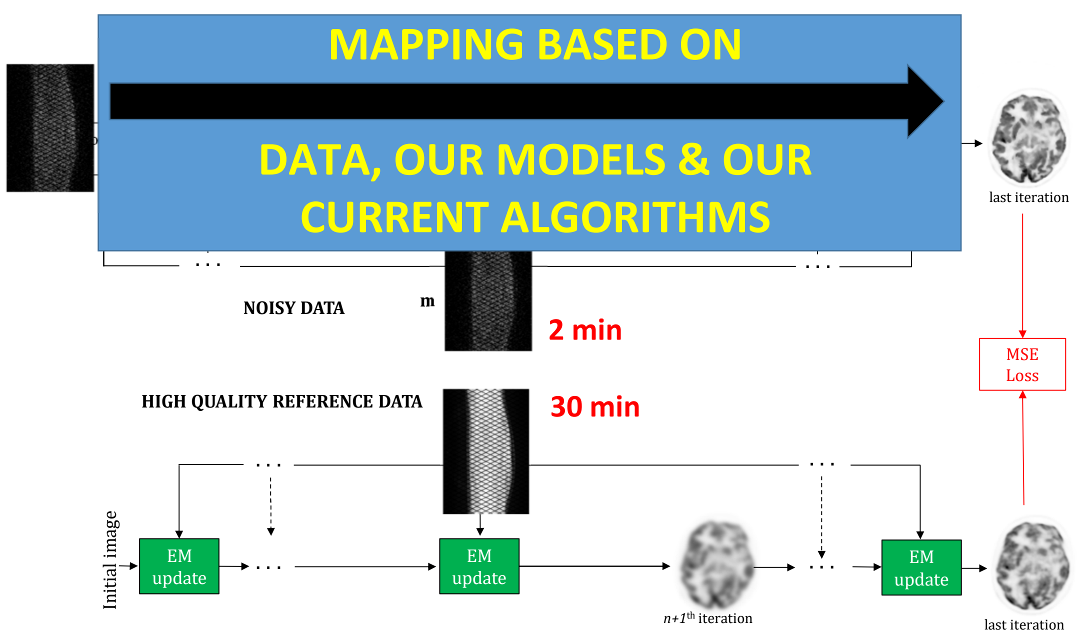
64

DL unrolled reconstruction for 3D PET



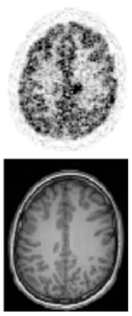
65

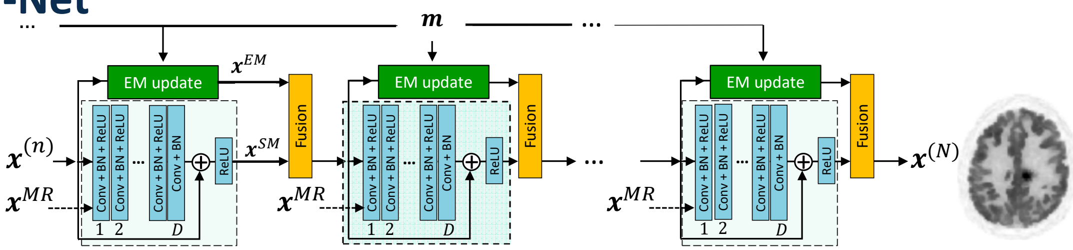
DL unrolled reconstruction for 3D PET



66

FBSEM-Net





$$\hat{\boldsymbol{x}} = \arg\max_{\boldsymbol{x}} \sum_{i=1}^{I} \left(m_i \ln(\boldsymbol{Ax})_i - (\boldsymbol{Ax})_i\right) - \beta U(\boldsymbol{x})$$

Given current $\boldsymbol{x}^n$:

$$\boldsymbol{x}^{SM} = \boldsymbol{x}^n - \gamma\beta\nabla U(\boldsymbol{x}^n) = F(\boldsymbol{x}^n)$$

Learned regularisation

Proximity operator

$$\boldsymbol{x}^{n+1} = \underset{\boldsymbol{x}}{\mathrm{argmax}}\, L(\boldsymbol{x}|\boldsymbol{m}) - \frac{1}{2\gamma}\|\boldsymbol{x} - \boldsymbol{x}^{SM}\|^2$$

Poisson log-likelihood with proximity to denoised update

Complete data log-likelihood as a surrogate for $L(\boldsymbol{x}|\boldsymbol{m})$, then can use EM update image

$$\delta_j = \frac{1}{\gamma}\frac{1}{(\boldsymbol{A}^T\boldsymbol{1})_j}$$

$$\boldsymbol{x}^{EM} = \frac{\boldsymbol{x}^n}{\boldsymbol{A}^T\boldsymbol{1}}\boldsymbol{A}^T\left(\frac{\boldsymbol{m}}{\boldsymbol{A}\boldsymbol{x}^n + \boldsymbol{\rho}}\right)$$

Simple quadratic penalty, so can use:

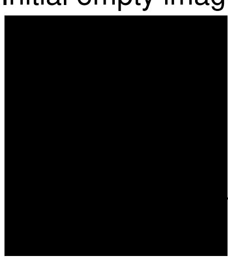
$$x_j^{n+1} = \frac{2x_j^{EM}}{\left(1-\delta_j x_j^{SM}\right) + \sqrt{\left(1-\delta_j x_j^{SM}\right)^2 + 4\delta_j x_j^{EM}}}$$

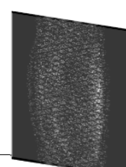
Mehranian and Reader, IEEE TRPMS 2020

67

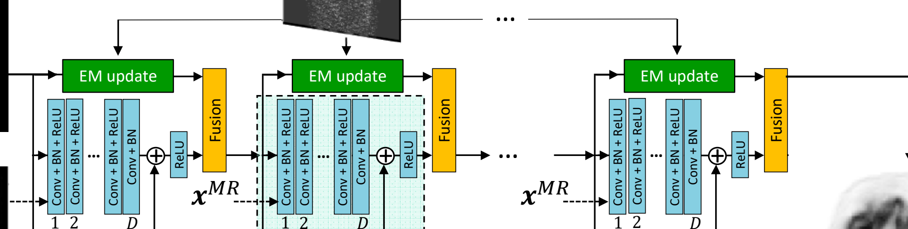
FBSEM-Net: real [18F]FDG data

Initial empty image





PET data
Low count (short scan, or low dose)



MRI

Learns ~77k parameters
N = 35 training datasets

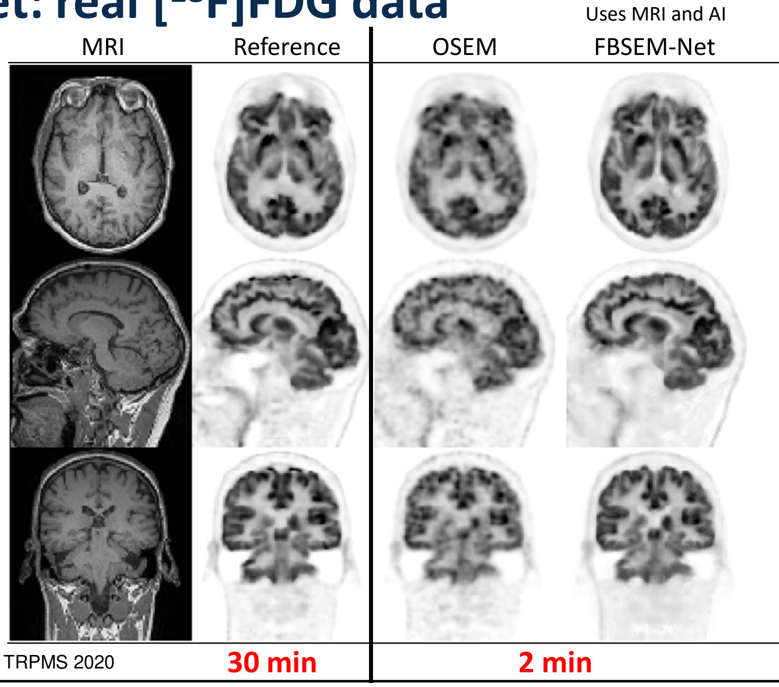
Uses MR, and the physics with DL

Reconstructed image of [18F]FDG

Mehranian and Reader, IEEE TRPMS 2020

68

FBSEM-Net: real [^{18}F]FDG data



Recent variations:

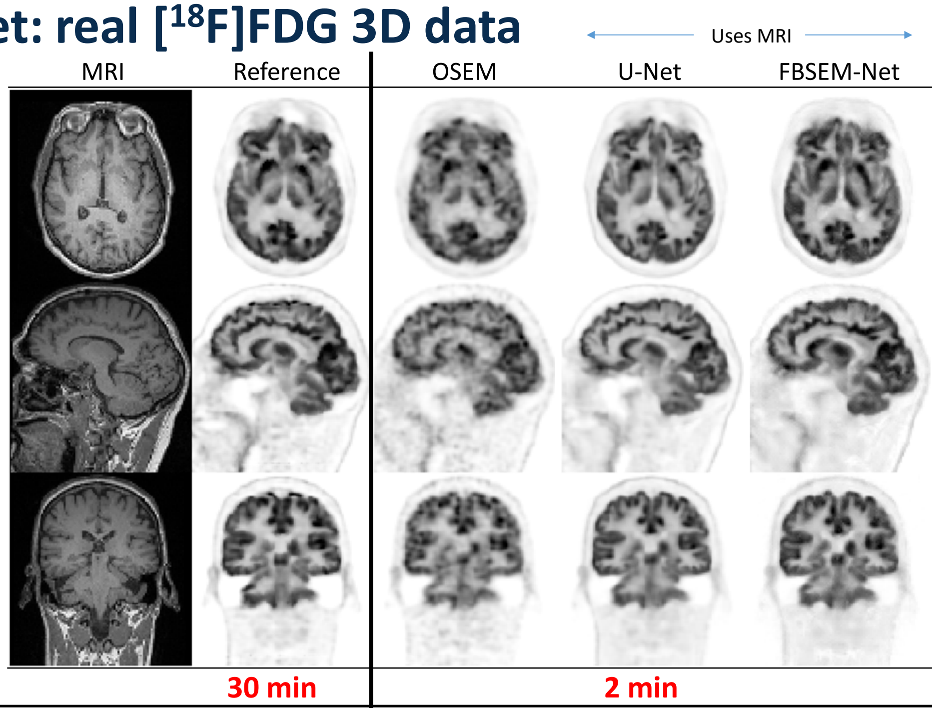
Sequential training [Corda d'Incan et al IEEE TRPMS 2021]

Using **transformers** [Rui Hu, Huafeng Liu 2022]

Mehranian and Reader, IEEE TRPMS 2020

69

FBSEM-Net: real [^{18}F]FDG 3D data



Mehranian and Reader, IEEE TRPMS 2020

70

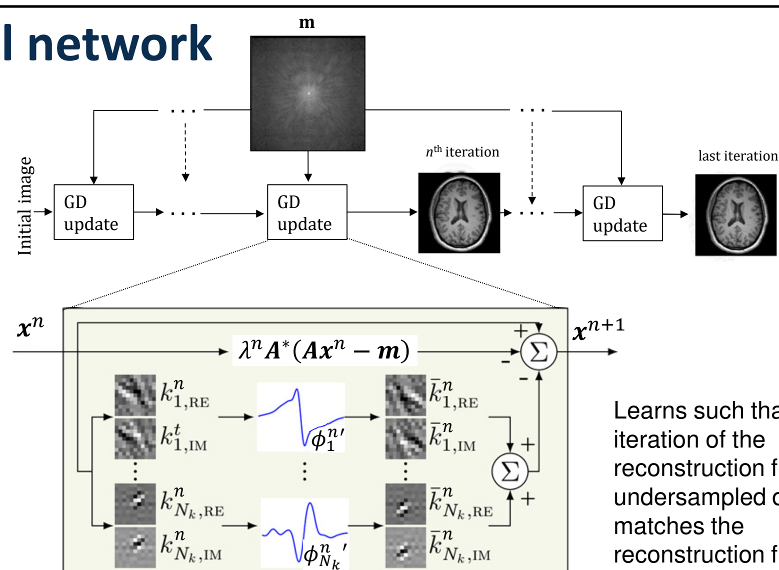
COMPARISON OF UNROLLED METHODS FOR PET

NAME	ARCHITECTURE (Reduced parameters)	LOSS FUNCTION	DATA SIZES 3D	NUMBER OF TRAINING PAIRS (LOW NUMBER)	BACKPROP
MAPEM-Net Gong et al 2019	CNN (U-Net) [>8 x 2 million parameters] **Iteration/module dependent**	MSE For end image	128x128x105	~18	Through all layers including EM update *(Memory intense)*
FBSEM-Net Mehranian & Reader 2020 *	CNN [77,000 parameters] **Same for all iterations/modules**	MSE For end image	114x114x128	~35	Through all layers, excluding EM update *(Memory intense)*
BCD-Net Lim et al 2018, **INN** Lim et al 2020	[10x4000 =40,000 parameters] **Iteration/module dependent**	MSE For current module compared to true/reference	200x200x112	~4	Training at iteration module only *(Not demanding)*

* New iteration-dependent target version: Corda-D'Incan et al TRPMS 2021
Sequential training also

71

MRI: variational network



Learns such that the last iteration of the reconstruction from undersampled data matches the reconstruction from fully sampled data.

$$x^{n+1} = x^n - \left(\sum_{i=1}^{N_k} (K_i^n)^T \phi^{n\prime}{}_i (K_i^n x^n) + \lambda^n A^*(Ax^n - m) \right)$$

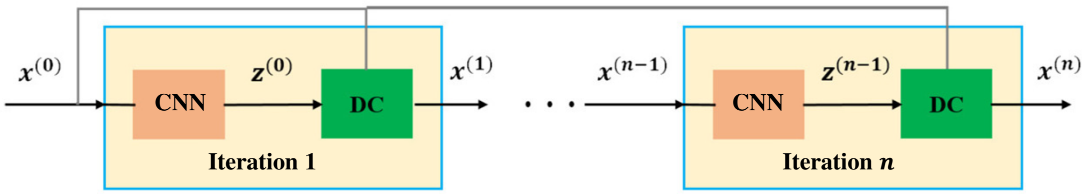
Representation of method of Hammernik *et al* 2018

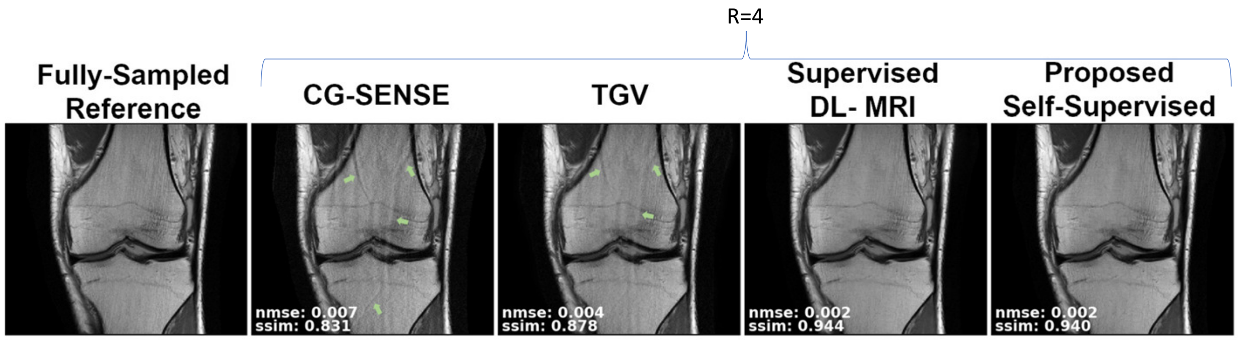
72

Yaman et al MRM 2020

Self-supervised MRI reconstruction

Supervised: images from fully-sampled data used for training
Self-supervised: half data used for k-space to reconstruct from, other half used for loss function



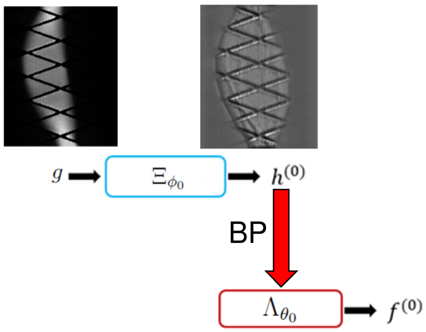


73

LPD

74

Towards Learned Primal Dual (LPD) – physics + data!



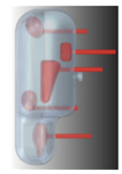


Figure Adapted from Massimiliano Colarieti-Tosti

Original method: Adler & Oktem IEEE TMI 2018

Guazzo & Colarieti-Tosti J Imaging. 2021

75

Towards Learned Primal Dual (LPD)

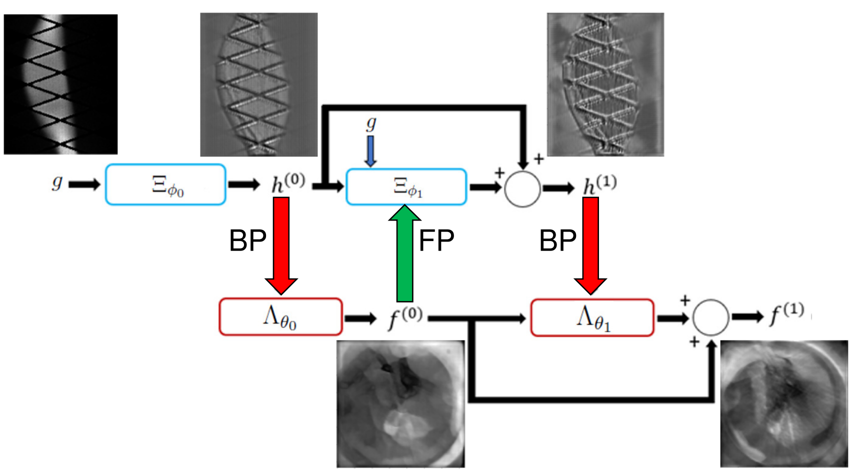


Figure Adapted from Massimiliano Colarieti-Tosti

Original method: Adler & Oktem IEEE TMI 2018

Guazzo & Colarieti-Tosti J Imaging. 2021

76

Towards Learned Primal Dual (LPD)

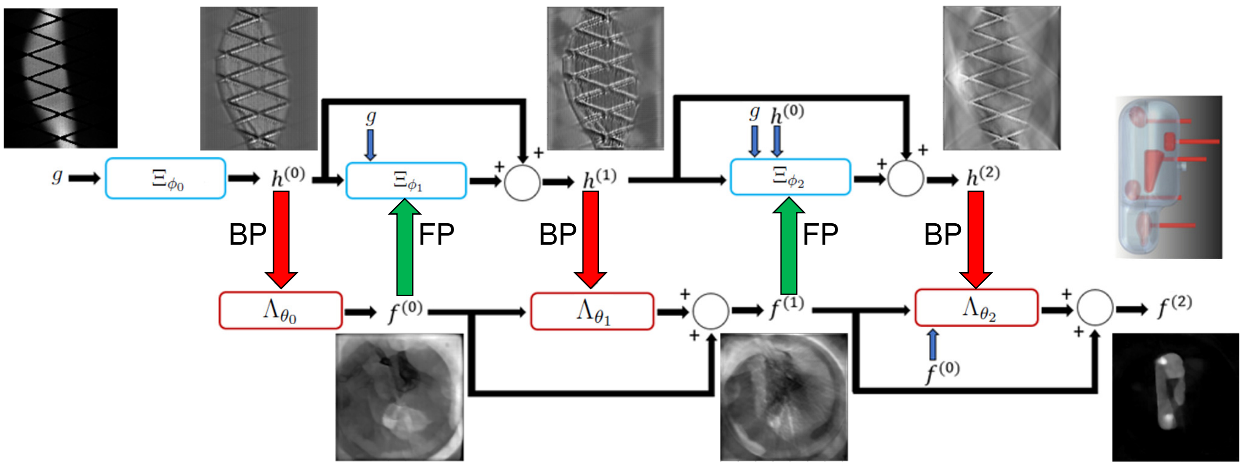


Figure Adapted from Massimiliano Colarieti-Tosti

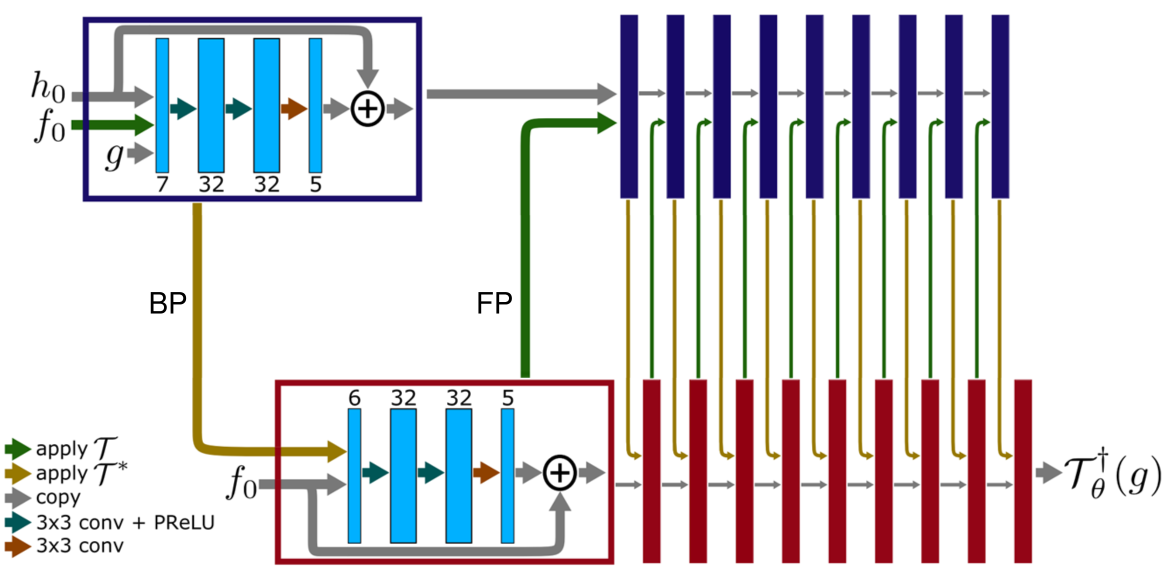
Original method: Adler & Oktem IEEE TMI 2018

Guazzo & Colarieti-Tosti J Imaging. 2021

77

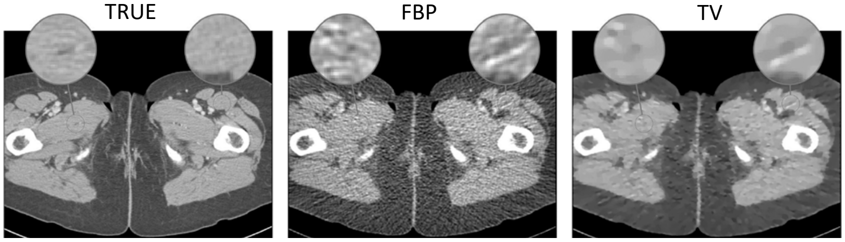
Original method: Adler & Oktem IEEE TMI 2018

Learned Primal Dual (LPD)



78

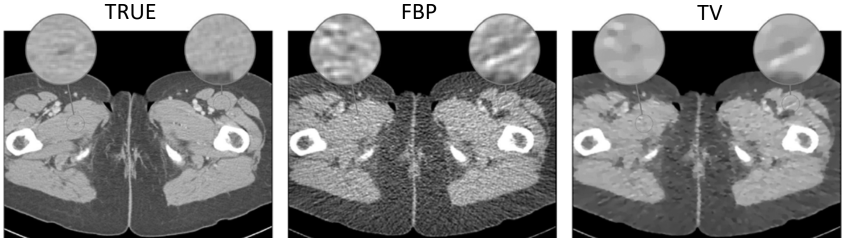
Learned Primal Dual (LPD)

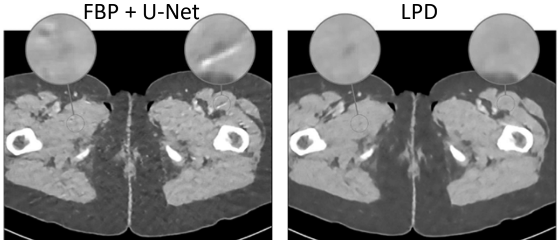


Adler & Oktem IEEE TMI 2018

79

Learned Primal Dual (LPD)



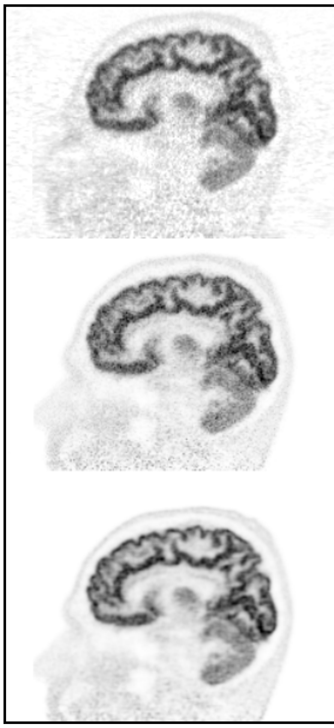


Method	PSNR	SSIM	Runtime	Parameters
FBP	33.65	0.830	**423**	**1**
TV	37.48	0.946	64 371	**1**
FBP + U-Net denoising	41.92	0.941	463	10^7
Learned Primal-Dual, linear	**44.11**	**0.969**	620	$2.4 \cdot 10^5$

- **2k training data pairs** to learn **~240k parameters** via MSE

Adler & Oktem IEEE TMI 2018

80



School of Biomedical Engineering and Imaging Sciences

REVIEW

END OF PRESENTATION

81