Post-Reconstruction Image Denoising and Artefact Removal for Low Count Positron Emission Tomography

da Costa-Luis, Casper

Awarding institution:
King's College London

Post-Reconstruction Image Denoising and Artefact Removal for Low Count Positron Emission Tomography

Thesis submitted to King's College London in fulfilment of the degree Doctor of Philosophy

Casper da Costa-Luis

Supervised by:

Professor Andrew J. Reader Professor Paul K. Marsden

March 2021

Abstract

Positron emission tomography (PET) is a powerful medical imaging modality for the brain, for cancer and for the heart. Image reconstruction is a crucial component to the success of PET, and methodology has undergone a number of significant advances, including progression from 2D to 3D PET reconstruction (improving signal to noise ratio), progression from analytical to iterative reconstruction methods (reducing variance) and advanced modelling of the PET data acquisition to improve image quality (improving image resolution). The latter includes resolution modelling (RM), which in PET accounts for effects including the positron range, photon acollinearity and limited detector resolution.

While notable improvements in image quality have been demonstrated, advances are still very much needed, and the aim of this thesis is to deal with noise and artefacts (such as the notorious ringing artefact introduced by RM, as well as partial volume effects) without introducing quantitative errors. Present approaches either leave noise and RM artefacts in the images, or else they compromise the spatial resolution of the end-point images. This thesis proposes novel deep learning (DL) based techniques to reduce noise and resolve artefacts without compromising resolution in a variety of scenarios, including in the case of quantification of small regions (e.g. lesions) and low-count PET imaging.

Furthermore, multimodality scans (specifically PET-MR) are used, where the jointly acquired data provides anatomical information which aids in the reduction of noise and artefacts while increasing resolution, thereby enabling low dose and/or reduced scan durations. The DL techniques developed are also robust enough to cope with highly limited training datasets and have built-in model consistency in order to constrain their outputs. Enforcing such constraints sets a maximum limit on errors in cases when a DL method fails to perform well on test data. A thorough comparison between the current most promising DL proposals for PET is also conducted, with the aim of providing much-needed guidelines for network architecture and design, for a given quantity of available training data.

Contents

List of Figures

List of Tables

Acknowledgements

Firstly, I would like to thank my supervisors Andrew J. Reader and Paul K. Marsden for their invaluable support & guidance. Popping in to Andrew's office (pre-pandemic) to ask a "quick" 5 min question would invariably lead to an instant hour-long discussion on a range of fascinating topics. Thank you for your boundless enthusiasm. Many thanks to Andrew P. King, Alexander Hammers, and Joel Dunn for feedback over the years and providing data; and to Pawel J. Markiewicz and Martín Belzunce for their tomographic reconstruction software.

I would also like to thank all of the science and mathematics teachers I have ever had (especially those from primary and secondary schools – spread out across three continents), without whom I would never have acquired the academic skill needed to write a thesis. In a break from tradition, I must also extend my thanks to my students. Imparting some of my knowledge and helping you on your own journeys has helped keep me sane and balanced; lending perspective to my own trajectory. Finally and most importantly: deepest and heartfelt gratitude to my mother, who has fought tenaciously against a hostile world to ensure it failed to dull a child's natural curiosity and eagerness to learn – a rare and impossible feat.

Funding

This work was supported in part by the King's College London and Imperial College London EPSRC Centre for Doctoral Training in Medical Imaging under Grant [EP/L015226/1], in part by the Wellcome EPSRC Centre for Medical Engineering at King's College London under Grant [WT 203148/Z/16/Z], in part by EPSRC under Grant [EP/M020142/1], in part by the National Institute for Health Research (NIHR) Biomedical Research Centre Award to Guy's and St Thomas' NHS Foundation Trust in partnership with King's College London, and in part by the NIHR Healthcare Technology Co-operative for Cardiovascular Disease at Guy's and St Thomas' NHS Foundation Trust. The views expressed are those of the author(s) and not necessarily those of the NHS, the NIHR or the Department of Health.

Chapter 1

Introduction to Positron Emission Tomography

PET is a powerful nuclear medical imaging technique. Images are acquired *in vivo*, allowing for non-invasive visualisation of metabolic processes, amongst other tasks. This includes qualitative diagnosis as well as quantitative monitoring of progression and treatment response of various diseases.

Clinical guildelines for evaluating cancerous lesion (tumour) progression are slow to evolve. Early recommendations rely primarily upon 1D anatomical measurements from an X-ray computed tomography (CT) scan [1], while a recently updated version of the response evaluation criteria in solid tumors (RECIST) guidelines (published nearly a decade later, in 2009) acknowledges PET as "adjunct to determination of progression," but continues to recommend a dedicated CT scan (stating that joint PET-CT often either lacks sufficient CT quality, or may bias an investigator without sufficient experience) [2]. Nevertheless, more recent recommendations indicate that PET is gaining traction in research [3] and clinical oncology [4], [5] – especially for lung cancer and lymphoma.

In addition to oncology, PET also has many applications in neuroscience; examination of brain functions, including quantification of cerebral blood flow, metabolism, and receptor binding [6]. For this reason, monitoring progression of conditions such as Alzheimer's disease (AD) or even mild cognitive impairment (MCI) can also be possible. Figure 1.1 shows reconstructions of central slices through three different brains.[1] Sensitivity is high enough that – in order to avoid unnecessarily stimulating regions of the brain – standard operating procedures frequently require patients to wear eye-masks and be in a silent environment during the tracer uptake period.

PET is also useful for planning surgery for epilepsy. With prevalence of around 1 in 150 people, epilepsy is one of the most common serious neurological disorders [8], [9]. A brain scan is required to determine if the type is *focal* (regions of hypointensivity within grey matter, amenable to surgical intervention) or *general*. Typically, a MRI scan is performed first. However, due to the comparatively low molecular sensitivity of magnetic resonance (MR), no anomalies might be detected and thus a follow-up PET scan may be required. There are also cases where simultaneous PET-MR (combining functional and anatomical information) is necessary for diagnosis: MR to identify hypoperfusion and PET to identify hypometabolism [10].

The use of positron emission for medical imaging dates back to 1950 [11]–[13]. The first clinical scan took place in 1952, and the first multi-detector positron imaging device was developed in 1962. Nearly a decade later – during 1968-72 – tomography was developed, with filtered backprojection (FBP) being used to reconstruct PET and CT images. Cylindrical PET scanners were proposed the year after, and shortly thereafter in 1978, the widely-used [^{18}F]FDG radiotracer was proposed.

Maximum likelihood expectation maximisation (MLEM) iterative reconstruction [14] was proposed for PET image reconstruction (Section 1.3) shortly after in the 1980s [15]. Unlike FBP, it can model and compensate for various noise and resolution degradation and effects. It should however be noted that pitfalls of modelling resolution include Gibbs ringing artefacts which affect clinically diagnostic measures such as maximum standardised uptake value (SUV_{max}) [16], [17].

[1] Data courtesy Colm McGinnity & Alexander Hammers, "Evaluation of Brain PET/MR versus PET-CT," REC 15/NE/0203, IRAS 178069. All data were acquired from the Siemens Biograph mMR scanner at King's College London & Guy's and St Thomas' PET Centre. Ten scans are included in this thesis: [^{18}F]FDG-PET with corresponding 3D T1 magnetisation-prepared rapid acquisition with gradient echo (MPRAGE). The study was approved by the institutional review boards and the research ethics committee, and written informed consent obtained from all study participants. The MPRAGE [7] data were acquired using a 5-channel head and neck coil with repetition time (TR) 1700 ms, echo time (TE) 2.63 ms, inversion time (TI) 900 ms, number of averages (NEX) 1, flip angle 9°, pixel bandwidth 199 Hz, reconstruction matrix size 224×256×176 and voxel dimensions 1.05×1.05×1.1 mm^3. Simultaneously acquired dual-point Dixon magnetic resonance imaging (MRI) was also used for PET attenuation correction. The Dixon data were acquired using the spoiled gradient-recalled sequence (SPGR) with T1 3.6 ms, TE 2.46 ms, NEX 1, flip angle 10°, pixel bandwidth 946 Hz, reconstruction matrix size 192×126×128, and voxel dimensions 2.06×2.06×3.12 mm^3.

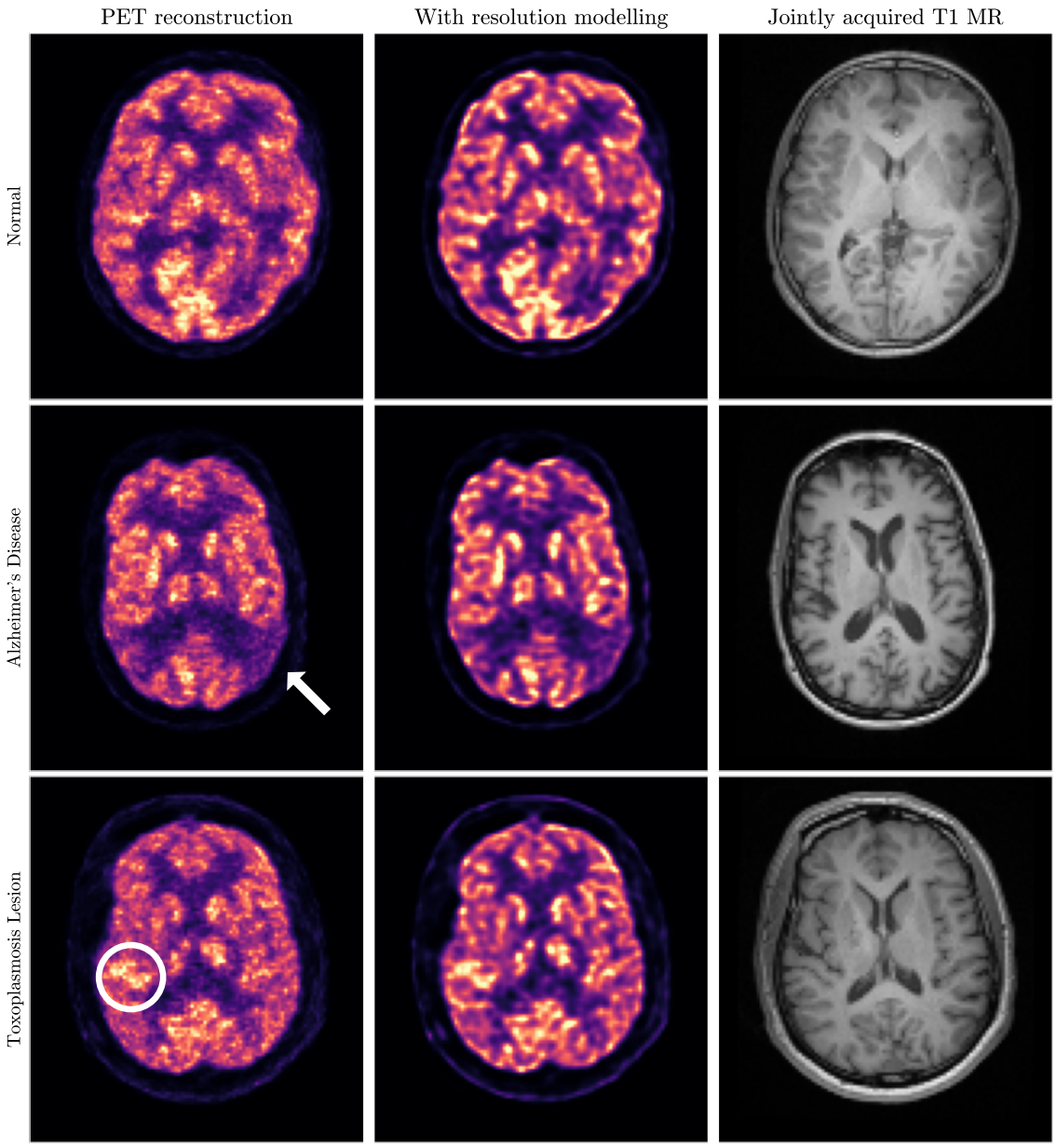


Figure 1.1: Central slices of three different PET-MR brain scans acquired on a Siemens Biograph mMR scanner as detailed in Section 3.2.1.2 (count levels between 400 M and 500 M). The top row was reported as normal (healthy), while the middle row shows hypo-intensities in the parietal/occipital lobes due to Alzheimer's Disease (AD). The bottom row shows a hyper-intense lesion caused by Toxoplasmosis. The different reconstruction methods for the left and middle columns are detailed in Section 1.3.1.

1.1 PET Radiotracers

PET uses radiotracers to track biochemical and physiological processed *in vivo*. Tracers are molecules which can be easily tracked. An ideal tracer should not affect the process it measures. Typically this means that tracers must be used in very small amounts yet still remain detectable by sensitive external equipment. One example of this is injecting smoke particles into wind tunnel experiments to visualise turbulence. In the case of PET, tracers are radioactive – so reducing injected dose is even more desirable due to the resultant increase in patient safety and reduction in cost [3], [18]. Additionally, the radiotracers used have a short half-life to reduce patient exposure (shown in Table 1.1). This means scans also need to be conducted soon after the radionuclides are produced by a cyclotron, and over a short period of time, before the activity decays.

Isotopic tracers are chemical compounds which are "labelled" (marked) by having one or more atoms replaced with an isotope of the stable atom. The resultant molecule should have effectively unaltered chemical properties for the purpose of any reaction being studied. However, special external equipment (such as mass or infra-red spectrometers) can detect the isotopes, and can therefore track (via blood samples) the labelled molecules through their pathway. PET uses tracers with positron-emitting radioactive isotopes (radionuclides), and detection is done non-invasively by radiation detectors. Molecular sensitivity is high enough that even picomolar tracer concentrations can be detected [19].

If a particular process needs to be studied, chemicals which specifically target the process are good candidates for labelling. One particularly successful radionuclide is [^{18}F]FDG. This molecule has a fluorine-18 isotope bound to a 2-deoxy-2-glucose molecule [20]. As an analogue to glucose (a sugar molecule), its uptake in living tissue corresponds to cellular metabolic activity. For example, lesions will often have different metabolic rates than surrounding healthy tissue, and thus will have different uptake rates of [^{18}F]FDG, resulting in differences in measured radioactivity. This difference will be visible in reconstructed images, allowing for qualitative as well as quantitative analysis. Other tracers used in PET include florbetaben ([^{18}F]FBB) and florbetapir ([^{18}F]FBP), both of which bind to β-amyloid plaques (which in turn indicate AD) [21]. Many other radiotracers exists, mostly labelled with ^{18}F, though ^{11}C (e.g. methionine) and ^{68}Ga (e.g. prostate-specific membrane antigen (PSMA)) are also used [6], [22]. For cardiac PET, ^{15}O, ^{13}N, and ^{82}Rb are also used. A summary of these radionuclides and their half-lives is given in Table 1.1.

Radioisotopes	Half-life
^{18}F	110 min
^{11}C	20 min
^{68}Ga	68 min
^{15}O	122 sec
^{13}N	10 min
^{82}Rb	75 sec

Table 1.1: Half-lives of well-known positron-emitting (neutron deficient) radionuclides.

Typical [^{18}F]FDG scan protocols require 60 min between administration of the tracer and the PET acquisition to allow time for sufficient uptake. The administered radioactivity is typically dependent on patient weight – circa 2.5 MBq/kg for 3D PET (roughly double for 2D) assuming 5 min per bed position [3], [23] – with associated radiation effective dose (ED) of 1−10 mSv. In the UK, the Administration of Radioactive Substances Advisory Committee (ARSAC) guidelines for PET further indicate that most cases should require under 4.5 MBq/kg and EDs of 1.6−7.6 mSv [22].

The data used in this research is based on [^{18}F]FDG (simulations as well as real patient data acquisitions). It should however be noted that the denoising and artefact reduction techniques investigated are broadly applicable to other tracers, other medical imaging modalities, as well as image post-processing and inverse problems in general.

1.2 PET Physics

Since PET is driven by the radioactive decay of isotopes labelling individual molecules, it is often called a molecular imaging modality. Resolution is however much coarser than a molecular level – typically a few millimetres. The radioactive decay causes emission of positrons, which in turn travel up to a few millimetres (positron range) before annihilating with a nearby electron. This annihilation produces two back-to-back photons (that may be acollinear due to momentum of the pair) which may be recorded almost simultaneously by two opposing detectors, as shown in Figure 1.2. There are a finite combination of detector pairs, and thus a finite number of lines of response (LoRs). Resolution degradation effects discussed further below include positron range, acollinearity, and detector size (summarised in Table 1.2) as well as random coincidences and scatter (depicted in Figure 1.4).

A closely related nuclear imaging technique is single photon emission computed tomography (SPECT). However, SPECT uses gamma-emitting radiotracers and directly measures these individual gamma (photon) rays. By comparison, in PET, positrons may travel some distance before annihilating. This may imply that PET should have lower resolution. However, PET in fact provides superior resolution to SPECT, primarily due to the additional localisation information provided by the resultant coincident pair of photons. Additionally, PET does not require beam limiting collimators (which limit sensitivity in SPECT).

Figure 1.2 shows a representation of a ring of photon detector blocks. The Siemens Biograph mMR scanner used to acquire the data used in this work has 8 rings of 56 blocks [24]. Each block is coupled to its own 3×3 array of avalanche photodiodes (APDs), which in turn localise the detected gamma ray position in an 8×8 scintillation crystal lutetium oxyorthosilicate (LSO) array. The field of view (FoV) is 59.4 cm transaxially and 25.8 axially, which is sufficient for whole-brain 3D scans in a single bed position. The mMR has a manufacturer-reported temporal resolution of 2.93 ns. After a photon is detected, a time window of 5.86 ns is opened during which a second detection in a different crystal is taken to be the coincident photon corresponding to a single positron-electron annihilation event. This event would have occurred somewhere along a virtual line (i.e. LoR) connecting the two detectors blocks. It should be noted that recent advances in detector technology has led to temporal resolutions of around 200 ps [25], thus enabling localisation along a given LoR. This localisation information can be inferred from the time difference in the detection of coincident photons, and is therefore referred to in the current literature as time of flight (ToF) [11].

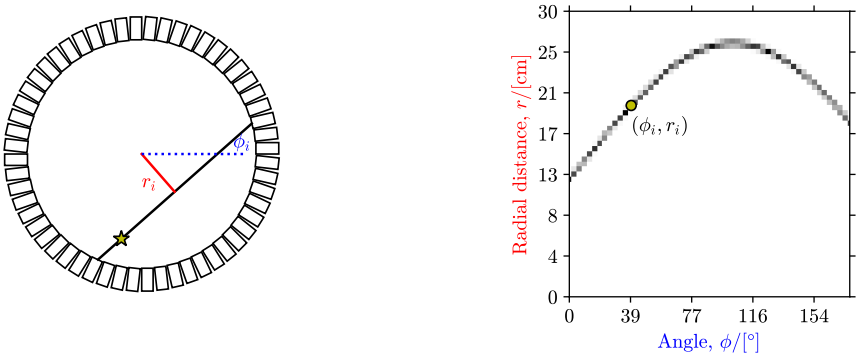


Figure 1.2: Diagram of a PET detector ring and i^{th} coincident photon pair (left panel) originating from a point source marked with a star. The source will emit multiple pairs, and the corresponding radial distances r and angles ϕ may be plotted in a sinogram (right panel).

The simplest method of reconstruction would be to overlay all acquired LoRs on top of each other (in the case of ToF, these lines will have a Gaussian – rather than uniform – intensity distribution). This is called backprojection. An improvement in spatial resolution is possible by using FBP [26] incorporating a high-pass ramp filter. Unfortunately, FBP will also emphasise high frequency noise, and is thus usually followed by post-smoothing (or truncation of the ramp filter) when used in clinical practice [27]. Figure 1.3 below shows the effect of this filter with and without the presence of noise on the Shepp-Logan phantom [28].

The main causes of limited PET resolution are listed in Table 1.2 in terms of the equivalent Gaussian full width at half maximum (FWHM) blur. Resolution loss due to scanner geometry is primarily due to the finite detector size, i.e. crystal face area [29]. Lack of penetration depth information and hardware decoding imperfections also further limit resolution, and improving this is often considered prohibitively expensive [29].

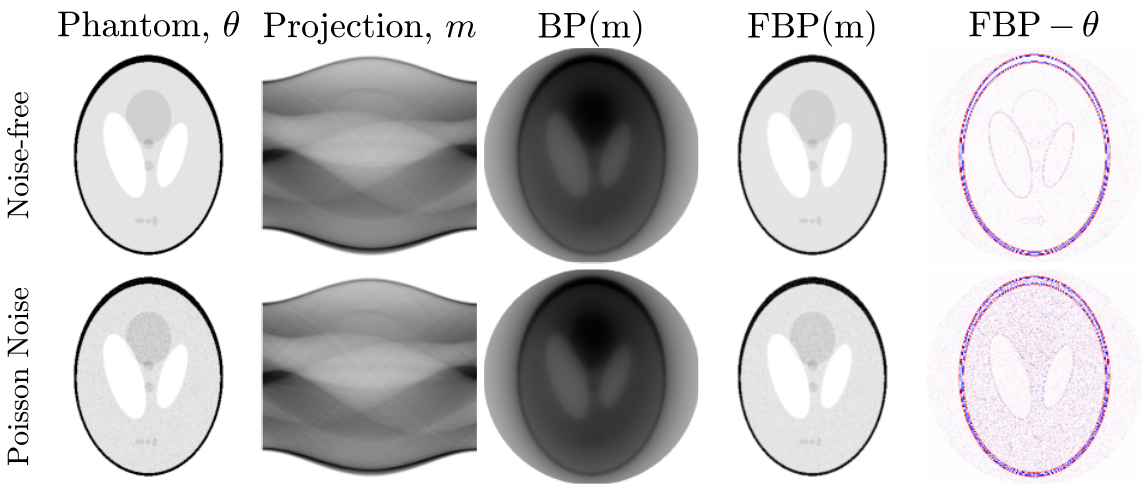


Figure 1.3: A 2D phantom (Shepp-Logan) of 1 M counts ($\boldsymbol{\theta}$, column 1) is projected (m, column 2) and then backprojected (BP) without (column 3) and with (column 4) a filter. The difference (error) between the FBP and phantom is shown in the last column (5). Results are shown without (row 1) and with (row 2) Poisson noise.

In any case, for scanner geometries with a resolution of under 4 mm, positron effects become significant. It should be noted that this is actually dependent on radionuclide – for example, ^{82}Rb causes nearly an order of magnitude more resolution loss than ^{18}F [30]. In [^{18}F]FDG-PET, resolution degradation due to acollinearity is proportional to detector ring diameter and accounts for up to 1 mm FWHM in the FoV centre [31]. Meanwhile positron range results in a comparatively modest Gaussian-equivalent FWHM of around 0.5 mm [32], [33], though slightly higher resolution is possible in PET-MR [34] due to positron range reduction in strong magnetic fields [35].

Effect	Approximate Resolution (Gaussian-equivalent FWHM)
Scanner geometry	3-7 mm
Photon acollinearity	1 mm (0.5°)
Positron range	0.5 mm

Table 1.2: Main causes of finite resolution in [^{18}F]FDG-PET.

Other important considerations when reconstructing images include detector block efficiencies due to geometric effects, interference, crystal efficiencies, and dead time [29], [36], [37]. In addition to such scanner-dependent effects, there are also object-dependent effects. The further photons need to travel in a dense medium, the more likely they are to be attenuated. At PET energies (511 keV), Compton effects are much more significant than photoelectric absorption [38]. The linear attenuation coefficient of photons of this energy varies depending on the absorbing medium – in particular, around $0.084\,\mathrm{cm}^{-1}$ and $0.105\,\mathrm{cm}^{-1}$ for soft tissue and bone, respectively (a notable exception are lungs, which have a coefficient of 0.02-$0.04\,\mathrm{cm}^{-1}$) [39]. Note the photon energy of 511 keV comes from the fact that positrons (rest mass $m_e = 511\,\mathrm{keV/c^2}$) annihilate with electrons (same rest mass), producing two photons of total energy given by $E = mc^2$, where total mass $m = 2m_e$. Due to the conservation of momentum, these photons (in their centre-of-mass frame of reference, which is assumed the as that of the detectors) must have equal and opposite momentum, p. Since for photons momentum is proportional to energy ($E = p/c$), this means that each photon has the same amount of energy, namely 511 keV. Photons of other energies would have different attenuation coefficients. In practice, where available, CT data can also be used to infer PET attenuation coefficients [38]. Equation (1.1) below shows how the coefficient characterises the exponential decrease in probability of transmission.

$$\mathrm{P}(x) = e^{-\mu x}, \tag{1.1}$$

where P is the probability of transmission;

- μ is the linear attenuation coefficient (depending on photon energy and object medium), and
- x is the distance travelled though the medium.

In general, μ is spatially variant, and Equation (1.1) must be integrated along each LoR to obtain the corresponding overall probability of transmission. Since both photons need to be detected, the transmission probability is the same for all points along an LoR.

Finally, detected LoRs may also be completely wrong due to random coincidences and scatter, as shown in Figure 1.4. Random coincidences are cases where photons from different annihilation events are received within the same detector coincidence window and thus incorrectly paired together. Scatter, meanwhile, refers to photons being deflected from their original paths by interaction with the object in the FoV almost entirely due to Compton scatter [38].

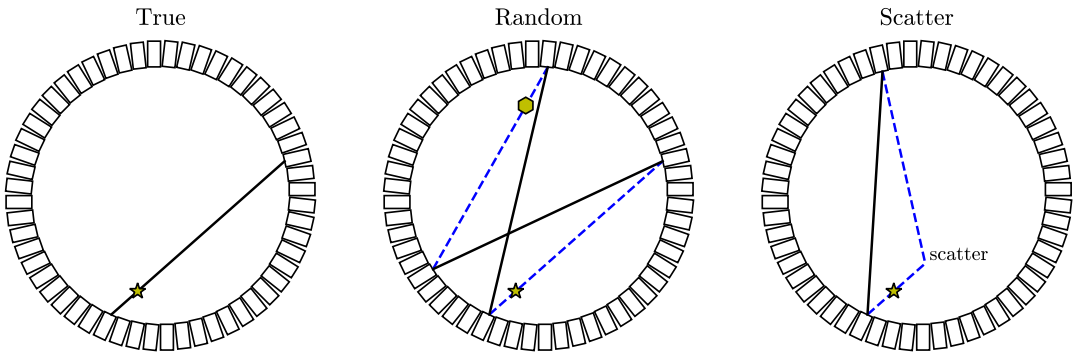


Figure 1.4: True, random and scatter coincidences.

It is important to keep in mind that radiotracers are difficult and costly to produce, and the radiation that they produce is harmful to patients. It is estimated that the radiation exposure of a typical PET scan reduces life expectancy by 1 to 3 weeks [40], [41]. Clinical PET-CT plans can easily result in a cumulative effective dose of hundreds of mSv and a dose-related life expectancy reduction of several months [42]. Methods enabling the reduction of the injected dose – as low as reasonably achieveable (ALARA) – are thus an active area of research [23]. Furthermore, reducing the overall scan time could be used to increase patient throughput, while decreasing frame durations for dynamic scans increases temporal resolution [43]. Unfortunately, both dose and time reductions results in fewer acquired counts. Suppressing noise and artefacts becomes particularly important in the case of such count reductions. Appropriately modelling and compensating for all the effects outlined above is thus crucial to obtaining images of sufficient clinical quality [43]. The following section addresses iterative reconstruction, a powerful method to solve inverse problems which can incorporate models of all of these effects.

1.3 Iterative Reconstruction

It is important to note that the scanner can provide individual photon pair detection data in the form of raw list-mode data (e.g. detector indices and times of detection) from which LoRs can be inferred. These LoRs are usually parameterised by their their radial distance and angle (as shown in Figure 1.2), and counted in a set of 2D histograms (called sinograms) for use in reconstruction algorithms. Native (span-1, i.e. uncompressed) mMR dimensions have 4 084 sinograms – each with 344 projection bins (radial coordinate) and 252 views (azimuthal angle). The commonly used span-11 axial compression [44] groups these together into 837 sinograms [36]. This is a compression factor of 4.88, yet causes minimal degradation in the resolution of reconstructed images [45]. The compressed sinograms reduce memory requirements as well as number of computational operations, thus making images quicker and easier to reconstruct.

1.3.1 Maximum Likelihood Expectation Maximisation

The concept of MLEM was proposed in 1977 [14], and a few years later in the 1980s was applied to tomographic imaging [15]. Unlike FBP discussed above in Section 1.2, iterative reconstruction can explicitly model and compensate for various resolution-degrading effects as well as noise properties. Additionally, it can also easily include models of attenuation and normalisation, which is important for tasks requiring accurate quantification [46].

In PET, the expected number of counts $\hat{\boldsymbol{m}}$ in the measured data (sinograms) are a function of the object $\boldsymbol{\theta}$:

$$\hat{\boldsymbol{m}} = q(\boldsymbol{\theta}), \tag{1.2}$$

where q models the imaging system, including noise;

$\boldsymbol{\theta}$ is the object, and

$\hat{\boldsymbol{m}}$ are the expected measured counts (sinogram data).

Note that q is a statistical model with parameters. The expected counts in each individual bin are given by line integrals along the corresponding LoR:

$$\hat{m}_i = \int_{P_i} \theta(x, y, z)\, dP_i + \hat{b}_i, \tag{1.3}$$

where $\theta(x, y, z)$ is the intensity (i.e. radioactivity concentration) of the voxel at the coordinates (x, y, z);

P_i is the path along the i^{th} LoR;

$\hat{b}_i$ is a corresponding additive background term, and

$\hat{m}_i$ is the expected counts in the i^{th} sinogram bin.

It should be noted that while the object itself has a continuous tracer distribution, q has a discrete range and $\hat{\boldsymbol{m}}$ is of finite length (i.e. a finite number of sinogram bins i). Throughout this work, the object is also considered as a collection of cartesian voxels in the FoV, with intensities given by their respective radioactivity concentration (in other words, x, y and z take discrete values). In general, the imaging system is decomposed into a linear operator – a matrix $\boldsymbol{P}$, mapping from image to sinogram space – and an additive background term $\hat{\boldsymbol{b}}$, as shown in Equation (1.4). The system matrix $\boldsymbol{P}$ is also commonly factorised into different components for attenuation, normalisation, and resolution modelling [47].

$$\hat{\boldsymbol{m}} = \boldsymbol{P}\boldsymbol{\theta} + \hat{\boldsymbol{b}} \tag{1.4}$$

$$= \boldsymbol{A}\boldsymbol{N}\boldsymbol{X}\boldsymbol{H}\boldsymbol{\theta} + \hat{\boldsymbol{s}} + \hat{\boldsymbol{r}}, \tag{1.5}$$

where $\boldsymbol{P}$ is the imaging system matrix;

- $\hat{\boldsymbol{b}}$ is an additive background term modelling the mean of the noise;
- $\boldsymbol{A}$ accounts for attenuation;
- $\boldsymbol{N}$ are normalisation factors;
- $\boldsymbol{X}$ maps from image to sinogram space (i.e. a 2D or 3D Radon transform);
- $\boldsymbol{H}$ applies a point spread function (PSF) model accounting for various resolution degradation effects;
- $\hat{\boldsymbol{s}}$ are expected (estimated) scatter events (Figure 1.4 right panel), and
- $\hat{\boldsymbol{r}}$ are expected (estimated) random coincidences (Figure 1.4 middle panel).

In addition to mapping between image and sinogram space, $\boldsymbol{P}$ encodes the probabilities of a decay event in each voxel in the FoV being successfully detected along each LoR. Note that for the case of low count PET – the primary focus in this work – it is fair to assume a linear model $\boldsymbol{P}$. For very high count rates, saturation of the detectors can occur, where dead time results in a non-linear model [48].

Given that the measured $\boldsymbol{m}$ are fundamentally integer numbers of counts, they can be modelled by a Poisson distribution with the mean $\hat{\boldsymbol{m}}$ [15]:

$$m_i \sim \text{Poiss}(\hat{m}_i). \tag{1.6}$$

Note that this assumes each bin i can be considered independently. The probability of obtaining the observed data is therefore given by the product of the probabilities for each bin:

$$\text{P}(\boldsymbol{m}|\hat{\boldsymbol{m}}) = \prod_i \frac{e^{-\hat{m}_i}\hat{m}_i^{m_i}}{m_i!}. \tag{1.7}$$

The task of reconstructing an estimate of the object $\hat{\boldsymbol{\theta}}$ involves maximising the likelihood $\mathcal{L}$ of obtaining $\boldsymbol{\theta}$ given the observations $\boldsymbol{m}$. The likelihood is defined to be given by the probability from Equation (1.7) above. When combined with the system projection matrix and noise from Equation (1.4), this results in the following likelihood function:

$$\mathcal{L}(\boldsymbol{\theta}|\boldsymbol{m}) = \prod_i \frac{\exp\left\{-\sum_j P_{ij}\theta_j - \hat{b}_i\right\}\left(\sum_j P_{ij}\theta_j + \hat{b}_i\right)^{m_i}}{m_i!}. \tag{1.8}$$

The maximum likelihood estimate (i.e. the most likely object given the measured data) is given by:

$$\hat{\boldsymbol{\theta}} = \underset{\boldsymbol{\theta}}{\operatorname{argmax}}\, \mathcal{L}(\boldsymbol{\theta}|\boldsymbol{m}). \tag{1.9}$$

However, given that the natural logarithm is a monotonic operator, it can be applied to the likelihood function without affecting the location of the maximum. Equation (1.9) is therefore equivalent to the maximum log-likelihood:

$$\hat{\boldsymbol{\theta}} = \underset{\boldsymbol{\theta}}{\operatorname{argmax}} \ln \mathcal{L}(\boldsymbol{\theta}|\boldsymbol{m}). \tag{1.10}$$

This is useful since a logarithm can be used to simplify the products and exponentials in Equation (1.8), resulting in:

$$\ln \mathcal{L}(\boldsymbol{\theta}|\boldsymbol{m}) = \sum_i \left[-\sum_j P_{ij}\theta_j - \hat{b}_i + m_i \ln \left\{ \sum_j P_{ij}\theta_j + \hat{b}_i \right\} - \ln m_i! \right]. \tag{1.11}$$

Note that $\ln m_i!$ can be ignored since it does not affect solution at the maximum, $\hat{\boldsymbol{\theta}}$. However, it should also be noted that alternative models exists – for example, the maximum a-posteriori (MAP) likelihood adds an extra object-dependent penalty function, $\beta(\boldsymbol{\theta})$, to Equation (1.11) [49].

Since (assuming a linear model) the log-likelihood is a convex function, the maximum occurs where the gradient is zero:

$$\mathrm{E}\left[\left. \frac{\partial \ln \mathcal{L}}{\partial \theta_j} \right|_{\hat{\boldsymbol{\theta}}} \right] = 0 \quad \forall j. \tag{1.12}$$

However, the gradient of Equation (1.11) is clearly given by:

$$\frac{\partial \ln \mathcal{L}}{\partial \theta_j} = \sum_i \left[-P_{ij} + \frac{m_i P_{ij}}{\sum_j P_{ij}\theta_j + \hat{b}_i} \right]. \tag{1.13}$$

Using vector notation to apply to all bins i, and using element-wise (Hadamard) division, Equation (1.12) and Equation (1.13) combine [50] to yield:

$$\mathrm{E}\left[\nabla \ln \mathcal{L}(\boldsymbol{\theta})|_{\hat{\boldsymbol{\theta}}} \right] = \mathrm{E}\left[-\boldsymbol{P}^\top \mathbf{1} + \boldsymbol{P}^\top \frac{\boldsymbol{m}}{\boldsymbol{P}\hat{\boldsymbol{\theta}} + \hat{\boldsymbol{b}}} \right] = \mathbf{0}. \tag{1.14}$$

While this is clearly consistent with the original $\mathrm{E}[\boldsymbol{m}] = \hat{\boldsymbol{m}}$ from Equation (1.4), a direct solution for $\boldsymbol{\theta}$ is still not possible. However, an iterative MLEM scheme can be used to converge upon a reasonable estimate [15]. Given an initial estimate $\boldsymbol{\theta}^{(k)}$, an updated estimate $\boldsymbol{\theta}^{(k+1)}$ can be calculated as follows:

$$\boldsymbol{\theta}^{(k+1)} = \frac{\boldsymbol{\theta}^{(k)}}{\boldsymbol{P}^\top \mathbf{1}} \circ \boldsymbol{P}^\top \frac{\boldsymbol{m}}{\boldsymbol{P}\boldsymbol{\theta}^{(k)} + \hat{\boldsymbol{b}}}. \tag{1.15}$$

The transposed system matrix $\boldsymbol{P}^\top$ represents backprojection. While the $\boldsymbol{P}^\top \mathbf{1}$ term is a result of the expectation maximisation derivation above, it can also be thought of as a normalising term (ensuring that when Equation (1.4) is satisfied, no further updates would occur, i.e. $\boldsymbol{\theta}^{(k+1)} = \boldsymbol{\theta}^{(k)}$). It is common to set the initial estimate $\boldsymbol{\theta}^{(0)} = \mathbf{1}$, though any set of positive values would ensure the reconstruction remains non-negative. Note that the multiplicative nature of the equation means that zero-value voxels are impossible to update. Such "zero trapping" effects may become significant and more likely as the number of counts decrease. This may occur when collecting inherently low count data, or due to increasing the number of subsets in ordered subsets expectation maximisation (OSEM) – the widely-used fast approximation of MLEM [51]–[53].

Note that penalised versions of maximu likelihood estimation (MLE) also exists [49]. Such algorithms however usually require at least one empirically-chosen hyperparameter which controls the strength of the penalty, and convergence is not necessarily guaranteed. Penalties can also introduce biases and artefacts. While convergence has been demonstrated for the case of Gibbs priors as well as MR-guided "pixel-by-pixel" priors [54], penalised expectation maximisation (EM) is not considered here. Instead, the focus will be on post-processing methods.

The problems with MLEM include the approximations made (for example, inaccuracies and omissions in the system matrix $\boldsymbol{P}$) and the choice of iteration number k. In ideal noise-free scenarios, convergence would occur as $k \to \infty$. However, in clinical practice the finite number of counts in the presence of noise means that there is a trade-off between bias and standard deviation of the reconstructed images (often referred to as a "bias-variance" trade-off). With the exception of early iterations, as k increases, so does the standard deviation (uncertainty in the reconstructed voxel intensities). For this reason, k is usually limited to at most a few hundred iterations in clinical practice.

While low count rates are less likely to saturate the system (making it safer to assume a linear system matrix $\boldsymbol{P}$), Cloquet and Defrise have shown that MLEM significantly deviates from the Cramer-Rao lower bound (CRLB) – the theoretical minimum standard deviation [55].

Inclusion of resolution modelling (RM) – using a PSF $\boldsymbol{H} \neq \boldsymbol{I}$ in Equation (1.5) – results in an improvement in contrast recovery as well as resolution, in addition to apparent spatial noise suppression (diminished voxel variance while increasing intervoxel covariance) [56]. Under certain conditions, this can mean better lesion detectability [56]–[59]. However, RM can also introduce Gibbs ringing artefacts (ripple effects near edges) which can greatly affect $\mathrm{SUV}_{\mathrm{max}}$ used in clinical practice to gauge tumour aggressiveness [60], [61]. The use of RM is therefore controversial, and inappropriate for some clinical tasks [59].

Alternatively, reconstructed images may be post-processed to remove inaccuracies and compensate for the shortcomings in MLEM. This is explored in the next section.

1.4 Post-processing

Theoretically, any post-processing method could be incorporated into iterative reconstruction. However, reconstruction focuses more on accurately modelling the underlying physics of the system, and modifications to MLEM tend to be in the form of regularising terms. Post-processing, meanwhile, focuses more on task-driven image-to-image mappings.

Denoising and artefact removal are particularly important for PET imaging, where there is a comparatively low signal to noise ratio (SNR) [62]. Instead of – or in addition to – tackling this problem during the reconstruction process (as discussed above), it is possible to address noise and artefacts with post-reconstruction processing. Furthermore, post-processing tends to be comparatively quick to run: images need only be processed once, rather than at every iteration.

While it may be tempting to take advantage of the numerous existing image denoising methods developed for non-medical qualitative tasks, it is important to keep in mind the unique requirements of PET – namely, the importance of quantitative accuracy. Unfortunately, most – if not all – post-processing methods have caveats. In many cases, Noise reduction is achieved at the cost of resolution, sensitivity, and contrast [63]. Some methods can also introduce edge inaccuracies as well as spatially-variant and highly localised bias. Since subtle changes in small regions are crucial for certain tasks (especially in oncology), such bias is highly undesirable.

1.4.1 Gaussian Smoothing

The simplest and clinically most widely used post-processing step is Gaussian post-smoothing (PS), i.e. convolution with a Gaussian kernel. This suppresses noise at the expense of reducing resolution. If the Gaussian kernel used is at least as wide as the PSF used in a resolution-modelled (RM) reconstruction, then Gibbs ringing is guaranteed to be removed [64]. Unfortunately PS works against the resolution gains of RM. The special case of using identical kernels is an example of the method of sieves, and results in images of better quality than if neither RM nor PS was used at all [64], [65]. The quality improvement is however minor, raising questions as to whether it is worth the extra effort and complexity. Figure 1.5 shows the effect of PS on ringing. Chapter 4 offers a more thorough exploration of PS and RM, as well as a machine learning approach to solving Gibbs ringing in the presence of noise.

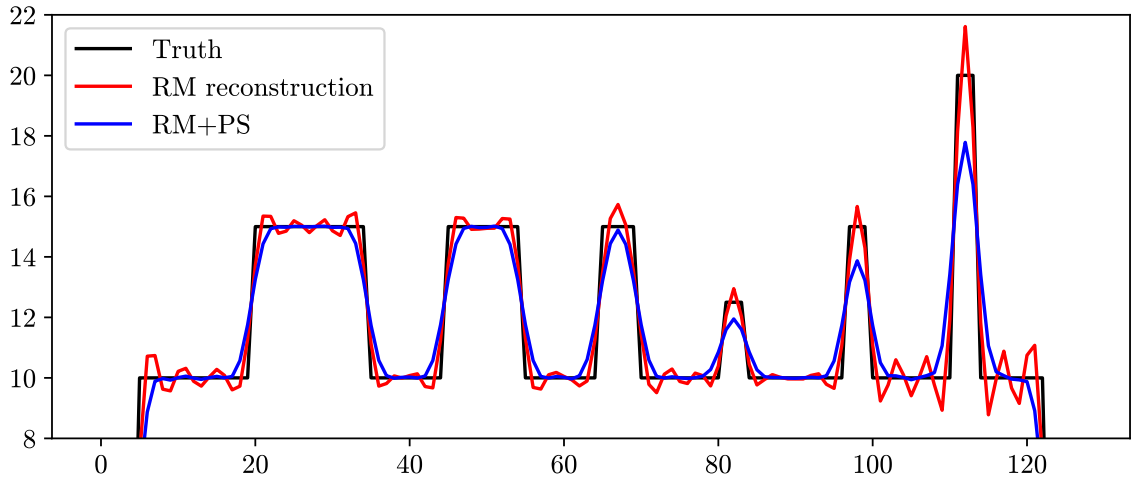


Figure 1.5: Profiles demonstrating the effect of smoothing on Gibbs ringing artefacts. The plot shows a 1D 'object' (black line), reconstruction of the object after a low-pass filter (red), and Gaussian smoothing of the reconstruction (blue).

1.4.2 Total Variation Denoising

TV denoising [66] is an alternative technique with aims to remove noise while retaining edge sharpness. As with smoothing, it also requires an extra hyperparameter, and can also be integrated into the reconstruction process. However, as a standalone post-processing step it also requires a discrepancy term (usually chosen to be the mean square error (MSE)) which quantifies the difference between the original and denoised images. TV is posed as an optimisation probelm, the general form of which is given in Equation (1.16) below. The hyperparameter β controls the strength of the denoising, where $\beta = 0$ recovers the original volume. The effect of TV is to discourage sudden changes in intensity between immediately adjacent voxels, thereby discouraging noise without completely removing edges.

$$\hat{\boldsymbol{\theta}} = \underset{\boldsymbol{\phi}}{\operatorname{argmin}} \left[E(\boldsymbol{\theta}, \boldsymbol{\phi}) + \beta \|\nabla \boldsymbol{\phi}\| \right], \tag{1.16}$$

where $\boldsymbol{\phi}$ is a denoised volume;

E is an error function (usually the MSE);

∇ is the spatial gradient operator, and

β is a hyperparameter ≥ 0.

There are several proposals for solving variants of this minimisation problem. When used in PET, TV is often integrated into the iterative reconstruction as a regularising term [67]. In related work, "adaptive-diffusive" TV has been proposed to reduce cone-bean CT ringing artefacts [68]. Mikhno et al. have also demonstrated that a spatially-weighted version of TV – which relies on per-voxel convergence rates – is capable of removing Gibbs ringing [69]. However, they also note that TV and its variants tend to add localised biases, especially in important regions of interest (RoIs) such as lesions. Unprocessed PET images tend to underestimate peak intensities. Meanwhile, TV-induced biases are unpredictable, sometimes overestimating and sometimes underestimating intensities, as shown in Figure 1.6). TV is therefore usually not used in clinical practice.

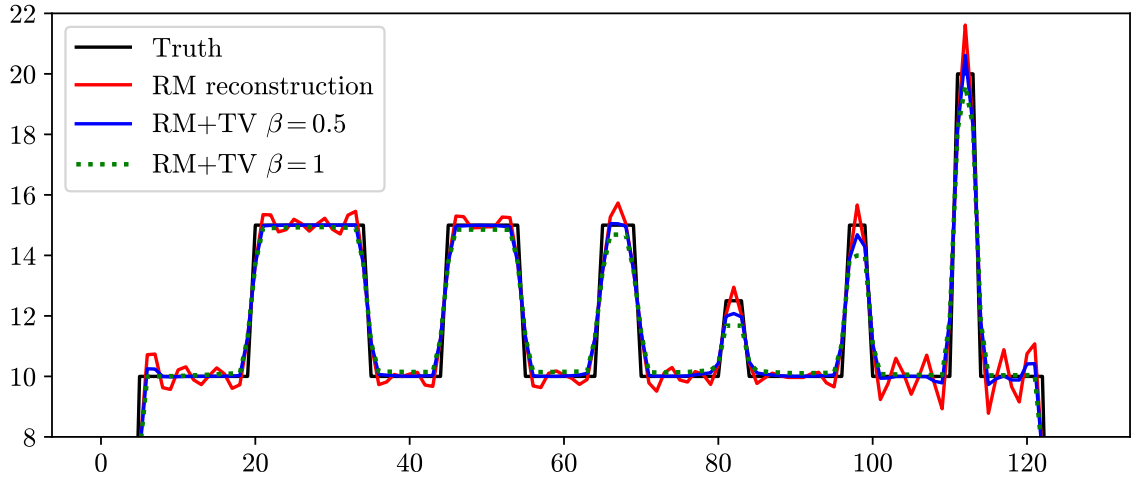


Figure 1.6: Profiles demonstrating TV denoising for different choices of hyperparameter β on the 1D object from Figure 1.5. For $\beta = 1$, TV overestimates the rightmost (small, high-intensity) object peak, and underestimates in the penultimate (small, moderate-intensity) peak.

1.4.3 Guided Filtering

Where available, higher-resolution anatomical information from other modalities (such as MR or CT) can be used to enhance the PET reconstruction [70], [71]. Instead of applying a spatially-invariant smoothing kernel, NLM guided filtering uses spatially-variant kernels $\boldsymbol{w}$. These kernels can be informed by another modality, as shown in Equation (1.17) below.

$$\mathrm{NLM}\left(\theta_j^{(\mathrm{PET})}, \boldsymbol{\theta}^{(\mathrm{guide})}\right) = \frac{\sum_{i \in N_j} w_{j,i} \theta_i^{(\mathrm{PET})}}{\sum_{i \in N_j} w_{j,i}}, \tag{1.17}$$

$$w_{j,i} = \exp\left\{-\frac{1}{2}\left(\frac{\theta_i^{(\mathrm{guide})} - \theta_j^{(\mathrm{guide})}}{\Omega}\right)^2\right\}, \tag{1.18}$$

where $\theta_j^{(\mathrm{PET})}$ is the j^{th} voxel of a noisy PET reconstruction;

$w_{j,i}$ is a weighting factor;

N_j is a neighbourhood around voxel j;

$\theta_i^{(\mathrm{guide})}$ is the i^{th} voxel from a guidance volume (for example, MRI), and

Ω is a hyperparameter ≥ 0.

The filter works under the assumption that if two nearby voxels have similar intensities in the guidance volume, then they should also have similar PET intensities. This assumption clearly cannot be expected to hold for distant voxels, and thus it is important to keep the neighbourhood N small. This neighbourhood can be viewed as a box filter. As an alternative, an attenuating weighting factor which decreases with distance could be used instead of a neighbourhood:

$$w'_{j,i} = w_{j,i} \exp\left\{-\frac{1}{2} d(j,i)^2\right\}, \tag{1.19}$$

where $d(j,i)$ is the Euclidean distance between voxels j and i.

In practice, it is likely that

$$\theta_i^{(\text{guide})} \neq \theta_j^{(\text{guide})} \quad \forall \{i, j\} \in \text{foreground}, i \neq j. \tag{1.20}$$

With this assumption, then in the limit of $\Omega \to 0$, the NLM filter becomes equivalent to the identity function and has no effect. Figure 1.7 below shows T1-guided filtering for different neighbourhood sizes and choices of hyperparameter Ω. Similar guided filtering can be integrated into iterative reconstruction, such as with kernel expecation maximisation (KEM) [72]. A full quantitative comparison of methods after simulated MLEM PET reconstruction in the presence of noise is given later in Chapter 3 and in particular line profiles in Figure 3.17.

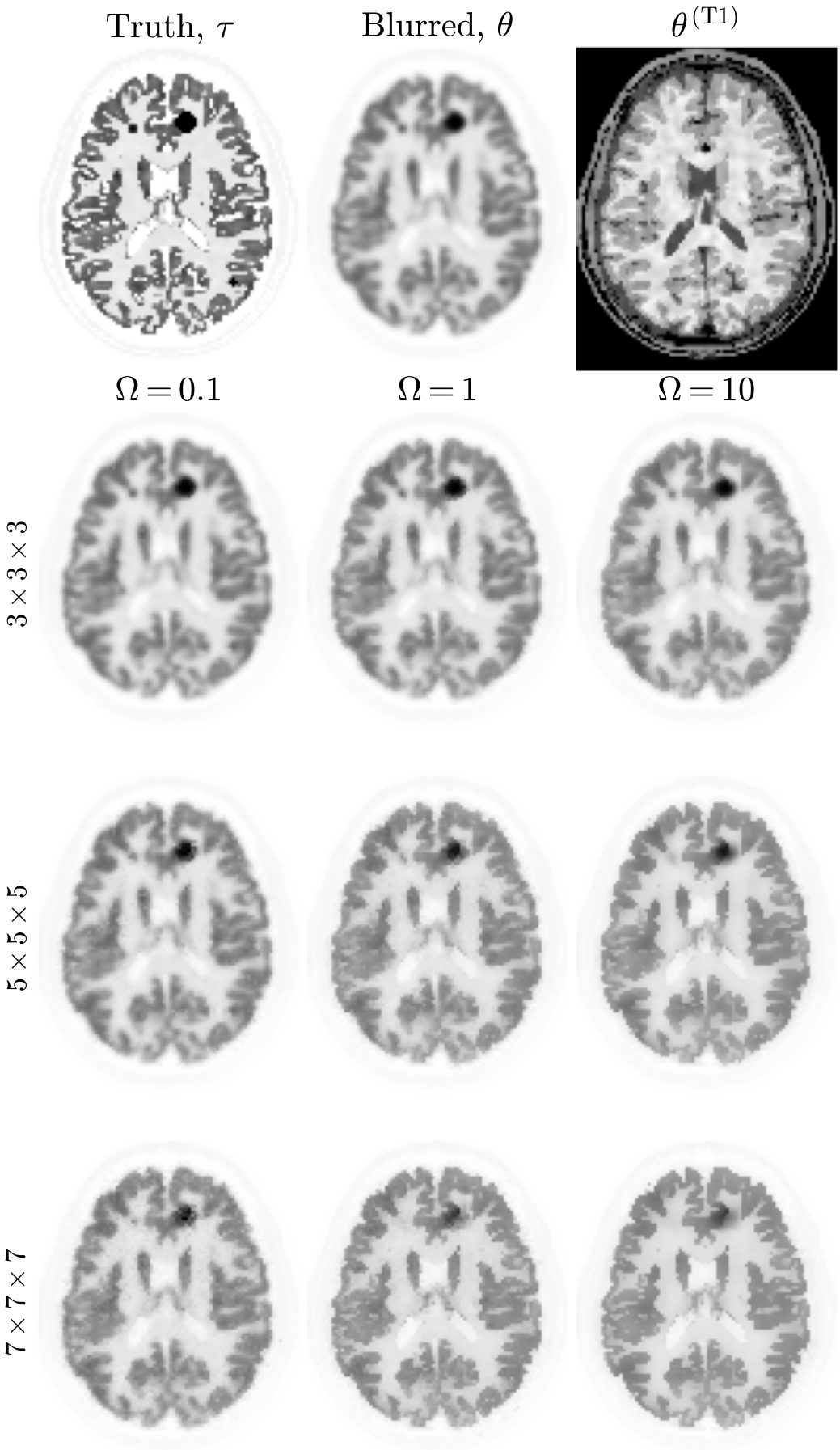


Figure 1.7: Central slices of 3D volumes demonstrating guided NLM filtering as per Equation (1.17). The truth $\boldsymbol{\tau}$ is convolved with a 4.5 mm FWHM Gaussian kernel to produce a blurred volume $\boldsymbol{\theta}$. The blurred volume is then NLM filtered using the corresponding T1-weighted image as guidance with different neighbourhood sizes (rows) and values of hyperparameter Ω (columns).

1.4.4 Registration

Often, registration and alignment (resampling) are necessary prerequisites for guided filtering. If the guiding modality is not acquired perfectly simultaneously as the PET data, then the two modalities are likely to be misaligned. Attempting NLM with misaligned images produces incorrect edges, as shown in Figure 1.8.

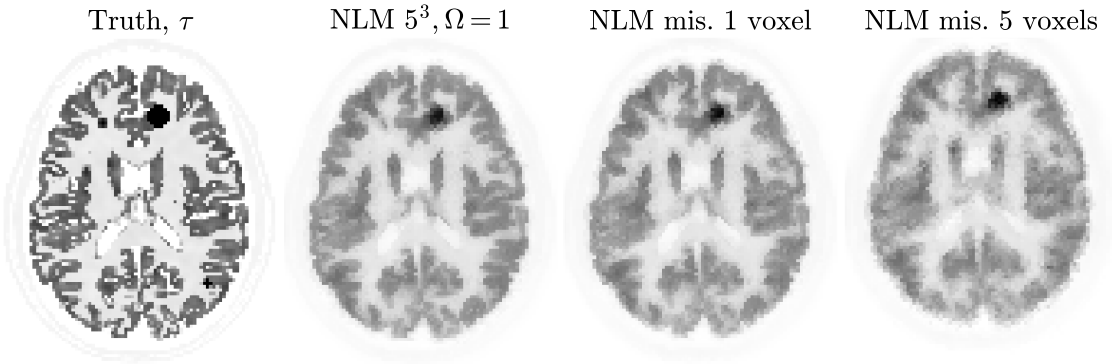


Figure 1.8: Effect of misalignment (misregistration) on NLM. The last two columns show the effect of shifting the PET volume upwards by 1 and 5 voxels, respectively, to the detriment of edge integrity.

In later chapters, alignment is performed using Statistical Parametric Mapping version 12 (SPM12)[2] [73], [74], which uses an algorithm based on [75] along with the normalised mutual information (NMI) objective function from [76]. As part of this work, a Python wrapper for the SPM12 library's co-registration functionality is made available at [77].

1.4.5 Machine Learning

There are also several more complex versions of guided and self-guided filtering. One example is block-matching in 3D (BM3D), where patches (blocks) across an image are grouped into batches based on their similarity, and each batch is denoised separately [78]. There are also many machine learning (ML) informed denoising proposals, including a modified version of block-matching [79].

Another interesting proposal is a residual convolutional neural network (CNN)-based single-image super-resolution (SISR) method applied to PET sinograms [80]. Based on simulations as well as pre-clinical data, the authors claim that "image-to-image" processing of sinogram data prior to reconstruction is better than image-to-image post-reconstruction processing.

Machine learning for image processing has a longer history outside the field of medical imaging, where the focus in mainly on full (3-channel) colour natural images. For example, Dong et al. showed impressive results using CNNs to perform super-resolution (SR) [81], and Ledig et al. used a generative adversarial network (GAN) to achieve similarly impressive SISR [82].

ML post-processing tailored specifically for PET will be explored in more detail in later chapters.

[2] https://www.fil.ion.ucl.ac.uk/spm/software/spm12/

1.5 Research Motivation

Broadly, this work focuses on two main areas: denoising low count PET, and reconstruction artefact removal.

Low counts could be a result of any combination of:

- low radiotracer doses;
- short scan durations, and
- high frame rates for dynamic PET (effectively many short scans).

Dose reduction decreases radiation exposure and therefore increases patient safety. Radiotracers are also difficult and expensive to produce, so reducing dose can also lower the monetary cost of PET scans. Reducing scan time would increase throughput in the clinic; allowing more patients to be scanned per day. Patient comfort is also increased: from the patients' and relatives' perspectives, quicker scans are also more convenient. Shorter scans also help reduce the burden on patients who are often already under a great deal of stress or have motor conditions which prevent them from staying still for the entire duration of a long scan. Artefacts due to motion are therefore reduced, thus potentially removing the need for complicated motion correction and registration techniques. Finally, reducing frame duration (dividing a single scan into multiple shorter durations) would allow for dynamic PET. Dynamic PET is an increasingly active are of research [83]. Rather than a single volume, the output of dynamic reconstructions would be a set of volumes showing the evolution of tracer distribution over time. Each frame would have the same low count problems associated with it as a single short scan duration.

All of the above are clinically desirable due to increased patient safety; increased throughput of patients, and improved time resolution for applications requiring temporal analysis. Unfortunately, both dose reduction and shortened scan durations result in fewer acquired counts, which in turn leads to a low SNR [62]. Reconstructed images therefore would be very noisy and may not be clinically useful.

In practice, there has to be a trade-off between signal reduction and dose and/or time reduction; often called the ALARA principle [23]. Increasing sensitivity of the imaging system would boost the signal. Broadly, there are three subject areas where sensitivity may be improved: Chemistry (improving radiotracers), Physics (improving detector hardware) and Mathematics (post-acquisition processing, including reconstruction and post-processing). This thesis focuses on the latter.

Secondly, the removal of reconstruction artefacts is important; most notably Gibbs ringing in lesions. Unfortunately, it is difficult to automatically distinguish between an artefact and a genuine signal. Artefact removal methods tend to be imperfect and also add bias or reduce resolution. Furthermore, most denosing post-processing methods tend to add more artefacts or biases, as well as sometimes also reducing resolution, further exacerbating the problem. The tasks of suppressing noise and artefacts are therefore strongly linked.

In light of the current literature as well as the increasing prevalence and impressive achievements of ML over the last decade, it seems highly likely that significant improvements should be possible for PET imaging.

However, care must be taken when applying methods inspired by natural image processing to a specialised medical context. For example, recently, Bal et al. suggested combining wavelet, curvelet and NLM denoising [84]. However, resultant images were clearly inferior to plain NLM, and their reported metrics (peak signal to noise ratio (PSNR), contrast, and "edge value") were inappropriate for evaluation of performance in the denoising task, so do not correspond well with a visual assessment of images.

In addition to suppressing with noise and artefacts, this thesis also aims to address the occasional use of inappropriate evaluation metrics in the current literature, as well as draw attention to the hazards of using methods developed without medical imaging in mind.

1.6 Summary of Chapters

The next chapter introduces the fundamental concepts of machine learning, starting with mathematical background (linear algebra, convolutions and notation) before addressing backpropagation and its use in training neural networks. Various types of networks (architectures) are described, including encoder-decoders, residual networks, concatenations, U-nets and the role of discriminators in adversarial training. This is followed by a full overview of applications to post-processing PET in the current literature.

Chapter 3 describes a novel proposal for post-processing called a "micro"-network due to its comparatively small size versus most other architectures in the current literature. After a careful re-analysis of the reasons behind the various common design choices of networks in the current literature (which are ofter meant for natural image processing and text recognition tasks), subtle design features and tweaks are proposed in light of the medical imaging task. This is followed by robust empirical testing on simulated as well as real patient data.

Since ML takes the concept of empirically-chosen hyperparameters to the extreme, it is often called a "black box." The reasons behind the specific values of the weights of a trained neural network are usually hard or impossible to explain. ML is thus viewed with suspicion, particularly in medical practice, where explainability and interpretability are strongly linked to robustness and trustworthyness [85], [86]. Chapter 4 aims to incorporate traditionally accepted iterative reconstruction into the training of a chain of micro-networks in order to introduce constraints on the "black box" and guarantee a level of robustness. The proposed network is constrained to operate in the null space of the imaging system, and thus produces output images which are compatible with MLEM. While these null-nets do not quite achieve the same level of performance as their unconstrained counterparts, marked improvement over traditional post-processing methods is demonstrated.

Chapter 5 embarks upon a much-needed thorough investigation of the most promising methods in the current literature (from Section 2.4). Most publications tend to introduce novel methods without a fair comparison to other state-of-the-art methods, and use significantly different datasets – making relative performance against competing methods impossible. This chapter deliberately investigates a diverse selection of architectural categories, including filling in gaps in the current literature. All networks investigated are trained on the same data and evaluated with appropriate metrics.

Finally, this thesis ends with a discussion and conclusion of findings, and suggests avenues for future research.

Chapter 2

Introduction to Machine Learning for Image Processing

The term artificial intelligence (AI) includes methods and algorithms to approximate human-level behaviour and reasoning to perform tasks. AI can be subdivided into two categories: knowledge based systems, and ML. The former involves hard-coded rules and logic written by human experts. Such approaches tend to become infeasible when applied to highly complex real-world problems. Meanwhile in ML, this task is at least partially automated without explicit human input. Formally, an ML algorithm accumulates experience and as a direct result improves its performance on a task as measured by some objective (loss) function [87, p. 2]. One could argue that traditional iterative methods such as MLEM could satisfy this definition. Perhaps to help further distinguish the two, it may be beneficial to further clarify that ML algorithms are always divided into learning (training) and prediction (validation and testing) stages, and that the original definition of ML only applies to the training phase. As with human reasoning, ML algorithms are comparatively very quick to run after training is complete. Meanwhile, MLEM does not have a fast prediction stage, and must be iteratively run on every new test dataset.

The clear benefit of using ML in PET is that both AI and prior knowledge can be used to their full extent. Iterative reconstruction uses a model of the physics of the system, while an ML-informed post-processing can remove artefacts and noise based on "experience." Unfortunately, it is the very automation of learning by experience – the distinguishing feature of ML – which is also often met with scepticism in the medical imaging community. ML algorithms are often called "black boxes," especially in the case of deep neural networks (NNs). Attempts to understand the internal workings of trained networks are an active area of (largely empirical) research [88].

This chapter introduces the basic concepts behind ML, with a focus on the building blocks of CNNs (which are well-suited to imaging tasks). An overview will also be given of the proposed applications of CNNs in the current state-of-the-art PET post-processing literature.

2.1 Mathematical Background

The methods discussed in this chapter rely heavily on basic linear algebra. The notations used are as follows:

- vectors: lower case bold ($\boldsymbol{x}$);
- matrices: upper case bold ($\boldsymbol{M}$), and
- multiplication: assumed to mean matrix multiplication, unless otherwise specified as element-wise (Hadamard product, $\circ$).

Images and volumes may be represented as "flattened" 1D array (vector) of pixels or voxels. Meanwhile, operations such as integral convolution may be represented as multiplication by circulant matrices[1]. These matrices would be likely be banded (in the case of finite convolutional kernel size) or at least diagonally dominant (for example, in the case of a Gaussian kernel). Spatially-invariant convolution would correspond to each row being identical (albeit shifted), as depicted in Equation (2.1) below.

$$\boldsymbol{M}\boldsymbol{x} = \begin{bmatrix} M_{11} & M_{12} & 0 & \cdots & 0 & M_{21} \\ M_{21} & M_{11} & M_{12} & \ddots & & 0 \\ 0 & \ddots & \ddots & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & \ddots & \ddots & 0 \\ 0 & & \ddots & M_{21} & M_{11} & M_{12} \\ M_{12} & 0 & \cdots & 0 & M_{21} & M_{11} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ \vdots \\ \vdots \\ x_J \end{bmatrix}. \tag{2.1}$$

where $\boldsymbol{M}$ is a (usually diagonally dominant) circulant matrix representing a convolution operation (in this case with a 1D kernel of width 3 and weights $\{M_{21}, M_{11}, M_{12}\}$), and

$\boldsymbol{v}$ is a "flattened" (1D) vector representation of an image (or volume) with J pixels (or voxels).

This equation explicitly shows a 1D kernel. However, a general N-dimensional convolution would have more off-diagonal bands which would act on spatially adjacent (when reshaped into the original image or volume) elements in $\boldsymbol{v}$.

[1] Any matrix $\boldsymbol{M}$ where adjacent rows are shifted, i.e. $M_{i+1,j+1} = M_{i,j}$ with wrap-around at boundaries, i.e. $M_{i+1,j+1-J} = M_{i,j}$.

These conventions are used here purely for ease of understanding, and do not necessarily correspond to implementation. For example, a convolution matrix is likely to be sparse, potentially taking up unnecessary memory and wasting computational time. Instead, optimised libraries are used for implementation. These use kernel weights directly and take advantage of computing hardware such as general-purpose graphics processing units (GPGPUs) to parallelise operations. In particular, code examples are provided using Python (2.7 or later) and Tensorflow 2.0 [89].

Furthermore, the gradient operator (∇) will also be used here. This transforms a function into a vector of partial derivatives, as defined in Equation (2.2) below.

$$\nabla_{\boldsymbol{x}} F = \begin{bmatrix} \partial F/\partial x_1 \\ \partial F/\partial x_2 \\ \vdots \\ \partial F/\partial x_J \end{bmatrix}. \tag{2.2}$$

Meanwhile, the gradient of the function with respect to a single (scalar) argument – for example, x_1 – is written as:

$$\frac{\partial F}{\partial x_1} = F', \tag{2.3}$$

$$\left.\frac{\partial F}{\partial x_1}\right|_{x_1=X} = F'(X). \tag{2.4}$$

Finally, recall the definition of modulo division, where

$$n (\mathrm{mod}\, d) \tag{2.5}$$

is defined to be the remainder when numerator n is divided by divisor d.

The rest of this chapter will make use of these notations and conventions.

2.1.1 Perceptrons

Artificial neural networks (ANNs) are computing systems (inspired by the working of biological systems such as brains and eyes) which can be trained to perform various tasks – such as image denoising. A perceptron is a basic ANN unit, and is a precursor to the more advanced networks in widespread use today.

Given an input vector $\boldsymbol{x}$, a perceptron yields a scalar output. The output is calculated as a dot product between the input vector and a set of trainable weights $\boldsymbol{w}$, followed by a binary thresholding:

$$\pi(\boldsymbol{x}) = \text{sign}(\boldsymbol{x} \cdot \boldsymbol{w} + b), \text{ and} \tag{2.6}$$

$$\text{sign}(y) = \begin{cases} 1 & \text{if } y > 0, \\ -1 & \text{otherwise,} \end{cases} \tag{2.7}$$

where $\boldsymbol{x}$ is an input vector;

$\boldsymbol{w}$ is a vector of optimisation parameters (trainable weights), and

b is a trainable offset (bias) parameter.

Basic boolean algebra can be performed by a perceptron. Assuming the input vector encodes a boolean array, where 1 represents `True` and -1 is `False`, perceptrons can perform logical operations such as `and`, `or`, `not and`, and `not or`. Since these operations are sufficient components to perform any other boolean operations, a few perceptrons are capable of performing complex boolean algebra.

If an output vector (rather than scalar) is required, then multiple independent perceptrons can be used. Such a model is called a perceptron layer. In fact, any boolean function can be represented by a two-layer perceptron. In general, multilayer perceptrons (MLPs) can model very complex processes. For example, images may be represented as vectors, meaning an MLP can be trained to perform complex image processing tasks.

2.1.1.1 Perceptron update rule

Training a perceptron involves iteratively updating the weights and biases. Initialisation of these parameters (before training) is typically randomised (sampled from a Gaussian or uniform distribution). Given a target output, the iterative update of the parameters of a perceptron is given by:

$$\boldsymbol{w}^{(k+1)} = \boldsymbol{w}^{(k)} + (T - \pi(\boldsymbol{x}))\beta\boldsymbol{x}, \text{ and} \tag{2.8}$$

$$b^{(k+1)} = b^{(k)} + (T - \pi(\boldsymbol{x}))\beta, \tag{2.9}$$

where T is the perceptron's target (desired) output;

β is an empirically-chosen step size (also called the *learning rate*), and

k is the iteration number.

This formulation therefore conforms to the definition of supervised learning, where target outputs are required for training. Given a sufficiently small β and linearly separable training examples, Equation (2.8) is guaranteed to converge [90]. This perceptron update equation is straightforward to apply to a single layer perceptron. The case of multiple layers will be considered later in Section 2.2.

2.1.1.2 Batch gradient descent

If the training data are not linearly separable, an alternative update equation can be used to converge on a best-fit estimate. This scheme requires a loss function (also called an objective or cost function). One example is the MSE:

$$\mathrm{MSE}(\boldsymbol{x}, \boldsymbol{y}) = \frac{1}{J}\sum_{j=1}^{J}(x_j - y_j)^2. \tag{2.10}$$

For a perceptron, the loss L is calculated across the entire batch of training data with all pairs of current and desired outputs:

$$L(\boldsymbol{w}, b) = \frac{1}{N}\sum_{n=1}^{N}\left[\pi_{\boldsymbol{w},b}(\boldsymbol{x}_n) - T_n\right]^2. \tag{2.11}$$

where N is total number of training examples.
The optimisation problem is therefore to minimise L, namely:

$$\hat{\boldsymbol{p}} = \underset{\boldsymbol{p}}{\operatorname{argmin}}\, L(\boldsymbol{p}), \tag{2.12}$$
$$\text{where } \boldsymbol{p} = \{\boldsymbol{w}, b\}.$$

This is also called least mean square (LMS). Note the similarity to the MLEM problem posed in Equation (1.9). For linear perceptron units, this loss function is convex, with only one global minimum (no other local minima). Therefore, taking inspiration from Equation (2.8), a sensible parameter update equation would be:

$$\boldsymbol{p}^{(k+1)} = \boldsymbol{p}^{(k)} - \beta\nabla_{\boldsymbol{p}}L. \tag{2.13}$$

When combined with Equation (2.11), this leads to Equation (2.14) below. Note that the gradient is taken with respect to the parameters $\boldsymbol{p}$ rather than input $\boldsymbol{x}$, so the sign function from Equation (2.7) does not need to be differentiable.

$$\boldsymbol{w}^{(k+1)} = \boldsymbol{w}^{(k)} + \frac{2\beta}{N}\sum_{n=1}^{N}(T_n - \pi(\boldsymbol{x}_n))\boldsymbol{x}_n, \tag{2.14}$$

$$b^{(k+1)} = b^{(k)} + \frac{2\beta}{N}\sum_{n=1}^{N}(T_n - \pi(\boldsymbol{x}_n)). \tag{2.15}$$

In general, gradient descent works well where:

- there are a large number of optimisation parameters $\boldsymbol{p}$, and
- the loss function is differentiable with respect to these parameters.

Gradient descent sacrifices the finite-iteration convergence guarantee of the perceptron update rule in order to remove the requirement of linearly separable training data [91].

2.1.1.3 Stochastic gradient descent

Disadvantages of batch gradient descent include slow convergence rates and, more importantly, the risk of converging to a local minimum if the objective function has multiple minima. One proposal to avoid local minima is to use a stochastic modification: instead of updating the parameters based on the entire batch of training data in Equations (2.14) and (2.15), updates can be computed on a per-sample basis:

$$\boldsymbol{w}^{(k+1)} = \boldsymbol{w}^{(k)} + 2(T_{n=k(\mathrm{mod}\,N)} - \pi(\boldsymbol{x}_{n=k(\mathrm{mod}\,N)}))\beta \boldsymbol{x}_{n=k(\mathrm{mod}\,N)}, \tag{2.16}$$

$$b^{(k+1)} = b^{(k)} + 2(T_{n=k(\mathrm{mod}\,N)} - \pi(\boldsymbol{x}_{n=k(\mathrm{mod}\,N)}))\beta. \tag{2.17}$$

One epoch is defined to be the number of iterations required to cycle through all training pairs (N). For sufficiently small β, this stochastic modification approximates the standard (whole batch) gradient descent method. However, due to the increased variability of the gradients of individual samples (compared to the batch total), stochastic gradient descent (SGD) is less prone to getting stuck in local minima. Stochastic methods also require less computational memory since each iteration uses only a subset of all available data, and also tend to converge faster since each epoch updates all parameters multiple times.

In practice, samples are often divided into mini-batches (more than one but less than N samples are used per iteration). This can achieve a balance between SGD and true whole batch gradient descent [92, Ch. 8].

2.2 Backpropagation

The SGD derivations above considered the training of a single perceptron, and are equally valid for an entire layer of perceptrons. For more complex cases with multiple layers and continuous thresholding functions, backpropagation is required.

2.2.0.1 Feed forward

For example, consider a model which transforms the input $\boldsymbol{x}_0$ to the output $\boldsymbol{x}_l$ by applying a sequence of functions f:

$$\begin{aligned} \boldsymbol{x}_l &= f_l(\boldsymbol{x}_{l-1}) \\ &= f_l(f_{l-1}(\boldsymbol{x}_{l-2})) \\ &= f_l(f_{l-1}(...(f_1(\boldsymbol{x}_0)))), \end{aligned} \tag{2.18}$$

where $\boldsymbol{x}_l$ is the output of the l^{th} layer.

Since each function's output is the argument of (fed into) a subsequent function, this is often called a "feed forward" model. The functions themselves are often called "layers," and could for example represent layers of perceptrons. If each layer l applied a scaling $\boldsymbol{W}_l$ and offset $\boldsymbol{b}_l$ followed by a thresholding function A_l, a 2-layer model would take the form:

$$\begin{aligned} \boldsymbol{x}_2 &= f_2(f_1(\boldsymbol{x}_0)) \\ &= A_2(\boldsymbol{W}_2(A_1(\boldsymbol{W}_1\boldsymbol{x}_0 + \boldsymbol{b}_1)) + \boldsymbol{b}_2), \text{ since} \\ f_l(\boldsymbol{x}_{l-1}) &= A_l(\boldsymbol{W}_l\boldsymbol{x}_{l-1} + \boldsymbol{b}_l), \end{aligned} \tag{2.19}$$

where $\boldsymbol{W}_l$ is a multiplicative weight matrix for the l^{th} layer;

$\boldsymbol{b}_l$ is the corresponding offset (bias) vector, and

A_l is the corresponding element-wise thresholding (activation) function.

The thresholding functions A are often called activation functions. It is usual for A to be non-linear, as otherwise the entire system becomes linear and equivalent to a single layer. A common choice of activation function is a rectified linear unit (ReLU) (except for, in order to avoid the vanishing gradient problem, the last layer of a network [93]. See Section 2.2.3 below for more details). ReLU performs element-wise replacement of all negative values with zero:

$$\text{ReLU}(z) = \begin{cases} z & \forall z \geq 0 \\ 0 & \text{otherwise.} \end{cases} \tag{2.20}$$

However, any continuous differentiable activation function is sufficient to satisfy the universal approximation theorem (UAT) requirement for a network to approximate any continuous mapping [94]–[96].

2.2.0.2 Error terms

All parameters (weights $\boldsymbol{W}$ and biases $\boldsymbol{b}$) are initially randomly chosen. The loss function L quantifies the error between the output $\boldsymbol{x}_l$ and the target (desired) output $\boldsymbol{T}$. The aim of the optimisation is to minimise the loss by updating the parameters. Recalling Equation (2.13), the update equation for a single parameter is given by:

$$p_l^{(k+1)} = p_l^{(k)} + \beta E_l^{(k)}; \tag{2.21}$$

$$E_l^{(k)} = -\frac{\partial L}{\partial p_l^{(k)}}, \tag{2.22}$$

where E_l is the error term for a parameter p_l in the l^{th} layer.

Note that this is identical to performing an exponential moving average (EMA) on parameter p_l. The error term can be formulated for entire parameter vectors ($\boldsymbol{p} = \boldsymbol{b}$) or matrices ($\boldsymbol{p} = \boldsymbol{W}$). For the final (output) layer, this means:

$$\boldsymbol{E}_l = [2(\boldsymbol{T} - \boldsymbol{x}_l) \circ A_l'(\boldsymbol{z}_l)] \nabla_{\boldsymbol{p}_l} \boldsymbol{z}_l, \tag{2.23}$$

$$\text{where } \boldsymbol{z}_l = \boldsymbol{W}_l \boldsymbol{x}_{l-1} + \boldsymbol{b}_l, \text{ and}$$

$$\nabla_{\boldsymbol{p}_l} \boldsymbol{z}_l = \begin{cases} \boldsymbol{x}_{l-1}^\top & \text{if} \quad \boldsymbol{p}_l = \boldsymbol{W}_l, \quad \text{or} \\ \mathbf{1} & \text{if} \quad \boldsymbol{p}_l = \boldsymbol{b}_l. \end{cases}$$

Note that $\boldsymbol{z}_l$ represents the pre-activation output of layer l, and $A'(\boldsymbol{z})$ is the derivative of A evaluated at $\boldsymbol{z}$ as per Equation (2.4). Note also the similarity between Equation (2.23) and the error term in Equation (2.14). Apart from the matrix-vector notation, the only change here is the explicit appearance of the activation function gradient $A'(\boldsymbol{z})$.

In general, l may not correspond to the last layer of the MLP. If there are a total of N_l layers, the error term for a parameter p_l in layer $l < N_l$ can be written by expanding Equation (2.22) using the differentiation chain rule:

$$\begin{aligned} E_l &= -\frac{\partial L}{\partial f_{N_l}} \frac{\partial f_{N_l}}{\partial f_{N_l-1}} \frac{\partial f_{N_l-1}}{\partial f_{N_l-2}} \cdots \frac{\partial f_{l+1}}{\partial f_l} \frac{\partial f_l}{\partial p_l} \\ &= -\frac{\partial L}{\partial f_{N_l}} \prod_{l+1}^{m=N_l} \left[\frac{\partial f_m}{\partial f_{m-1}}\right] \frac{\partial f_l}{\partial p_l}. \end{aligned} \tag{2.24}$$

As with gradient descent, a requirement for backpropagation is that all functions (L and f) be differentiable. Thus all terms in Equation (2.24) are straightforward to compute. Using the MSE as the loss function L and recalling the definition of the layer function f_l from Equation (2.19), the general formula for the error term is:

$$\boldsymbol{E}_l = \left[\left(2(\boldsymbol{T} - \boldsymbol{x}_{N_l})^\top \circ \prod_{l+1}^{m=N_l} \circ\, A_m'(\boldsymbol{z}_m^\top) \boldsymbol{W}_m\right)^\top \circ A_l'(\boldsymbol{z}_l)\right] \nabla_{\boldsymbol{p}_l} \boldsymbol{z}_l. \tag{2.25}$$

In this formulation, multiplication operations are performed in order (left-to-right) – including scalar, matrix, and Hadamard operations. Note also that the Hadamard product operator $\prod \circ$ is evaluated here from the upper limit N_l first. The order is important as the matrix multiplications convert input vectors to the required shape for the preceding layer, while the Hadamard multiplications apply the chain rule element-wise.

In this manner, the errors are propagated "backwards" through the MLP from the last layer in order to update parameters. The entire term in square brackets from Equation (2.25) can also be re-used when calculating the error term for the preceding layer ($\boldsymbol{E}_{l-1}$). In practice, this means that fewer computational operations are required if these intermediate results are temporarily stored.

Training is often terminated if there is no significant decrease in loss L with increasing epochs. Specific thresholds and alternative conventions will be discussed in later chapters.

2.2.0.3 Data nomenclature

The equations considered above make use of input and desired output training data pairs $\{\boldsymbol{x}_0, \boldsymbol{T}\}$. Each training pair could be, for example, a low quality and corresponding higher quality PET image. However, it is good practice to also have validation and test data [97]. The definitions of these are as follows:

Training Input and desired output pairs of data used for iterative backpropagation of errors (Equation (2.21) and Equation (2.25)).

Validation Data pairs used to calculate an alternative loss L (usually after each training epoch) for diagnostic/evaluation purposes.

Test Data pairs used to calculate an alternative loss L after the entire training process is complete (i.e. unseen during training) for diagnostic/evaluation purposes.

The term *overfitting* refers to the case where training loss decreases yet test loss increases with increasing epochs [87, p. 67]. Due to this phenomenon, validation loss is understood to be more representative of test loss than training loss. The advantage of using validation data over test data is that this diagnostic can be evaluated at every epoch. Diverging training and validation losses indicates a lack of sufficient amounts of training data. The model effectively memorises the training examples but loses the ability to generalise to unseen (test) data. An example of this phenomenon is shown in Figure 2.1.

If sufficient amounts of validation data are used and overfitting does not seem to occur, one could argue that test data may not be required. Alternatively, *cross-validation* could be used. Cross-validation refers to the periodic cycling of data pairs between the training and validation datasets after each epoch. In such a scenario, it becomes crucial to use distinct test data to evaluate performance post-training to determine if overfitting has occured.

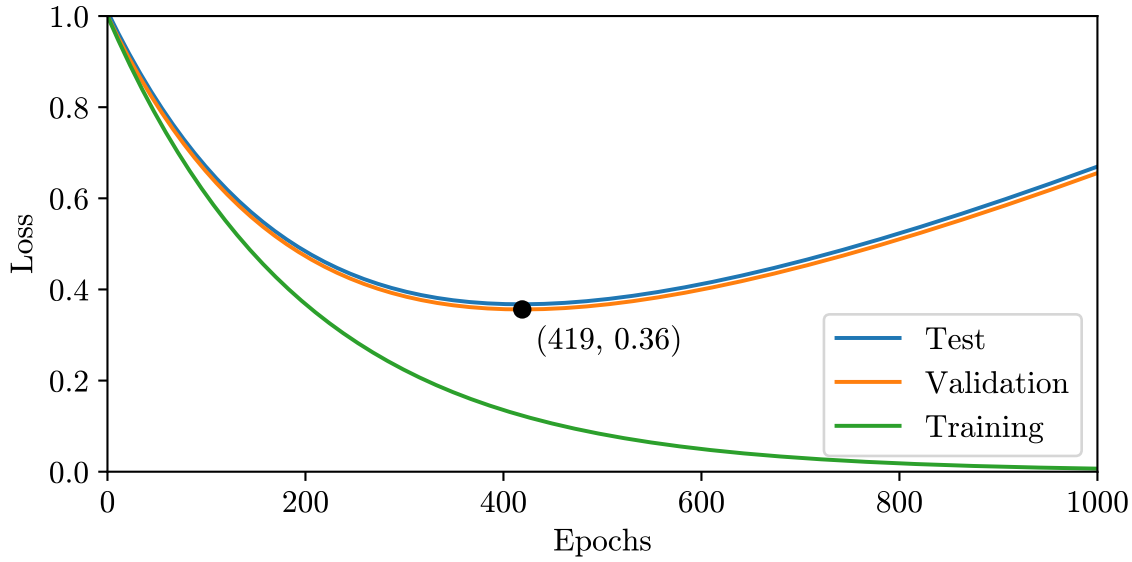


Figure 2.1: Overfitting: training loss continues to decrease while validation (and test) loss increase after 419 epochs.

Ideally, training, validation and test data should each capture the full range of possible inputs and outputs.

2.2.1 Momentum

There have been many proposed stochastic optimisation algorithms. A common modification to SGD is the inclusion of the concept of momentum. If the loss function L is thought of as a hyperplane parameterised by the weights and biases, the task of optimisation is to find the lowest point on this hyperplane (see Equation (2.12)). Ideally the hyperplane should be convex, rendering the gradient descent similar to a rolling marble gradually settling in the bottom of a bowl. The step size β in Equation (2.21) can be thought of as controlling the velocity of this descent.

If, however, the surface is not smooth, there is a risk of being trapped in local minima. If the local minima has dimensions greater than the step size β, it will be impossible to escape. Continuing the analogy, one approach to this problem is allowing the marble to have some momentum which would allow it to continue in the same direction for a short distance even if rolling uphill. The argument is that this will allow escape from small local minima especially where the general surface outside the minima is steep.

The simplest approach would be to constrain the acceleration between subsequent iterations k, for example:

$$|\Delta E_l^{(k+1)}| = |E_l^{(k+1)} - E_l^{(k)}| < \mu, \tag{2.26}$$

where μ is an empirically-chosen hyperparameter.

The classical approach [98] however is to incorporate a velocity term into the iterative update procedure. The parameter update formula (see Equation (2.21)) is altered into a two-step process [99] given by:

$$v^{(k+1)} = \mu v^{(k)} + \beta E^{(k)}, \tag{2.27}$$

$$p^{(k+1)} = p^{(k)} + \mu v^{(k+1)}. \tag{2.28}$$

where $\mu \in [0, 1]$.

Alternatively, Nesterov accelerated gradient (NAG) [100] can be rewritten as a modification to argument of the error term [101]. Using NAG, $E^{(k)}$ is evaluated at $p^{(k)} + \mu v^{(k)}$ instead of at $p^{(k)}$. Explicitly, this replaces Equation (2.27) with:

$$v^{(k+1)} = \mu v^{(k)} + \beta \left. E^{(k)} \right|_{p=p^{(k)}+\mu v^{(k)}}. \tag{2.29}$$

The Adaptive moment estimation (Adam) method [102] is a widely-used and more recent proposal. This takes first as well as second order moment terms (therefore introducing another hyperparameter), and performs bias-correction for both.

2.2.2 Regularisation

Practical issues with backpropagation include slow convergence (requiring many epochs) as well as complete failure to converge. Failure may be due to vanishing gradients (see Section 2.2.3 below) or – conversely – explosion of parameters beyond the limits of computational memory (numerical overflow). For example, single-precision floating-point numbers [103] are used in all implementations here, and cannot represent values greater than 3.4×10^{38} Furthermore, in the case of noisy and/or limited amounts of training data, overfitting is likely to occur. This means that with increasing epochs, L decreases when evaluated on the training data – but increases on validation data.

One definition of regularisation is any strategy intended to increase generalisability (decrease test error) without increasing training error [92, Ch. 7]. In other words, regularisation helps prevent overfitting without harming model performance. However, regularisation in the strictest sense is a modification of the optimisation search space. Such modifications can often also help with convergence. While stochastic methods are unlikely to find the precise global minimum of the unaltered loss function L, regularisation decreases the chance of this even further. However, it should be noted that – as with MLEM image reconstruction – perfectly matching the training data is not desirable when noise is present.

There are different approaches to regularisation, the most popular of which are discussed below.

2.2.2.1 Parameter regularisation

In the current ML literature, parameter regularisation is usually implemented by adding an ℓ_1 and/or ℓ_2 norm of all of the trainable parameters to the loss function [92], [104]. Equation (2.12) is altered to be:

$$\hat{\boldsymbol{p}} = \underset{\boldsymbol{p}}{\operatorname{argmin}} \left[L(\boldsymbol{p}) + \lambda_1 \|\boldsymbol{p}\| + \frac{\lambda_2}{2} \|\boldsymbol{p}\|_2^2 \right], \tag{2.30}$$

where λ_1 and λ_2 are positive hyperparameters controlling the strength of regularisation.
The effect is to make the parameter search space more convex by favouring certain values (in this case, values close to zero). While this makes it very unlikely for the true global minimum of the unaltered loss L to be found, overfitting is also less likely. The optimisation goal would ideally find another minimum very close to the global minimum (i.e. effectively the same training error) while not overfitting (i.e. comparable validation and test error).

In general, ℓ_2 regularisation is particularly good at stabilising underdetermined problems, while ℓ_1 is better at encouraging sparsity (i.e. parameter values set to zero) [92].

2.2.2.2 Early termination

Even if overfitting does occur, it may still be possible to obtain a decent model with reasonable generalisability. Parameter values can be saved at each epoch only if validation loss has reached a new minimum. Post-training, the model state corresponding to minimum validation loss can thus be restored (see the marking on Figure 2.1). Technically, the number of training epochs is a hyperparameter. Early termination (stopping) is therefore a form of regularisation applied to the epoch number rather than directly to the objective function.

2.2.2.3 Data augmentation

Data augmentation is a substitute for providing more training data in order to better represent the true data distribution. The is done by appending "fake" data pairs to the training dataset. When using images, augmentation techniques include affine transformations, masking (including cropping), and addition of noise.

Care must however be taken that augmentations do indeed fall within a realistic distribution, as otherwise overfitting may occur. Augmentation may also make the model insensitive to certain features – for example, rotational augmentations will train the model to be rotationally invariant. In medical imaging, size and shape of features tend to be important, so typically only reflection and rotation are proposed if at all. Other affine transformations such as scale and shear are occasionally used with natural image tasks outside the field of medical imaging.

2.2.2.4 Normalisation

Normalisation refers to rescaling and/or offsetting data in order to resemble a desired distribution. Input normalisation means applying normalisation to the entire space of input data $\boldsymbol{x}_0$ before being fed into the network. Provided that the original distribution information is irrelevant to the model's task, input normalisation can help when processing inputs with large differences in intensity distributions [105].

Alternatively, normalisation can be performed by a layer function f_l. Frequently, this is done using batch normalisation (BN), where mean and variance of layer outputs across the mini-batch dimension (i.e. number of training pairs used in the iteration) are set to 0 and 1, respectively.

2.2.2.5 Dropout

The idea behind dropout is to introduce a probability of ignoring backpropagation pathways, effectively temporarily "switching off" individual perceptron units for one epoch. This means that the associated parameters do not update for units which are switched off, nor do error contributions propagated via them [92]. This strategy encourages units to work more independently of each other, as well as encouraging redundancy – effectively training several narrower MLPs whose contributions can be averaged together for prediction (validation and testing). Dropout has been shown to reduce the likelihood of overfitting [106].

2.2.2.6 Implicit regularisation

In practice, over-parameterised deep neural networks are observed not to overfit as much as theoretically expected, even when no explicit regularisation is employed. There are suggestions that this is due to regularisation being implicit in the training process, especially in the case of SGD, where the effect becomes more pronounced when smaller mini-batch sizes are used [107], [108]. The observed implicit regularisation effect is still an open question, and more recent work suggests that that sequences of layers performing rank reduction may be the main underlying mechanism [109], [110].

2.2.3 Discussion

A common issue with backpropagation is the vanishing gradient problem [111]. This occurs when error terms become small or zero (vanish). Since the terms are calculated as products of gradients, as soon as a zero is encountered the entire associated error propagation pathway is effectively switched off. If, as is common practice, the weights $\boldsymbol{W}$ are initialised close to zero, small values get exponentially smaller as they are multiplied together. As a result, the problem of vanishing gradients becomes worse as the number of layers increases.

The activation function may also contribute to the vanishing gradient problem. ReLU has a zero output and zero gradient for all negative inputs, meaning pathways are easily switched off. If there are a large number of alternative pathways – many adjacent perceptrons in the same layer – this may not harm performance. Alternatively, it may be the case that only a few pathways are required to fit the training data, in which case vanishing gradients effectively act to prune the computational graph, avoiding unnecessary computations.

Both regularisation and SGD momentum can help combat the vanishing gradient problem [99], [112]. Regularisation is also primarily used to decrease the chance of overfitting. Both momentum and regularisation strategies also affect the speed of convergence – usually for the better – but potentially result in suboptimal local minima solutions. Such strategies also usually require tuning of extra hyperparameters.

Advanced stochastic optimisers such as Adam are capable of delivering good results with very little manual fine tuning required [102]. It is however interesting to note that in recent machine learning competitions, winning models tend to use plain SGD with very bespoke learning rate schedulers [113]. It appears that while Adam can be used to quickly train a variety of different models, once a model is selected a slight performance boost should be possible by using a more finely tuned optimiser. It should be noted that alternative algorithms also exist. For example, limited-memory Broyden-Fletcher-Goldfarb-Shanno (L-BFGS) is quasi-newton method (unlike first order methods such as Adam and NAG) and requires large mini-batch sizes to correctly approximate the Hessian matrix – making it perhaps more appropriate for small optimisation problems [114], [115].

Overfitting – decreasing error on the finite number of training data samples, yet increasing error on the entire dataset – indicates a reduction in a model's robustness (ability to generalise) [87, p. 67]. In practice, the entire dataset may be unknown. As a substitute, distinct test data samples can be used instead. Test data is therefore essential to evaluate a trained model's performance. Additionally, validation data can help with evaluation and regularisation (particularly for early termination) during the training process [97].

In addition to architecture and training regime choices, the choice of training data itself is also crucial. For example, the deep image prior (DIP) approach [116] is a form of transfer learning which relies on training a regularised network to map pure noise to a natural image. Training is terminated before complete convergence, and the network is used as a regularising term (in lieu of, for example, TV) in a traditional denoising task for any other image. Despite the DIP being trained on pure noise inputs and unrelated target and test images, the approach has proven surprisingly effective due to a regularised network's apparent high impedance to noise and low impedance to signal.

When training a network, there is often a trade-off between accuracy (i.e. sensitivity and specificity) and robustness to the training examples. The problem likely due to weights storing information learned from multiple different inputs. Any new input altering these weights will thus negatively affect performance for all prior inputs. Such a scenario is called catastrophic interference (CI) [117], [118]. Proposed solutions for CI thus tend to centre around constraining weights to each learn from one input.

There are various types of ANNs, including MLPs, belief networks, recurrent neural networks (RNNs) (including, for example, Boltzmann machines) and CNNs. The latter in particular are naturally well-suited to tackling image processing tasks, and are therefore detailed in the next section.

2.3 Convolutional Neural Networks

Perceptrons are often referred to as *neurons* in the current literature. This is because of some (limited) similarities to the operation of biological neurons. As MLPs consist of a set of interconnected neurons, they are also commonly referred to as NNs.

The problem with NNs is that they tend to have very many parameters, and thus take a long time to train in order to perform seemingly simple tasks. CNNs are a subset of NNs which are inspired by retinal ganglia and therefore well-suited to imaging tasks. Neurons in the human brain are capable of switching between states in around 10^{-3} s, yet humans can perform complex recognition tasks in around 10^{-1} s. Since it is hypothesised that a few hundred switches are insufficient to perform such complex tasks, it seems likely that biological neural systems perform highly parallel processing [87, p. 82]. NNs also aim to imitate the highly parallelisable operations which are likely performed by the human brain, and CNNs take this further by imitating image processing operations performed by the retina. Small convolutional kernels act as local feature detectors. When combined with non-linear activation functions (which control thresholding or sensitivity), these convolutional layers can be chained together to perform complex tasks such as object detection, segmentation, denoising and artefact removal.

The basic idea behind a CNN is to replace the multiplicative weights of an MLP's layers with convolution operations. However, Equation (2.19) demonstrated that perceptron layer weights can be represented by a matrix $\boldsymbol{W}$. Meanwhile Equation (2.1) demonstrated that a convolution operation can also be represented by a matrix. This means that the SGD backpropagation algorithms discussed above also apply to CNNs.

Convolutional matrices are typically sparse, and more specifically, circulant. The circulant matrix multiplication with an image vector convention outlined in Equation (2.1) – while being a mathematically convenient notation – is not particularly easy to visualise. It is easier to consider input images directly as 2D matrices, and convolutional kernels as smaller 2D templates applied to each spatial region of the input, as shown in Figure 2.2. Note that this can generalise to any other number of dimensions (e.g. 3D volumes).

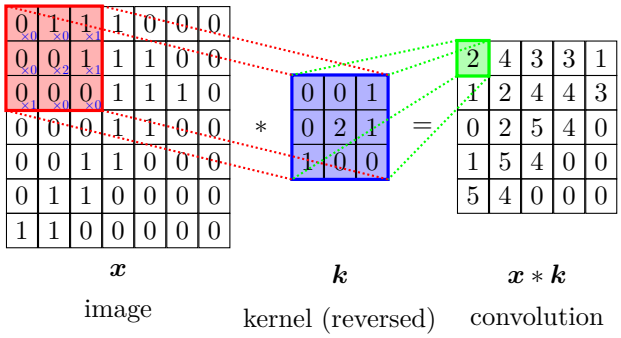


Figure 2.2: Visualisation of 2D convolution without padding.

The convolution depicted does not use padding – the output image is smaller than the input image as the kernel is not allowed to exceed the input boundaries. Alternative padding strategies essentially augment the input image so that the output shape is the same as the unaltered input. Common strategies are shown in Figure 2.3. In the current literature, the two most commonly used are unpadded (also called "valid") and zero-padding.

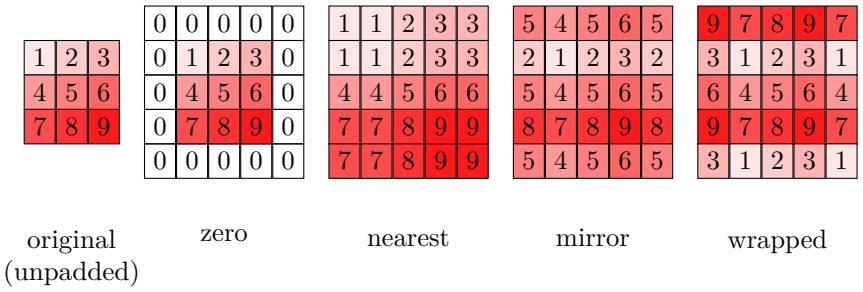


Figure 2.3: Visualisation of 2D padding strategies.

As with retinal ganglia, convolution can act as very powerful feature detectors, as demonstrated in Figure 2.4. The idea behind CNNs is to chain together local feature detectors with activations/thresholding to perform complex tasks.

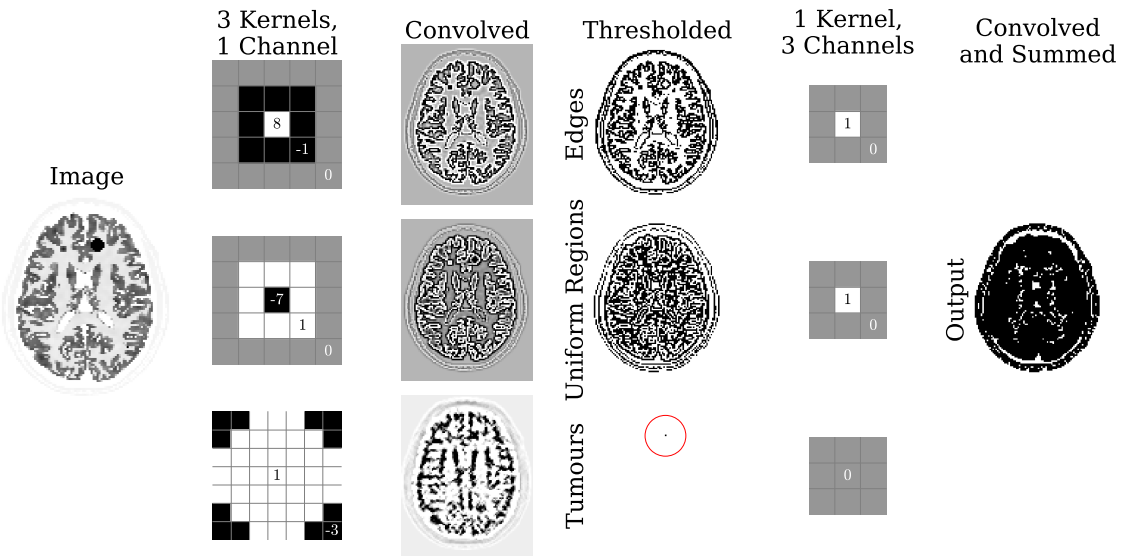


Figure 2.4: Convolutions as feature detectors. An input image (column 1) is convolved with 3 different kernels (column 2) yielding 3 different images or channels (column 3), each of which has a different bias added followed by ReLU thresholding (column 4). By careful selection of the kernel weights and the biases, features (namely, edges, uniform regions, and large tumours) have therefore been detected. Channel reduction is also possible: convolving each channel with a different kernel (column 5) and element-wise adding the resultant images yields a single-channel output image (column 6) – in this case a segmentation mask corresponding of all foreground pixels.

It should be noted that in a CNN, each convolution operates on the entire input layer. If the input consists of multiple 3D volumes (i.e. 4D), the convolution matrix will map the multi-channel 3D input to a single 3D output channel. Furthermore, a single layer in a CNN may consist of multiple independent (multi-channel) convolutions. Each convolution would result in an output channel. Channels are grouped together to form the overall layer's output. In this way, a convolutional layer is a many-to-many channel mapping. Biases $\boldsymbol{b}$ are most commonly applied on a per-channel basis (i.e. a single bias value is added to all elements of a channel). The reasoning behind this convention is that each (multi-channel) convolution acts as a detector for just one feature, so just one corresponding bias is required to control sensitivity to the feature.

The term convolution (Conv) layer will be used here to refer to a multi-channel convolution followed by addition of biases (resulting in $\boldsymbol{z}$ from Equation (2.23)). In contradiction, some of the current literature occasionally includes the subsequent non-linear activation function A in the definition of a convolutional layer (resulting in $\boldsymbol{x}$). Since activation functions are discussed separately here, it makes more sense to separate the two concepts in this work. Conversely, an alternative viewpoint is to include biases as part of the activation function A, since both biases and activation functions work together to perform non-linear thresholding. However, one could argue that the kernel scaling also contributes to this thesholding – as do all preceding layers. Since biases are also usually per-convolutional scalars which control the response of individual output channles, it makes more sense to include them in the definition of convolution.

There are a number of other features of CNNs which are worth defining. These are summarised below.

Layer height The spatial size (i.e. dimensions excluding number of channels) of a layer's output.

Layer width The number of output channels of a layer (i.e. for a Conv layer, the number of convolutions).

Network depth The number of Convs layers.

Receptive field The spatial dimensions of a region in the network's input which could affect a single output element.

The receptive field of a network extends over the entire input for an MLP, but tends to be much smaller for a CNN. Increasing the receptive field of a CNN can be achieved by increasing layer densities – i.e. increasing their kernel sizes, thereby decreasing the sparsity of the circulant weight matrices – making them closer to perceptrons. This greatly increases the number of optimisation parameters and loses the benefits of CNNs over ANNs. For example, a single image-to-image mapping perceptron layer with a receptive field $\mathcal{O}(N^2)$ has $\mathcal{O}(N^4)$ parameters (where N is the width of the input image). Alternatively, a deep CNN can be used. Applying a chain of convolutional kernels of a small width (usually 3), the same receptive field can be achieved with just $\mathcal{O}(N^2)$ parameters – scaling linearly rather than quadratically with field size.

Note also that a network must have sufficient width and depth in order to meet the UAT requirements for approximating any continuous mapping [95], [96].

2.3.1 Layers

There are several common types of layers, some of which are summarised below.

Fully connected (or dense) A layer of perceptrons, i.e. each output element is a function of all input elements.

Convolution (Conv) One (or more) multi-channel convolution(s) followed by element-wise addition of a per-channel bias.

2.3.1.1 Spatial sampling

Layers which alter the spatial resolution are given below.

Strided Conv Spatial downsampling using convolutional kernels which are applied every $i > 1$ elements.

Transposed Conv Spatial upsampling using fractional strided Conv. Can also be thought of as applying the transpose of an integer-strided Conv matrix.

Pooling Spatial downsampling by grouping nearby elements and outputting one (e.g. the maximum) element per group.

Linear interpolation Bilinear or trilinear spatial downsampling or upsampling.

2.3.1.2 Activation functions

There are functions which control the sensitivity (thresholding) of the previous layer. For this reason, both BN and dropout can be thought of as special types of activation functions.

Batch normalisation Offset and scaling normalisation (i.e. zero mean and unit standard deviation) across the mini-batch dimension (i.e. number of training pairs used in an iteration of backpropagation). When used, BN is applied before any other activation function.

Dropout Randomly treats elements as zero with a probability $\alpha \in (0, 1)$ during training (every iteration), but scales by α during prediction (validation and test). The constant α is an empirically chosen hyperparameter.

Sigmoidal Element-wise $A(z) = (\exp\{-z\} + 1)^{-1}$.

Hyperbolic tangent Element-wise $A(z) = tanh(z) = \frac{\exp\{z\}-\exp\{-z\}}{\exp\{z\}+\exp\{-z\}}$.

Exponential linear unit Element-wise $A(z) = \begin{cases} z & z > 0, \\ \exp\{z\} - 1 & z \leq 0. \end{cases}$

Rectified linear unit Element-wise $A(z) = \begin{cases} z & z > 0, \\ 0 & z \leq 0. \end{cases}$

Leaky rectified linear unit Element-wise $A(z) = \begin{cases} z & z > 0, \\ \alpha z & z \leq 0, \alpha > 0, \end{cases}$ where the constant α is an empirically chosen hyperparameter.

There are also a few special types of layer which can operate on multiple previous layer outputs. The two most widely used of these are residual connections and concatenation connections. In the current literature, both are often called skip connections. Due to this ambiguity, the term "skip" will be avoided here.

A concatenation simply means appending the outputs of two or more layers together (along the channel dimension). Meanwhile, residual connections perform element-wise addition of outputs of multiple layers [119].

Since convolutions are essentially weighted summations, a residual can be implemented as a concatenation followed by a convolution layer (with unit kernel spatial dimensions and zero bias).

Residual *blocks* refer to groups of layers which have a residual connection between the group's input and output layers. If a residual connection is present between the input and output of the entire network, the overall architecture can be called a residual network (ResNet).

2.3.2 Visualising network architectures

A layer's height and width (i.e. output spatial size and number of channels) can be represented visually using rectangular blocks with a corresponding height and width. An example network diagram is given in Figure 2.5. This shows a 64-channel input followed by a 32-convolution layer with stride 2 (see Section 2.3.1.1), followed by a sigmoidal activation function. The next layer concatenates (along the channel dimension) the convolution and activation layer outputs.

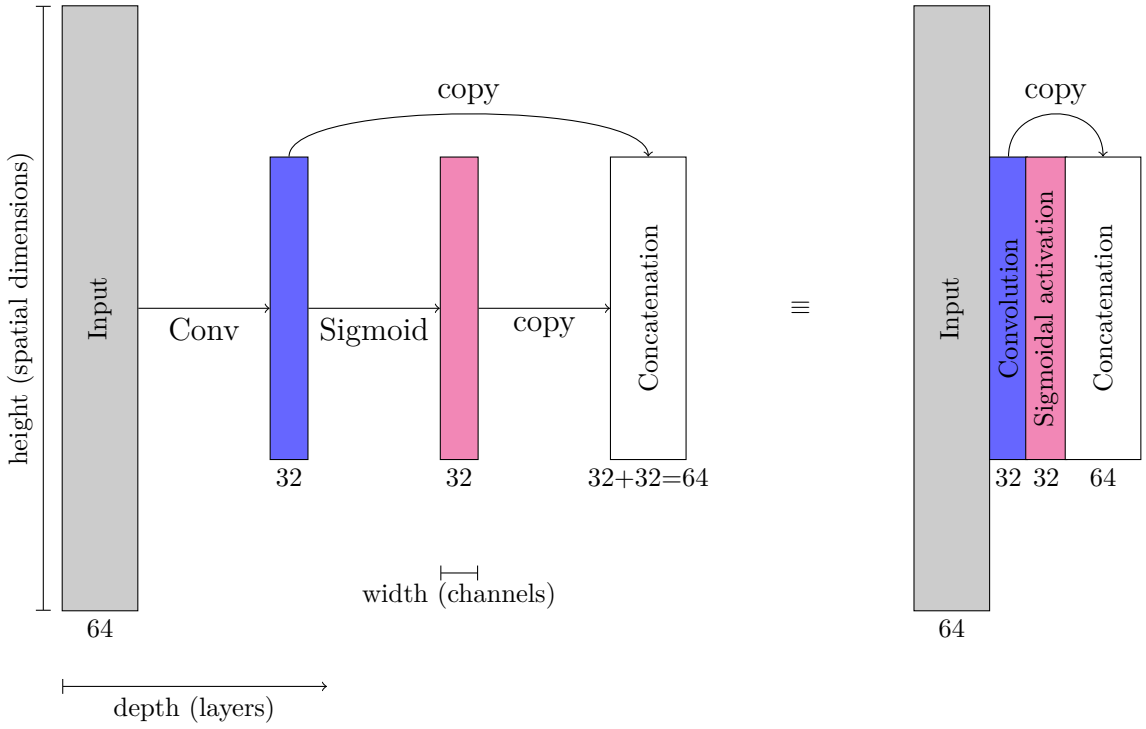


Figure 2.5: Visual representation of a simple network. Each block represents the output of a layer, with the number below each block signifying its width (the number of output channels), and the height corresponding to the spatial size. An equivalent compressed version – hiding operations between adjacent layer outputs – is also shown.

In this context, a deep neural network (DNN) is taken to mean a CNN with a large depth (i.e. number of layers). "Large" is itself an arbitrary term. Some proposals call 10 layers "shallow" [120] while others claim that any more than 1 layer is "deep" [121]. It should also be noted that this is different from the term *deep learning* – any end-to-end mapping from raw data to desired output without any manual feature extraction from the raw data [122].

2.3.2.1 Convolutional Encoder-Decoder Networks

In mathematics, the term *encoding* refers to a mapping from one space to another. Dimensionality reduction is not a requirement for encoding. Conversely, in the current literature on neural networks, the term is often used to refer to layer(s) which also perform spatial downsampling. This is due to the assumption that such downsampling will force the layer to produce a compressed version of its input. The term *compression* in turn implies encoding into a relatively small latent space (which implies dimensionality reduction, i.e. a compressed sensing method).

Convolutional encoder-decoders (CEDs) are networks which perform encoding and decoding. *Autoencoders* are CEDs with the same overall network input and output dimensions. As with the latter, once again it is usually understood that spatial downsampling and upsampling will occur in intermediate layers, though strictly speaking this is not a requirement.

The idea behind the CED architecture is to force noise to be discarded via compression, followed by a restoration of the original dimensions via decoding, resulting in a denoised network output. It should be noted that the combination of convolutions and downsampling can also quickly increase the network's receptive field.

2.3.3 U-Nets

U-nets [123] are CEDs with skip (either residual or, more commonly, concatenation) connections forming a characteristic U-shaped architectural diagram. Skip connections are placed between pair of layer outputs of the same spatial dimensions. Each pair consists of an output from the encoder and the decoder parts.

The extra connections are designed to ensure high spatial frequency details – which may not be noise – do not have to be lost. They also help alleviate the vanishing gradient problem: deep encoding layers producing null values will not result in the overall network producing a null output.

2.3.4 Adversarial Networks (Discriminators)

One trained, networks often produce outputs which look artificial and unnatural. Simply minimising an objective function during training does not guarantee that generated images will look realistic. To alleviate this problem, a second network can be trained to distinguish between the target (desired) and generated (artificial) outputs. This discriminator network usually has a binary cross entropy (BCE) loss function, given by:

$$\mathrm{BCE}(\boldsymbol{x}, \boldsymbol{y}) = \frac{1}{J} \sum_{j=1}^{J} \left[-y_j \log x_j - (1 - y_j) \log (1 - x_j) \right], \tag{2.31}$$

where $\boldsymbol{x}$ and $\boldsymbol{y}$ are vectors of probabilities (elements $\in [0, 1]$) of being a target (rather than generated prediction).
The discriminator loss function L_D is subtracted from the generator's loss function L_G to produce a modified loss L'_G. Both networks are trained side-by-side (every iteration of backpropagation, L_D and L'_G are alternately used to update the corresponding network's parameters). In this way, the generator and discriminator must compete with each other in an adversarial manner.

Together, the generator and discriminator networks are called a GAN [124].

2.4 Application to Post-processing PET: Denoising and Artefact Reduction

This work focuses on applications of CNNs to PET denoising and artefact reduction. There have been several proposals in this area, a few of which are explored in detail in Chapter 5. In PET and medical imaging in general – unlike natural image processing – small, high spatial frequency components are often very important. Simply assuming such components are noise and removing them is something which CNNs are prone to doing, and any proposed method must take this into consideration. Network instabilities causing unpredictable false positives and negatives as well as artefacts can be hazardous in medical imaging [125].

One approach would be to train an autoencoder architecture to map one reconstructed noise realisation to another (of the same noise level). Normally, a CED is trained to be an identity operator – the training input and targets are identical – but a relatively short intermediate layer forces encoding into a latent space (i.e. a smaller number of dimensions than the input). This acts as a bottleneck, and is intended to force a loss of less relevant high spatial frequency information such as noise. Unfortunately, the suppressed high frequency components may also represent genuine signal (such as sharp edges). In low count PET, one has the advantage of being able to generate multiple noise realisations from the same object. A network trained to map different noise realisations to each other would not require a bottleneck to remove noise – the differences in the training data pairs should in theory provide enough information to perform this task. Without a bottleneck, high spatial frequencies corresponding to genuine signal are also more likely to be preserved. Current proposals for low count PET include the *Noise2Noise* architecture [126] and a "consensus loss" modification [127]. It should however be noted that noisy targets are theoretically acceptable only where the noise has zero mean [126]. Since the noise properties in PET reconstruction result in a positive bias, the Noise2Noise approach may not be completely valid.

Dynamic PET reconstructions can also be denoised using data from adjacent frames, or even using the whole-scan reconstruction as a prior. One recent proposal uses a modified version of the DIP approach [116]. Instead of random noise, the whole-scan reconstruction is used as the DIP network input [128].

More recent work suggests combining multiple different networks – for example, a noise-detecting network being used to focus the attention of a denoising residual U-net [129].

Anatomical information from jointly-acquired (or coregistered from separate acquisitions) modalities can also be used to enhance PET images. Chapter 5 will explore the specifics of CNNs architectures, focusing on PET and MR-guided PET post processing.

2.4.1 Related Work

Due to the safety hazards associated with nuclear medicine, healthy volunteer PET data are relatively uncommon. For the same reason, very high quality (i.e. high count) clinical data – which may serve as an approximation of the ground truth – is also not readily available. By comparison, other medical imaging modalities such as MR and ultrasound tend to have much more healthy volunteer and high quality data readily available. Supervised machine learning methods typically require large amounts of training data pairs. Primarily due to the more readily available data, there exist a larger body of literature on medical image denoising for these modalities.

Many post-processing networks – especially in MR – are used for segmentation purposes. Since segmentation images are by definition more uniform than denoised ones, CNNs (which as discussed tend to suppress high spatial frequency details) have been applied to this problem with great success. Other research areas of interest closely related to segmentation include classification and prediction. For example, AD prediction using [^{18}F]FDG PET [130].

Image super-resolution is another possibility which is closely related to denoising and artefact removal. For example, Song et al. recently investigated the use of CNNs for MR-guided PET super-resolution, where spatial coordinates were injected into input channels [131]. Injection of coordinates improved performance (in terms of PSNR, structural similarity index (SSIM) and contrast to noise ratio (CNR)) for very deep (20-layer) networks, while shallower 3-layer networks were not able to effectively use this additional information.

Muckley et al. have shown that networks trained to remove Gibbs ringing artefacts and noise from natural images can successfully remove noise and artefacts diffusion MRI, thereby allowing a halving of acquisition time without noticeable degradation in image quality [132]. Chapter 4 will explore this concept in more detail in the context of PET imaging.

Meanwhile, there is increasing interest in integrating ML into various aspects of traditional iterative reconstruction [133], [134]. For example, Hwang et al. [135] use maximum likelihood activity and attenuation (MLAA) reconstruction (with ToF) to generate estimated attenuation maps. They then use a 2D CNN to fuse the joint reconstructions (both activity and attenuation) into a better (equivalent to CT-derived) attenuation map for use in OSEM. Three CNNs were considered: convolutional autoencoder (CAE), U-net, and a "hybrid" (U-net with fewer concatenation connections). The "hybrid" was found to perform best in terms of root mean square error (RMSE), despite the fact that all networks were identical apart from the concatenations (i.e. the U-net should be the most powerful).

ML-informed regularisation terms can also be directly incorporated into the reconstruction process. Gong et al. have suggested an using the alternating direction method of multipliers (ADMM) algorithm to iteratively reconstruct images while using a DIP network for regularisation [115]. Alternatively, a denoising U-net could be used as a regularising term [136]. Kim et al. suggest a similar approach using patch-based denoising CNN [137] which shows promise at removing bias regardless of noise level. This is done by applying a local linear fitting (LLF) function (modulating the input image with the denoised image as with guided filtering). Results are shown to outperform TV and NLM guided filtering. However, Kim et al. show that the standard unrolled approach (no LLF/bias correction) is inferior to plain NLM. This is counter-intuitive since the far superior CNN should be able to outperform NLM even with a small bias due to mismatched noise levels (assuming normalised inputs). The more recent block coordinate descent (BCD) net strategy by Lim et al. goes further by using a different CNN for iteration [138]. Lim et al. claim superior performance to the single-network methods since their BCD net is designed to tackle Poisson noise (rather than segmentation or general Gaussian noise as is the case with the U-net and CNN methods, respectively). Despite being an arguably safe modification to standard clinical OSEM (the regulariser's weight can be made arbitrarily small to recover the original OSEM reconstruction), the BCD net results still showed unpredictable false positives and negatives – in certain cases leading to inferior activity recovery than OSEM or TV denoising. An updated proposal by Gong et al. called MAPEM-net also uses a distinct network (in this case, a U-net) per iteration [139], incorporating basic distance-driven projectors to facilitate backpropagation [27]. The preliminary results were comparable to CNN-based post-reconstruction processing – though further validation and testing is still required.

There has also been an increasing amount of interest in using deep networks to perform the entire reconstruction from raw scanner data (so called end-to-end reconstruction). This would effectively require the network to learn an approximation of the inverse system matrix. The networks can only learn an approximation since an uncompressed version of this matrix would be prohibitively large – approximately 1 PB even assuming span-11 compression and 1-byte bins ($[344 \times 344 \times 127] \times [837 \times 344 \times 252]$). However, given the sparsity of this matrix, learning a compressed version may be viable. Proposals for sinogram-to-image end-to-end reconstruction include *AUTOMAP* (a 3-layer perceptron followed by a 2-layer CNN) [140] and *DeepPET* (a 31-layer CNN) [141]. Both networks work on cropped 2D sinograms in order to make the inverse problem more tractable. Admittedly, *AUTOMAP* was intended for direct reconstruction of MR images (where the inverse system matrix is essentially an inverse undersampled Fourier transform, which in turn can be emulated by a single perceptron layer), and results for PET sinograms were disappointing. *DeepPET* however uses a very different architecture and focuses on whole-body PET. Unfortunately, while managing to produce images with low noise, *DeepPET* also produces many misleading false positive and false negative features, and traditional maximum likelihood reconstruction remains clearly clinically superior – especially for brain imaging.

Alternatively, several competitive post-processing CNN-based methods have also been proposed. These have the advantage of being much quicker to apply, and thus facilitate easy use in practice – both in the clinic and retrospectively on large datasets. The following chapter investigates one such post-processing CNN.

Chapter 3

Micro-Networks

This chapter focuses on post-reconstruction MR-assisted image quality enhancement for low count PET. The proposed CNN is intended to perform denoising and artefact removal without introducing bias. The aim is to transform low count reconstructions into clinically useful images for tasks where they would otherwise have been of prohibitively low quality. As this is a specific problem, the proposed post-processing network is designed from first principles with this aim in mind (rather than starting with a pre-existing generic denoising network architecture and making modifications to it). The resultant network has a relatively low complexity (at least an order of magnitude fewer parameters than those commonly found in the current literature), and a novel name is therefore proposed for it: micro-net (μ-net) [142].

3.1 Motivation

As discussed in Section 1.5, noise suppression is especially important in low count PET imaging.

Methods which aim to reduce noise also tend to reduce resolution (such as with PS) and can also introduce bias (such as certain regularising methods). Where available, higher resolution anatomical information from another modality (such as MR) can be used to assist with the task of denoising and artefact removal. Traditional joint reconstruction for PET-MR incorporates regularisation terms. One example would be the inclusion of a TV penalty term into the reconstruction of both modalities [143]. Such methods slow down the reconstruction and also are sensitive to empirically-chosen hyperparameters. Alternatively, CNNs can be used as regularisers during reconstruction [133]. As described in Section 2.3, CNNs are well suited to joint image processing. Additionally, networks can be applied as a learned post-processing step. Such a network would be much easier to implement and quicker to run in clinical practice than regularised iterative reconstruction (e.g. [144]), and far more powerful than traditional post-processing methods such as smoothing and guided filtering.

Unfortunately, CNNs often require large amounts of labelled data to train (one proposed smartphone image denoising dataset contains 24,000 image pairs requiring half a terabyte of storage [145]). This can be difficult and expensive to obtain, especially in nuclear medical imaging. Furthermore, the copious amount of denoising and artefact-reducing work outside the field of medical imaging may not be directly applicable here. In particular, algorithms designed to work on natural images (where no human life is at risk) are often designed with a much higher false positive and false negative tolerance than would be acceptable in a medical context. While there is increasing interest in ML-informed image quality improvement in PET (see Section 2.4), the field is still relatively understudied. Proposed network architectures are seldom presented with any justification of choice of hyperparameters. For example, some of the works discussed propose a U-net (originally designed for segmentation [123]) which is slightly tweaked to perform denoising [146]. Other proposals to automate "architecture engineering" can be prohibitively time-consuming [147]. Instead, this chapter focuses on designing an architecture from first principles with the specific task of PET post-processing. A more thorough comparison of existing state-of-the-art proposals is conducted in Chapter 5. It should be noted that early in this research project, data from only four clinical patient scans was available for training and testing [142], meaning any proposed network would have to be exceptionally robust against overfitting. Chapter 5 includes a more thorough analysis for larger training patient datasets.

Many post-processing networks in the current literature also operate on 2D slices or even small patches rather than full 3D volumes. This may be because such networks are often based on proposals for 2D natural image processing, or because of prohibitively large computational memory requirements. Backpropagation (Section 2.2) requires storing outputs of all intermediate layers in order to calculate partial derivatives, meaning a network with many layers operating on images or volumes with a large number of elements can easily saturate available memory. While using smaller 2D patches can alleviate this problem, it also removes the ability to perform partial volume effect correction using data from adjacent slices. Patching and re-stitching patches together also results in a computational overhead, and can be a source of artefacts if boundaries are not handled appropriately.

Note that PET denoising is a very different task from consumer camera/natural image denoising. PET reconstructions have a much higher level of and different distribution of noise. It is crucially important to not mistake a small genuine signal as noise, since such signals could correspond to a small lesion. Conversely, it is also crucial to avoid preserving noise in a misleading way – for example, retaining a small noise spike which can be mistaken for an operable tumour. Reconstructions also have higher noise than other modalities such as MR and CT. Therefore, techniques which are in widespread use for other imaging tasks will not necessarily work well for PET.

Finally, robustness to unseen (test) data is particularly important in medical imaging, and demonstrating that overfitting does not occur is vital if a model is to be used in clinical practice. *Occam's razor* is related to model generalisability and avoiding overfitting; stating that the simplest model which meets the performance target is best. This is exemplified in decision trees – a precursor to neural networks – where the well-known iterative dichotomiser 3 (ID3) algorithm has an inductive bias towards shorter trees [87, p. 65]. Similarly, wide residual networks (WRNs) have been shown to outperform much deeper networks of even "thousands of layers" [148]. Given this, it is surprising that most of the current literature concerning CNNs focuses on increasing network depth. Depth is a key advance in deep learning, allowing the approximation of nearly any function mapping. However, needlessly complicated models which ignore the underlying function mapping should be avoided.

A re-exploration of architectural design choices is required in the context of this PET imaging task, and the CNN proposed in this chapter aims to address the problems outlined above.

3.2 Methods

3.2.1 Datasets

The data used for training, validation and testing consists of simulations as well as real clinical patient reconstructions. The main advantage of simulation is that the ground truth is known, allowing for comprehensive analysis of results. However, simulations are ultimately artificial and can only approximate real patient data. Details of each dataset are given below.

3.2.1.1 Simulations

Phantoms used in simulations are derived from MR-based segmentations of 20 subjects in the *BrainWeb* dataset [149]. Each subject is modified to have [^{18}F]FDG PET-like intensities, given in Table 3.1 below.

Type	Intensity scale factor
White matter	1
Grey matter	4
Skin	0.5
Hyperintense lesions	6 to 8

Table 3.1: [^{18}F]FDG PET intensity contrast ratios used for phantom-based simulations.

Hyperintense spherical lesions of random size (between 5 to 15 mm in diameter) and varying position and sharpness are also introduced into each phantom. Scaling and zero-padding of the ground truths are also performed to match the dimensions and resolution of reconstructions from the Siemens Biograph mMR scanner: $2.09\times2.09\times2.03\,\mathrm{mm}^3$ voxel size, and image dimensions $344\times344\times127$. Attenuation maps are generated with factors of 0.13 and $0.0975\,\mathrm{cm}^{-1}$ for bone and tissue, respectively, and added to scanner manufacturer-provided hardware maps. Finally, some randomised structure is introduced into both PET and MR segmentations in order to produce a more realistic non piece-wise constant ground truth phantom $\boldsymbol{\tau}$. Randomisation also ensures that a simple mapping from MR to ground truth PET images is not possible. The randomised structure is given by Equation (3.1), below. As part of this work, code to download and produce the full ground truth phantom dataset is made freely available at [150].

$$\boldsymbol{\tau} = \boldsymbol{\phi} \circ (\mathbf{1} + \gamma[2G_\sigma(\boldsymbol{\rho}) - \mathbf{1}]) \tag{3.1}$$

where $\boldsymbol{\tau}$ is used as a realistic ground truth phantom for the simulations,

$\boldsymbol{\phi}$ is a *BrainWeb*-based segmented phantom,

γ is an intensity parameter empirically chosen to be 1.5 for PET and 1 for MR segmentations,

G_σ is a circulant matrix representing 3D Gaussian smoothing of $\sigma = 1$ voxel, and

$\boldsymbol{\rho}$ is of the same size as $\boldsymbol{\phi}$ with random uniform distributed elements $\in [0, 1)$.

Each phantom is forward projected into sinogram space as per Equation (1.5) (in preliminary work, using *APIRL* [151], but later in this chapter with the more widely-used *NiftyPET* [152]), with a Gaussian matrix operator $\boldsymbol{H}$ of 4.5 mm FWHM simulating resolution degradation effects. The projection is then used as a mean for a Poisson noise model (Equation (1.6)). Simulations correspond to the Siemens Biograph mMR scanner (specifically, 837 span 11 sinograms $\hat{\boldsymbol{m}}$), accounting for photon attenuation and normalisation (including geometry, crystal efficiencies, and dead time effects as described in [153] and [152]). Three different count levels are chosen for each phantom: 3 M (very low), 30 M (low), and 300 M (full). The maximum count level is chosen to be comparable to that of real data (for a scan of 20 min with 370 MBq injected activity). These simulated counts include 26 % randoms and 28 % scatter. Ten Poisson noise realisations are generated for each count level, followed by MLEM reconstruction. (Note that preliminary work at the start of this chapter uses *APIRL* projectors, three noise realisations at 4.3 M, 43 M and 301 M counts instead, as well as 500 M in one experiment.) Reconstructions are done with resolution modelling (300 iterations) and without (100 iterations). The factor of roughly 3 times the number of iterations is proposed to compensate for the slower convergence rates in resolution modelling [154], [155]. Note that since both projectors use the Siddon algorithm [156], [157] – infinitesimally narrow LoRs – a Gaussian matrix operator $\boldsymbol{H}$ of 2.5 mm FWHM is actually used in the "no resolution modelling" case to emulate more realistic wider LoRs.

3.2.1.2 Patient data

Clinical PET-MR head scan data are obtained from 23 patients on the Siemens Biograph mMR.[1] While the acquired count levels vary between 400 M to 500 M across the scans (averaging 430 M counts per acquisition), the listmode data is randomly sampled with replacement (using the bootstrap method from [36]) to produce ten realisations at each of the three count levels used in the simulations above to ensure consistent count levels and similar distributions. Randoms are estimated through variance reduction of delayed coincidences [158], while scatters are updated at each iteration using a fully 3D voxel-driven scatter model (VSM) based on a single-scatter model for the *NiftyPET* results [152].

[1] Data courtesy Colm McGinnity & Alexander Hammers, "Evaluation of Brain PET/MR versus PET-CT," REC 15/NE/0203, IRAS 178069. All data were acquired from the Siemens Biograph mMR scanner at King's College London & Guy's and St Thomas' PET Centre. Ten scans are included in this thesis: [^{18}F]FDG-PET with corresponding 3D T1 MPRAGE. The study was approved by the institutional review boards and the research ethics committee, and written informed consent obtained from all study participants. The MPRAGE [7] data were acquired using a 5-channel head and neck coil with repetition time (TR) 1700 ms, echo time (TE) 2.63 ms, inversion time (TI) 900 ms, number of averages (NEX) 1, flip angle 9°, pixel bandwidth 199 Hz, reconstruction matrix size 224×256×176 and voxel dimensions 1.05×1.05×1.1 mm^3. Simultaneously acquired dual-point Dixon MRI was also used for PET attenuation correction. The Dixon data were acquired using the SPGR with T1 3.6 ms, TE 2.46 ms, NEX 1, flip angle 10°, pixel bandwidth 946 Hz, reconstruction matrix size 192×126×128, and voxel dimensions 2.06×2.06×3.12 mm^3.

The reconstructions are performed with and without resolution modelling as described in the case of the simulations (above). In lieu of a ground truth, the original raw listmode data (without bootstrap sampling) is also reconstructed for each patient for use as a high quality reference. MR images (MPRAGE T1 reconstructions [7]) are obtained directly from the scanner and registered to the full count PET reconstructions using SPM12 as described in Section 1.4.4.

3.2.2 Evaluation metrics

Multiple noise realisations at the same count level allows for the calculation of standard deviation σ across realisations (averaged across voxels). Bias b can also be calculated against the ground truth (if known, i.e. for simulations) or full count reconstructions (for patient scans). NRMSE ϵ, bias and standard deviation are all normalised as in [159], such that the resultant metrics:

- can be quoted as percentages;
- remain consistent with with $\epsilon^2 = \sigma^2 + b^2$, and
- avoid element-wise division (thereby avoiding inaccuracies from low intensity values near machine floating point precision).

The formulae are as below:

$$b(\boldsymbol{\theta}, \boldsymbol{T}) = \frac{100\%}{\sqrt{\sum_j T_j^2}} \sqrt{\sum_j \left(T_j - \mathop{\mathrm{E}}_r \{\theta_{r,j}\}\right)^2}, \tag{3.2}$$

$$\sigma(\boldsymbol{\theta}, \boldsymbol{T}) = \frac{100\%}{\sqrt{\sum_j T_j^2}} \sqrt{\sum_j \mathop{\mathrm{Var}}_r \{\theta_{r,j}\}}, \text{ and} \tag{3.3}$$

$$\epsilon(\boldsymbol{\theta}, \boldsymbol{T}) = \frac{100\%}{\sqrt{\sum_j T_j^2}} \sqrt{\sum_j \mathop{\mathrm{E}}_r \left\{(T_j - \theta_{r,j})^2\right\}}, \tag{3.4}$$

where $\theta_{r,j}$ is the j^{th} voxel of the r^{th} reconstruction (from the r^{th} noise realisation),

$\mathop{\mathrm{E}}_r \{\cdot\}$ is the mean operator across r,

$\mathop{\mathrm{Var}}_r \{\cdot\}$ is the variance operator across r,

T_j is the j^{th} target (or, if available, ground truth) voxel,

b is normalised bias,

σ is normalised standard deviation, and

ϵ is normalised root mean squared error (NRMSE).

Note that PSNR is another commonly used image quality metric. However, for very noisy images, PSNR will treat noise spikes as if it they were a favourable signal (see Figure 3.1). PSNR is thus a highly inappropriate metric for low dose PET quality analysis.

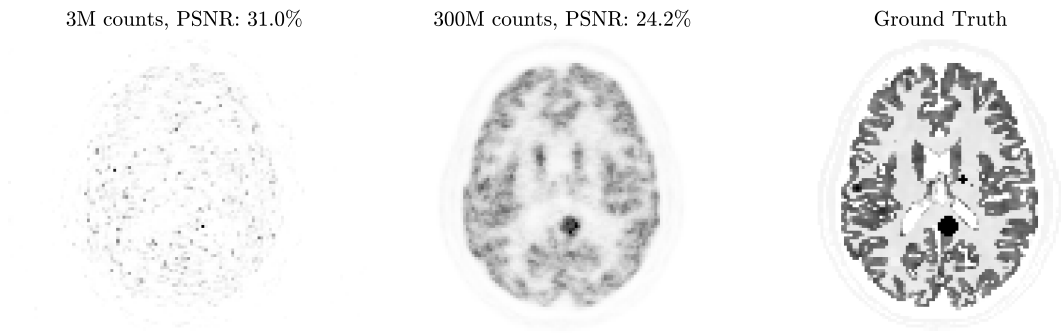


Figure 3.1: Peak signal-to-noise ratio (PSNR) can be higher for very noisy images. The ground truth is simulated including lesions as per Section 3.2.1.1 and a central slice is shown (right). Simulated very low (left) and low (middle) reconstructions show higher (better) PSNR for the lower quality image due to noise spikes.

3.2.3 Reference methods

PS and NLM are used as reference post-processing methods. Both have a single optimisation hyperparameter, which is set so as to minimise the NRMSE between the input $\boldsymbol{\theta}$ (low or very low count PET and corresponding T1-weighted MR) and the target $\boldsymbol{T}$ (full count or ground truth PET). The respective optimisation problems are given by:

$$\hat{\sigma} = \underset{\sigma}{\operatorname{argmin}}\, \epsilon\left(G_\sigma(\boldsymbol{\theta}), \boldsymbol{T}\right), \tag{3.5}$$

$$\hat{\Omega} = \underset{\Omega}{\operatorname{argmin}}\, \epsilon\left(\mathrm{NLM}_\Omega\left(\boldsymbol{\theta}, \boldsymbol{\theta}^{(\mathrm{T1})}\right), \boldsymbol{T}\right). \tag{3.6}$$

where $\hat{\sigma}$ is the optimal PS hyperparameter,

$\hat{\Omega}$ is the optimal MR-guided NLM filtering hyperparameter, and

NLM is as defined in Equation (1.17).

In the interest of fairness, these reference methods are optimised on the training data (rather than the entire available dataset) to allow for direct performance comparison to machine learning methods.

As a further comparison in later sections within this chapter, a U-net is modified to have some of the advantages of the proposed μ-net, and is discussed in more detail later in Section 3.3.1.3.

3.2.4 Architecture

In addition to low count PET and corresponding MR inputs, the network proposed here is designed to accept additional input volumes (in the form of channels) such as those from other existing post-processing methods. Initially, a single-layer, single-kernel network is trained (effectively an optimal Gaussian-like smoothing filter), and more layers are gradually added until the network starts to overfit. The final proposed network has three convolutional layers. Since the network is fairly small, it can accept more input channels without exhausting GPGPU memory. In particular, NLM-filtered volumes are also provided as inputs to the network. The PS volumes, however, are not provided as additional inputs as smoothing is trivially achievable by a CNN. Any current or future proposed method can also be provided as an input channel, thus theoretically guaranteeing that the network should be able to at least match – if not outperform – other competitive methods.

As outlined in Section 2.3, a network's receptive field can be increased by either increasing individual kernel sizes or increasing network depth (number of layers). While network depth scales linearly with receptive field width, kernel size scales quadratically. However, both network depth and kernel size are linearly related to total number of optimisation parameters. Therefore, in order to reduce the complexity of optimisation and reduce the network's memory requirements, increased depth may be more favourable than kernel size. However, for the small network used here, computational memory is not a constraint. For this reason, increasing kernel size instead of network depth may be a more appropriate strategy to increase receptive field. The additional parameters in larger kernels should allow for more more complex, powerful processing.

Figure 3.2 below shows the feed forward network containing l convolutional layers and activation functions (following the conventions set out in Figure 2.5). The network converts a low quality PET (and corresponding MR, i.e. 2-channel) input to a high quality PET output volume.

The number of Conv layers l, as well as the number of kernels (i.e. output channels or layer width) n and kernel spatial width s are all important hyperparameters. Additionally, the choice of activation functions A and loss function is also investigated here. The advantage of (re)investigating these choices means that it should be possible to design a bespoke network from first principles to tackle this particular task. The Adam optimiser (discussed in Section 2.2.3) is used in all cases owing to its demonstrated robustness in a wide variety of scenarios.

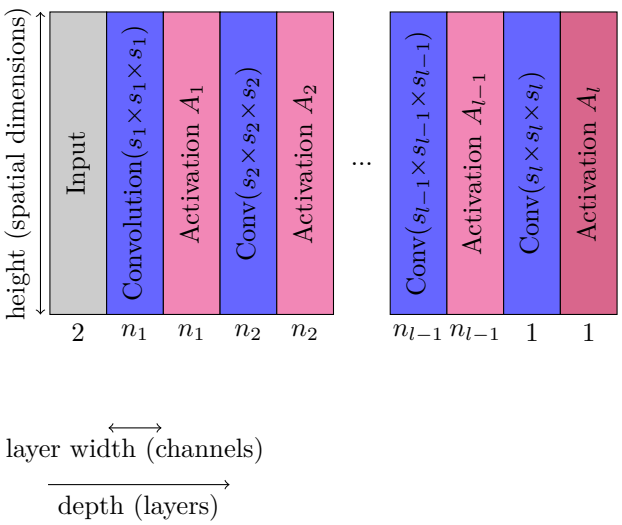


Figure 3.2: Visual representation of a generic MR-guided PET post-processing feed forward network architecture. Each block represents the output of a layer, with the number below each block signifying the number of output channels. The layer operations themselves are hidden as per the convention set out in Figure 2.5. There are l convolutional layers in total, with the number and width of kernels in a layer given by n_l and s_l, respectively. Overall, the network depicted converts two input channels (PET and MR) into a higher quality single-channel PET output.

The investigations starts with $l = 1$ layer (a $3 \times 3 \times 3$ convolution) followed by a sigmoidal activation. Layers and activation functions are gradually appended to this architecture until there is no longer any improvement in the trained network's minimum validation loss. No down nor up-sampling is performed throughout. Since the overall training dataset and network size is (initially) small, there are low demands on computational memory. This means that mini-batches are not required – meaning that gradient descent does not have to be stochastic. This can potentially solve the CI problem with a very different approach – having all weights affected by all inputs, thus leaving the optimisation algorithm to find a robust, generalisable solution.

There are an extremely large number of possible hyperparameter choices. In order to simplify the problem, it is first observed that for a single-layer (and single-kernel) network, a kernel spatial width of $s_1 = 5$ is optimal (see Figure 5.12 later). This is unsurprising since such a network essentially applies a single learned post-smoothing kernel which is likely to approximate a Gaussian. For the voxel dimensions (2.09×2.09 mm in the transverse plane) for the Siemens Biograph mMR, a 4.5 mm FWHM Gaussian resolution-modelling kernel would have a standard deviation given by:

$$\sigma = \frac{4.5 \text{ mm}}{2.09 \text{ mm voxel}^{-1} \times \sqrt{8 \ln 2}} = 0.914 \text{ voxels}. \tag{3.7}$$

The majority (99.6 %) of the weights in such a kernel lie within a spatial width of 5 voxels. For this reason, for all network architectures considered in this chapter, the kernel spatial width for the first layer is set to be $s_1 = 5$. (Note that a more thorough investigation of a post-processing network capable of removing resolution-modelling artefacts can be found later in Chapter 4.) Subsequent Conv layers have a spatial width 3 in keeping with standard practice in the current literature. Furthermore, the final Conv layer uses a single kernel of spatial width $s_l = 1$ in order to perform a simple weighted average over its input channels.

As a further starting point, sigmoidal activation functions are initially used after each Conv layer. Since sigmoids have a range $\in [0, 1]$, care must be taken to ensure that the network's targets also fall within this range. If the last layer of the network is a sigmoid, all target images must be pre-scaled such that their maximum value is 1. Since the targets are always MLEM reconstructions or ground truth PET volumes – and therefore non-negative – there is no need to further scale to account for the minimum allowable value.

Each of the following choices of hyperparameters are considered here for investigation/optimisation:

- number of layers, l
- number of kernels per layer, n_l
- for a training dataset consisting of one or more patients, the number of reconstructions (noise realisations) of each patient used for training, R
- Adam optimiser learning rate
- loss function (NRMSE, MSE, ℓ_1, and adversarial (discriminator) loss)

Further analysis of the impact of using more patient datasets for training can be found later in Chapter 5.

3.3 Results

3.3.1 Simulations

3.3.1.1 Preliminary study: ground truth simulations

A starting experiment (based on work presented in [160]) is the case of recovering the ground truth from full count PET-MR. Specifically, the network training data consists of a single phantom simulation: a 500 M count resolution-modelled PET reconstruction and corresponding MR volume. While this may seem to be a very small dataset to to use for training, the network itself is also extremely small (few trainable parameters), with $n = \{9, 18, 1\}$ and $s = \{5, 3, 1\}$. Note that the number of optimisation parameters in a layer j is given by:

$$\#\text{params}_j = (n_{j-1} s_j^d + 1) n_j, \tag{3.8}$$

where n_j is the number of kernels (i.e. output channels) in layer j;

s_j is the kernel width in layer j, and

d is the number of spatial dimensions (in this case, 3).

This network therefore has only 6.67 k weights and biases in total (note also the similarity to the first CNN-based super-resolution proposal [81], where a 3-layer network has kernels $n = \{64, 32, 3\}$ of width $s = \{9, 1, 5\}$). By comparison, there are at least an order of magnitude more non-zero voxels in the training target (ground truth) volume, making memorisation (overfitting) unlikely. Moreover, compressing the target image (into a `*.zip` file) still results in a size greater than a file containing the uncompressed weights and biases. The comparatively low number of weights and biases means that even a compressed target cannot be memorised. Such a network – deliberately small by design – is therefore called a micro-net (or μ-net) in the rest of this work.

The network architecture is shown in Figure 3.3. Note that targets must be pre-scaled to be in the range $[0, 1]$ to remain within the domain of the final sigmoid activation function.

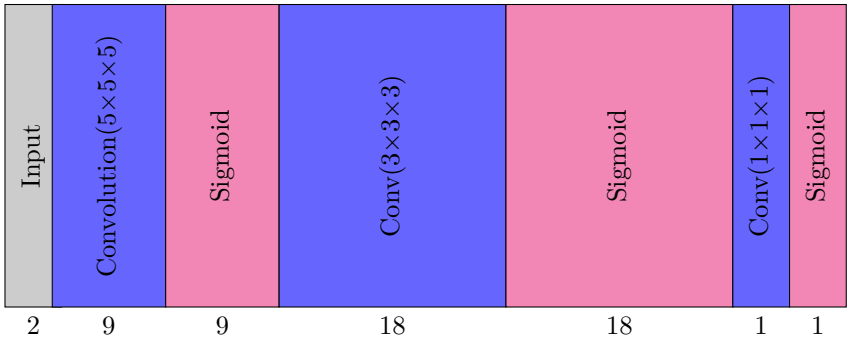


Figure 3.3: Visual representation of MR-guided PET post-processing architecture which outputs a ground truth volume.

At the start of training, the weights and biases must be assigned starting values. *He* normal initialisation [161] is used as it is found to reduce loss by a factor of 3 compared to *LeCun* uniform initialisation [162]. The former method entails initialising weights $\boldsymbol{w}_l$ by random normal sampling with standard deviation $\sqrt{2/n_{l-1}}$, while biases are set to zero. This helps prevent saturation of activation functions with very large positive or negative values.

The Adam optimiser is used with a learning rate of 10^{-3} to minimise an NRMSE loss function. A central slice from a test dataset (a different phantom) is shown in Figure 3.4. The network demonstrates impressive recovery abilities, including for a small PET-unique lesion (0). The line profiles also demonstrate suppression of Gibbs ringing (lesion 3) and partial volume effect (PVE) correction. This is possible due to the high quality MR input volume.

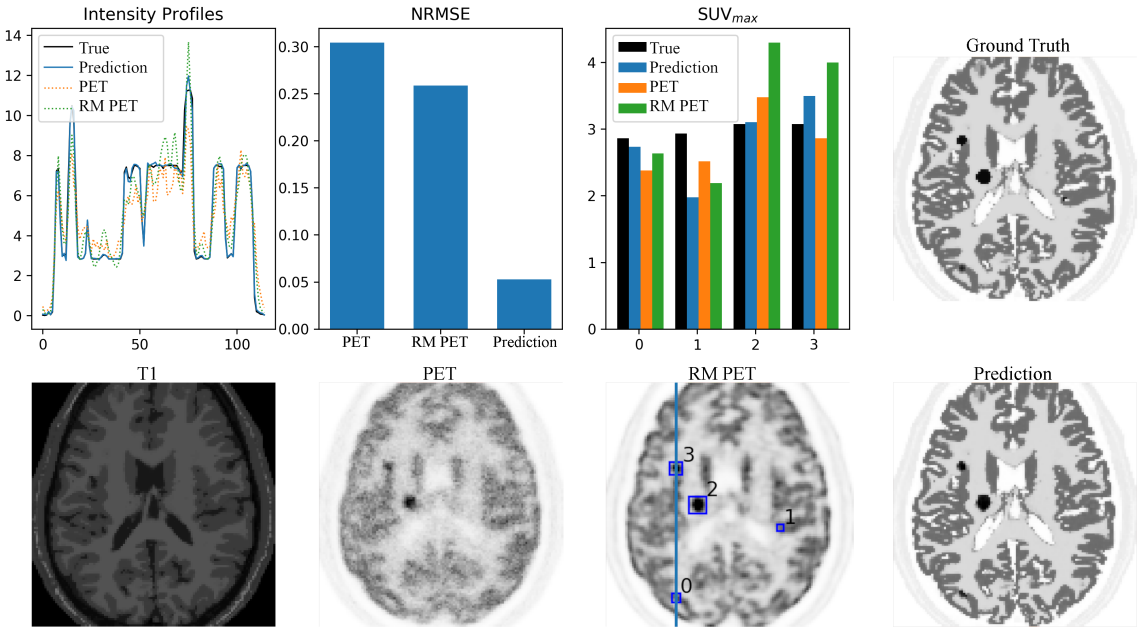


Figure 3.4: Central slice of a phantom from the validation dataset. The network aims to predict the ground truth from a 500M count PET reconstruction and corresponding T1 MR (bottom row, first two images). The ground truth and resolution-modelled (RM) reconstruction are shown for reference. SUV_{max} values are calculated for the lesions numbered 0-3.

Unfortunately, such a study would be an impossible task for clinical patient data, where the ground truth is unknown, and the MR is of lower quality (and possibly misaligned). The network above – trained on very artificial ground truth simulation data – cannot be simply applied to real patient data.

3.3.1.2 Reference methods

Both PS and NLM guided filtering (Equation (1.17)) have a hyperparameter. In the case of PS, it should be noted that smoothing using a kernel at least as large as the RM PSF has long been proposed as a way of obviating ringing artefacts [64], [65], [163]. However, to provide a fair comparison to CNN-based methods, both PS and NLM hyperparameters can also be optimised to minimise the NRMSE against a target volume.

The optimal hyperparameters (for PS and NLM) are shown in Figure 3.5 and Figure 3.6 below, where the known ground truth $\boldsymbol{\tau}$ is used as a target. In both cases, the PET reconstruction is considered as an input both with and without RM. As expected, the optimal PS FWHM is larger for the lower count (higher noise) level. However, the optimal NLM hyperparameter is only slightly larger. This too is expected since the same MR guidance volume is used.

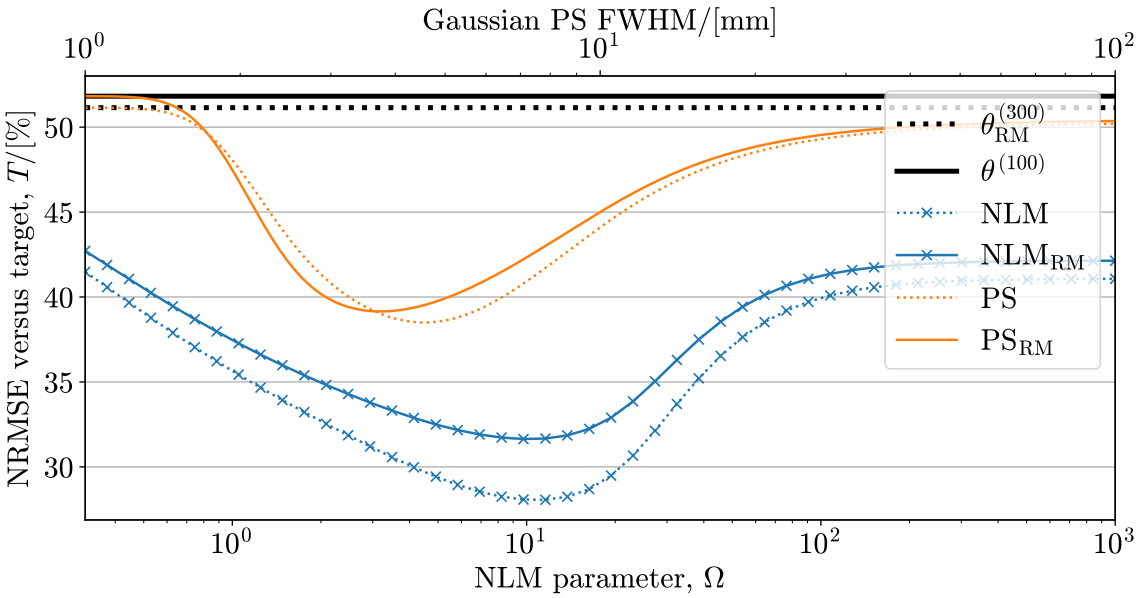


Figure 3.5: Optimal hyperparameters for PS (orange, top axis) and NLM (blue, bottom axis) for 30 M count inputs minimising NRMSE against the known ground truth $\boldsymbol{T} = \boldsymbol{\tau}$. The optimal FWHM (i.e. Equation (3.5)) and Ω (i.e. Equation (3.6)) are 3.7 mm (4.6 mm) and 9.1 (11.5) for the input PET (RM) volumes. For both PS and NLM, very small parameter values have little effect. Increasing parameter values gradually decreases the NRMSE until an optimum is reached, at which point NRMSE increases again and eventually plateaus due to over-smoothing/filtering.

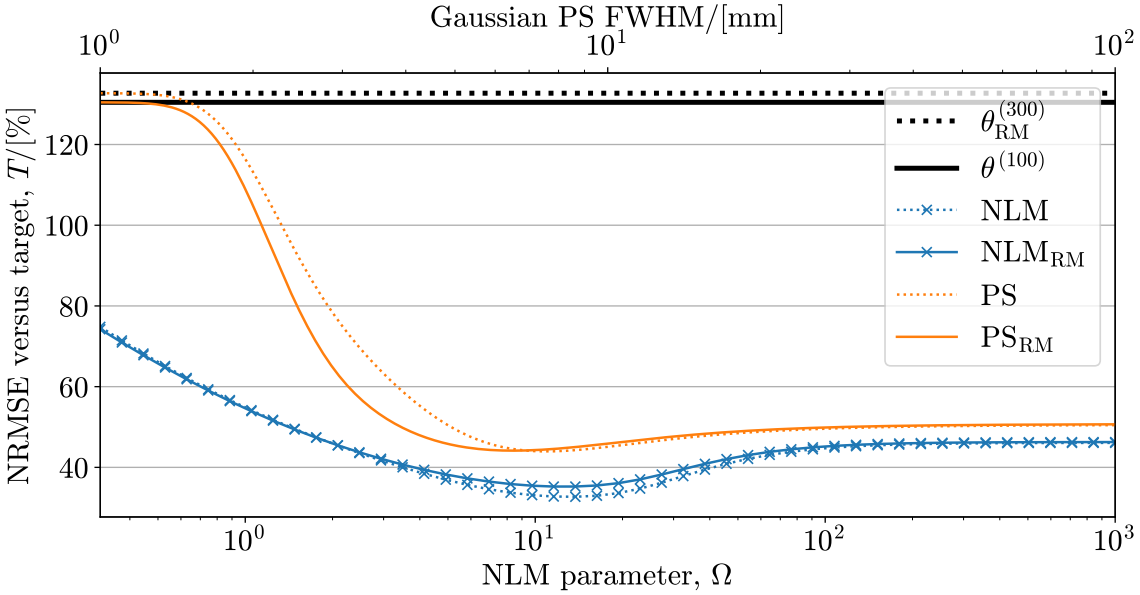


Figure 3.6: Optimal hyperparameters (similar to Figure 3.5) for 3 M count inputs. The optimal FWHM and Ω are 6.5 mm (7.2 mm) and 18.3 (18.3) for the input PET (RM) volumes. Unsurprisingly, more smoothing/filtering is required to reach the optimal NRMSEs for these noisier inputs, and the optimal NRMSEs are also larger.

If the full (300 M) count PET is used as a target instead, the optimisation at the two different input count levels is given in Figure 3.7 and Figure 3.8, respectively. This is a more practically realistic optimisation since ground truths are not known in clinical practice. All of the optimal hyperparameters are found to be larger as the methods must work harder to deal with the noise in the target.

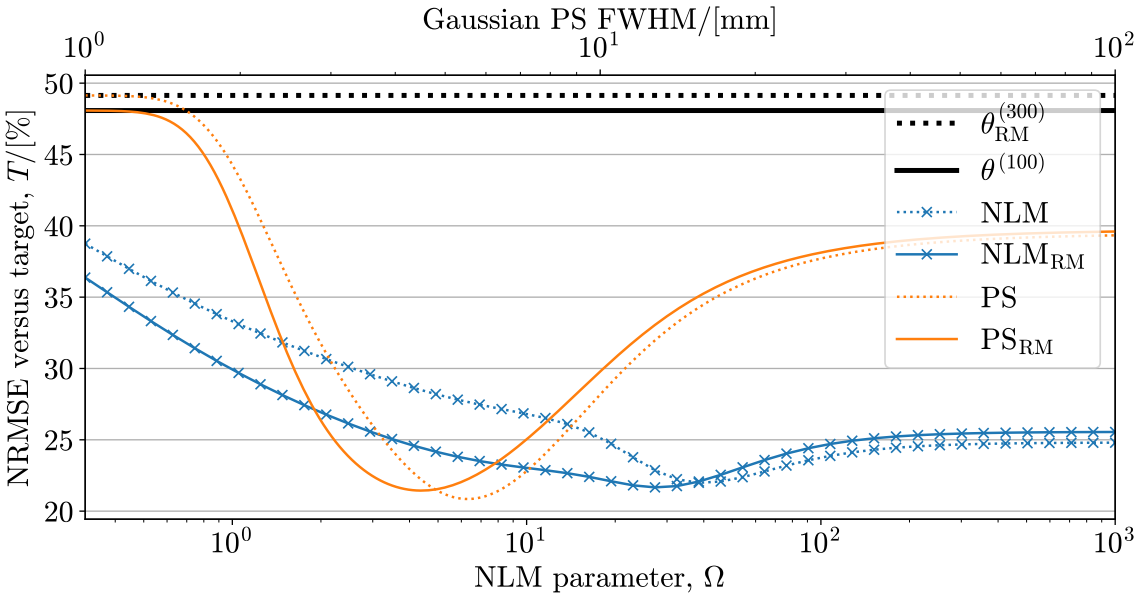


Figure 3.7: Optimal hyperparameters for PS and NLM for 30 M count inputs minimising NRMSE against a full (300 M) count target (since ground truth is unknown in clinical practice). The optimal FWHM and Ω are 4.6 mm (5.7 mm) and 29.2 (36.8) for the input PET (RM) volumes. Compared to when the ground truth is used as a target (Figure 3.5), the optimal NRMSEs are lower as expected (low count reconstructions more closely match the full count reconstructions than they do the ground truth). However, the optimal parameter values are larger, implying over-smoothing/filtering.

It should be noted that for the NLM method, a $5 \times 5 \times 5$ neighbourhood is used for 30 M count inputs, while a larger neighbourhood of $7 \times 7 \times 7$ is required for the lower quality 3 M count PET volumes to ensure outputs do not contain block-like artefacts. The NLM method is written to work on 3D volumes and run on a GPGPU (see Appendix A.1), but still takes a prohibitively long time to compute for much larger neighbourhoods. Furthermore, logically only relatively small neighbourhoods should affect the value of a voxel.

Corresponding endpoint images for the above methods are shown in the study below.

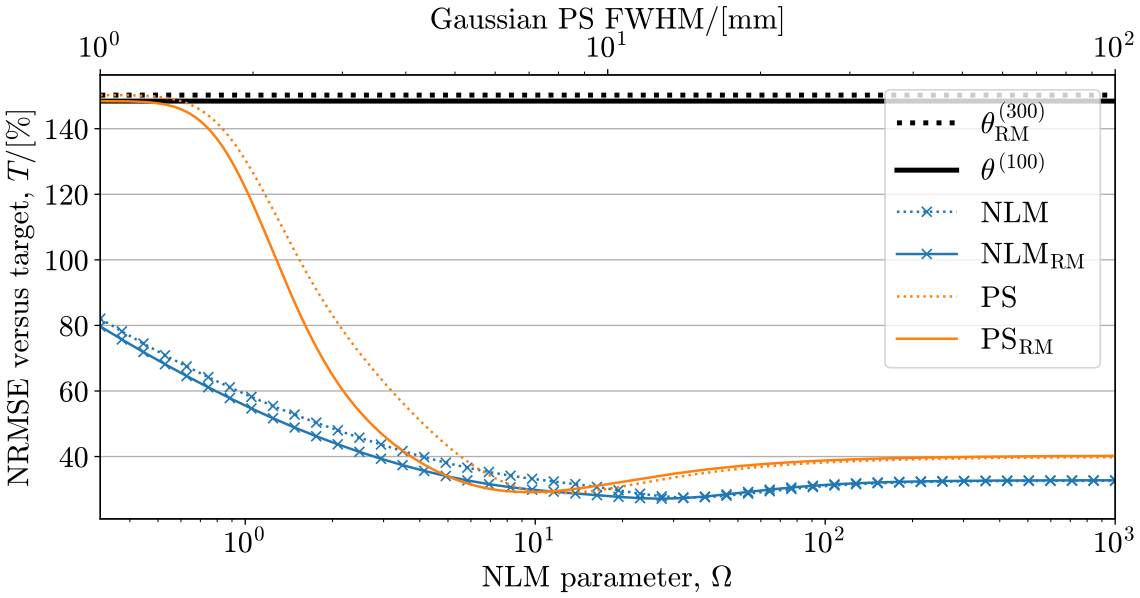


Figure 3.8: Optimal hyperparameters (similar to Figure 3.7) for 3 M count inputs. The optimal FWHM and Ω are 7.2 mm (8.1 mm) and 36.8 (58.6) for the input PET (RM) volumes.

3.3.1.3 Study: micro-networks

After the preliminary μ-net study (predicting ground truth phantoms Section 3.3.1.1), a more realistic follow-up task is to upgrade low count reconstructions. The preliminary network considered above can be easily modified to accept additional 3D input channels without exhausting computational memory. PET reconstructions both with and without RM can be provided in addition to the MR. Furthermore, element-wise products between the MR and the PET reconstructions can also be provided as additional input channels, with the hypothesis that such multiplication would help modulate the PET data with the higher resolution anatomical information provided by the MR in such a way that would be otherwise difficult to achieve with a small network. While it may be common practice to normalise network inputs and target outputs (i.e. enforce unit standard deviation and zero mean), in this case there is a logical non-negativity constraint on the PET data so full normalisation is not desirable. Instead, the PET and MR volumes are scaled to have unit standard deviation (and only in the case of the MR also zero mean) in order to have similar magnitudes and thus make this simple multiplicative modulation more meaningful. In clinical practice, the scale factor used on the target PET outputs during training may be saved for later application during testing (i.e. to un-normalise the network outputs for the purpose of quantitative PET tasks).

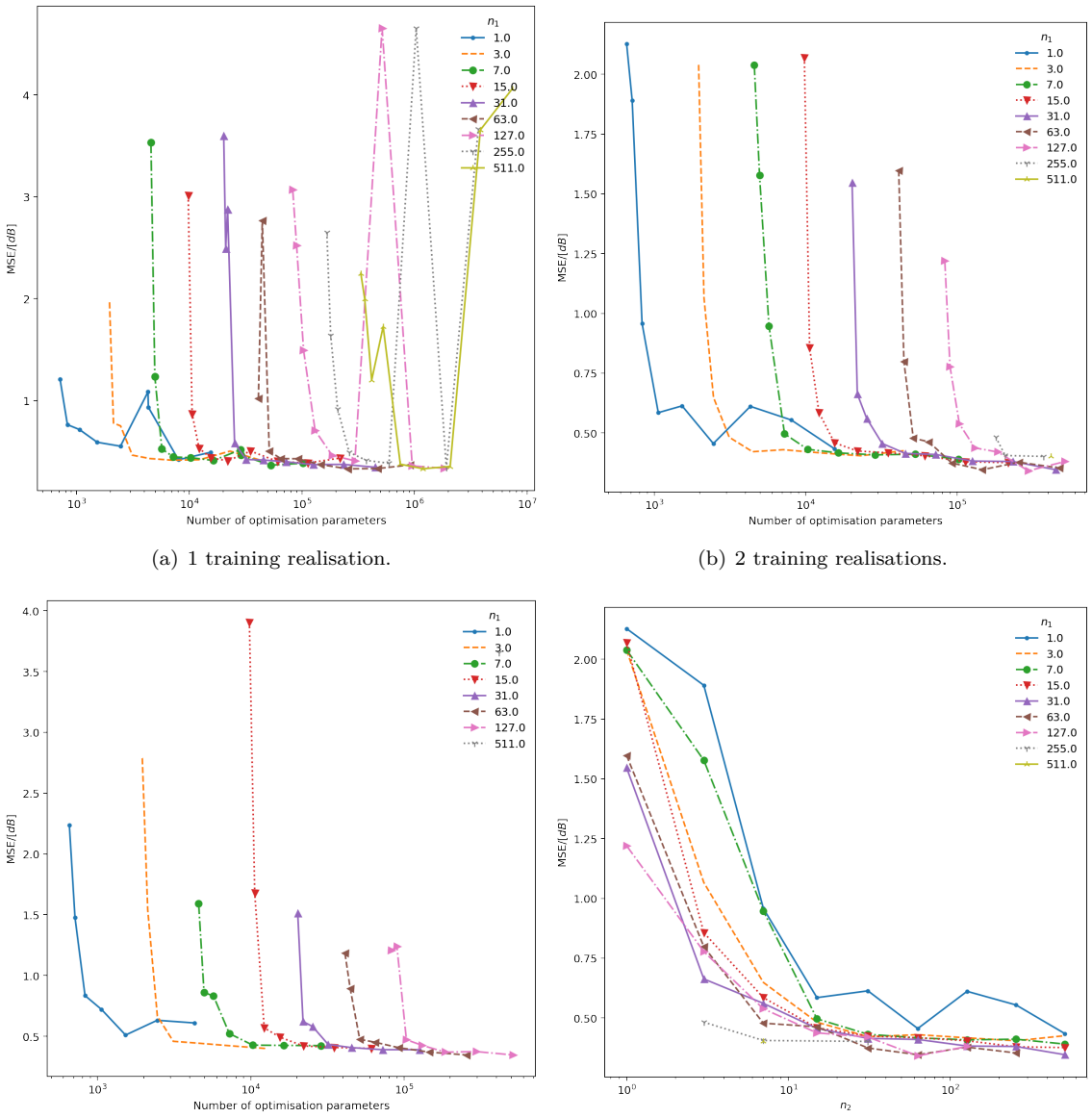
In order to fully utilise the additional input channels, the network requires more parameters. To emulate a real-world scenario where there are far fewer training samples than test data, this proposed network is also trained on just one phantom, validated on another, and tested on the remaining 18 in the simulation dataset. Various choices for the number of layers (from 1 to 6) are empirically investigated, and for this limited amount of training data, 3 layers are still found to be optimal [142]. A more thorough investigation of number of layers can be found later in this section.

Given a 3-layer network and a single training phantom, there still remain a number of other considerations. Firstly, the number of kernels per hidden layer (i.e. excluding input & output layers, which have a fixed number of channels) are systematically and independently varied (from 1 to 511 in powers of 2, less 1). However, using reconstructions of more than one noise realisation for each phantom may train the network to better recognise and remove noise. In order to investigate this, each considered network architecture is re-trained with $R = 1$, 2, or 3 noise realisations – shown below in Figures 3.9(a) to 3.9(c), respectively. The figures show the different choices for number of kernels in the first layer n_1 as different curves, labelled in the legend. By varying the number of kernels in the second layer n_2, these curves can be traced out, gradually reducing the total error. Rather than showing n_2 directly, the total number of weights and biases are given on the horizontal axis. This makes it easier to distinguish the curves. There appears to be a slight reduction in minimum MSE when $R = 2$, while there is no further significant improvement with $R = 3$. Given the similar performance, using fewer realisations is advantageous due to increased training speed. As a result, $R = 2$ seems to be the best choice. For the case of $R = 2$, an alternative plot showing n_2 explicitly on the horizontal axis is shown in Figure 3.9(d).

The curves are not monotonically decreasing, most likely due to the optimisation difficulty for very large networks.

The best choice is a 3-layer network with $n = \{63, 63, 1\}$ kernels, which is found to produce the lowest validation loss. The resultant architecture is shown in Figure 3.10.

Furthermore, as a quick test to ascertain a reasonable learning rate to use for the Adam optimiser, a small hyperparameter search space is chosen. For $R = 2$, the number of kernels in the hidden layers are set to be the same ($n_1 = n_2$) and varied from 1 up to 127. Each architecture is trained for 2 min at a range of learning rates (varying from 10^{-5} to 10^{0} in 50 logarithmic steps), as shown in Figure 3.11. Based on these results, a learning rate of 10^{-2} is used for the rest of this work (contrary to the widely used default in most Adam implementations of 10^{-3}). Note that since the training loss is comparatively flat around this learning rate for many architectures, it is likely to continue to work similarly well even after other small modifications are made to the network. As training progresses, Adam will also adapt, thereby decreasing the significance of the initial choice of learning rate.



(a) 1 training realisation.

(b) 2 training realisations.

(c) 3 training realisations.

(d) 2 training realisations, as per Figure 3.9(b) but with n_2 shown explicitly on the horizontal axis.

Figure 3.9: Final validation MSE for a 3-layer μ-net trained on one phantom. For a fixed number of kernels in the first layer n_1, the number of kernels in the second layer n_2 is varied and the network completely re-trained in order to trace out the various line series shown in all four panels. While only one ground truth target phantom was used, panels (a), (b) and (c) used reconstructions of $R = 1$, 2, and 3 different noise realisations as inputs, respectively. These panels also show the total number of parameters in the network on the x axis since this makes it easier to distinguish the lines from each other. Panel (d) however makes the independent variables explicit, showing the same data as in (b) but with the number of kernels in the second layer n_2 on the x axis instead. The results indicate that $R = 2$ realisations (panels (b) or (d)) and $n_1 = n_2 = 63$ kernels is best (lowest MSE value on the brown dashed line with left-pointing triangle markers).

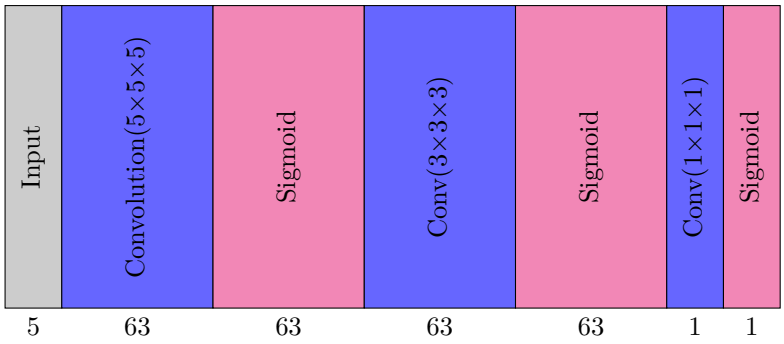


Figure 3.10: Visual representation of a post-processing architecture which maps low count PET, $\mathrm{PET}_{\mathrm{RM}}$, MR, PET $\circ$ MR, and $\mathrm{PET}_{\mathrm{RM}}\circ$ MR to a higher count PET reconstruction. The two element-wise product channels are supplied as a simple manually MR-modulated PET.

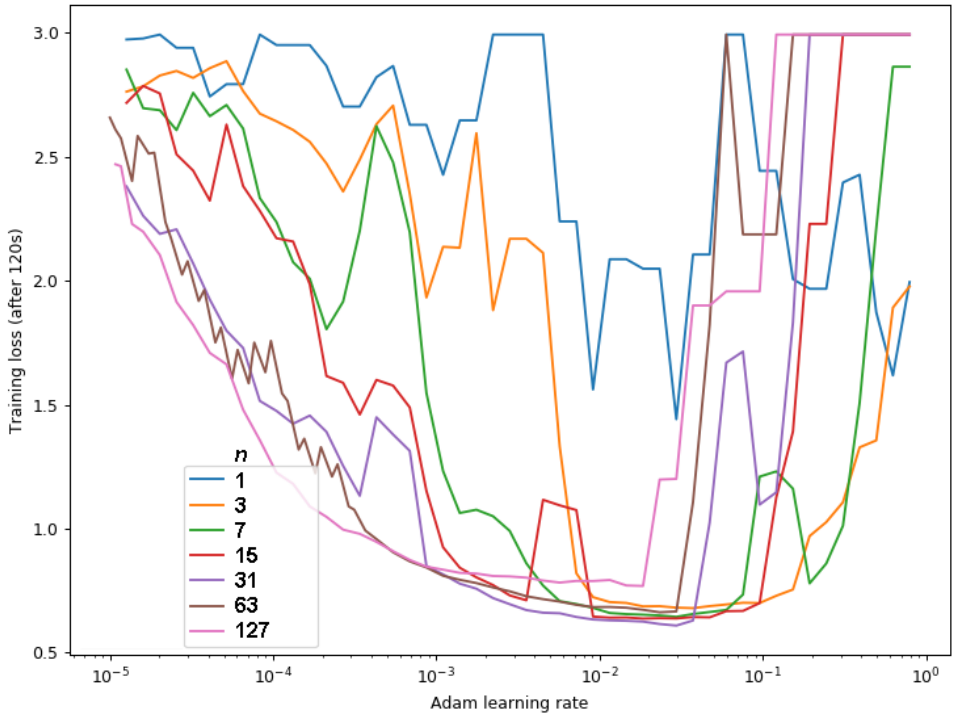


Figure 3.11: Basic experiments on the effect of Adam optimiser learning rate on training loss after 120 s for various choices of number of kernels n per layer (only for the same number of kernels in each layer, i.e. $n = n_1 = n_2$). As training progresses further, Adam should adapt (thereby decreasing the significance of the initial learning rate). It is nevertheless interesting to note that a learning rate of around 10^{-2} works well to quickly reduce loss within this limited amount of training time for all architectures.

Central slices from one phantom in the test dataset are shown in Figure 3.12. The metrics shown, however, are calculated across the entire test dataset (18 phantoms, 3 noise realisations each). The μ-net produces a images of a very comparable (and even slightly lower) standard deviation compared to the target full count PET image, and manages to recover small lesions and fine detail. Optimal guided filtering by comparison produces less natural-looking images with larger errors, while resolution modelling combined with optimal PS has a higher level of noise.

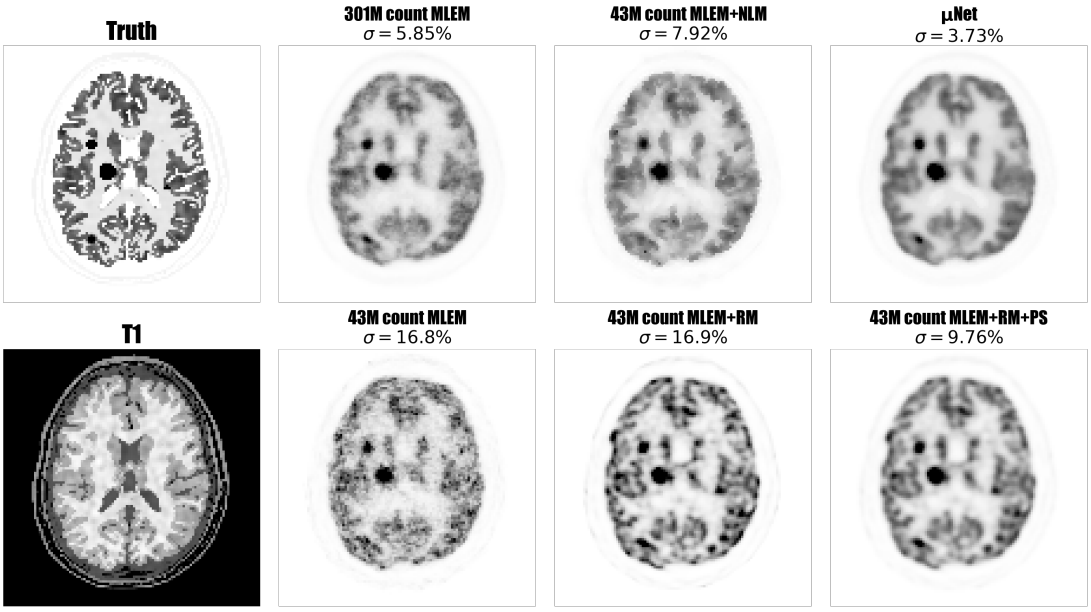
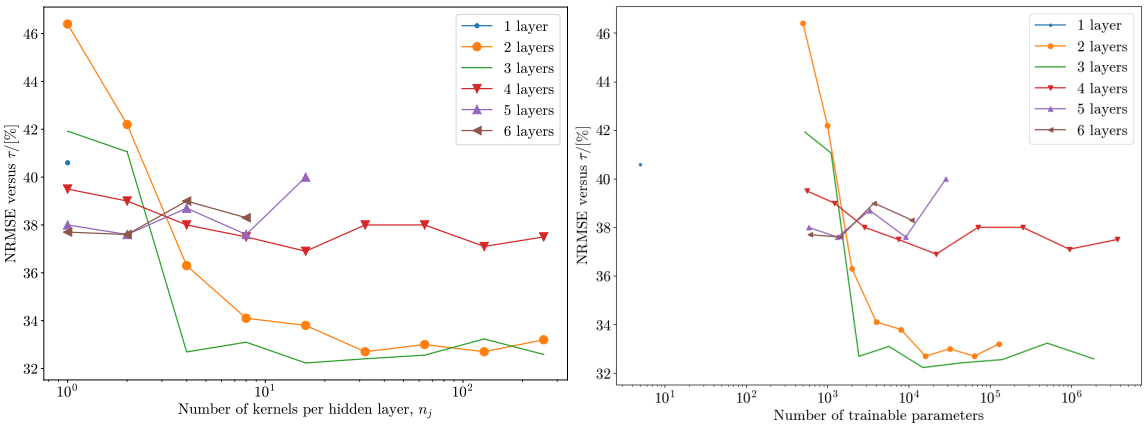


Figure 3.12: Central slice of a phantom from the **test** dataset. Interestingly, the μ-net prediction based on 43 M count data has a lower standard deviation (3.73 %) than the 301 M count target (5.85 %).

However, rather than providing element-wise products of PET and MR volumes as network inputs, it seems more sensible to use the more advanced NLM guided filter directly as a network input pre-processing step (optimal PS, meanwhile, is trivially achievable by the network and is therefore not inserted as an additional input channel). Since NLM is invariant to the intensity scale of the guidance volume (see Equation (1.17)), input volumes also no longer need to be scaled for modulation purposes. In order to remove the need to scale the output volumes to be within the range of a sigmoid, the final activation is also replaced with an exponential linear unit (ELU). This enforces a weaker non-negativity constraint (theoretically allowing values as low as -1, though in practice no significant negative values are observed below) without placing an upper bound on target values – which is more in keeping with the target PET reconstructions.

The training data still uses two realisation from one phantom, but the different input channels mean that another hyperparameter search is required to determine an optimal architecture. Since a new hyperparameter search is required anyway owing to the change of inputs, this is also an opportunity to use a different reconstruction framework. At this point, the reconstruction framework is switched from *APIRL* to *NiftyPET*, the number of noise realisations for evaluation of standard deviation are increased from 3 to 10, and the count levels are slightly modified as described in Section 3.2.

The number of kernels for all hidden layers n are set to be the same in order to reduce the hyperparameter search space. For each choice of depth (total number of layers, from 1 to 6), the number of kernels n is initially set to 1 for all hidden layers. The number of kernels per layer n is then increased from 1 up to 256 in powers of 2 to produce the curves in Figure 3.13 below (showing the NRMSE calculated across the test dataset). Due to memory constraints, it is only possible to reach up to $n = 16$ and 8 kernels per layer for $l = 5$ and 6 layers, respectively.



(a)

(b) Equivalent to Figure 3.13(a) but using an alternative horizontal axis.

Figure 3.13: Effect of varying number of layers (network depth) and number of kernels per hidden layer (width) on **test** NRMSE (for 3 M→300 M counts mapping, calculated versus truth $\boldsymbol{\tau}$). For each choice of depth, the number of kernels n is initially set to 1 for all hidden layers. The number of kernels per layer n is then increased from 1 up to 256 in powers of 2 to produce the curves above. Due to memory constraints, it is only possible to reach up to $n = 16$ and 8 kernels per layer for $l = 5$ and 6 layers, respectively.

The drastic increase in error from 3 to 4 layers is potentially due to increased optimisation difficulty rather than overfitting [164]. The final architecture uses $n_1 = n_2 = 32$, is depicted in Figure 3.14 below.

To investigate the importance of each input channel, networks are also re-trained on various combinations of inputs. Information is omitted by setting input elements to zero (the alternative – leaving out input channels – would alter the number of parameters in the first layer and therefore change the architecture, resulting in an unfair comparison). Finally, the effect of replacing NRMSE with an ℓ_1-norm loss function is also investigated as the latter has been proposed as a way to encourage less blurring [165].

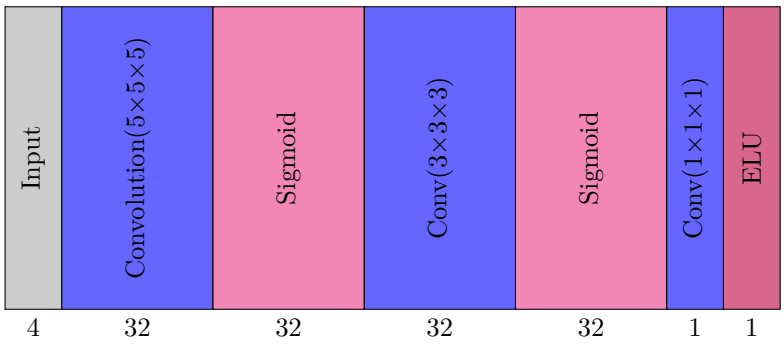


Figure 3.14: Visual representation of a post-processing architecture which maps low count PET, PET_{RM}, MR, and NLM (MR-guided filtering of PET_{RM}) to a higher count PET reconstruction.

As a further comparison method, a U-net is modified to have as many of the advantages of the proposed μ-net as possible. These advantages include accepting the same outputs and multi-channel inputs, as well as performing fully 3D convolutions. ReLU activation functions are replaced with ELUs to help eliminate vanishing gradients. Optimisation details (choice of optimiser, parameter initialisation, and NRMSE loss) are kept the same as for the μ-net. Specifically, the U-net comprises of an *encoder* and *decoder*, and a final residual layer, as shown in Figure 3.15. The encoder consists of 4 convolutional layers (with stride 2). The decoder repetitively performs trilinear upsampling (scale factor 2), concatenation with the corresponding encoder layer, and convolution (stride 1). The number of kernels per convolution layer are increased with U-net depth: $n = \{32, 64, 128, 256, 128, 64, 32, 1\}$. ELU activation functions are inserted for each multi-channel convolution output. The final residual layer adds the decoder's single-channel ($n_8 = 1$) output (element-wise) to the NLM input channel (as this is the "best" input in terms of NRMSE).

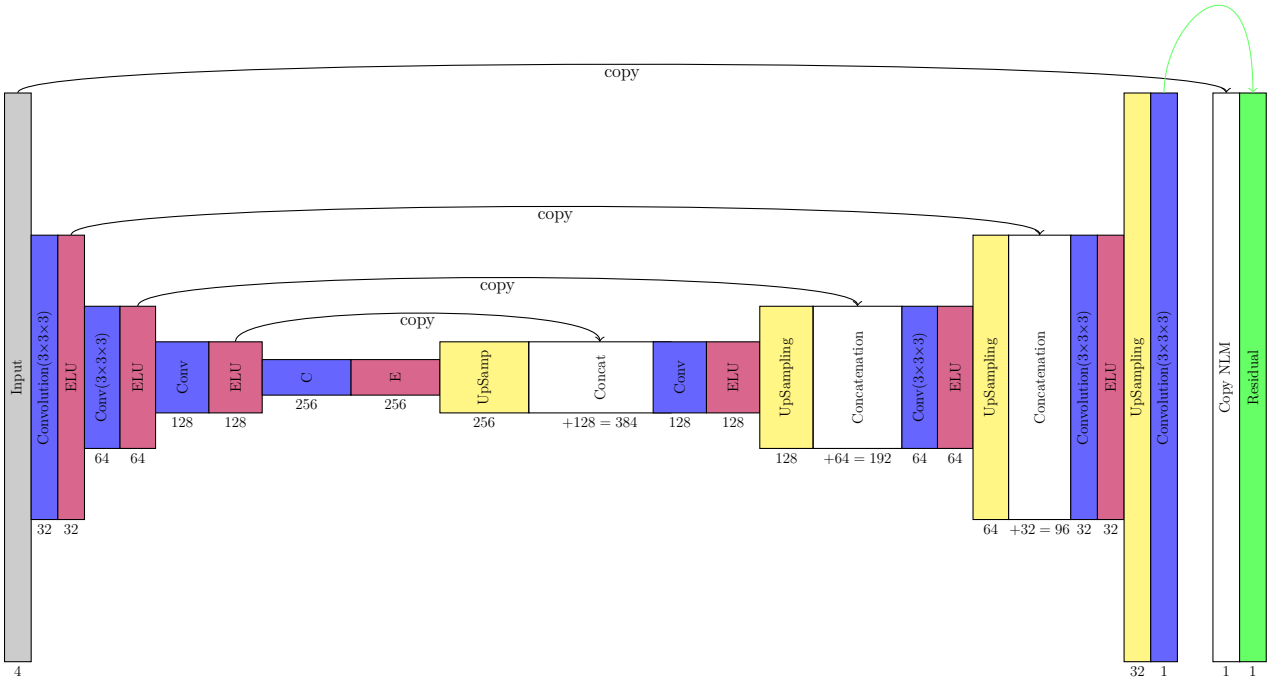


Figure 3.15: Visual representation of a post-processing U-net with the same task as Figure 3.14. Convolutions use kernel width 3 and – when downsampling – stride 2, while (trilinear) upsampling uses a scale factor of 2. The final residual layer performs element-wise addition between the last layer and the input NLM channel.

Figure 3.16 shows central slices from one test phantom for all considered inputs and post-processing methods.

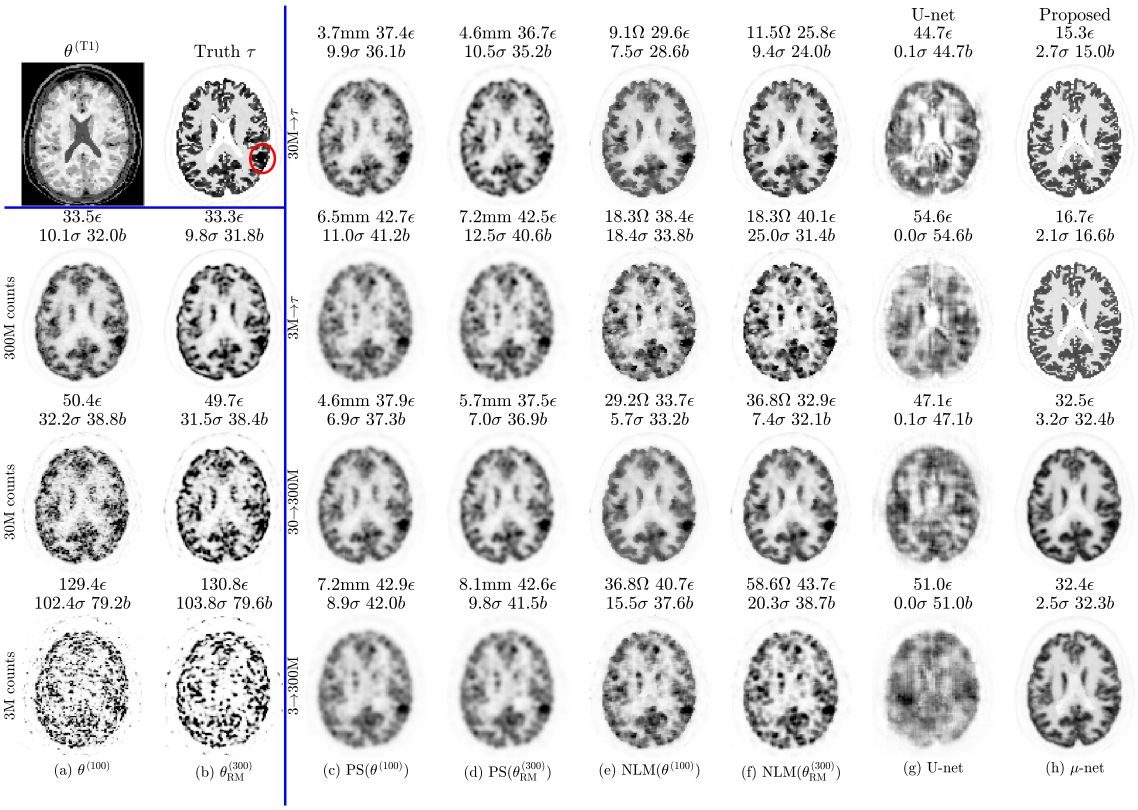


Figure 3.16: Simulation **test** data: cropped central slices from one set of MLEM reconstructions of subject 6 at different count levels without (a) and with (b) resolution modelling. For comparison (c)-(f) and proposed (h) methods, optimisation is performed to minimise NRMSE between the training input and target. This is given by the row titles, which are labelled according to "input → optimisation target." NRMSE ϵ and bias b metrics are calculated versus the known ground truth $\boldsymbol{\tau}$. Standard deviation σ is across 10 realisations. Optimal values are given in panel titles for smoothing FWHM (mm) and NLM hyperparameter (Ω) as obtained from Figures 3.5 to 3.8. All images use a common colourscale so are directly comparable to each other.

Profiles including the lesion in Figure 3.16 are shown in Figure 3.17. Note that the μ-net simultaneously suppresses noise, partial volume, and ringing effects to match the standard count reconstruction.

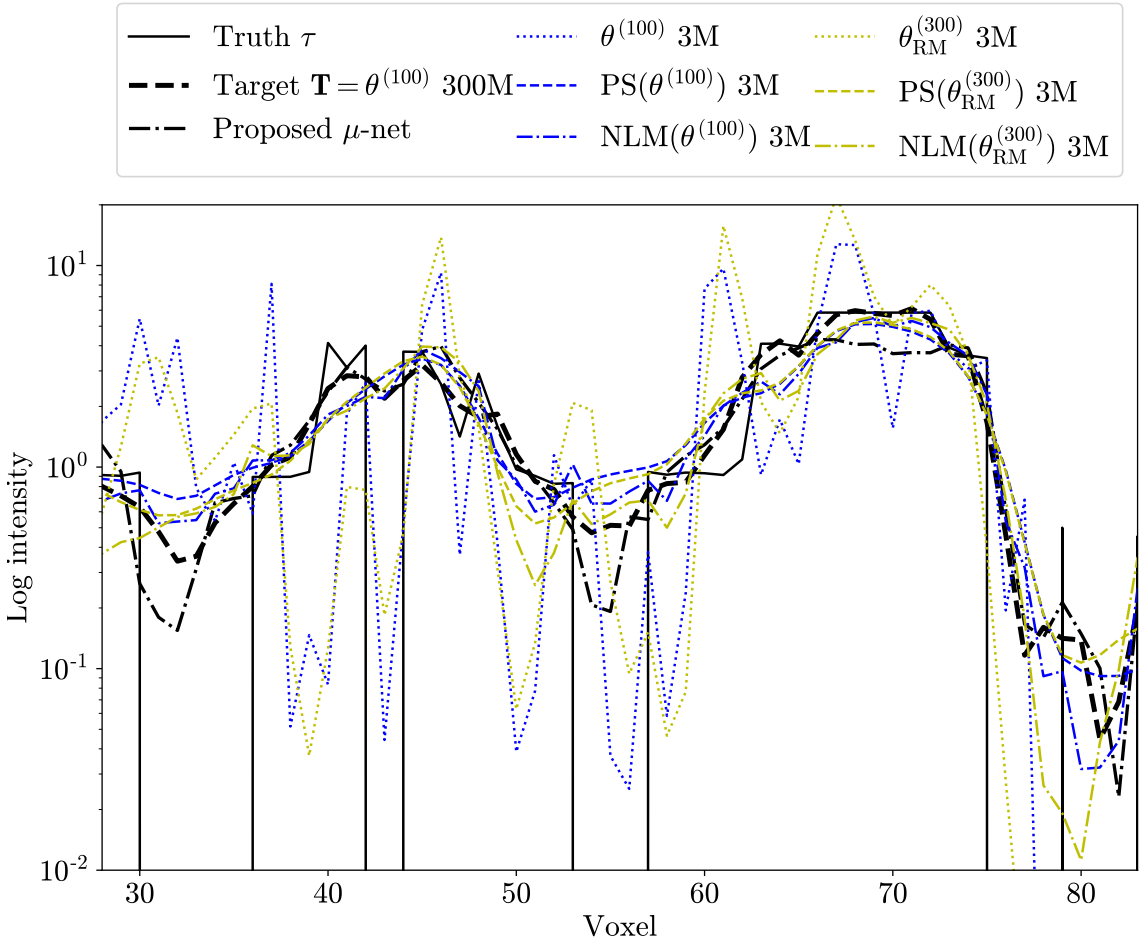
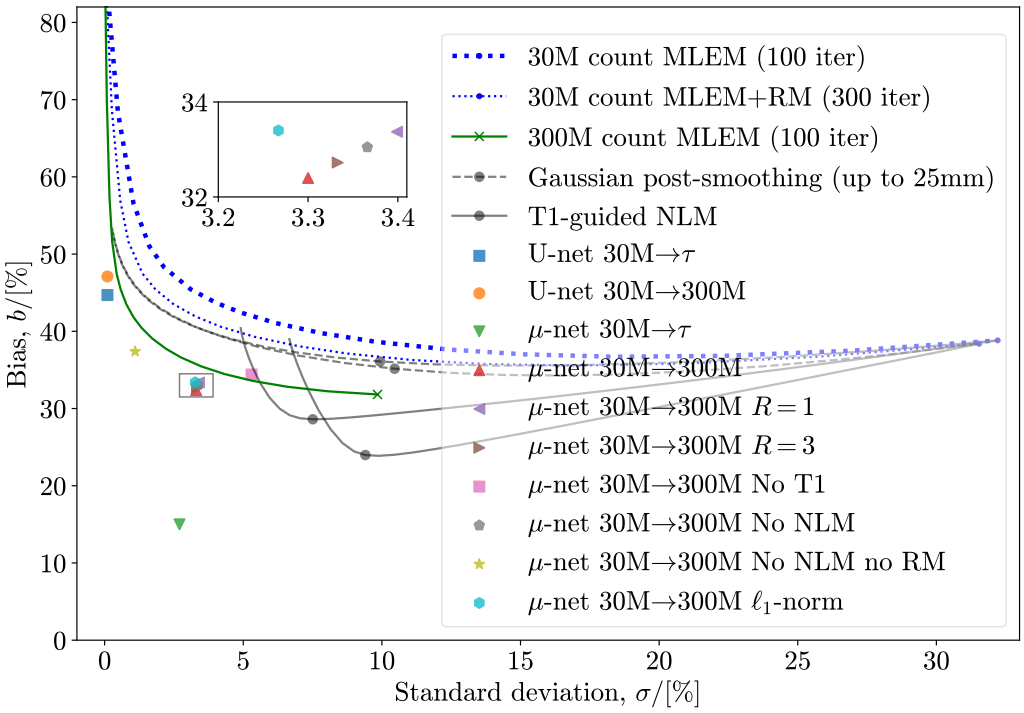
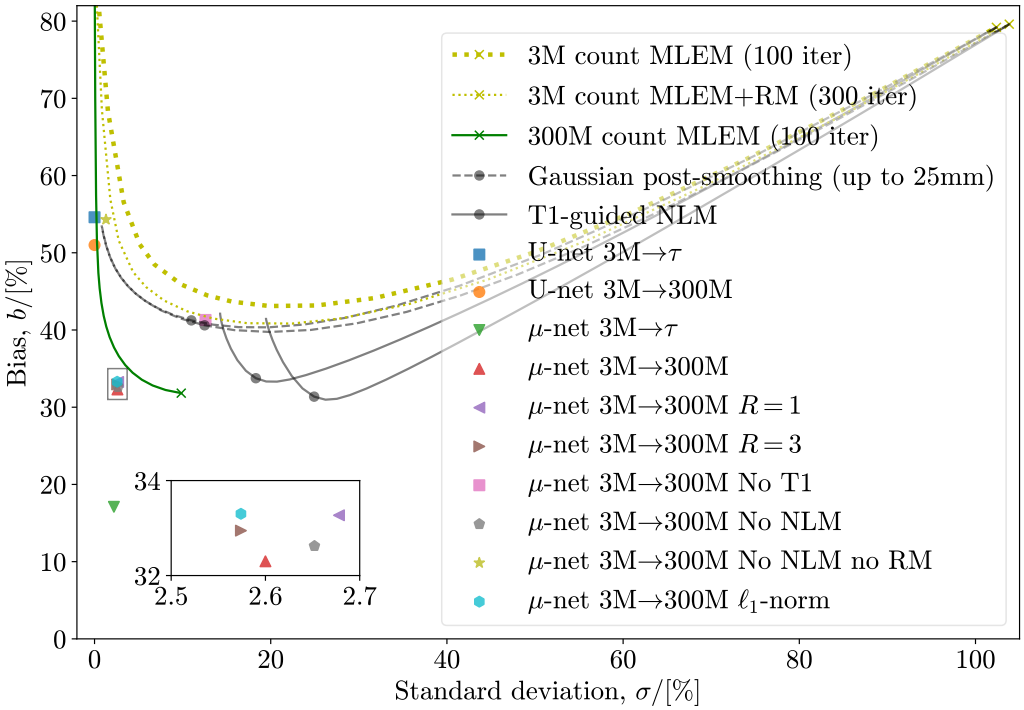


Figure 3.17: Test data profiles (horizontal line through the lesion circled in Figure 3.16 $\boldsymbol{\tau}$) for 3 M→300 M counts mapping.

Bias-variance curves for low (30 M) and very low (3 M) count MLEM reconstructions are shown in Figure 3.18. The endpoints of the reconstructions are used as network inputs. The separate cases of full (300 M) count endpoint and ground truth $\boldsymbol{\tau}$ target network outputs are considered. Increasing Gaussian PS (in steps of 0.1 mm) and NLM ($\Omega \in [10^{-5}, 10^{5}]$ in logarithmic steps – increments on the exponent of 0.01) guided filtering results of input PET volumes are also shown, with optimal (closest to the origin, identical to minimal NRMSE) values clearly marked. The proposed network's prediction based on low count inputs has comparable bias and much lower standard deviation compared to the target standard count reconstruction.



(a) Low (30 M) to full (300 M) count.



(b) Very low (3 M) to full (300 M) count.

Figure 3.18: Test bias versus standard deviation. Distance from the origin corresponds to NRMSE (note that the axes have different scales). Curves show the trade-off with increasing MLEM iterations (dotted for the [very] low count inputs and solid green for the full count target) starting in the top left and finishing at endpoints marked with crosses. Gaussian PS of increasing FWHM (solid black) and NLM filtering with increasing Ω (dashed black) are also applied to the MLEM endpoints, and optimal (i.e. minimum NRMSE as per Section 3.3.1.2) values are marked with circles. Unless specified otherwise, $R = 2$ realisations of one patient dataset are used to train each network. Meanwhile 10 realisations are used for calculating test standard deviation.

Zeroing inputs has a detrimental effect on test error in all cases. For example, with the low (30 M count) inputs, excluding MR information (also excluding MR-guided NLM; purely supplying MLEM and MLEM+RM, 45 % NRMSE) is slightly worse than not including NLM and MLEM+RM (purely supplying MLEM and MR, 39 % NRMSE). This is interesting as it implies that (for the given noise level) RM is less important for quality improvement than MR information. Furthermore, it is interesting to note that re-training the network with more ($R = 3$) realisations evidently has negligible improvement, while using fewer ($R = 1$) has very little detriment. The very low count results (Figure 3.18(b)) make it clearer that omitting resolution modelling information harms network performance more than omitting MR information does. There is also a slight improvement as training realisations R increase from 1 to 2, and a negligible improvement from 2 to 3.

Meanwhile, using an ℓ_1 loss function results in a slightly higher NRMSE (compared to NRMSE being optimised for directly).

Ideally the networks should be re-trained several times in order to produce confidence intervals to verify these results, as explored in Chapter 5 later.

As a final investigation, further modifications to the loss function are considered. Specifically, the training method is changed to an adversarial regime (GAN [124], Section 2.3.4). This can be posed as a modification to the loss function – making it a partially learned objective function. The learned component is a network called a discriminator. The discriminator D used here is a 3D network with 2 convolutional layers, shown in Figure 3.19. D consists of a convolution layer of 64 kernels of width 3 and stride 8 followed by a sigmoid activation function and a fully connected layer outputting one logit.

Incorporation of a discriminator or adversarial loss can increase similarity to desired outputs (particularly visually). Discriminator networks aim to distinguish between target and prediction outputs. Adversarial training involves using a discriminator's feedback to enhance the prediction network's output consistency with the desired targets. Discriminators have been used in patch-based PET denoising U-nets in 2D [166] and 3D simulations [165].

After 4 000 epochs of training the post-processing network μ, the discriminator is trained alongside for a further 2 000 epochs in an adversarial training regime as in [166], [167]. In this manner, feedback from D is used to further refine μ's parameters so that its output to more closely resembles its targets.

The discriminator D is trained using BCE loss (Equation (2.31)). For comparison, a TV loss term is also considered (aimed at enhancing edges by considering the spatial gradient). Finally, the generator network is also retrained using both an adversarial and a TV term, with the overall modified generator's loss function L'_G given by Equation (3.9).

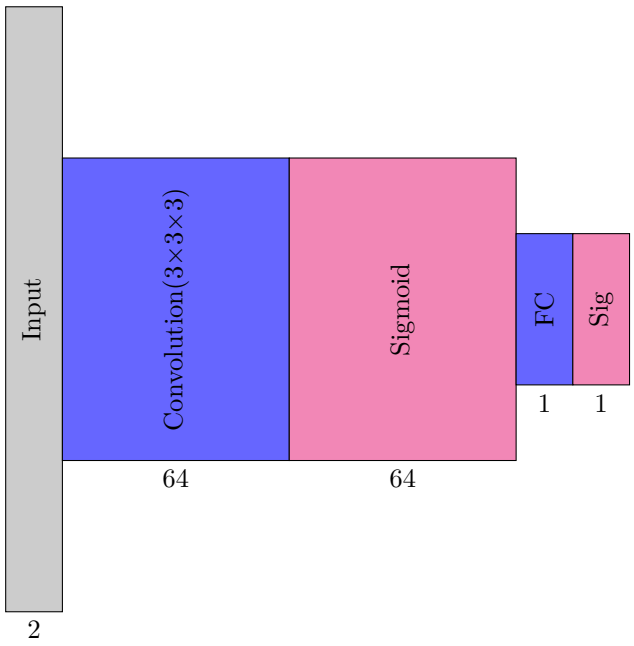


Figure 3.19: Visual representation of discriminator network architecture which outputs probability of the input being a real target (rather than post-processing network prediction). The first convolution uses a stride of 8, while the fully connected (FC) layer performs an unpadded convolution with the same kernel dimensions as its input. The final output is thus a single value $\in [0, 1]$ to be interpreted as a probability.

$$L'_G = \epsilon\left(\mu(\boldsymbol{\theta}), \boldsymbol{T}\right) - 0.00005|\nabla\mu(\boldsymbol{\theta})|^2 - 0.01 \times [\mathrm{BCE}(D(\mu(\boldsymbol{\theta})), \mathbf{0}) + \mathrm{BCE}(D(\boldsymbol{T}), \mathbf{1})], \tag{3.9}$$

where L is the overall loss,

$|\nabla \cdot|$ represents the magnitude of the forward-difference spatial gradient (with respect to voxel indices),

D represents application of the discriminator to yield a probability of whether the input is a target (1) or generated prediction (0).

The entire analysis is also repeated replacing the μ-net with a U-net (shown in Figure 3.20) and retraining.

The b-σ curves shown in Figure 3.21 are for the extreme case of recovering the simulation ground truth $\boldsymbol{T} = \boldsymbol{\tau}$ from very low (3 M) count data. It is clear that using a TV loss term has a detrimental effect on network prediction bias, but this can be compensated for by incorporating an adversarial discriminator D. The lowest overall NRMSE is obtained when using both TV and adversarial loss terms. Corresponding images of endpoints are depicted in Figure 3.22. Adversarial training increases standard deviation while decreasing bias, resulting in an improved or at least similar NRMSE while also improving visual similarity of the output to the target. However as the improvements are quantitatively marginal, in this extreme noise reduction case, a simpler network – i.e. a μ-net – may be favourable over the alternatives.

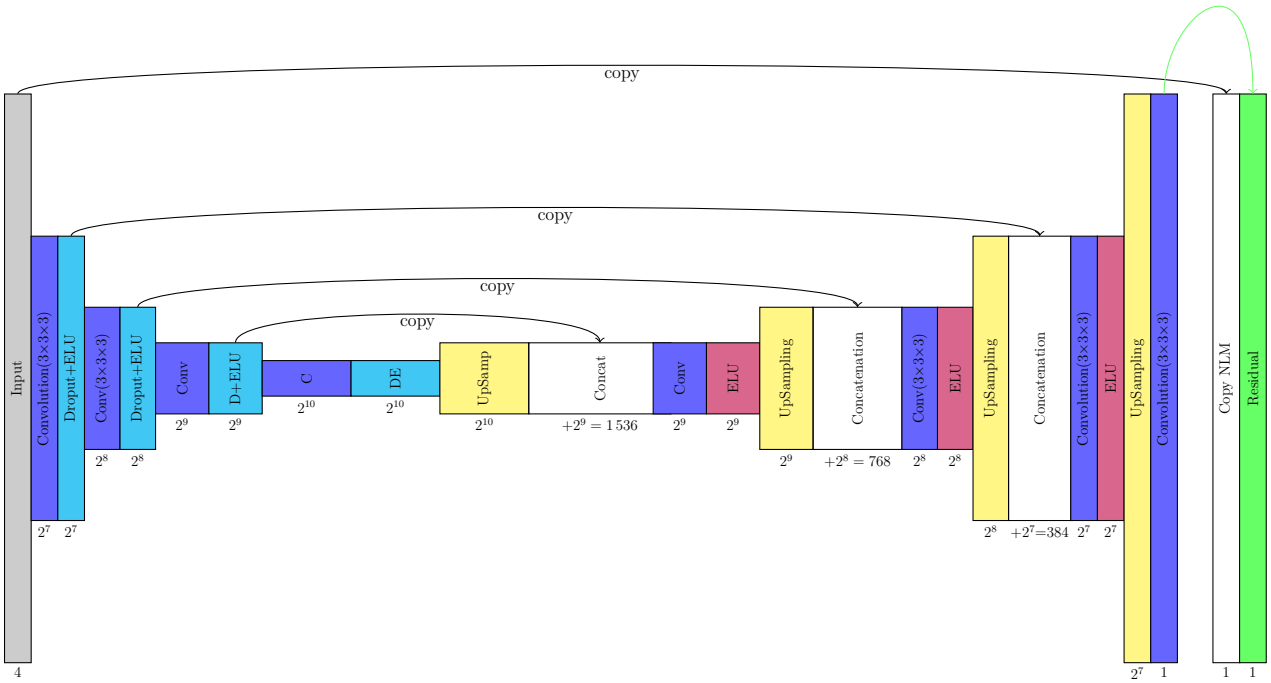


Figure 3.20: Visual representation of a post-processing U-net (compare to Figure 3.14). Convolutions use kernel width 3 and – when downsampling – stride 2 and 20 % dropout, while (trilinear) upsampling uses a scale factor of 2. The final residual layer performs element-wise addition between the last layer and the input NLM channel.

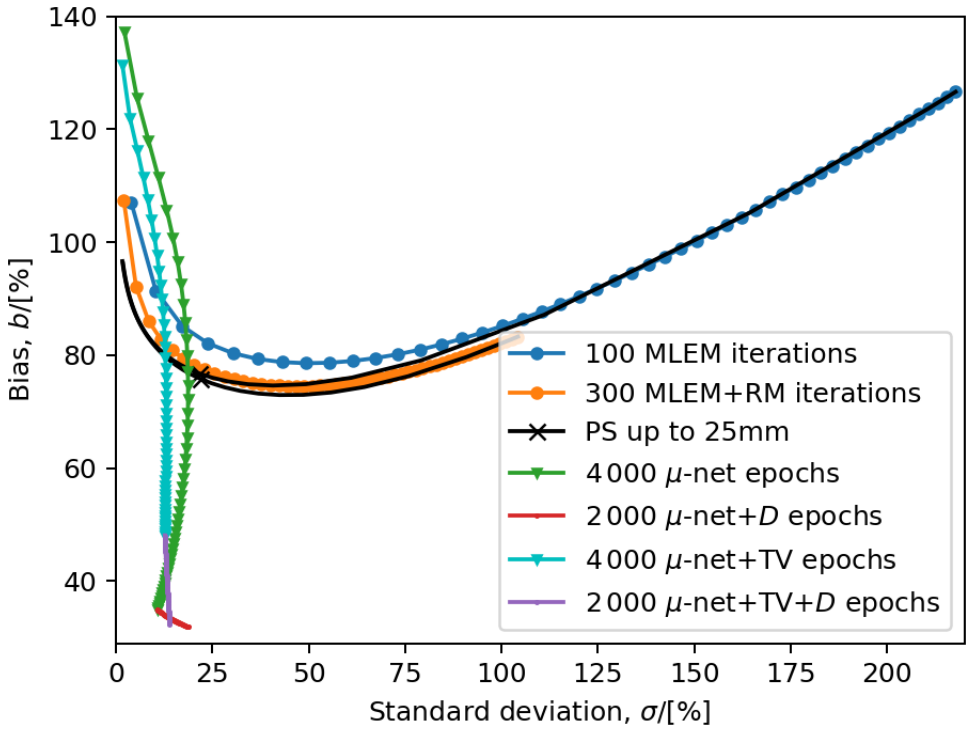


Figure 3.21: Standard deviation σ versus bias b (calculated against the simulated ground truth $\boldsymbol{\tau}$) with increasing MLEM iterations for **test** data. The reconstruction endpoints (along with T1 volumes) serve as inputs to the network. Gradual Gaussian post-smoothing (PS) of up to 25 mm full width at half maximum (FWHM) of the endpoints are also shown, with minimal NRMSE marked with crosses.

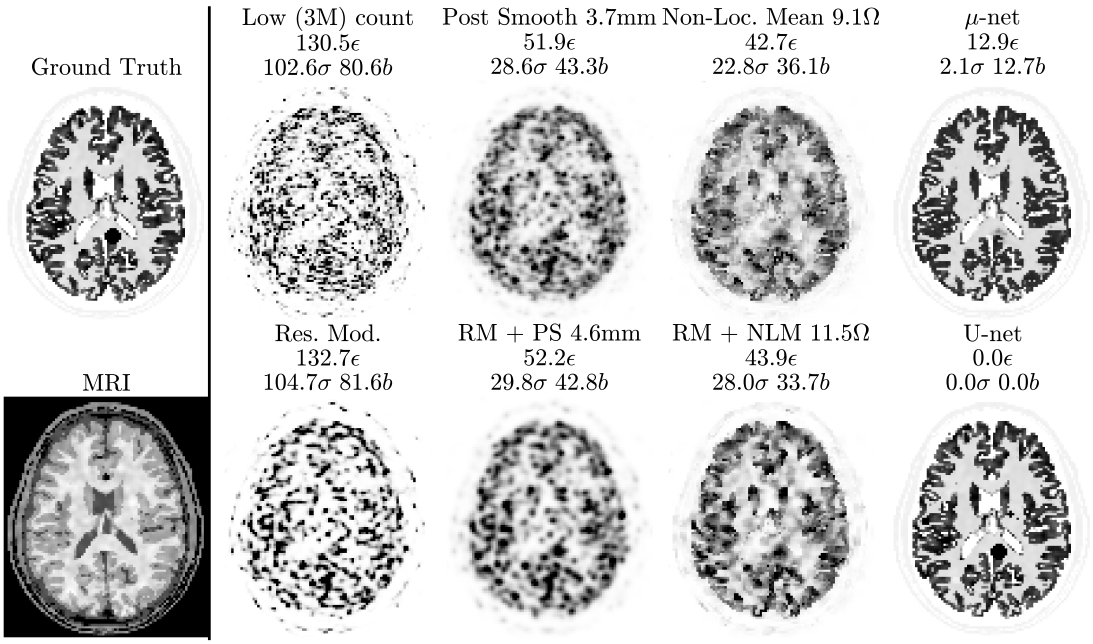


Figure 3.22: Central slices from 3D endpoints of one **training** simulation subject. Bias b, standard deviation σ and NRMSE ϵ are calculated across 10 realisations.

Corresponding test data (not used during training) results are shown in Figure 3.23. Note the poor performance of the U-net compared to the μ-net due to the former overfitting on the limited amount of training data from Figure 3.22.

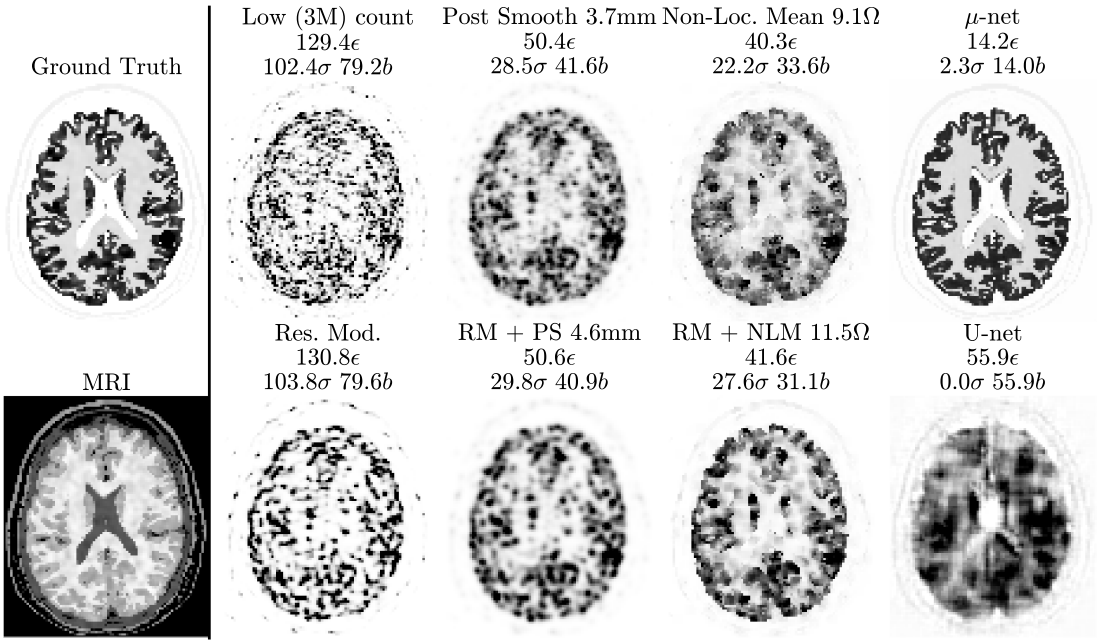


Figure 3.23: Central slices from 3D endpoints of one **test** simulation subject.

3.3.2 Patient data

All of the experiments based on phantom simulations in Section 3.3.1.2 and Section 3.3.1.3 are repeated for real patient data. Note that NRMSE and bias-variance curves cannot be calculated against the ground truth as the latter is unknown.

3.3.2.1 Reference methods

As with the case of simulations, both PS and NLM have hyperparameters with are optimised over the available clinical data. As the ground truth is unknown, such optimisation is only possible using the full count reconstructions as a reference. The optimisation hyperparameters at the two different input count levels are given in Figures 3.24 and 3.25, respectively.

These curves look remarkably similar to the simulation results in Figures 3.7 and 3.8 earlier in terms of curve shape and optimal hyperparameter values, indicating the simulations were likely good approximations of real data. Also similar to the simulation case, a larger NLM neighbourhood width of 7 (rather than 5) is used for very low (3 M) count inputs to avoid block-like artefacts.

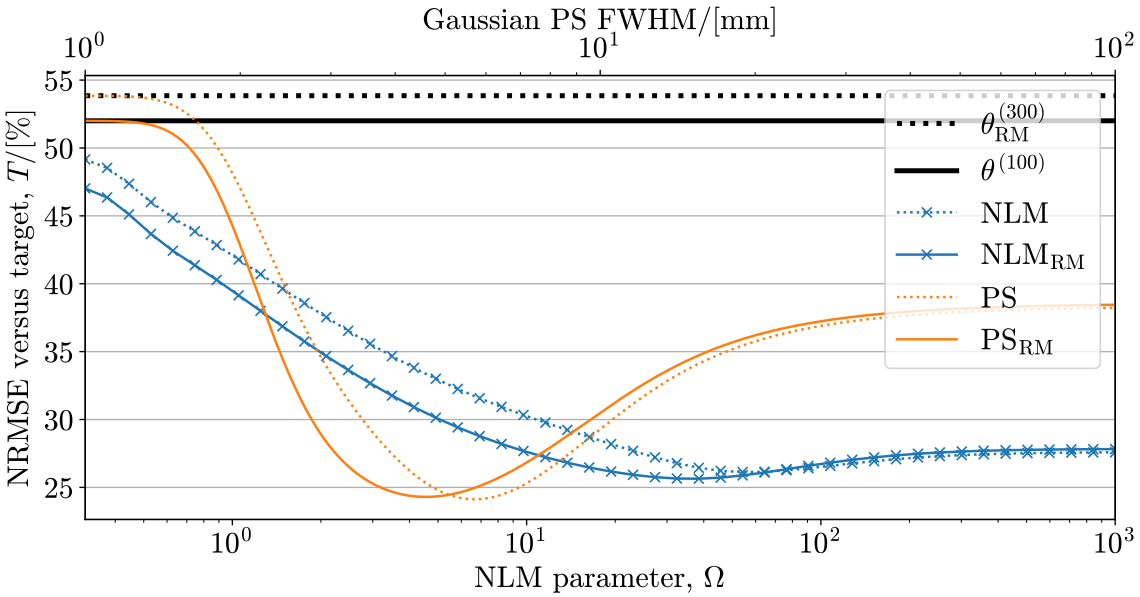


Figure 3.24: Optimal hyperparameters for PS and NLM for 30 M count inputs minimising NRMSE against a full (300 M) count target clinical patient reconstruction (comparable to simulations in Figure 3.7).

Corresponding endpoint images for the above methods are shown in the study below.

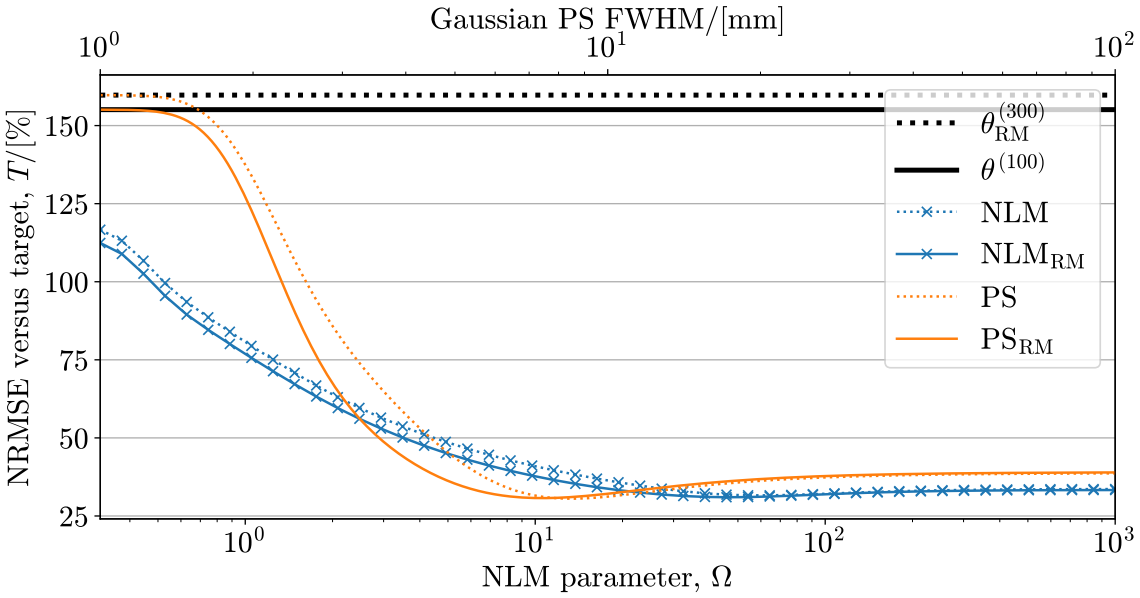


Figure 3.25: Optimal hyperparameters (similar to Figure 3.7) for 3 M count inputs (comparable to simulations in Figure 3.8).

3.3.2.2 Study: micro-networks

All of the simulation-based experiments from Section 3.3.1.3 are repeated here with patient data.

The first model proposed in Figure 3.10 is now trained on real patient data. Similar results for validation data are shown in Figure 3.26. As the ground truth is unknown, it is not possible to show the true NRMSE and bias. However, σ values are calculated across 3 disjoint low count data sets.

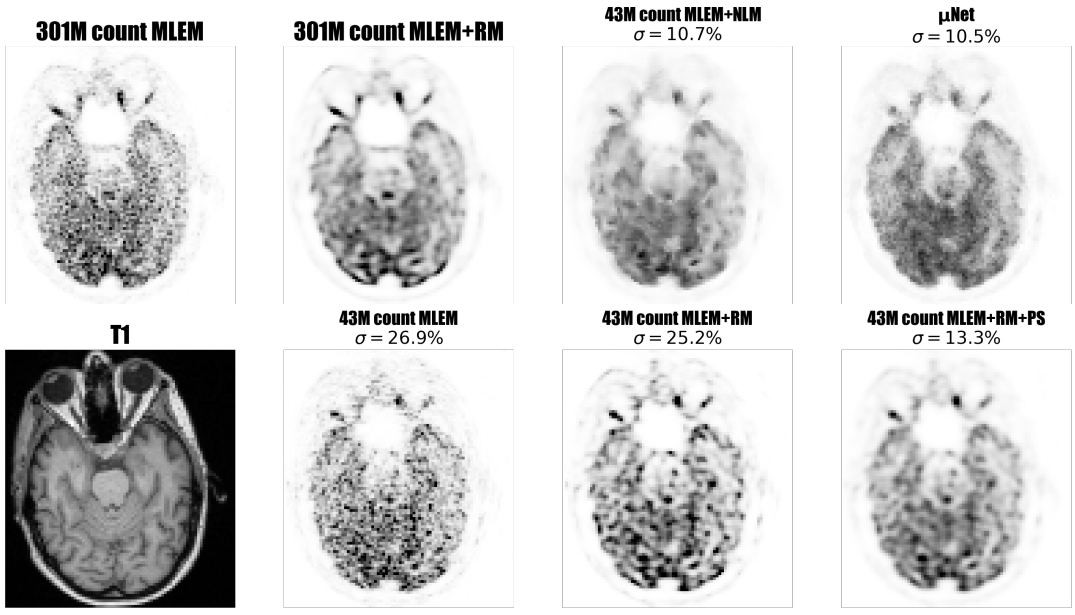


Figure 3.26: Central slice from the test dataset (comparable to simulation results in Figure 3.12).

It is interesting to note that while the μ-net (top right) produces the lowest σ, it has more visual (apparent) noise than the comparison (43 M count) methods. This means the network is not performing simple smoothing, but considering each voxel independently (i.e. there is reduced intervoxel covariance). On the other hand, it may also be that the network is deliberately adding noise in order to match the noise properties of the training targets. A more quantitative analysis is given in Section 5.3.2 later, including approximation of bias, NRMSE, difference images, performance in lesion RoIs, and comparison against a wide range of competitive methods.

The network architecture from Figure 3.14 is considered next. as with the first simulation study, this network incorporates an NLM input, a different (ELU) final activation which removes the need for scaling, and a choice of hyperparameters obtained from the earlier optimisation in Figure 3.13.

Test results after retraining on patient data are shown in Figure 3.27. Once again, U-net (architecture as shown in Figure 3.15) results are included for comparison. Note that since the ground truth is unknown, metrics are calculated with reference to the full count reconstruction $\boldsymbol{\theta}_{\text{full}}^{(100)}$.

Finally, the loss function is further modified; incorporating an adversarial training regime (using the discriminator from Figure 3.19) and TV term, as show in Equation (3.9).

Real data results are show in Figure 3.28 and Figure 3.29 for one training and one test patient dataset, respectively. Once again, U-net results (retraining the architecture depicted in Figure 3.20) indicate strong overfitting.

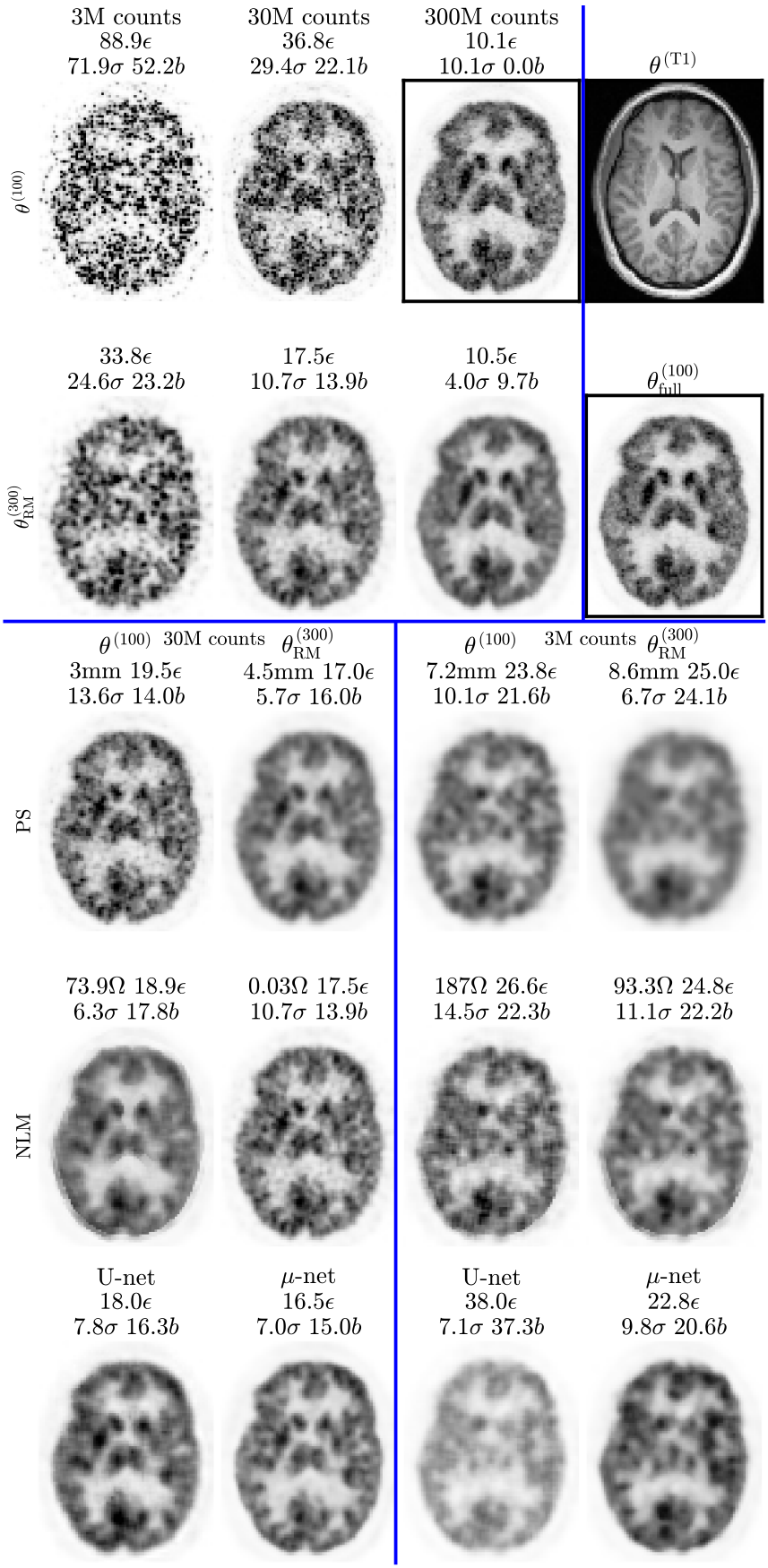


Figure 3.27: Patient data **test** results: cropped central slices from MLEM reconstructions of patient 2. The top left panel of 6 images are standard MLEM reconstructions at various bootstrap-sampled count levels without (top row) and with (bottom row) resolution modelling. The top right image in the black box is the 300 M reconstruction target $\boldsymbol{T}$ for comparison. Meanwhile the two images in the top right panel are the T1-weighted MR and raw (no bootstrap sampling) reconstructions. The lower panels show low (30 M count, left panel) and very low (3 M count, right panel) results of various post-processing methods: PS and NLM applied reconstructions with and without RM, as well as U-net and μ-net outputs.

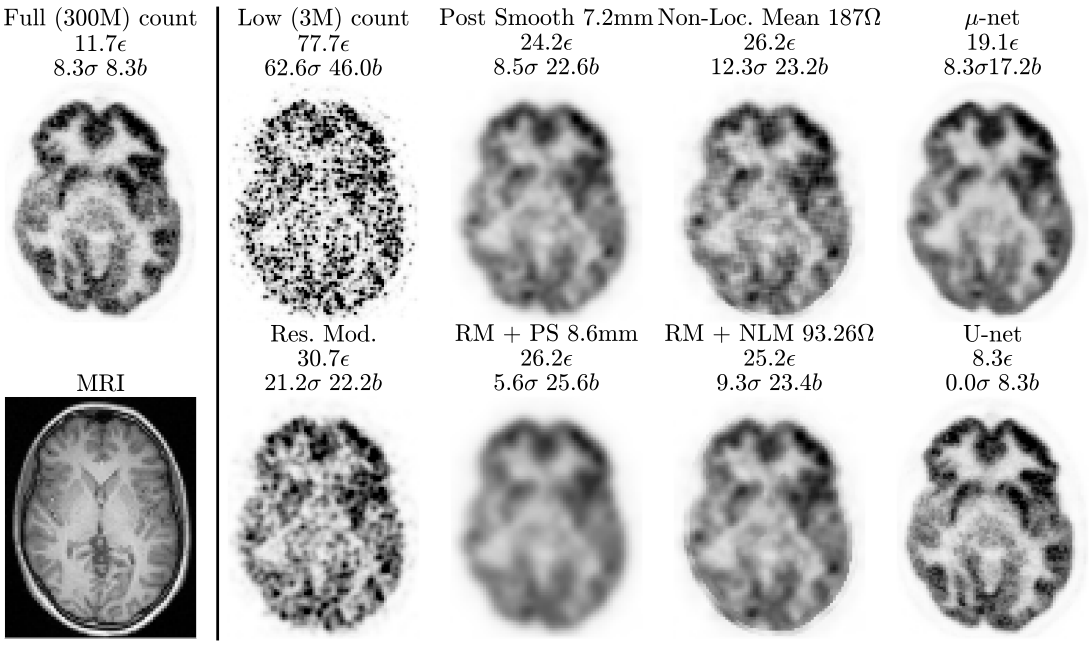


Figure 3.28: Central slices from 3D endpoints of one **training** patient dataset (analogous to simulation results from Figure 3.22). The patient suffered from epilepsy, and grey matter hyperintensities are visible in the frontal cortex. Bias b, standard deviation σ, and NRMSE ϵ are calculated using the full reconstruction ($\boldsymbol{\theta}_{\text{full}}^{(100)}$, not shown) as a reference.

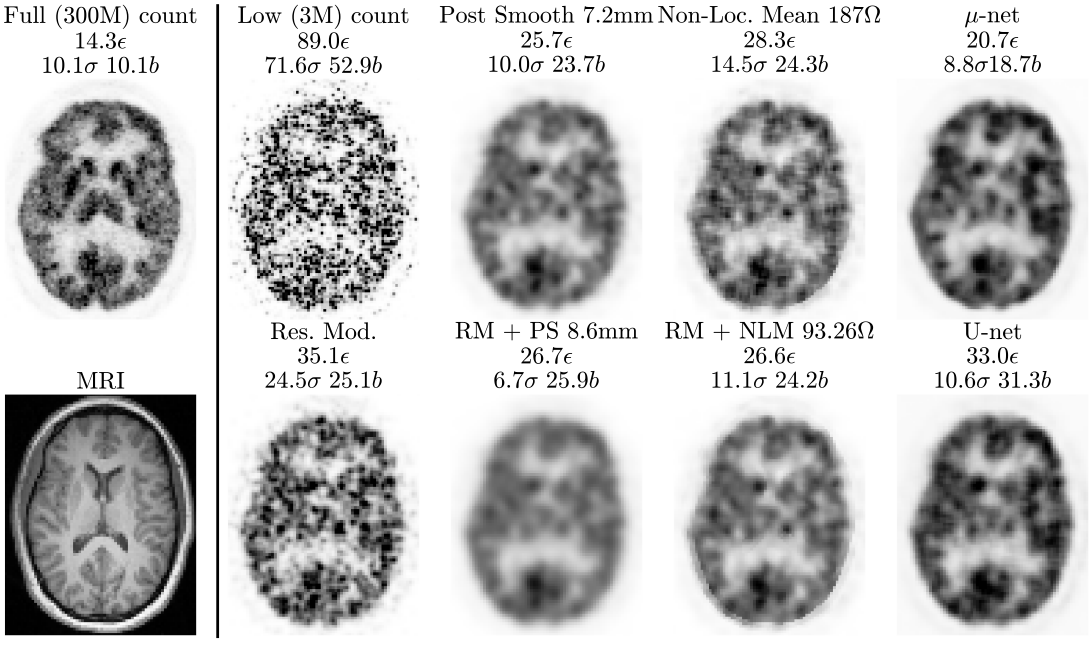


Figure 3.29: Central slices from 3D endpoints of one **test** patient dataset (analogous to simulation results from Figure 3.23). Error metrics (b, σ, and ϵ) are calculated using the full reconstruction ($\boldsymbol{\theta}_{\text{full}}^{(100)}$, top right panel in Figure 3.27) as a reference.

3.4 Discussion

A number of different inputs, loss functions, training regimes, activation functions and layer depths & widths are considered in the sections above. Note that whenever an architectural or dataset change is made, the network is retrained. Additionally, when the reconstruction software is changed from *APIRL* to *NiftyPET*, the entire optimal hyperparameter search is redone.

A summary of the architectures considered in this chapter is shown in Table 3.2, below.

Short Name	Preliminary	μNet	μNet(-P)	μNet(GAN)
Trainable parameters	6.67 k	147 k	43.7 k	309 k
" (inference)	6.67 k	147 k	43.7 k	43.7 k
Training voxels	30.1 M	150 M	120 M	120 M
Voxels per parameter	4.51 k	1.02 k	2.74 k	388
" (inference)	4.51 k	1.02 k	2.74 k	2.74 k
Count upgrade factor	∞	7	10; 100	10; 100
Last activation	Sigmoid	Sigmoid	ELU	ELU
Loss function	NRMSE	NRMSE	NRMSE	NRMSE; TV; adversarial
Input modalities	PET; T1	PET; PET_{RM}; T1; PET∘T1; PET_{RM}∘T1	PET; PET_{RM}; T1; (NLM)	PET; PET_{RM}; T1; (NLM)
Reconstruction framework	APIRL	APIRL	NiftyPET	NiftyPET

Table 3.2: Overview of the four experiments considered in this chapter. Each (three convolution layer) network has the number of kernels in each layer optimised in order to minimise loss (using one training and one validation simulated phantom/bootstrap sampled patient reconstruction). Differences in the number of network parameters between training and inference are due to a discriminator being used solely in the training process for the GAN.

While the primary objective here is to post-compensate for degradation due to noise, the μ-net can also suppress artefacts, including PVEs and ringing. Results above indicate that choice of hyperparameters is crucial for network performance. An optimal combination can produce results far superior to more complex architectures – even where the more complex architecture can theoretically fully model the smaller one. This is clearly exemplified by the consistent relative lower error of the proposed μ-net compared to U-nets (note that Chapter 5 includes a larger training data set to reduce the likelihood of overfitting, and considers specific U-net proposals from the current literature). The proposed μ-net architecture has clearly impressive noise-suppression abilities, reducing NRMSE by up to 89 % compared to MLEM and 72 % compared to RM+PS (Figure 3.23).

The CNN proposed in this chapter has a relatively low complexity, and is therefore called a micro-net (μ-net). The network uses low count PET along with corresponding MR reconstructions to predict a full dose PET reconstruction. Due to its minimal memory requirements, the network can operate directly on 3D reconstructed volumes. The relatively low number of optimisation parameters also means that overfitting is most likely obviated (as discussed in Section 3.3.1.1). This is clearly demonstrated when comparing training with test predictions (such as Figure 3.22 versus Figure 3.23, and Figure 3.28 versus Figure 3.29) – error metrics remain consistent for the μ-net, while the U-net fails to cope with unseen test data.

This finding is perhaps not so surprising in the context of related work. For example, the first published CNN-based super-resolution proposal [81] was for a 3-layer network (with a design similar to that of the μ-net here), and noted that larger networks did not perform as well. It should be noted however that there do exist larger architectures that have been shown to produce good results in some cases [82], [168]. In any case, a validation study involving assessment by clinical experts on a much larger patient cohort would be required to ascertain whether low-count PET and CNNs are robust enough for use in clinical practice.

An advantage of optimising hyperparameters – in particular layer depth and width – is that the resultant network has relatively low memory and computational overhead, enabling use of full 3D volumes and lower training times. This can greatly ease practical implementation – both for the purpose of future research as well as use in the clinic. In contrast, there is a prevailing notion in the current literature that increasing depth combined with a reducing kernel size – while maintaining a constant receptive field with fewer traininable parameters – increases expressivity due to non-linearities. However, this presumes that the true underlying phenomenon requires many non-linearities to accurately model. This may not be the case with low count PET imaging. Furthermore, a large receptive field – while useful for segmentation and detection as per the original intention of U-nets – may be undesirable for PET denoising. Using an overly complex model leads to overfitting, similar to fitting a high order polynomial to noisy linear data. Nevertheless, deeper, more complex networks from the current literature as well as regularisation and performance assessment as a function of training data size may be found in Chapter 5.

The proposed CNN can be trained to approximate standard (300 M) count reconstructions from low (30 M) count data. The resultant NRMSE is at least 36 % lower (calculated over 10 realisations against the ground truth simulations) compared to MLEM used in clinical practice (Figure 3.16). In contrast, a smaller decrease of 25 % or 33 % in NRMSE is obtained when an ideally optimised (using knowledge of the ground truth) PS or NLM is performed. A 26 % NRMSE decrease is obtained with both RM and optimised PS. More pronounced improvements are observed for low count real patient datasets; where error metrics are calculated against the complete reconstructions of raw data (Figure 3.27). Best-case decreases of 47 % and 49 % for PS and NLM methods respectively are observed compared to a decrease of 51 % for the U-net and 55 % for the proposed μ-net.

The improvements are even more pronounced for the case of very low (3 M) count input PET data: for simulations (Figure 3.23), the decrease in NRMSE is 89 % for the proposed μ-net compared to 61 % (PS) and 69 % (NLM). For patient data, the decreases are 71 % (PS) and 68 % (NLM) versus 77 % (μ-net). In both cases, however, the U-net fails at test time due to overfitting on the training data.

Overfitting to training data is demonstrated to occur as the network size is increased. In an extreme case, a U-net (which produces better predictions for training data) is shown to completely fail on test data due to overfitting to this case of very limited training data. Meanwhile, the resultant images from the proposed μ-net (which has low training data requirements) have lower noise, reduced ringing and partial volume effects, as well as sharper edges and improved resolution compared to conventional MLEM.

The simulations results clearly show that application of a μ-net always produces lower NRMSE than post-smoothing or NLM filtering (see Figure 3.18). The micro-network predictions in Figure 3.16(h) also show much less noise – a reduction in standard deviation σ by a factor of up to 3 compared to rivals (c)-(f) – and lower bias. The exception is the case of mapping 30 M→300 M, where a slightly higher σ than NLM is compensated for by the lower bias to still produce a lower overall NRMSE (visible in Figure 3.18(a)). This reduction is achieved without sacrificing image resolution.

The network biases make it possible to trivially correct for spatially-invariant (global) bias in the input PET images. Meanwhile, robustness to the spatially-invariant component of noise is achieved by having a small architecture: micro-nets are not very dense; instead consisting of small local kernels which are applied to the whole input – i.e. spatially-invariant. As the kernels are optimised over the entire input, they must be able to cope with the various instances of noise found over the whole volume. The training phase should result in kernels optimised for the "average" region. Kernels should thus be able to compensate for spatially-invariant noise irrespective of the chosen loss function. Since micro-nets have a small receptive field (a small neighbourhood width of 7 input voxels which are able to affect an output voxel) applied over a large volume (two orders of magnitude wider than the receptive field) it seems logical that they should not be able to compensate for spatially-variant noise. However, it is possible that based on the features detected in different regions, kernels may indeed be activated by (and thus "aware of") different regions, thereby handling both spatially-variant noise and bias.

Using an ℓ_1 loss or adding a TV loss term typically harms test performance. However, using an NRMSE loss along with a TV term and a discriminator results in a comparable overall NRMSE but better qualitative performance. Similar though less pronounced findings are observed for patient data, where NRMSE can be calculated against the full count data set.

The novelty of the contributions of this chapter lie mainly in a combination of various approaches. Each individual consideration is not unique in and of itself, but rather works together in unison. The primary considerations are summarised as follows:

Activation functions Using sigmoidal – and for the final layer, ELU – activation functions (Section 2.3.1.2) introduces non-linear kernel sensitivity control without making it too easy to set negative values to zero (discarding information). Where non-negativity is desirable, ELU provides a weaker constraint than ReLU but is far less susceptible to the vanishing gradient problem. The benefit (particularly for µ-nets) outweighs the increased training time.

Fully 3D Using 3D volumes (rather than 2D slices) means adjacent slice information is available to kernels, resulting in a superior ability to correct PVEs and distinguish between signal and noise.

Multiple realisations For a given input noise level, training on more than one noise realisation of the same patient ($R > 1$) further increases robustness to noise at the chosen level.

No patches, no non-unity strides, and no augmentation Working directly on the full volumes (without subdivision into small regions) and not pooling or downsampling circumvents boundary-related issues. Convolving with unity stride also helps use all available training data without needing to resort to data augmentation techniques. Note that supplementary figures in [144] indicate that a 3D patch overlap of 8 voxels in all directions is required for a patch-based network to achieve similar performance.

Competitive inputs Even a relatively simple operation such as an element-wise product between two channels requires a fully connected (dense) layer (same number of parameters as input voxels). To avoid this unnecessary vast increase in optimisation parameters, product images – or indeed any advanced post-filtering method such as NLM guided filtering – can be pre-computed and supplied as inputs. This results in better joint edge modulation across modalities.

Few kernels (optimal network depth and width) A comparatively low number (63) of kernels n are used in each layer in order to avoid redundant parameters and preclude the possibility of overfitting (memorising the training data rather than learning features). The number of optimisation parameters (see Equation (3.8)) is comparatively small ($\mathcal{O}(10^5)$) in total. The smaller parameter search space also decreases the optimisation difficulty. Networks with other values of n ranging from 1 to 2 048 are also trialled and found to perform either similarly or not as well.

3.5 Summary

This work proposes the use of a relatively low-complexity CNN (a micro-net) as a post-reconstruction MR-guided image processing step to reduce noise and reconstruction artefacts while also improving resolution in low count PET scans. The CNN is designed to be fully 3D, robust to very limited amounts of training data, and to accept multiple inputs (including competitive denoising methods).

The proposed CNN can be trained to approximate standard (300 M) count reconstructions from low (30 M) and very low (3 M) count PET data in conjunction with MR reconstructions. The resultant NRMSE is consistently lower compared to MLEM and PS used in clinical practice – both for simulated phantoms as well as bootstrap-sampled patient data. The proposed μ-nets also consistently outperform NLM guided filtering and competitive post-processing U-nets – in the latter case due to demonstrable increased robustness against overfitting (techniques to improve U-net performance are explored in Chapter 5). The improvements are more pronounced for the case of very low (3 M) count input PET data. Meanwhile GANs – which can be used to augment data sets [169] – have been recently applied to low dose PET [165], [170]. The results in this chapter indicate that improvements from using an adversarial training regime are minimal and largely qualitative rather than quantitative.

Future work will need to consider the impact of mismatched noise levels (testing on different noise levels than used for training), as well as using one architecture to compensate for noise and artefacts at different noise levels and at different iterations of MLEM (rather then re-training a network for each case).

It would be particularly interesting in future work to extend the network to include joint modality (synergistic) post-processing such as PET-guided undersampled MR reconstruction, or even modality generation such as PET prediction based on MR.

Increasing the number of training data sets will also produce a more robust network with even better resolution recovery and artefact suppression properties, and is explored in more detail in Chapter 5. Chapter 5 also investigates more of the concepts introduced in Chapter 2. In particular, more focus is given to concepts such as dropout, batch normalisation, strided convolution, upsampling, residual connections, and adversarial training; and whether such concepts are more likely to favour deeper architectures such as U-nets.

Most importantly, future validation studies on much larger patient cohorts are also required to comprehensively ascertain robustness for use of low-count PET and CNNs in the clinic.

Chapter 4

Data Consistency and Null-space Networks

This chapter focuses on the (often avoided [56], [59]) RM component of iterative reconstruction in PET. Specifically, the objective is to constrain a post-processing network to remove ringing artefacts while retaining (or even improving) resolution in an extremely robust manner (i.e. without misleadingly inserting non-existent features or removing relevant ones). The general technique which can help to achieve this is often referred to in the current literature as "data consistency." However in the case of PET, the underlying listmode count data are noisy, and thus it is undesirable for a reconstructed volume to be fully consistent with the acquired data. Rather than enforce consistency with the noisy data, this chapter concerns enforcing consistency with one of the deterministic parts of the model – namely the PSF. It should be noted that even when noise-free, consistency does not preclude the possibility of artefacts – instead, it simply introduces a constraint on artefacts.

4.1 Motivation

As discussed in Section 1.3.1, MLEM iterative reconstruction is a powerful technique with capabilities including compensating for resolution degradation effects by modelling the PET system's PSF. However, this leads to Gibbs ringing artefacts (overshoots and undershoots near edges) which can compound catastrophically in certain cases (e.g. spherical lesions with overshoots of 70 % [154]). Such ringing can potentially lead to misdiagnosis of tumour aggressiveness [60], [61]. The use of RM is therefore controversial, and inappropriate for some clinical tasks [59].

Rather than tackle both denoising and artefact reduction together (as done in Chapter 3), the aim here is to focus solely on the RM aspect of PET.

As detailed in Section 1.4, the simplest and clinically most widely used post-processing step is PS. Ringing artefacts are guaranteed to be removed if the smoothing kernel is at least as wide as the RM reconstruction PSF [64], [65]. However, PS works against the resolution gains of RM – suppressing noise at the expense of reducing resolution. The special case of using identical kernels (an example of the method of sieves) results in images of slightly better quality [64], [65] than if neither RM nor PS was used at all (see Figure 1.5). The minor improvement raises questions as to whether it is worth the extra effort and complexity.

The problem addressed here is whether it is possible to optimise a CNN as an alternative to PS in a rigorously constrained manner. Specifically, the network's output should be guaranteed to remove ringing artefacts while simultaneously retaining resolution. Ideally, the network should also ignore other artefacts, noise, and even features – thereby enabling training on simulations and testing on real patients. In related work, Lucas et al. have conducted a review of deep learning (DL) for inverse problems (including GANs) [171]. They conclude that future challenges include engineering knowledge about inverse problems into the DL architecture is required.

An interesting type of CNN which could potentially satisfy these properties is a null-space network. Ringing artefacts are caused by a sharp drop in recovery of high frequency components. The unrecoverable data exists in the null-space of the imaging system. A deep null-space network [172] has the ability to fill in this null-space, thereby removing artefacts in a "data consistent" manner. Null-nets have recently been applied to FBP of sparse photoacoustic tomography (PAT) [173] and undersampled Radon transforms [174]. Note that the null-net approach is distinct from that of the unrolled/unfolded iterative reconstruction methods using DL which have been proposed for PET [133], [138], [139], [175]. The unrolled methods incorporate an ML-informed regularisation term into the system model, thereby subtly altering the model altogether. Null-nets, meanwhile, aim to regularise while remaining consistent with the original model.

Theoretically, the true system PSF model is full rank (or at least this is certainly the case when modelled by a Gaussian). This means that it should be fully invertible, and there is no null-space. However, in practice due to sensitivity limitations (machine precision and noise), the singular value decomposition (SVD) spectrum would have many effectively zero values, meaning solutions are non-unique [17].

Null-nets are often called "data consistent," and applied in cases where the acquisition has low noise but is missing data (subsampled as is the case with MR sequences). Data or model-based deep learning (MoDL) based techniques for inverse problems have been applied to MRI [176]. The network serves to effectively fill in missing data without introducing inconsistencies with the acquired data. More formally, the network's predicted MR volume – when forward projected (Fourier transformed) and subsampled – should match the originally acquired k-space readouts. As outlined above, such consistency is undesirable for the PET reconstruction process. The difference in PET is that the data is relatively complete (not subsampled) but noisy. Instead, a null-net can be repurposed to ensure model consistency – or more accurately, consistency with the RM part of the PET system model. Formally, the network's predicted PET volume should match its input (the MLEM reconstructed volume) – when both are blurred with the same PSF kernel. In other words, smoothing with the PSF kernel is invariant to the network's effect. The specifics of the network are discussed in the following section.

Note that in the context of null-nets, "data consistency" refers to a post-processing step which is invariant to (the whole or part of the) forward model. This is not to be confused with the term when applied to the iterative reconstruction process. In the latter case, "data consistent" refers instead to part of the objective function.

4.2 Methods

4.2.1 Theory

This section outlines what a null-space network is, and how it can be used to regularise MLEM iterative reconstruction (and in this case, Richardson-Lucy (R-L)).

A key concept for this chapter is the Moore-Penrose pseudo-inverse [177]. The pseudo-inverse of a matrix $\boldsymbol{P}$ is written as $\boldsymbol{P}^{+}$. It is a generalisation of the inverse of a square matrix ($\boldsymbol{P}^{-1}$) – the pseudo-inverse can be computed for non-square, non-full-rank matrices. Formally, the following Moore-Penrose conditions for the pseudo-inverse must be satisfied:

$$\boldsymbol{P}\boldsymbol{P}^{+}\boldsymbol{P} = \boldsymbol{P}, \tag{4.1}$$

$$\boldsymbol{P}^{+}\boldsymbol{P}\boldsymbol{P}^{+} = \boldsymbol{P}^{+}, \tag{4.2}$$

$$(\boldsymbol{P}\boldsymbol{P}^{+})^{*} = \boldsymbol{P}^{+}\boldsymbol{P}, \text{and} \tag{4.3}$$

$$(\boldsymbol{P}^{+}\boldsymbol{P})^{*} = \boldsymbol{P}\boldsymbol{P}^{+}, \tag{4.4}$$

where $\boldsymbol{P}^{+}$ is the Moore-Penrose pseudo-inverse, and

$\boldsymbol{P}^{*}$ is the conjugate transpose of marix $\boldsymbol{P}$.

Solutions of a linear system $\boldsymbol{m} = \boldsymbol{P}\boldsymbol{\theta}$, if they exist, satisfy Equation (4.5) [178], below.

$$\boldsymbol{\theta} = \boldsymbol{P}^{+}\boldsymbol{m} + (\boldsymbol{I} - \boldsymbol{P}^{+}\boldsymbol{P})\boldsymbol{x}, \tag{4.5}$$

where $\boldsymbol{x}$ is an arbitrary vector and $(\boldsymbol{I} - \boldsymbol{P}^{+}\boldsymbol{P})\boldsymbol{x}$ is in the null-space.

The PET system model (Equation (1.4)) meanwhile has an additional background term modelling the mean of the noise. Moreover, for ill-conditioned systems (such as in PET), multiple solutions exist.

Nevertheless, one can consider a single part of the PET system model, namely the PSF modelling matrix $\boldsymbol{H}$ (excluding the projection into sinogram space and additive terms, i.e. $\boldsymbol{m}$ is a blurred volume and $\boldsymbol{P} = \boldsymbol{H}$ in Equation (4.5)). In the case of purely performing resolution recovery, MLEM iterative reconstruction is equivalent to the R-L algorithm, given by Algorithm 1.

Algorithm 1: Richardson-Lucy (RL) algorithm. Note that the symmetric Gaussian kernel used in the G function means that the adjoint is also G.

Input: m: blurred volume, σ: Gaussian blur parameter, k: number of iterations
Output: θ: reconstructed volume
$\theta \leftarrow \text{ones}(\text{shape}(m))$
for $i \leftarrow 1$ **to** k **do**
  $\theta \leftarrow \theta \times G_\sigma(m \div G_\sigma(\theta))$ `// G is a Gaussian blur; × & ÷ are element-wise`
end

The proposed null-net approach is to apply a residual operation $\boldsymbol{\mathcal{V}}$ (given in Equation (4.6)) to the reconstructed data. $\boldsymbol{\mathcal{V}}$ in turn incorporates an arbitrary operator $\boldsymbol{M}$ which can represent the application of a CNN.

$$\boldsymbol{\mathcal{V}} = \boldsymbol{I} + (\boldsymbol{I} - \boldsymbol{H}^{+}\boldsymbol{H})\boldsymbol{M}, \tag{4.6}$$

where $\boldsymbol{\mathcal{V}}$ is a residual (identity plus another operation) post-processing matrix;

$\boldsymbol{I}$ is the identity matrix, and

$\boldsymbol{M}$ may represent an arbitrary operation such as the application of a CNN.

Note that since $(\boldsymbol{I} - \boldsymbol{H}^{+}\boldsymbol{H})$ is a projector onto the null-space (the kernel of $\boldsymbol{H}$; $\ker(\boldsymbol{H})$), the post-processing operation $\boldsymbol{\mathcal{V}}$ satisfies the following property:

$$\boldsymbol{H}\boldsymbol{\mathcal{V}}\boldsymbol{\theta} = \boldsymbol{H}\boldsymbol{\theta} = \boldsymbol{m}. \tag{4.7}$$

In this case, Schwab et al. [172] propose that for any "classical" (i.e. orthogonal to the kernel of $\boldsymbol{H}$; $\ker(\boldsymbol{H})^{\perp}$) reconstruction operation $\boldsymbol{\mathcal{R}}$, the post-processing operation $\boldsymbol{\mathcal{V}}$ performs regularisation. This regularisation occurs in the null-space of the system and is thus data (or model) consistent. This is an important proposal as it means $\boldsymbol{\mathcal{R}}$ may be a Kullback-Leibler (KL) solution (as is the case with MLEM and R-L) and remain compatible with Equation (4.5) even though the latter is a least squares (LS) solution. The overall regularised reconstruction operator is given by Equation (4.8).

$$\boldsymbol{N} = \boldsymbol{\mathcal{V}}\boldsymbol{\mathcal{R}}. \tag{4.8}$$

More concretely, substituting Equation (4.8) into Equation (4.6) and rewriting using function notation ($\boldsymbol{\mathcal{R}} \to \hat{\boldsymbol{\theta}}$, $\boldsymbol{N} \to N(\hat{\boldsymbol{\theta}})$, $\boldsymbol{M} \to M(\cdot)$, $\boldsymbol{H} \to G(\cdot)$, and $\boldsymbol{H}^{+} \to \mathcal{R}_k(\cdot)$ – note that the last subsitiution is valid as per [173], [174]), the proposed post-processing null-net N is given by:

$$N(\hat{\boldsymbol{\theta}}) = \hat{\boldsymbol{\theta}} + M(\hat{\boldsymbol{\theta}}) - \mathcal{R}_k(G(M(\hat{\boldsymbol{\theta}}))), \text{where} \tag{4.9}$$

$$\hat{\boldsymbol{\theta}} = \mathcal{R}_k(\boldsymbol{m}), \tag{4.10}$$

and $\mathcal{R}_k$ performs k MLEM iterations (in this case R-L, Algorithm 1),

- G applies the forward model (in this case Gaussian smoothing),
- M applies a CNN,
- $\hat{\boldsymbol{\theta}}$ is an MLEM reconstructed (in this case, R-L) volume, and
- $\boldsymbol{m}$ is the raw input data (in this case a blurred image).

4.2.2 Training

Consistency is enforced by incorporating MLEM reconstruction (in this case, R-L) into the training of the network M. The loss function minimises the difference between a ground truth $\boldsymbol{\theta}$ and the null-net regularised simulated reconstruction $N(\mathcal{R}_k(G(\boldsymbol{\theta})))$ from Chapter 3 above. Using NRMSE for the loss function:

$$L(N;\boldsymbol{\theta}) = \sum_n \sqrt{\|\boldsymbol{\theta}_n - N(\mathcal{R}_k(G(\boldsymbol{\theta}_n)))\|^2 / \|\boldsymbol{\theta}_n\|^2}, \tag{4.11}$$

where $\boldsymbol{\theta}_n$ is the n^{th} ground truth volume.

This loss function modification is the only difference between the post-processing networks considered in the previous chapter and the null-space network here. Note that while the $\mathcal{R}_k(G(\boldsymbol{\theta}_n))$ term can be pre-computed, the null-net N itself also appears in the loss function. N as defined by Equation (4.9) includes a term $\mathcal{R}_k(G(M(\hat{\boldsymbol{\theta}})))$ which may not be pre-computed since it includes M (a CNN containing all the trainable parameters). This internal CNN output undergoes k iterations of R-L which must be performed at each training epoch, making training slow for large k.

Once trained, N satisfies $G(N(\hat{\boldsymbol{x}})) = G(\hat{\boldsymbol{x}})$ theoretically for any arbitrary ground truth $\boldsymbol{x}$ (not just $\boldsymbol{x} = \boldsymbol{\theta}$). However in practice – primarily due to the thresholding effects of activation functions – intensity ranges of $\boldsymbol{x}$ must lie within the range seen in the training data $\boldsymbol{\theta}$.

Following from concepts introduced in Chapter 3, M is implemented as a 4-layer fully 3D μ-net. Each convolution layer uses unit stride, kernel width 3, and zero-padding. The first 3 layers use 32 kernels and ReLU activation functions, while the final layer has 1 kernel and an ELU (a weak non-negativity constraint since – for PET imaging – negative values are not expected). The architecture is shown in Figure 4.1. The Adam optimiser is used with learning rate 10^{-3} and ℓ_1 convolutional kernel regularisation with weighting factor 10^{-4}.

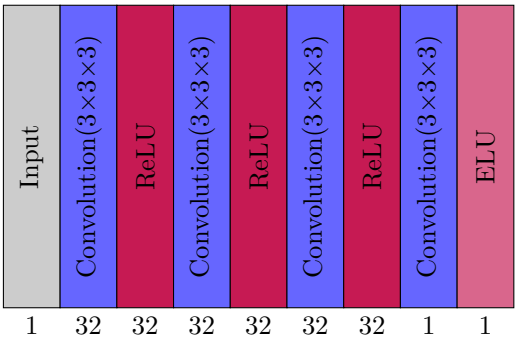


Figure 4.1: Visual representation of a post-processing μ-net M intended to operate in the null-space of a resolution-modelling Gaussian smoothing operation. The layer operations themselves are hidden as per the convention set out in Figure 2.5.

For comparison, a network C (with the same architecture as M) is trained on the same data. NRMSE is also used as the loss for C, i.e.:

$$L(C;\boldsymbol{\theta}) = \sum_n \sqrt{\|\boldsymbol{\theta}_n - C(\mathcal{R}_k(G(\boldsymbol{\theta}_n)))\|^2 / \|\boldsymbol{\theta}_n\|^2}, \tag{4.12}$$

where C is a CNN with the same architecture as M (not a null-net N).

The network C thus performs the same post-processing task as N, but without the consistency constraint of Equation (4.9).

The crucial difference between directly applying a post-processing network C or using N is the latter's residual reformulation. Note that $\mathcal{R}_k(G(M(\hat{\boldsymbol{\theta}}))) - M(\hat{\boldsymbol{\theta}})$ from Equation (4.9) produces solely the estimated Gibbs ringing artefacts in null-space. Therefore the overall effect of N is to remove such artefacts from the reconstructed volume $\hat{\boldsymbol{\theta}}$ while ignoring all other effects and features. This means that the N should be able to achieve very good performance even if unseen (test) data contain significantly different features from the training data.

4.3 Results

4.3.1 Simulations

All 20 *BrainWeb* ground truth phantoms from Section 3.2.1.1 are used in training. Additionally, a 21$^{\text{st}}$ "phantom" is used consisting of noise (uniform random sampling followed by 3D median filter of width 3). The reason for this extra phantom is to augment the features in the training data set, potentially increasing robustness to unseen (test) data. Two *BrainWeb* phantoms are reserved for validation and testing, while the remaining 19 phantoms are used in training (minimising Equation (4.11), using a Gaussian RM kernel with a FWHM of 4.5 mm). Training is terminated when validation loss fails to decrease for 1000 epochs.

Note that the training volumes are noise-free, and in fact are not even particularly PET specific in the sense that no sinogram projections are performed. A central slice of the test subject is shown in Figure 4.2, with metrics including NRMSE (ϵ) and mean structural similarity index (MSSIM) [179] (see Equation (4.17) below) measured against the ground truth.

$$\mu_x = \sum_i w_i x_i, \tag{4.13}$$

$$\sigma_x = \sqrt{\sum_i w_i (x_i - \mu_x)^2}, \tag{4.14}$$

$$\sigma_{x,y} = \sum_i w_i (x_i - \mu_x)(y_i - \mu_y), \tag{4.15}$$

$$\text{SSIM}(\boldsymbol{x}, \boldsymbol{y}) = \frac{\left(2\mu_x\mu_y + (0.01L)^2\right)\left(2\sigma_{x,y} + (0.03L)^2\right)}{\left(\mu_x^2 + \mu_y^2 + (0.01L)^2\right)\left(\sigma_x^2 + \sigma_y^2 + (0.03L)^2\right)}, \tag{4.16}$$

$$\text{MSSIM}(\boldsymbol{\theta}, \boldsymbol{T}) = \frac{1}{J}\sum_j^J \text{SSIM}(\boldsymbol{\theta}_j, \boldsymbol{T}_j). \tag{4.17}$$

where w_i is the i^{th} component of a Gaussian kernel of 4.5 mm FWHM,

μ_x is the weighted mean across $\boldsymbol{x}$,

σ_x is the weighted standard deviation across $\boldsymbol{x}$,

$\sigma_{x,y}$ is weighted covariance of $\boldsymbol{x}$ and $\boldsymbol{y}$,

L is the peak-to-peak (intensity range) of the reference $\boldsymbol{y}$,

$\boldsymbol{\theta}_j$ is the j^{th} region of interest of the input volume (in this case, foreground voxels only, i.e. whole-brain), and

$\boldsymbol{T}_j$ is the corresponding region in the target (reference) volume.

Note that the CNN C output (top row, third column) is designed to directly match the ground truth (top row, first column). The null-space network N output, meanwhile, is designed to match the smoothed versions (i.e. the bottom row, rightmost difference image should be zero).

The naïve CNN reduces NRMSE drastically from 33.2 % to 16.3 %, and increases MSSIM from 0.95 to 0.99. The null-net meanwhile produces a more modest improvement – 25 % and 0.97, respectively. Nevertheless, when forward modelled (Gaussian smoothed) the null-net and pre-processed results are almost identical, as expected. The smoothed CNN result – while slightly closer to the ground truth – is nonetheless not as consistent, as seen in the bias images (bottom row). This indicates a potential issue with robustness – if run on significantly different test data, the CNN results may be poorer.

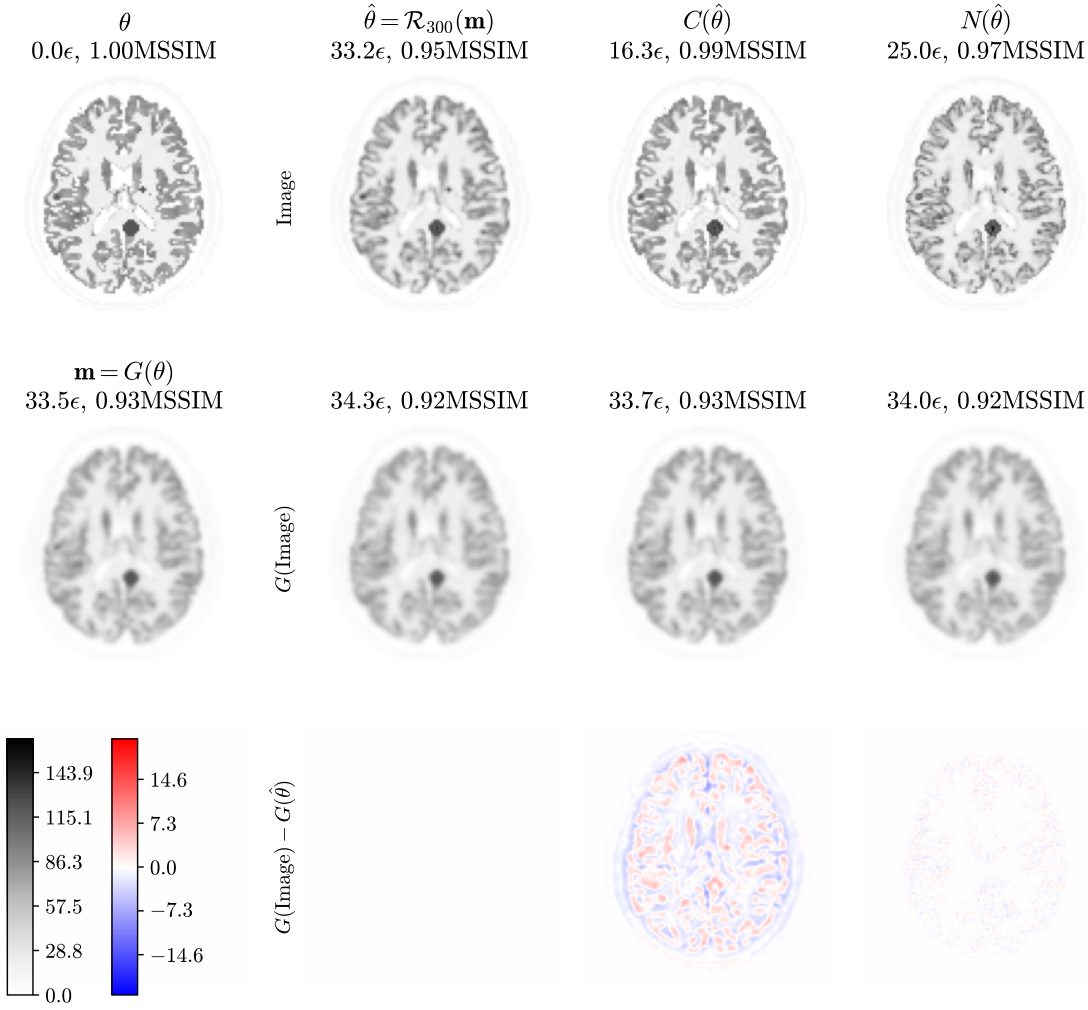


Figure 4.2: Central slice of *BrainWeb* based test subject resolution recovery simulations. Metrics (NRMSE ϵ and MSSIM) are measured against the ground truth foreground (i.e. whole-brain). The ground truth (top row, leftmost) and Gaussian smoothed version (middle row, leftmost) are show in the first column. The top row also includes three methods to recover resolution from the Gaussian smoothed truth: Richardson-Lucy (second column), a post-processing CNN (third column), and null-space network N (fourth column). The middle row shows Gaussian smoothed versions of the top row, while the bottom row shows the difference between these smoothed versions and the smoothed ground truth. The grey scale applies to the top and middle rows, while the blue-red scale applies to the difference images in the bottom row.

An interesting test would be to use a completely different dataset in order to check whether consistency is still apparent. Since in practice for test data the true PSF is not known, this different dataset should also use a different smoothing kernel. For this purpose, the *BigBrain* [^{18}F]FDG-PET phantom [180] is registered with the *BrainWeb* data. Resolution degradation is achieved by Gaussian smoothing with 4.5 mm FWHM. Resolution recovery results for this test phantom are depicted in Figure 4.3. Here, the unconstrained CNN metrics are only slightly better than the null-net N. Nevertheless, the unconstrained network C performs surprisingly well considering the large change in test data.

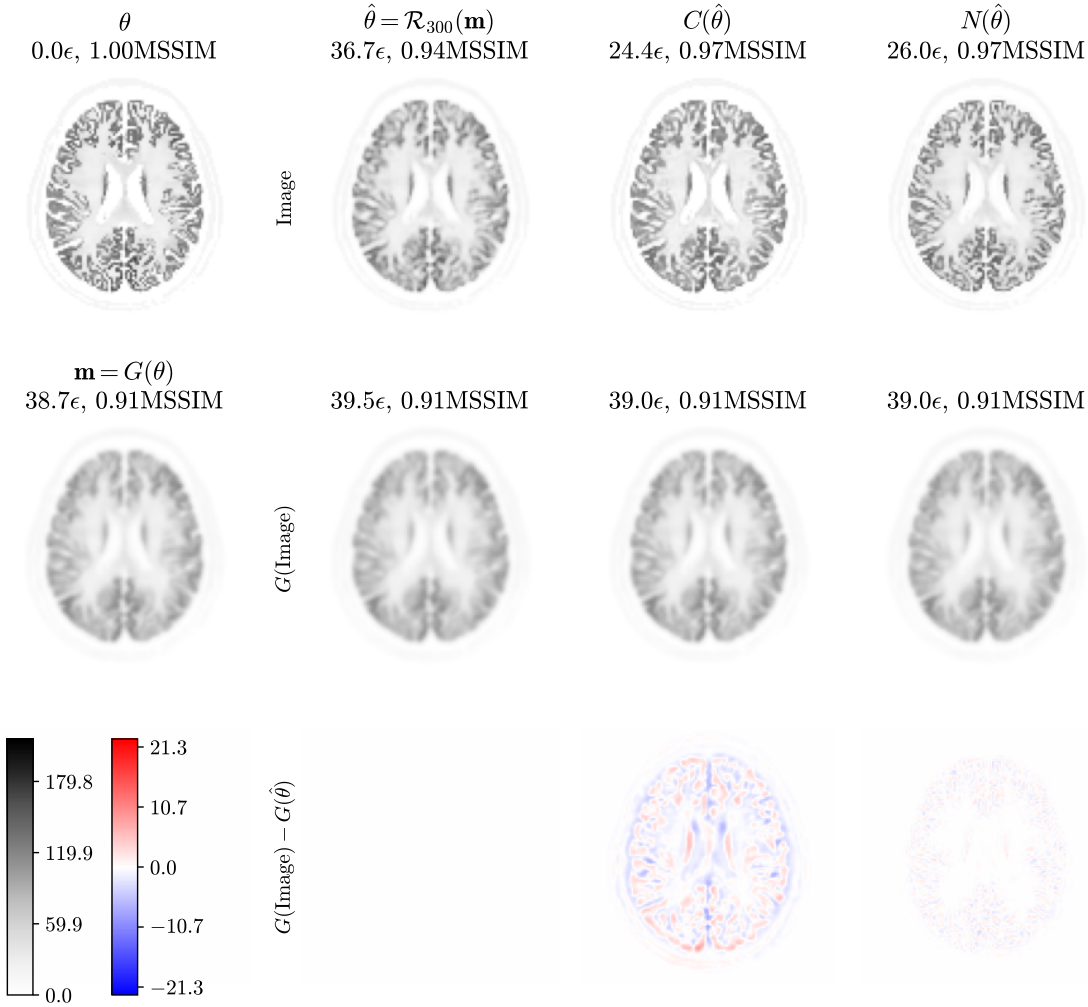


Figure 4.3: Central slice of *Big Brain* based test subject (similar to Figure 4.2).

Finally, full MLEM RM PET simulated reconstructions can be used as an input volumes. These inputs have noise which none of the post-processing methods in this chapter are explicitly trained with. However, at least in the case of the null-net, this noise should not cause a post-processing failure. The results are shown in Figure 4.4 for both 30 M and 300 M count reconstructions. In this case, both C and N produce quantitatively worse results. However, the consistency enforced on N (via Equation (4.9)) constrains these errors.

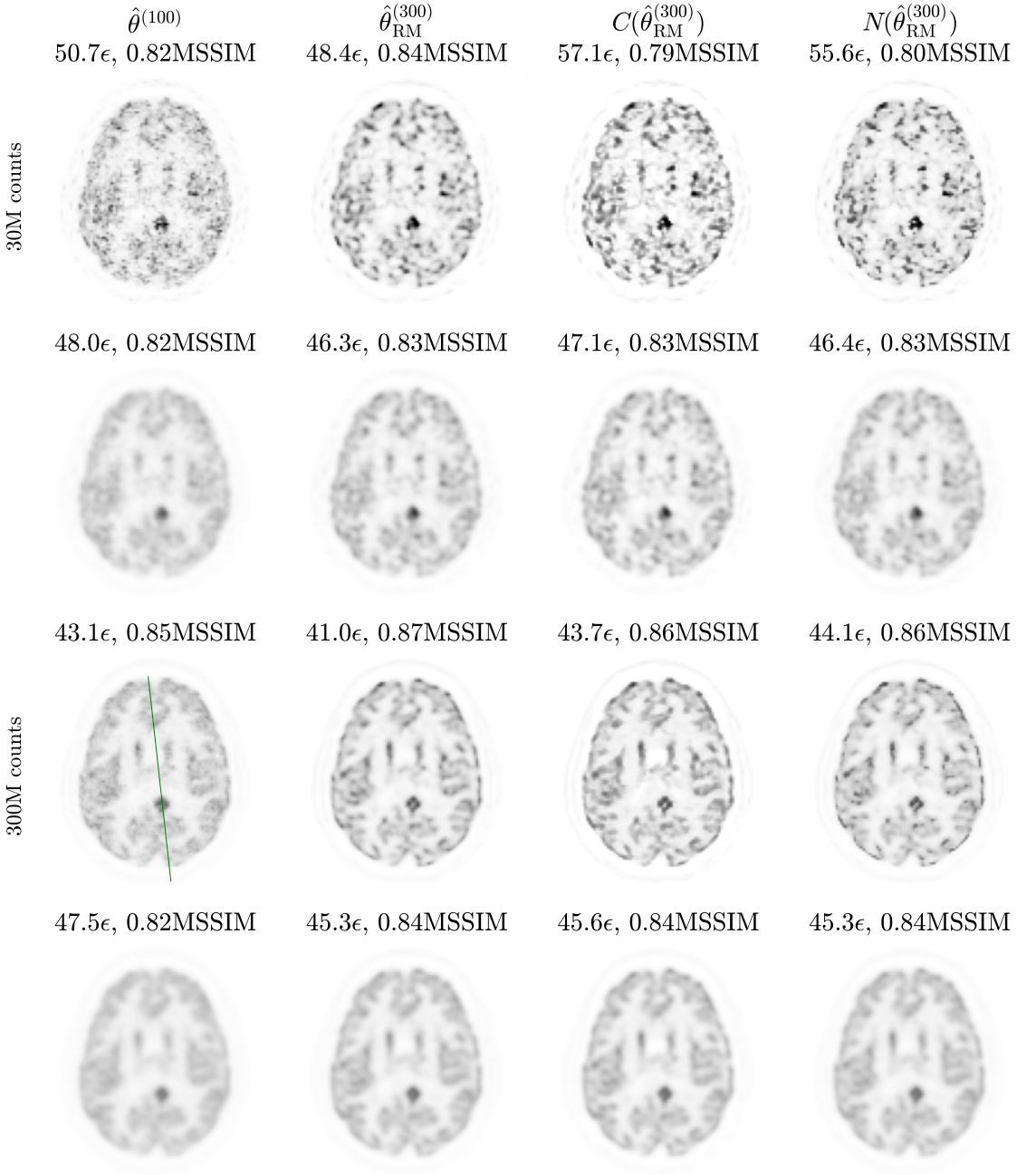


Figure 4.4: Central slice of resolution-modelled PET reconstruction simulation and post-processing. Compare with ground truth and noise-free results in Figure 4.2. Ringing is clearly visible (especially with grey matter hyperintensities) in the resolution-modelled reconstructions regardless of application of a post-processing network.

The CNN increases the whole-brain NRMSE and decreases the MSSIM. Meanwhile, these metrics are barely affected by the null-net for the 300 M count case. Indeed, it is difficult to see much visual difference from the input RM reconstruction. Line profiles through the 300 M count images are shown in Figure 4.5 which make the differences clearer. The CNN clearly has an – often excessive – sharpening effect, while the the null-net approach produces a less aggressive effect, following the input RM reconstruction more closely.

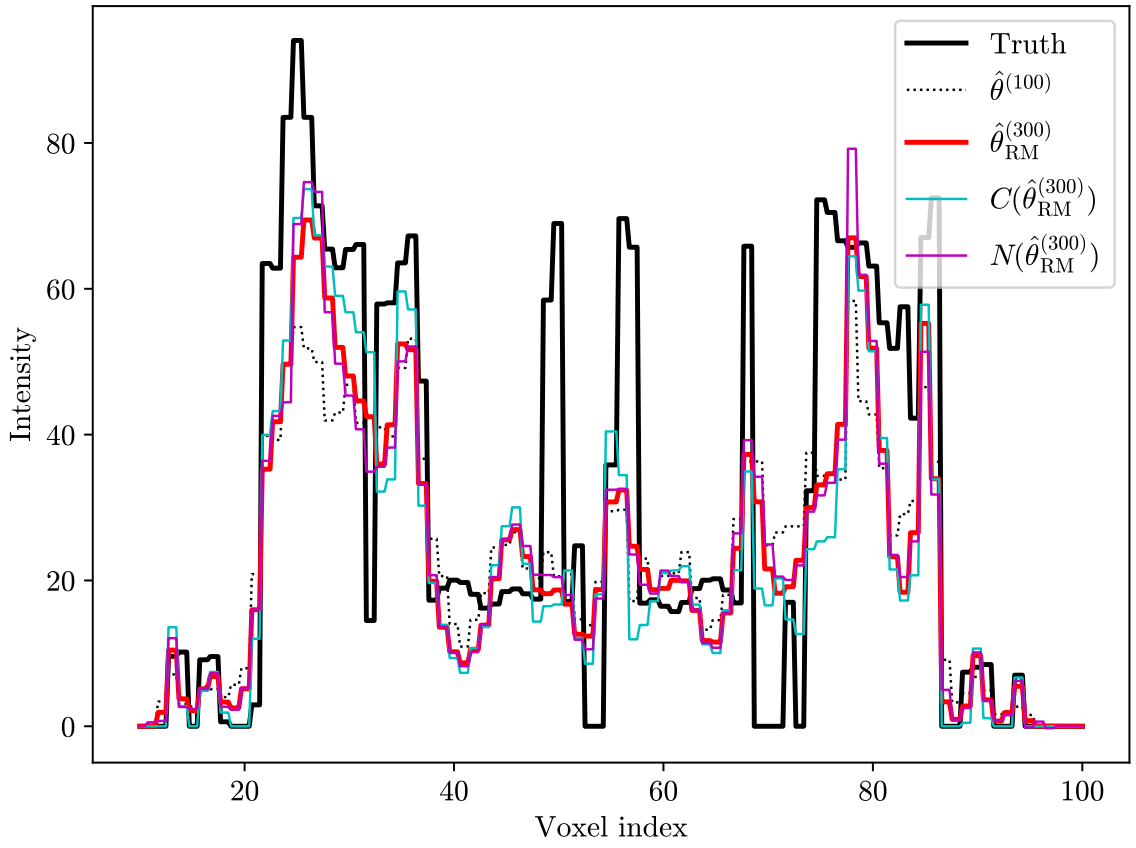


Figure 4.5: Line profiles through 300M count PET MLEM test results from Figure 4.4.

4.3.2 Real Patients

Since N is constrained to operate in the null-space, it is also possible to apply the network (trained on simulations as described above) and test on real data. This is somewhat similar to domain transfer learning, where inconsequential features are ignored (i.e. do not interfere with the task). Real patient data is bootstrap-sampled at 30 M and 300 M counts and reconstructed as described in Section 3.2.1.2. Metrics are measured against the foreground (whole-brain) 3D reconstruction of the raw full (circa 430 M) count scan. Once again, both neural networks degrade results (NRMSE is larger and MSSIM lower in the output [third, C and forth, N] columns compared to the input [second] column) – with the unconstrained C performing worse. Profiles corresponding to the 300 M count results are given in Figure 4.7.

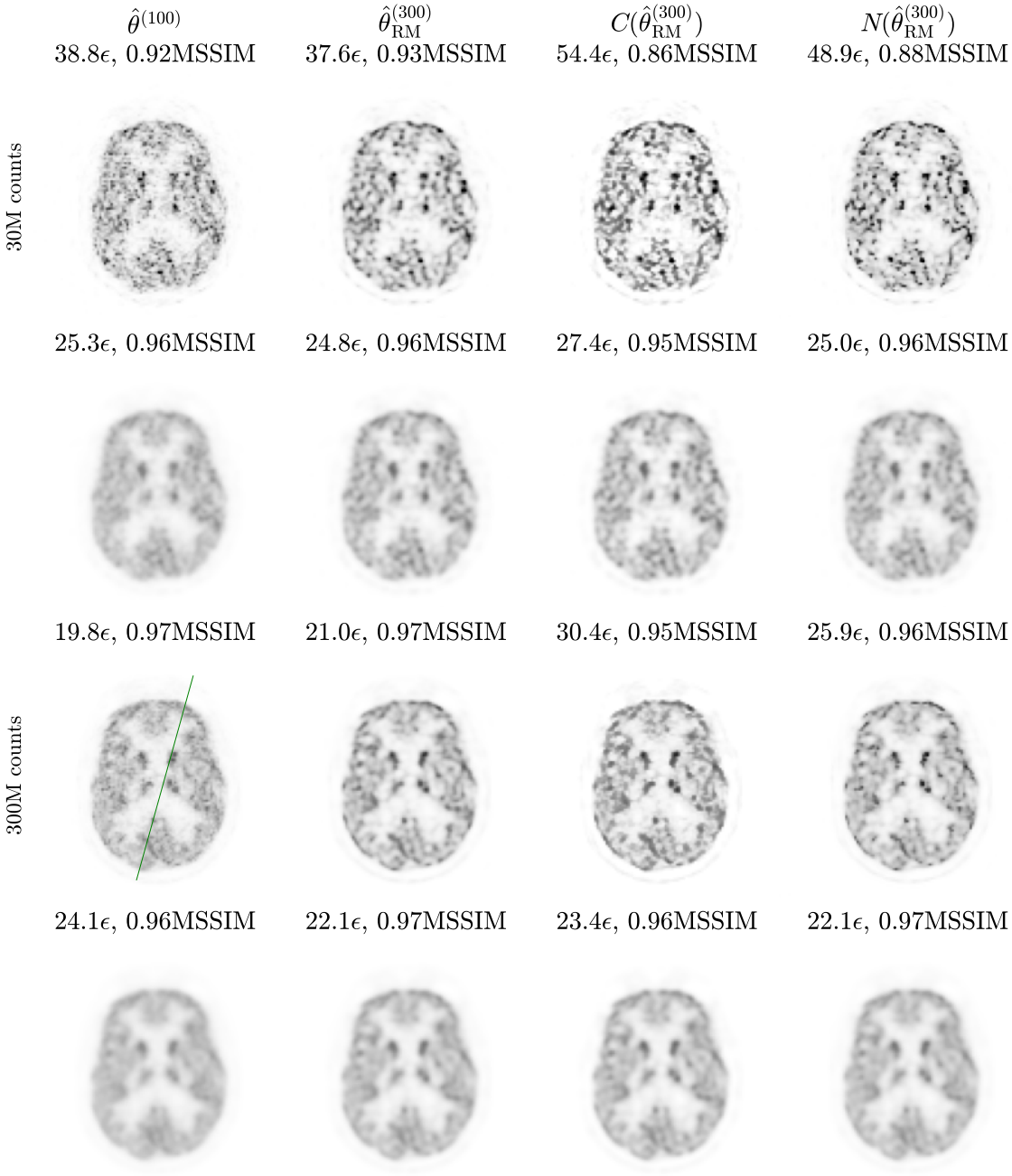


Figure 4.6: Central slice of resolution-modelled clinical PET reconstruction and post-processing. NRMSE (ϵ) and MSSIM are calculated against the full (circa 430M) count reconstruction. As with the analogous simulated phantom results in Figure 4.4, both networks (third and fourth columns) degrade NRMSE and MSSIM compared to the input (second column).

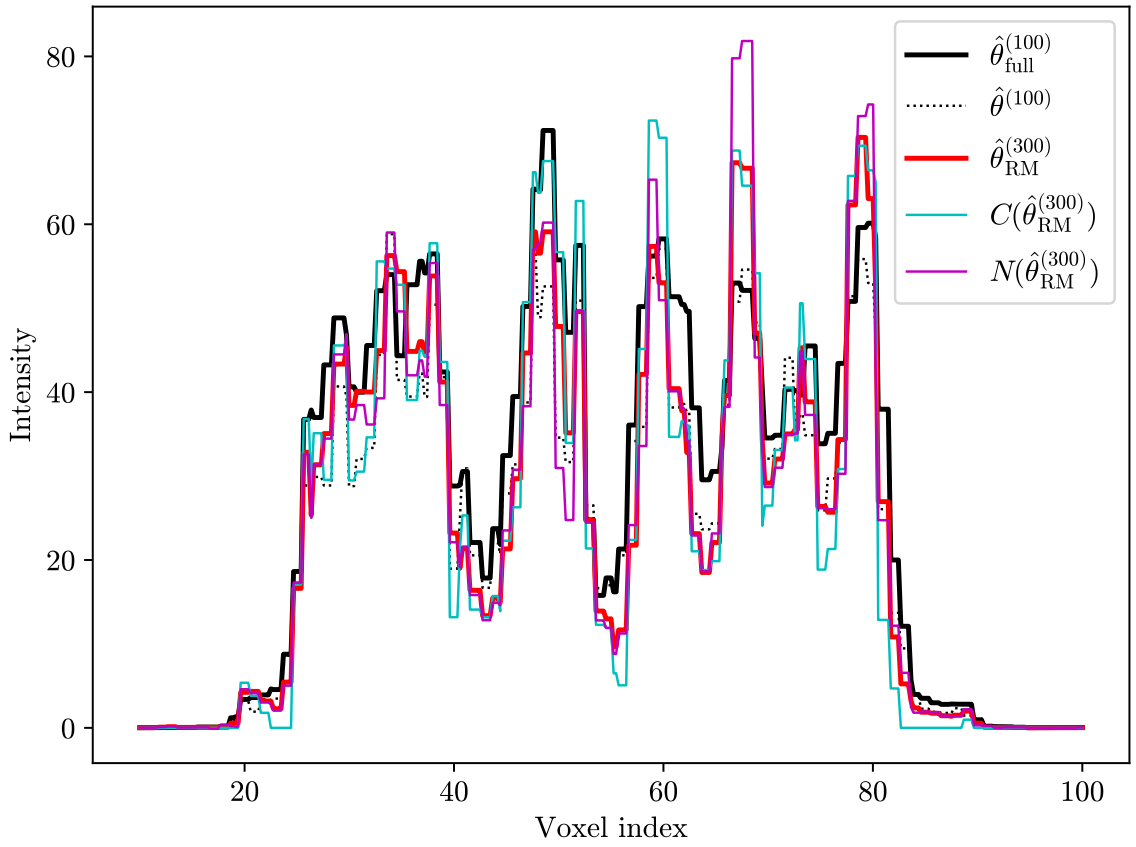


Figure 4.7: Line profiles through 300M count real patient results from Figure 4.6.

4.4 Discussion

The noise-free simulation results on a test dataset similar to the training set (i.e. both based on *BrainWeb*) show that an unconstrained CNN outperforms a constrained null-net (Figure 4.2 indicates lower NRMSE and higher MSSIM – 16.3 % and 0.99 for C compared to 25.0 % and 0.97 for N, respectively). This is surprising since theoretically both approaches should produce the same optimal result. If the optimisation problem is itself difficult, then constraining the search space should help. However, the null-net does not really constrain the search space – instead, the design (a residual network with an MLEM/R-L component) is what imposes a constraint on the overall network output. In fact, backpropagation through this iterative R-L can lead to vanishing or exploding gradients, making optimisation more difficult. Meanwhile when the noise-free test case is based on a different (*BigBrain*) dataset, the null-net performance is relatively unaffected, while the CNN degrades to roughly the same as the null-net (Figure 4.3).

It should however be noted that in all cases, the input and outputs of the null-net are essentially identical when forward-modelled (blurred). When the test data is replaced with MLEM reconstructions of simulated noise realisations, both neural networks degrade their inputs (in terms of qualitative appearance as well as NRMSE and MSSIM). This is fairly unsurprising since neither network was trained on noisy reconstructions. However, there is a limit on the degradation caused by the null-net due to the model consistency constraint. This is found both for the simulated (Figure 4.4) and real clinical (Figure 4.6) reconstructions.

At this stage, neither the CNN nor null-net results appear to be robust or trustworthy enough for clinical application. Nevertheless, the null-net approach – while potentially less powerful than an unconstrained CNN – is far more robust to unseen test data. This guarantee is particularly important in a clinical setting, where incorporating techniques which can sensibly limit the capabilities of "black box" networks will likely be a requirement for approval and adoption in practice. It should be noted that this null-net formulation is a post-processing step which runs an iterative reconstruction again, thereby potentially doubling total reconstruction time.

Future work should investigate incorporating the PET system's projectors into the null-net's iterative reconstruction component. While this will significantly increase training time, the network should cope far better with noisy data and thus may enhance – rather than degrade – its inputs. However, care does need to be taken when using null-nets – for example, consistency with noise is not desirable, so the training data really should always be noise-free. In the context of PET, the applications may be limited to Gibbs ringing RM reconstruction artefact removal.

4.5 Summary

The task of the proposed null-net is to transform MLEM (in this case, just the resolution modelling or R-L component) reconstructions into artefact-free ground truth predictions in a robust manner consistent with the forward model. Consistency means the forward model (smoothing with the system's PSF) should produce identical results whether applied to the input or to the predictions of the null-net.

The null-net demonstrates better generalisability than an unconstrained CNN due to the consistency constraint (as evidenced by the metrics in Figure 4.6 being consistently better for N than C). Nevertheless, both networks degrade their inputs when run on noisy data so are not currently suitable for clinical PET imaging tasks. Future work should consider incorporating the PET projectors into the null-net to potentially obviate this issue. An alternative avenue for progress would be to replace $M(\hat{\boldsymbol{\theta}})$ from Equation (4.9) with any other "classical" regularised reconstruction.

In any case, the demonstrable increase in robustness caused by the null-net formulation may be key to achieving widespread adoption of DL in clinical PET.

Chapter 5

CNNs for Low Count PET Post-Processing: Comparison of Current Approaches

This chapter compares the most promising state-of-the-art CNNs for PET and MR guided PET post-processing methods, focusing primarily on denoising and artefact removal in low count scans. The aim is to find a consensus or at least some patterns suggesting which methods work best. In the current literature, CNNs are frequently proposed without justification of hyperparameters and design choices. Important considerations include number of layers (depth), number of kernels per layer (width), spatial size/resolution/downsampling (height), as well as concatenation (skip) and residual connections.

Most of the current proposals suggest the use of "deep" networks. The term "deep" is frequently used in machine learning literature in subtly different ways, potentially causing confusion. The convention used in this thesis defines DL as a subset of ML. Specifically, DL is distinguished from general ML as it incorporates mostly automatic feature selection with minimal manual (human) intervention [122]. ANNs (and more specifically CNNs) are also subsets of ML, but do not necessarily need to be part of DL. Raw inputs could first be pre-processed and features extracted using traditional and/or manual methods before being fed into a network to produce desired outputs. "Deep networks," meanwhile (not to be confused with "deep learning"), is used here to refer to networks with a "large" number of layers. Since layers can function as automatic feature detectors, extractors, and modifiers, one could argue that in most cases deep networks are also capable of deep learning.

In the context of PET post-processing, however, raw listmode data is first histogrammed (put into sinogram bins, often with some level of compression such as span-11 [44]) before being reconstructed via iterative methods (see Section 1.3). The resultant volumes are potentially post-filtered using – for example – Gaussian smoothing, TV denoising, or guided NLM. Only at this stage is a CNN applied to further improve image quality. Even if the CNN is used to completely replace more traditional post-filtering methods, it is still operating on highly processed (post-reconstruction) data. Therefore, PET post-processing is not viewed as a DL task (regardless of the CNN depth) in this work. Figure 5.1 shows how the PET post-processing methods considered in this chapter relate to machine learning in general.

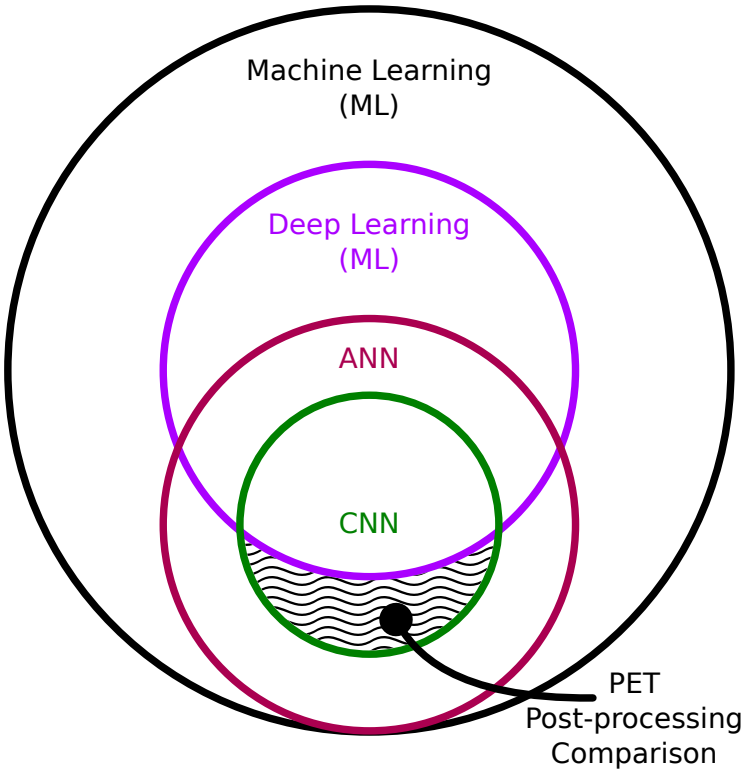


Figure 5.1: Convention used in this chapter for CNN-based PET post-processing in relation to machine learning.

It should be noted that there has been related recent work on genuinely deep learning in PET with end-to-end reconstruction methods such as *AUTOMAP* [140] and *DeepPET* [141] (see Section 2.4.1). Such methods are comparatively fast during test-time since they avoid having to run MLEM iterative reconstructions. However, so far such methods have produced poor results inferior to standard MLEM. While *DeepPET* produced promising results for piecewise-constant phantoms simulations, real patient brain results were clearly unusable in clinical practice [141]. This chapter focuses instead on simpler, more robust post-reconstruction processing as these require less training data, are far quicker to train, and according to the current literature still outperform end-to-end methods in terms of error metrics.

5.1 Motivation

The clinical uses of PET imaging were highlighted in Chapter 1, including the benefits of low count scans (such as increased safety, increased patient throughput, reduced cost, and shorter dynamic frame times). The pitfalls of low count PET are an increase in noise as well as zero trapping effects (see Section 1.3.1) and thus decrease in image quality and clinical utility. In such cases, image post-processing (detailed in Section 1.4) becomes especially important. A simple solution is post-smoothing, sacrificing resolution to reduce apparent noise (diminishing voxel variance while increasing intervoxel covariance) [56]. Alternatively as discussed in Section 2.3, CNNs are particularly well-suited to this image processing tasks. Notably, jointly acquired images from another modality (such as MR or CT) can easily be incorporated into a multi-channel input for a CNN. In the literature discussed in this chapter, the injection of higher resolution anatomical information has led to demonstrable improvements in the quality of network outputs.

However, it is generally accepted that there is no formula for designing a neural network (and proposals to automate architecture engineering are often prohibitively time-consuming [147]). This is evidence by the existence of multiple competing proposals for low count PET. Important differences include hyperparameters such as depth (number of layers), width (kernels per layer), and height (layer output spatial dimensions); as well as overall architecture choices (such as inclusion of residual connections, concatenation, and downsampling). The optimal choices for each of these considerations are very problem-specific. However, given the specific task of PET image denoising, it should be possible to devise at least a rough guide or set of principles to follow. Networks architectures can be designed to solve ordinary differential equations (ODEs) using a specific well-known scheme (such as Euler, Runge-Kutta or leap frog) [181]. It is therefore reasonable to expect that a network architecture could also be designed to solve a specific denoising problem.

A further complication is that PET denoising is considerably different from other denoising imaging tasks. PET images have a much lower SNR than, for example, natural images acquired with a consumer digital camera. Proposals based on other denoising tasks are thus unlikely to work reliably in a clinical PET context without major modification. Small hyper-intense (and hypo-intense) regions can easily be mistaken as noise and thus removed by DL methods. However, in PET such regions often correspond to lesions, and are thus the most important part of the image – precisely what should *not* be removed. Networks should be designed to be robust against this difficult issue before they can be considered for use in the clinic. This is especially problematic as lesions might also be PET-unique (e.g. appearing only in PET images but not in MR), meaning other input channels derived from other modalities might not assist in their detection and recovery.

Furthermore, the noise properties of low count PET images are so different that even commonly used image quality metrics fail to appropriately assess quality. For example, the frequently-used PSNR is a metric which by definition ranks extremely noisy images as being better than the ground truth (as evidenced by Figure 3.1). Part of a thorough comparison thus involves choosing appropriate metrics.

In recent related work in DL-informed reconstruction of MR and CT, it has been shown that one can design nearly undetectable perturbations in a network's inputs in order to produce catastrophic artefacts in the outputs [125].

While not investigated, the researchers hypothesised that such perturbations have a small yet significant probability of occurring naturally in real test data. However, others have shown that denoising autoencoders can grossly emphasised hitherto effectively undetectable artefacts when the inputs have structured noise, as is the case in fluorescent microscopy [182]. The authors in [125] further hypothesise that networks need to be re-trained on each required specific task such as subsampling patterns and ratios (broadly analogous to PET scanner geometry and count levels). This chapter focuses on principles for choosing a network architecture for a specific task (rather than robustness or re-training for a wider variety of tasks).

This chapter compares the most promising state-of-the-art proposals in the current literature for low count PET. The effect of providing other input modalities – in particular, MR images – is also investigated here. The aim is to ascertain what architectural choices and hyperparameters are best suited to this task, as well as suggest avenues for future improvement. Additionally, in order to fill in unexplored gaps in the current literature, new networks are also created and analysed.

5.1.1 Existing Comparisons

For low count PET post-processing, others have compared a U-net's performance on [^{18}F]FDG lung reconstructions before and after skip connections are removed (the latter resulting in a CAE) [146]. They also compare the effect of adding a GAN component during the U-net's training. Results were shown to be mixed. All architectures demonstrated comparable MSE, while simpler architectures had better SNRs. However, standardised uptake values (SUVs) were shown to be superior for more complex architectures. However, the comparison did not investigate performance against amount of training data, nor was there consideration of any simulation data for comprehensive evaluation metrics. Appending a CT input channel was reported to have either no impact or a slight improvement depending on the architecture. Interestingly, the authors further concluded that different architectures were required depending on whether quantitative accuracy or visual quality was sought. A key omission in [146] was the effect of varying network depth (number of layers). Additionally, on a fundamental level the primary comparison was between a heavily modified U-net loosely based on [123] and a CAE arbitrarily chosen to have half as many convolutional layers. Considering this, it is very interesting that the CAE managed similar performance at all. A number of additional shortcomings of the earlier comparison are addressed here, namely:

- analysis using MSSIM (defined in Equation (4.17)) as a more appropriate metric:
  - MSSIM is quantitative yet also corresponds well to visual quality,
  - is designed to overcome the shortcomings of NRMSE [179], and
  - is also robust against noise, unlike SUV and PSNR;
- investigation of multiple architectures based on the current literature;
- investigation of a wider range of architectures (filling in gaps in the literature) including a thorough investigation on the effect of:
  - varying depth (number of layers),
  - overall residual connections,
  - intermediate concatenation (skip) connections, and
  - downsampling;
- use of distinct training, validation, and test datasets;
- use of brain images (which are more difficult improve than lung images [141]);
- use of simulation data for more comprehensive analysis against known ground truths (in addition to real data).

Regarding architecture, it has also been demonstrated in [135] that – for an MLAA task – a U-net's error decreases when skip/concatenations are removed. Meanwhile, for classification and segmentation tasks, comparisons have included cases with the maximum possible skip (concatenation) connections – called *DenseNets* [183]. Each layer in these networks receive (as input) the outputs of all preceding layers. The term "dense" in this sense is unfortunately confusing since "dense" is a synonym for "fully connected." Both terms are usually used to refer to a type of layer, where each output voxel is connected to all input voxels. *DenseNets*, meanwhile, have all layers connected to all previous layers. Recently, Dolz et. al proposed a *HyperDense-Net* for multi-modal MR segmentation [184], which augments multiple pathways (two separate DenseNets) with inter-pathway connections.

It should be noted that the DL proposals considered below are all supervised methods. Unsupervised (or self-supervised) alternatives such as Noise2Noise [126], Noise2Void [182] and DIP [115], [116], [128] approaches are not compared here. This is because supervised methods outperform unsupervised methods – albeit with the caveat that higher quality target training data is required.

5.2 Methods

Six different DL proposals are considered, summarised in Table 5.1 and available at [185]. Additionally for reference, varying levels of PS and MR-guided NLM filtering are also used for comparison.

Note that each network investigated here is also given distinct validation and test datasets, whereas the original proposals frequently lack either validation or test data. Where feasible (i.e. training time is low enough), networks are also re-trained multiple times on the same data in order to help ensure that results are not affected by poor random initialisation. This also naturally allows for estimation of the network's variability (results can be shown on a bias-standard deviation graph).

Regarding loss functions, number of epochs and learning rate scheduler, each network should be trained as described in the originating literature. However, since these hyperparameters are likely specifically optimised to the amount and type of data originally provided, slight modifications are made to improve performance on the data used here. Since more training data is available here, the number of training epochs are appropriately increased. Furthermore, in the last phase of the training (from the beginning if there is only one phase) restrictions on the number of epochs are completely removed. Instead, the validation data is used to determine when to terminate training. This should result in all networks considered performing at least as well as shown in the original proposals.

Short Name	μNet	ResUNet-C	ResUNet-X	ResUGAN-TV
Trainable parameters	10.9 k	487 k	1.95 M	906 k
" (inference)	10.9 k	487 k	1.95 M	888 k
Unique training voxels			812 M	
Voxels per parameter	74.4 k	1.67 k	417	896
" (inference)	74.4 k	1.67 k	417	913
Count upgrade factor			10	
Loss function	$\sqrt{\ell_2}$	ℓ_1	ℓ_1	ℓ_2; $\ell_2\nabla$; TV; adversarial
Input modalities			FDG-PET; T1	

Short Name	μNet-P	LA-UGAN-AC
Trainable parameters	42.2 k	64.0 M
" (inference)	42.2 k	58.5 M
Unique training voxels	812 M	
Voxels per parameter	19.2 k	12.7
" (inference)	19.2 k	13.9
Count upgrade factor	10	
Loss function	$\sqrt{\ell_2}$	ℓ_1; adversarial
Input modalities	FDG-PET; T1; (FDG-PET·T1; NLM)	FDG-PET; T1

Table 5.1: Overview of networks implemented for comparison here based on the current literature. "Inference" means only including prediction parameters (i.e. excluding any discriminator parameters which are used exclusively during training). Input modalities in parentheses are deterministically computed from the other inputs (i.e. FDG-PET·T1 and NLM are both pre-computed from FDG-PET and T1). "Unique training voxels" means the size of the input dataset before any such deterministic data augmentations and pre-processing. "Count upgrade factor" is the ratio of acquired counts between the training input and target data.

Short Name	µNet	ResUNet-C	ResUNet-X	ResUGAN-TV
Trainable parameters	43.7 k	487 k	1.95 M	905 k
" (inference)	43.7 k	487 k	1.95 M	888 k
Unique training voxels	120 M	747 M	420 M	36.1 M
Voxels per parameter	2.75 k	1.53 k	216	39.9
" (inference)	2.75 k	1.53 k	216	40.6
Count upgrade factor	10; 100	100	200	10
Input modalities	FDG-PET; $\text{FDG-PET}_{\text{RM}}$; T1; (NLM)	FBB-PET; T1; T2; T2-FLAIR	FDG-PET; T1; T2-FLAIR	FDG-PET

Short Name	µNet-P	LA-UGAN-AC
Trainable parameters	139 k	256 M
" (inference)	139 k	234 M
Unique training voxels	120 M	571 M sim; 902 M real
Voxels per parameter	866	2.23 sim; 3.52 real
" (inference)	866	2.44 sim; 3.86 real
Count upgrade factor	10; 100	4
Input modalities	FDG-PET; $\text{FDG-PET}_{\text{RM}}$; T1; (NLM)	FDG-PET; T1; FA-DTI; MD-DTI

Table 5.2: Overview of original network proposals from the current literature. Discrepancies from Table 5.1 are primarily due to either dimensionality (original proposals in 3D rather than 2D) or input channels (different input modalities other than PET and T1-weighted MR).

Note that all implementations here (Table 5.1) have more unique training data voxels per trainable network parameter (referring to the row marked "Voxels per parameter") than the original proposals (*µNet* [164], *ResUNet-C* [186], *ResUNet-X* [187], *ResUGAN-TV* [166], *µNet-P* [142], [164], and *LA-UGAN-AC* [170], as summarised in Table 5.2). This means that performance of the comparison implementations should match or exceed the original publications. Note that "unique" is used to mean excluding affine augmentations. The other main differences from the original proposals are spatial dimensions (2D versus 3D) and input modalities. For a fair comparison, all implementations here are in 2D and based on PET and T1-weighted MR input channels. All networks thus have the same opportunity to find patterns and correlations between the two dimensions in order to aid with feature detection and post-processing. It seems unlikely that adding a third spatial dimension and more input modalities would change the relative performance (i.e. comparative ranking) of these networks, though this could be investigated in future work.

All other details are faithfully implemented as per the original proposals. This includes choice of loss function(s), network depth, number and width of kernels in each layer, training regimes and regularisation methods. Specific details for each method are given below. The implementations of all methods are available for download at [185].

5.2.1 μNet

A modified version of the *μNet* from Chapter 3 (specifically, [164]) is shown in Figure 5.2. The architectural differences from the original proposal are summarised in Table 5.3, while the differences in training data are summarised in Table 5.4. These changes are not under investigation here, and are made for the purposes of conducting a fair comparison to the other rival methods described in the following sections.

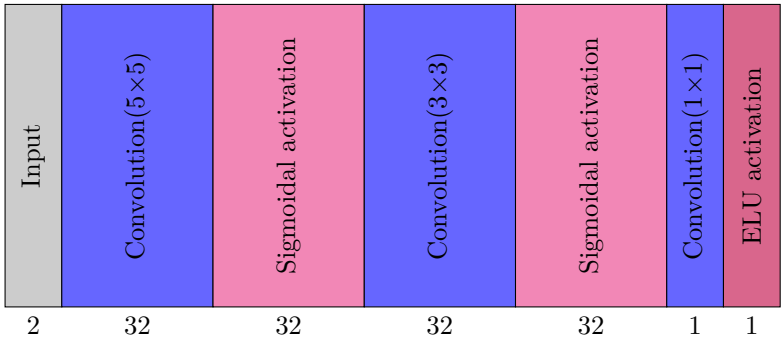


Figure 5.2: Visual representation of *μNet*: an MR-guided PET denoising architecture. Each block represents the output of a layer, with the number below each block signifying the number of output channels. The layer operations themselves are hidden as per the convention set out in Figure 2.5.

Property	Implemented	Original
Input channels	2	4
Input spatial dimensions	128×128	$344 \times 344 \times 127$

Table 5.3: Differences between *μNet* architecture implementation here and the original proposal.

Property	Implemented	Original
Input channels	FDG-PET; T1	FDG-PET; T1; $\text{FDG-PET}_{\text{RM}}$; (NLM)
Input image width	268 mm	719 mm
OSEM iterations × subsets	7×14	100×1; 300×1
Count upgrade factor	10	10; 100
Training data	30 simulations; 30 real patients	2 simulations; 2 real patients
Validation data	3 simulations; 3 real patients	3 simulations; 3 real patients
Test data	27 simulations; 27 real patients	24 simulations; 24 real patients

Table 5.4: Differences between data used for *μNet* training here versus the original proposal.

The network training uses the Adam optimiser with a learning rate of 10^{-3} to minimise an NRMSE loss function. There is no fixed maximum number of epochs, and training is terminated if the validation loss fails to decrease for over 10 000 epochs.

5.2.2 μNet-P

Another modified version of a *μNet* from Chapter 3 (in this case based on both [142] and [164]) is shown in Figure 5.3. Instead of the first convolutional layer directly operating on the just the PET and MR input channels, two additional channels are also provided: a simple element-wise product between the PET and MR, as well as an MR-guided NLM filtered version of the PET channel.

The architectural differences from the original proposal are summarised in Table 5.5, while the differences in training data are summarised in Table 5.6. These changes are not under investigation here, and are made for the purposes of conducting a fair comparison to the other rival methods described in the following sections. Note that the extra input channels compared to the other comparison networks (2 more) are both deterministically computed based on the other inputs. This pre-computation could be implemented as a non-trainable layer just before the first convolutional layer. This means the overall network effectively requires the same 2 input channels as the other networks considered in this chapter – making the comparison a fair one.

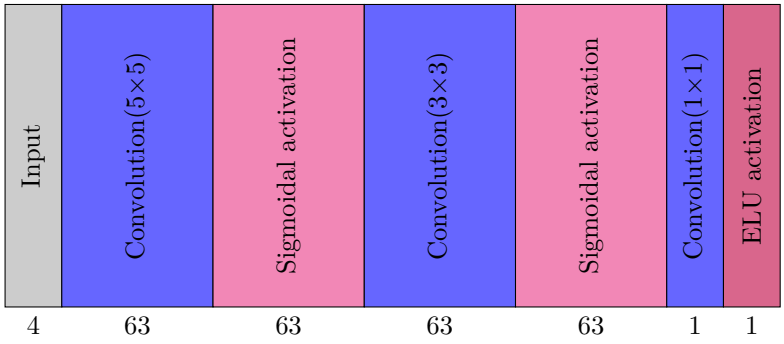


Figure 5.3: Visual representation of *μNet-P*: an MR-guided PET denoising architecture. Similar to Figure 5.2 albeit with additional pre-processing to augment the input channels, and wider hidden layers.

Property	Implemented	Original
Input spatial dimensions	128×128	$344 \times 344 \times 127$

Table 5.5: Differences between *μNet-P* architecture implementation and original proposal.

Property	Implemented	Original
Input channels	FDG-PET; T1; (FDG-PET·T1; NLM)	FDG-PET; T1; $\text{FDG-PET}_{\text{RM}}$; (NLM)
Input image width	268 mm	719 mm
OSEM iterations × subsets	7×14	100×1; 300×1
Count upgrade factor	10	10; 100
Training data	30 simulations; 30 real patients	2 simulations; 2 real patients
Validation data	3 simulations; 3 real patients	3 simulations; 3 real patients
Test data	27 simulations; 27 real patients	24 simulations; 24 real patients

Table 5.6: Differences between data used for *μNet-P* training versus the original proposal.

As before, training also uses the Adam optimiser with the same learning rate of 10^{-3} to minimise an NRMSE loss function, and is terminated when the validation loss fails to decrease for over 10 000 epochs.

5.2.3 ResUNet-C

This architecture is taken from Chen at al [186]. It is a U-net (i.e. incorporates downsampling and upsampling with "skip" connections between layers of similar spatial resolution). The network consists of 15 Conv-BN-ReLU blocks interspersed with 3 pairs of maximum-value pooling (MaxPool) (downsampling) and linear interpolation (lerp) (upsampling) operations. Network width (i.e. number of convolutional kernels and thus channels) doubles as sampling halves, varying from a width of 16 to 128. A visualisation of the architecture is given in Figure 5.4.

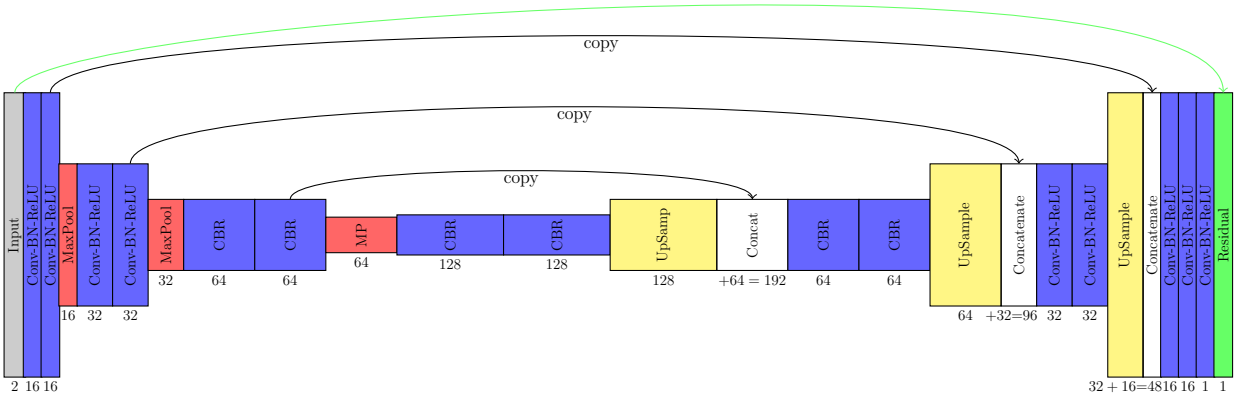


Figure 5.4: Visual representation of *ResUNet-C*: an MR-guided PET denoising residual U-Net architecture. The numbers below each layer signify the number of output channels of the layer. Conv-BN-ReLU (CBR) layers are combined into a single block for ease of representation. Convolutions use kernel width 3, while pooling and (bilinear) upsampling use a scale factor of 2. The final residual layer performs element-wise addition between the last layer and the input PET channel.

There are a couple of minor differences in the implementation used here (as depicted in Figure 5.4) compared to the original proposal. These alterations are made for the purposes of conducting a fair comparison to rival methods, and are summarised in Table 5.7. Furthermore, differences in the data used are summarised in Table 5.8.

Property	Implemented	Original
Input channels	2	4
Input spatial dimensions	128×128	256×256
Training epochs	-	100
Validation-based training epoch tolerance	200	-

Table 5.7: Differences between *ResUNet-C* architecture implementation and original proposal.

Property	Implemented	Original
Input channels	FDG-PET; T1	FBB-PET; T1; T2; T2-FLAIR
Input voxel width	2.09 mm	1.17 mm
Input image width	268 mm	300 mm
OSEM iterations × subsets	7×14	2×28
Count upgrade factor	10	100
Training data	30 simulations; 30 real patients	32 real patients
Validation data	3 simulations; 3 real patients	8 real patients
Test data	27 simulations; 27 real patients	-

Table 5.8: Differences between data used for *ResUNet-C* training versus the original proposal.

The network training in this case uses an Adam optimiser with a learning rate of $2{\times}10^{-4}$ to minimise an ℓ_1 loss function, and training is terminated if the validation loss fails to decrease for over 200 epochs.

5.2.4 ResUNet-X

This architecture is also a U-net, taken from Xu at al [187]. It is heavily inspired by the previous network (*ResUNet-C*), but incorporates wider layers (double the number of kernels) and was proposed for the specific case of [^{18}F]FDG (rather than [^{18}F]FBB) PET-MR, thus making it more relevant here. This more recent proposal also uses mean-value pooling (MeanPool) instead of MaxPool; leaky rectified linear unit (LReLU) instead of ReLU, and includes residual connections around each pair of Conv-BN-LReLU. A visualisation of the architecture is given in Figure 5.5.

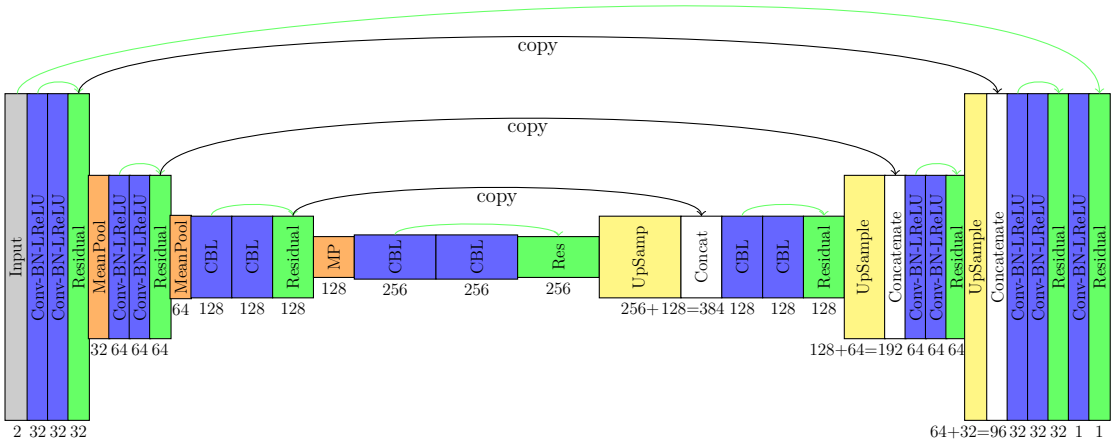


Figure 5.5: Visual representation of *ResUNet-X*: an MR-guided PET denoising residual U-Net architecture inspired by *ResUNet-C* (Figure 5.4). Conv-BN-LReLU (CBL) layers are combined into a single block (and Residual is abbreviated to Res) for ease of representation. Changes include doubling the number of kernels, adding more residual connections, and replacing MaxPool with MeanPool downsampling.

There are minor differences in the implementation used here (as depicted in Figure 5.5) compared to the original proposal. Once again, these alterations are made for the purposes of conducting a fair comparison to rival methods, and are summarised in Table 5.9. Furthermore, differences in the data used are summarised in Table 5.10. The original implementation called for spatially adjacent slices to be represented in the channel dimension (so-called "2.5D"). One adjacent slice on either side was demonstrated to be superior (in terms of NRMSE, SSIM, and PSNR) to 2D. Two or more adjacent slices did not result in further significant improvements.

The network training initially uses the root mean square propagation (RMSProp) optimiser with a learning rate of 10^{-3} gradually decaying to $2.5{\times}10^{-4}$ over 120 epochs. An ℓ_1 loss function is used. Once again, no hard limit is set on the maximum epochs, and training is instead terminated when the validation loss fails to decrease for over 1200 epochs.

Property	Implemented	Original
Input channels	2	3
Input spatial dimensions	128×128	$256 \times 256 \times 3$
Training epochs	-	120
Validation-based epoch tolerance	1 200	-

Table 5.9: Differences between *ResUNet-X* architecture implementation and original proposal.

Property	Implemented	Original
Input channels	FDG-PET; T1	FDG-PET; T1; T2-FLAIR
Input voxel width	2.09 mm	1.17 mm
Input image width	268 mm	300 mm
OSEM iterations × subsets	7×14	2×28
Count upgrade factor	10	200
Training data	30 simulations; 30 real patients	21 real patients
Validation data	3 simulations; 3 real patients	3 real patients
Test data	27 simulations; 27 real patients	-

Table 5.10: Differences between data used for *ResUNet-X* training versus the original proposal.

5.2.5 ResUGAN-TV

This architecture is loosely based on a U-net, taken from Kaplan and Zhu [166], using residual layers instead of concatenations. The network training regime also makes use of a discriminator network. A visualisation of the architecture is given in Figure 5.6.

There are some differences in the implementation used here (as depicted in Figure 5.6) compared to the original proposal. Once again, these alterations are made for the purposes of conducting a fair comparison to rival methods, and are summarised in Table 5.11. Furthermore, differences in the data used are summarised in Table 5.12. The original implementation called for PET-only data (without MR inputs), and worked on 16×16 patches with an overlap of 2. The modified version for comparison operates directly on full 128×128 slices, and thus the discriminator requires a final averaging (MeanPool) layer to yield a single probability of the slice being a desired full count (not generated) image.

Property	Implemented	Original
Input channels	2	1
Input spatial dimensions	128×128	16×16
Extra discriminator layer	MeanPool	-
Training epochs (GAN)	100 (-)	100 (unspecified)
Validation-based epoch tolerance (GAN)	0 (1 000)	-
ℓ_1-regularisation parameter weight	10^-3	unspecified

Table 5.11: Differences between *ResUGAN-TV* architecture implementation and original proposal.

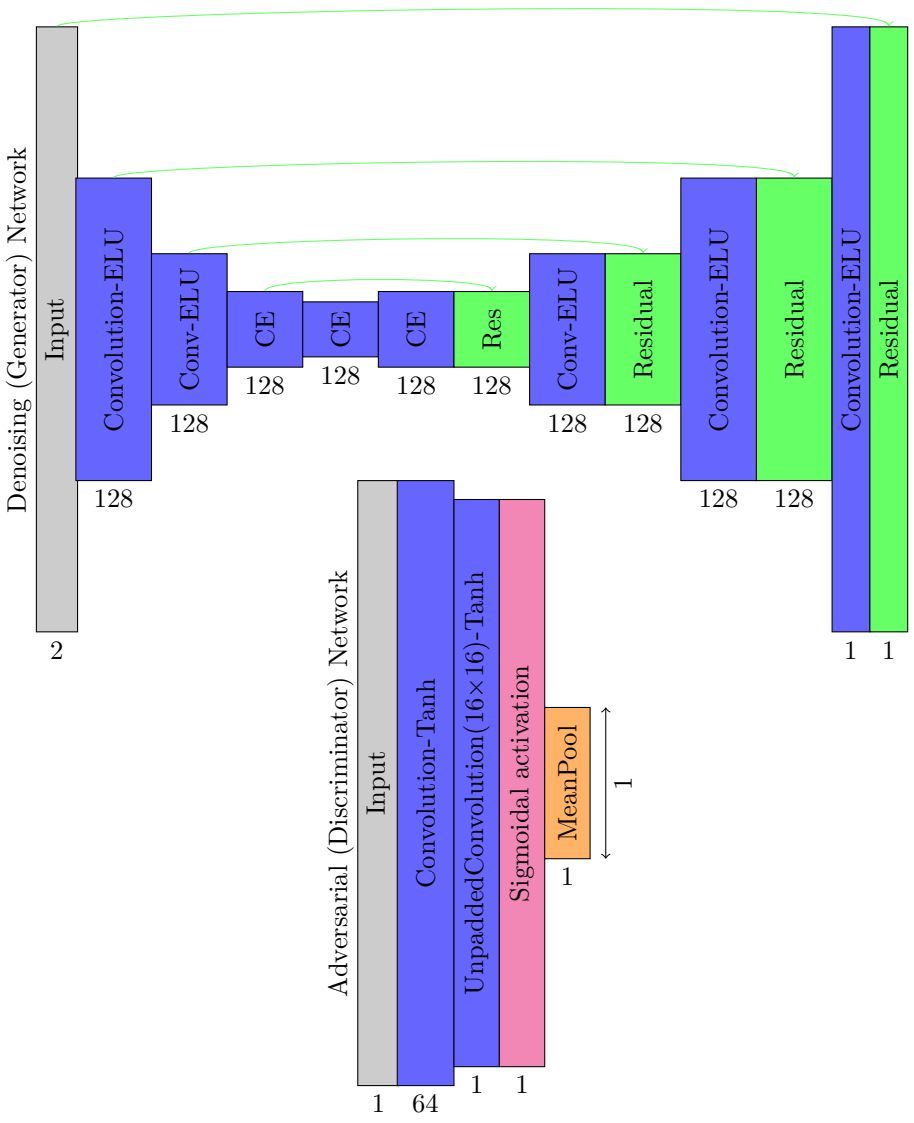


Figure 5.6: Visual representation of *ResUGAN-TV*: a PET denoising residual U-Net architecture incorporating a discriminator (generative adversarial) network (GAN) and total variation (TV) denoising in its loss function. Conv-ELU (CE) layers are combined into a single block for ease of representation. Convolutions are stride 2 (or 1/2) for the denoising (generator) network, and stride 1 for the adversarial (discriminator). All convolutions use zero-padding and kerenls of shape 3×3, except for the unpadded 16×16 layer in the discriminator.

Property	Implemented	Original
Input channels	FDG-PET; T1	FDG-PET
Input voxel width	2.09 mm	2 mm
Input image width	268 mm	32 mm
OSEM iterations × subsets	7 × 14	likely 3 × 17
Training data	30 simulations; 30 real patients	1 real patient
Validation data	3 simulations; 3 real patients	-
Test data	27 simulations; 27 real patients	1 real patient

Table 5.12: Differences between data used for *ResUGAN-TV* training versus the original proposal.

Notably, the loss function includes a gradient term and a total variational (TV) denoising term in addition to the adversarial loss, as shown in Equation (5.1). The empirically chosen constants in the equation are provided in the original work [166]. The network training initially uses the Adam optimiser with a learning rate of 10^{-3} for 100 epochs to minimise the loss function without the discriminator term. The learning rate is then reduced by a factor of 10 and the discriminator is trained alongside until the validation loss fails to decrease for over 1000 epochs.

$$L(\hat{\boldsymbol{\theta}}, \boldsymbol{T}) = \mathrm{MS}(\hat{\boldsymbol{\theta}} - \boldsymbol{T}) - 5{\times}10^{-5}\,\mathrm{MS}(\nabla\hat{\boldsymbol{\theta}}) + 0.075\,\mathrm{MS}(\nabla\hat{\boldsymbol{\theta}} - \nabla\boldsymbol{T}) - 0.1\,\mathrm{BCE}(D(\hat{\boldsymbol{\theta}}), \mathbf{1}), \tag{5.1}$$

where $\hat{\boldsymbol{\theta}}$ is the output of the denoising (generator) network,

$\boldsymbol{T}$ is the target full count image,

$D(\cdot)$ represents application of the discriminator network,

MS is the mean squared, and

BCE is binary cross-entropy.

5.2.6 LA-UGAN-AC

This architecture is also loosely based on a U-net, taken from Wang et al. [170], using convolution stride to control downsampling and upsampling. The network training regime also makes use of a discriminator network. Once trained, the denoising network's output is then concatenated with its inputs to form an augmented input for a second denoising network. This proposed network cascading is called "auto-context" (AC) by Wang et al. The denoising networks also use a "locality adaptive" (LA) first layer. The LA layer is simply a 1×1 convolutional layer with a single output channel. What makes the LA layer unique is that it uses 256 different kernels, each assigned to a distinct 8×8 spatial region of the input (rather than a single kernel operating over the entire 128×128 input). A visualisation of the architecture is given in Figure 5.6.

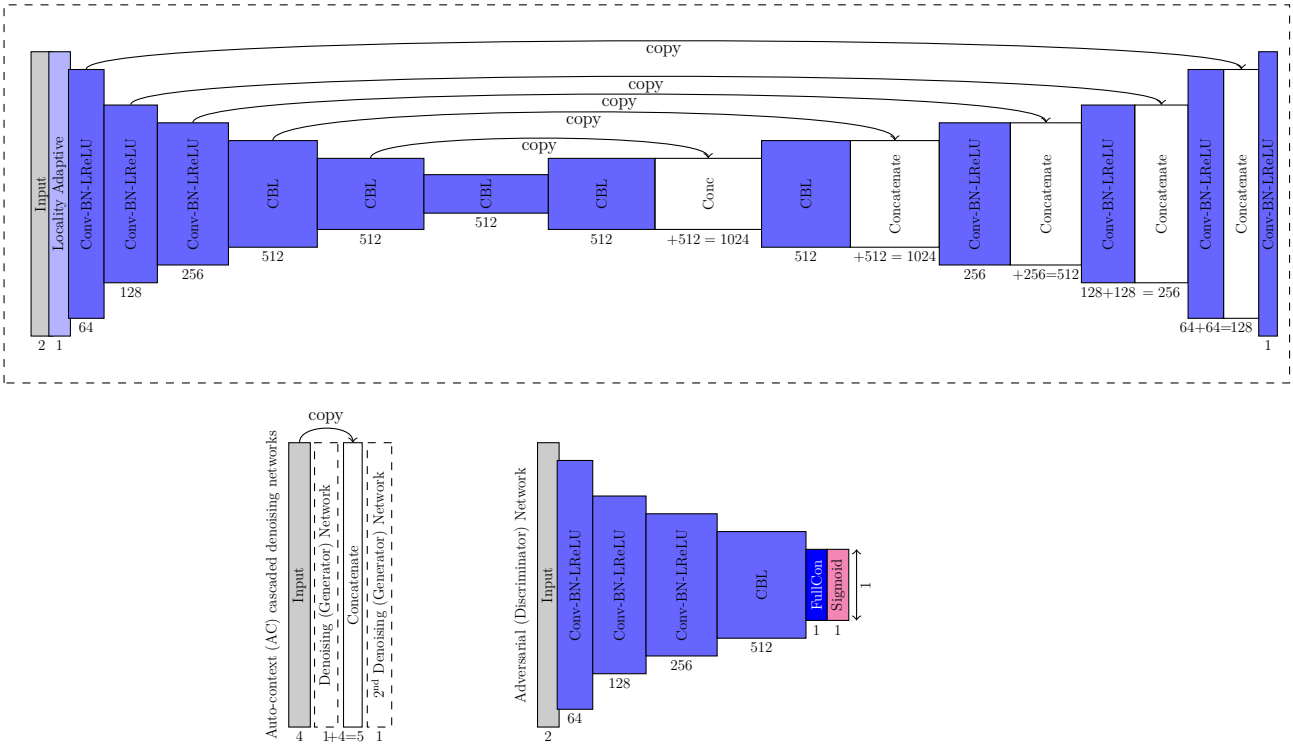


Figure 5.7: Visual representation of *LA-UGAN-AC*: an MR-guided PET denoising U-Net architecture. The numbers below each layer signify the number of output channels of the layer. Conv-BN-LReLU (CBL) layers are combined into a single block for ease of representation. Convolutions use kernel width 4 and a stride of 2 (or 1/2). LReLU uses a negative slope of 0.2. "Auto-context" refers to cascading – appending the output prediction of one trained network to the input channels, and using this augmented input set of data for another network.

This proposed network has some curious design choices. Firstly, the LA layer forces a linear combination of different input modalities at the very start of the network(s). Intuitively, this does not appear to be a good idea, since it seems reasonable to expect that full use of the MR channel information would only be possible with the non-linearities afforded by more layers. Intriguingly, the authors indicated that the LA layer learned to effectively perform a weighted summation of the PET and MR inputs, with a relative weighting of 85 % and 15 %, respectively.

Another curious choice is the cascading (AC) strategy. The overall effect appears to be to double the network size while insisting that an intermediate output from its central layer is in image-space. While this may serve as a form of regularisation and thus make training easier, research has shown that – given sufficient training data – it is best to train end-to-end without forcing intermediate outputs. For example, Wu et al. demonstrated that a 15-layer CNN outperforms 3 cascaded 5-layer networks in terms of SSIM of CT images [188].

Property	Implemented	Original
Input channels	2	2 simulations; 4 real patients
Input spatial dimensions	128×128	$64 \times 64 \times 64$
Training epochs (AC)	400 (-)	200 (200)
Validation-based epoch tolerance (AC)	0 (1 000)	-
ℓ_1-regularisation parameter weight	10^-3	unspecified

Table 5.13: Differences between *LA-UGAN-AC* architecture implementation and original proposal.

Property	Implemented	Original
Input channels	FDG-PET; T1	FDG-PET; T1; real only: fractional anisotropy (FA)-DTI; mean diffusivity (MD)-DTI
Input image width	268 mm	134 mm
OSEM iterations × subsets	7×14	3×21
Count upgrade factor	10	4
Training data	30 simulations; 30 real patients	19 simulations; 15 real patients
Validation data	3 simulations; 3 real patients	1 simulation; 1 real patient
Test data	27 simulations; 27 real patients	-

Table 5.14: Differences between data used for *LA-UGAN-AC* training versus the original proposal.

A LA-UGAN network is trained alongside a discriminator using the Adam optimiser using an ℓ_1 loss function. The learning rate is initially set to 10^{-3} for 200 epochs, after which it decays to 0 over the next 200 epochs. At this point, the prediction images are concatenated with the inputs, thereby forming an augmented input for a different LA-GAN network. Such cascading of networks is referred to by the authors as auto-context (AC). This AC network is then trained alongside a second discriminator without a fixed maximum number of epochs, and training is terminated if the validation loss fails to decrease for over 1000 epochs.

5.2.7 Grid Search

There is a significant gap between the 3-layer *μNet* (Section 5.2.1) and the other much deeper methods under investigation – 8 layers in the next deepest *ResUGAN-TV* network (Section 5.2.5). A summary of the number of convolutional layers for each network is given in the top half of Table 5.15. In order to bridge the gap in this investigation (essentially performing a hyperparameter grid search), additional network with an intermediate number of layers are also trained. These are summarised in the lower half of the same table.

	Short Name	Number of Convolutional Layers: Denoiser (Generator)	Number of Convolutional Layers: Adversarial (Discriminator)
Literature	μNet	3	-
	ResUNet-C	15	-
	ResUNet-X	15	-
	ResUGAN-TV	8	2
	LA-UGAN-AC	26	5
Grid Search	CNN-2	2	-
	CNN-3	3	-
	CNN-4	4	-
	ResCNN-5	5	-
	ResUCNN-5	5	-
	ResUNet-5	5	-
	CNN-7	7	-
	CED-7	7	-
	ResUNet-7	7	-

Table 5.15: Overview of network depths investigated.

The naming convention for the "grid search" networks is as follows:

- Network type:
  - CNN: Convolutional neural net; a sequence of convolutional layers;
  - CED: Convolutional encoder-decoder; a CNN with spatial downsampling (via stride-2 convolution) and upsampling (via bilinear interpolation);
  - UCNN: a CNN with concatenation connections forming a U-like shape;
  - UNet: Both a CED and UCNN;
- Prefix:
  - Res: overall concatenation between PET input channel and the penultimate layer, followed by a final Conv(1×1) layer;
- Suffix:
  - number: number of convolutional layers.

All convolutional kernels have a width of 3, and are followed by a sigmoidal activation function. An exception to this rule is the last convolutional layer (or last two layers in the case of "Res" networks), which uses a kernel width of 1 followed by a ELU activation function. Finally, *CNN-1* and *CNN-2* use a larger kernel width of 5 in their first layer as they would otherwise have a relatively narrow 3×3 receptive field – the wider 5×5 field of view allows for potentially increased smoothing.

The network training uses the Adam optimiser with a learning rate of 10^{-2} to minimise an NRMSE loss function. An ℓ_1-regularisation parameter weight of 10^{-6} is also used. Training is terminated if the validation loss fails to decrease for over 10 000 epochs.

For example, the network *CNN-3* differs from the *μNet* (Section 5.2.1) only in the first convolutional layer (3×3 rather than 5×5 - see Figure 5.2) and the inclusion of ℓ_1-regularisation.

A summary of the different networks is given below, showing the number of kernels in each convolutional layer. Where appropriate, the indices of the layers for copy-and-concatenate operations are also given (index 0 is the input layer), along with scale factors (for down/upsampling). Note that residual connections are not used. Concatenation is more general and powerful. A convolutional layer with a single kernel of spatial width 1 (i.e. number of parameters given by number of input channels) following a concatenation is capable of duplicating the effect of a residual layer (i.e. when all kernel weights are set to 1). A concatenation-and-convolution step can learn an optimal weighted summation.

```
defaults:
  kernel_width: 3
  activation: Sigmoid
  last_layer_override:
    kernel_width: 1
    activation: ELU
CNN-1:  # single filter
  kernels: [1]
  kernel_widths: [5]
CNN-2:  # single hidden layer
  kernels: [32, 1]
  kernel_widths: [5, 1]
CNN-3:  # micro-net but kernel width 3 and l1 regularisation
  kernels: [32, 32, 1]
CNN-4:  # extra layer
  kernels: [32, 64, 32, 1]
ResCNN-5:  # overall skip/concatenation (and extra convolution to condense)
  kernels: [32, 64, 32, 1, 1]
  copy_concatenation_layers: [(0, 4)]
ResUCNN-5:  # also internal concat
  kernels: [32, 64, 32, 1, 1]
  copy_concatenation_layers: [(0, 4), (1, 3)]
ResUNet-5:  # also down/upsampling (U-net)
```

```
  kernels: [32, 64, 32, 1, 1]
  downsampling_factor: [1, 2, 0.5, 1, 1]
  copy_concatenation_layers: [(0, 4), (1, 3)]
CNN-7:  # more layers (approaching Chen2019/Xu2020)
  kernels: [16, 32, 64, 32, 16, 1, 1]
CED-7:  # also down/upsampling (CED)
  kernels: [16, 32, 64, 32, 16, 1, 1]
  downsampling_factor: [1, 2, 2, 0.5, 0.5, 1, 1]
ResUNet-7:  # also concatenation (U-net)
  kernels: [32, 64, 128, 64, 32, 1, 1]
  downsampling_factor: [1, 2, 2, 0.5, 0.5, 1, 1]
  copy_concatenation_layers: [(0, 6), (1, 5), (2, 4)]
```

5.3 Results

5.3.1 Simulations

BrainWeb-based simulation data is generated in a similar manner to that described in Section 3.2. Specifically, a count level of 30 M is used for the inputs and 300 M for the targets. Three independent PET noise realisations are generated for each of the 20 available subjects, each reconstructed using OSEM with 14 subsets and 7 iterations.

Each network is trained on this data as described in the sections above.

The whole-volume (including voxels outside the brain) training and validation NRMSE evolution with epochs is shown in Figure 5.8. Training is terminated (typically based on validation loss) as prescribed by each method (outlined in Section 5.2 above). Note that in all cases, validation loss has either stabilised or is increasing. Upon training termination, each network's parameters are restored to their values corresponding to the minimum validation loss.

Most methods (apart from the *μNet* and *ResUGAN-TV*) show a large divergence between the training and validation losses, strongly indicating overfitting as discussed in Section 2.2. The most complicated method – *LA-GAN-AC* – does however have a much lower training error than the rest, implying a possibility of outperforming all other methods if overfitting can be obviated (e.g. by providing a much larger training dataset). Interestingly as discussed in Section 5.2.6, the original proposal utilised an even smaller training dataset than used here, and should have thus been susceptible to chronic overfitting.

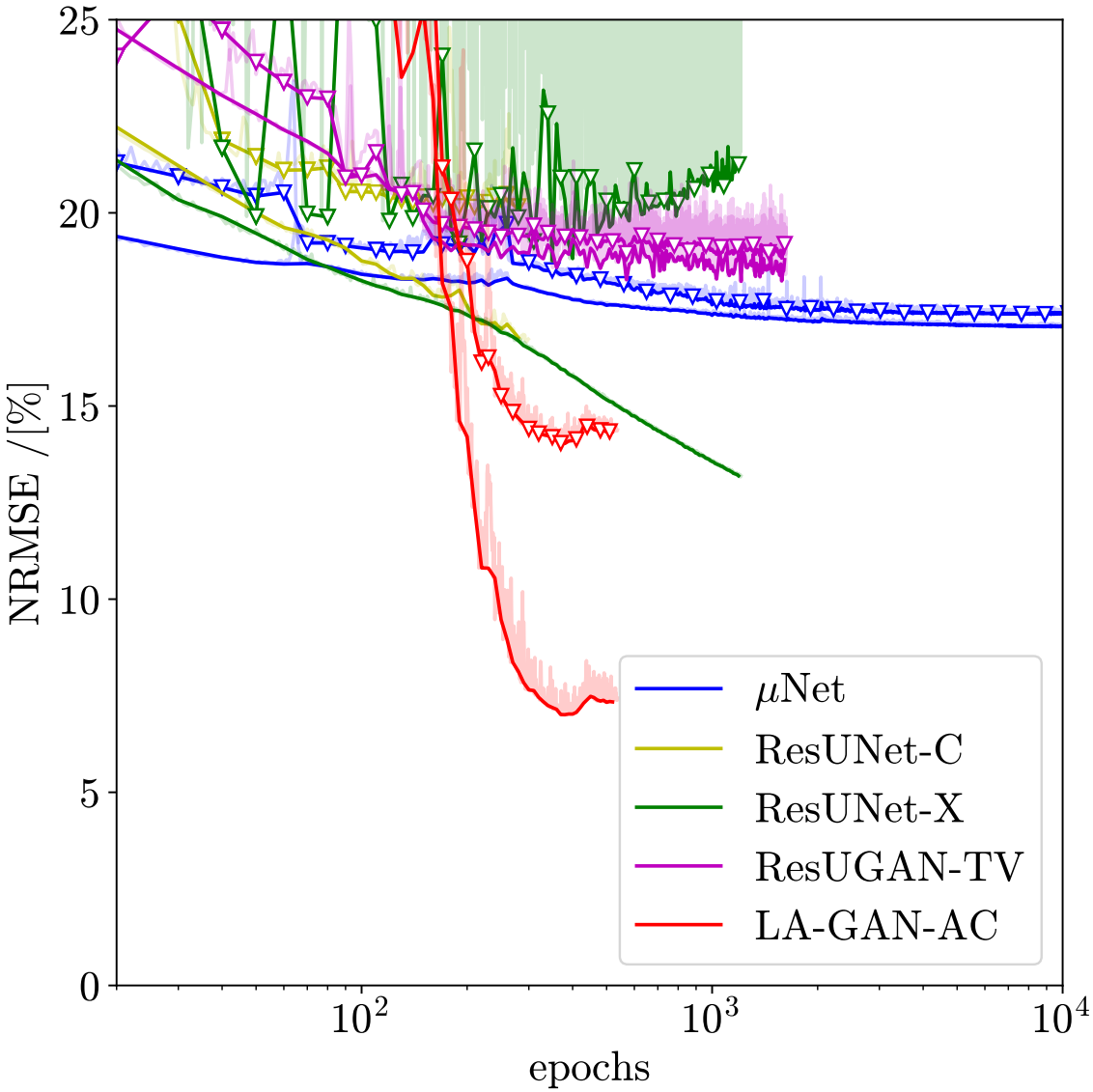
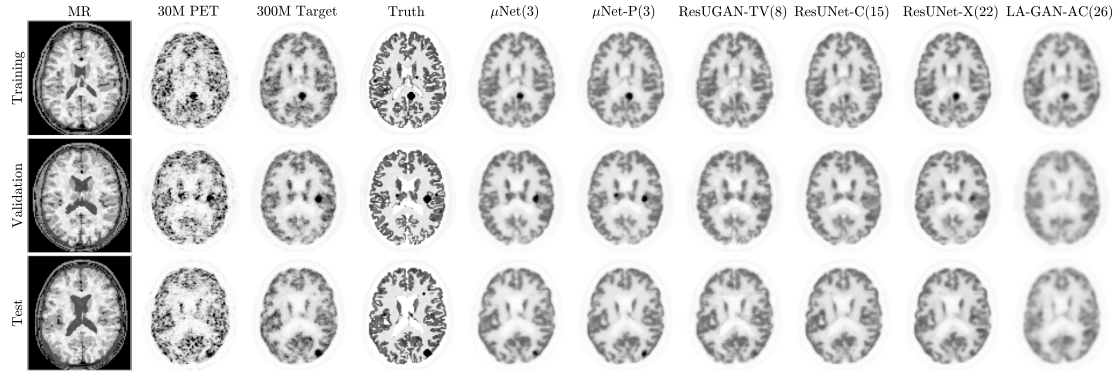
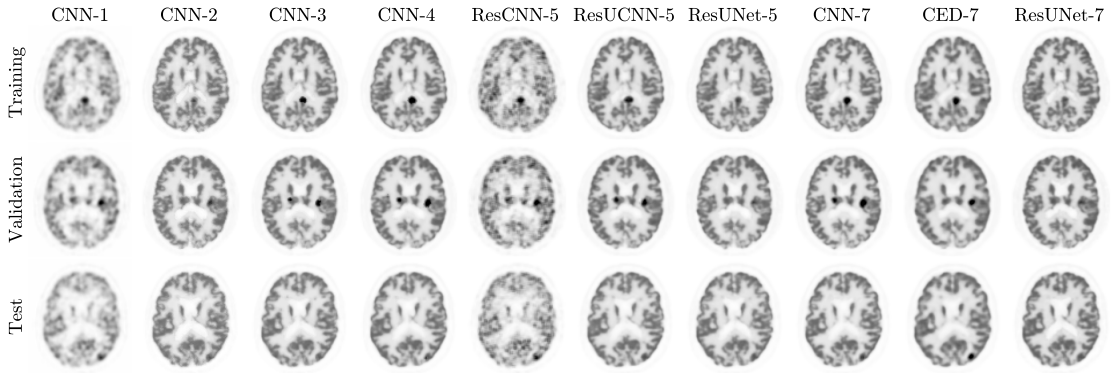


Figure 5.8: Whole-volume training (no markers) and validation (triangles) NRMSE against training epochs. Pale lines represent actual NRMSE values, while solid lines represent a moving minimum with a 10-epoch window.

Three different subjects are selected at random – one each from the training, validation and test datasets. Central slices for these subjects are shown in Figure 5.9. The first four columns of Figure 5.9(a) show the 2 input channels (MR and low count PET), target (full count PET), and ground truth images. The remaining columns correspond to the outputs of networks proposed in literature (based on the inputs). For the hyperparameter grid search networks (Section 5.2.7), outputs are shown in Figure 5.9(b). The intensity display scale is the same for all PET images (apart from the 30 M count input, which is scaled by a factor of 10), making them directly comparable. Qualitatively, note that while most methods seem to reduce the visible noise and improve edges, most methods also remove the lesions (false negatives, especially in the test datasets). A more quantitative assessment of whole-brain versus lesion accuracy is included later below.



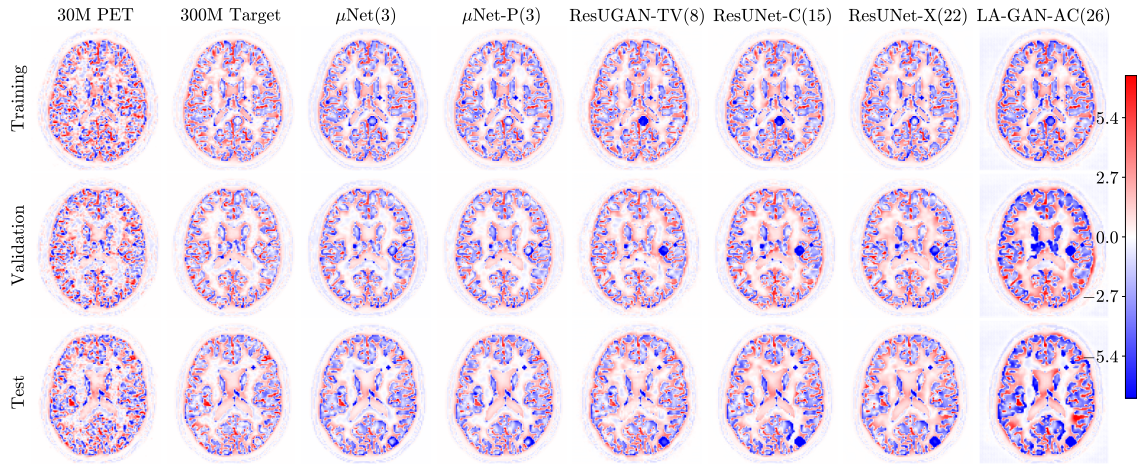
(a) The first four columns show input (MR and 30 M count PET), target (300 M count PET), and ground truth images. The remaining columns show output (networks proposed in literature) images. The number of convolutional layers in each network are parenthesised in the column headings.



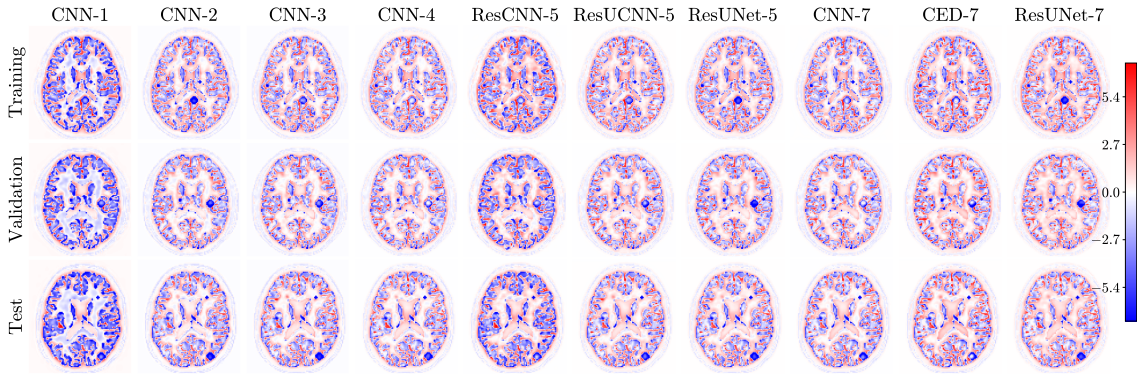
(b) Hyperparamter grid search network output images. The number of convolutional layers in each network are given in the column headings.

Figure 5.9: Training (first row), validation (second row) and test (third row) simulation datasets. For each dataset, a central slice of one noise realisation of one subject is shown.

Corresponding bias and standard deviation PET images (calculated across 3 noise realisations) are shown in Figure 5.10 and Figure 5.11, respectively.



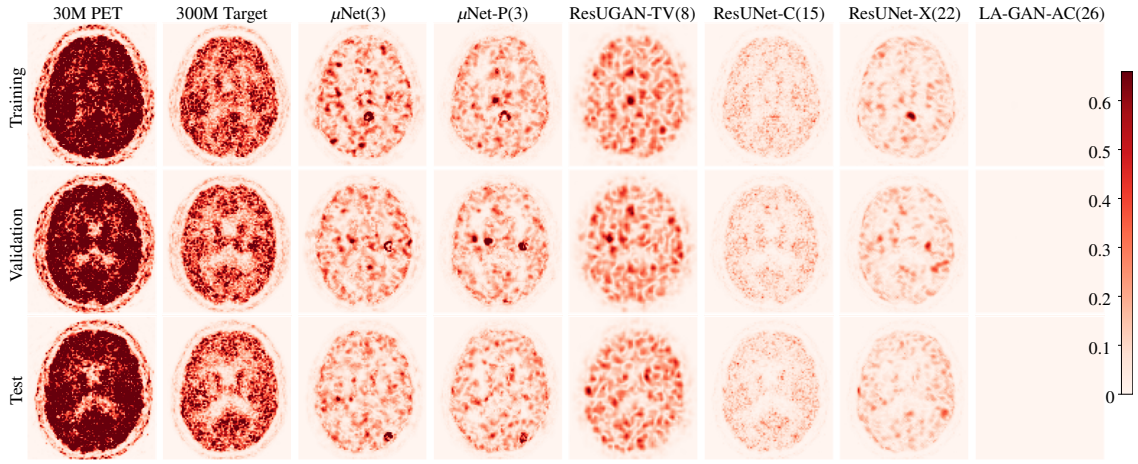
(a) Bias images corresponding to Figure 5.9(a).



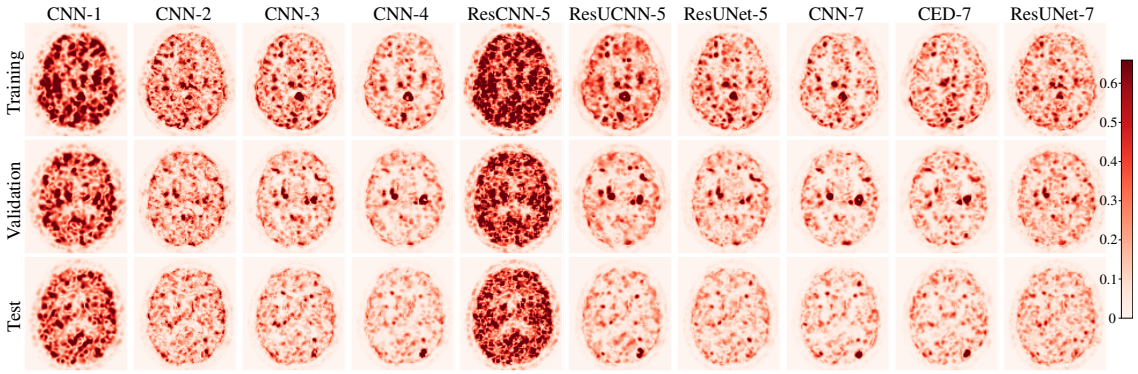
(b) Bias images corresponding to Figure 5.9(b).

Figure 5.10: Bias images corresponding to Figure 5.9. All images use a common colourscale so are immediately comparable.

For the test results (Figure 5.10, bottom row) there is clear bias for most methods in lesions, with larger networks also showing high bias elsewhere (especially putamen and grey matter in general for the *LA-GAN-AC*). At the other end of the complexity spectrum, the simple *CNN-1* has a uniform high bias everywhere. Interestingly, the *ResCNN-5* also has a high bias in most regions. The image with the lowest bias (both overall and for the lesions) appears to be the input 30 M count reconstruction – nearly matching the target 300 M count image. Depending on the clinical task, *μNet*, *μNet-P*, and *CED-7* may be acceptable too since they retain low bias in the very centre of the large lesion (thereby ensuring accurate $\mathrm{SUV_{max}}$).



(a) Standard deviation images corresponding to Figure 5.9(a).



(b) Standard deviation images corresponding to Figure 5.9(b).

Figure 5.11: Standard deviation images corresponding to Figure 5.9. All images use a common colourscale so are immediately comparable.

In general, all methods greatly reduce standard deviation in most areas apart from lesions. Exceptions are *CNN-1*, *ResCNN-5* and *ResUGAN-TV*, which have moderate standard deviation in all areas. Remarkably low standard deviation in all areas is also visible in *ResUNet-C* and *LA-GAN-AC*, the latter being effectively zero. Considering that these two methods have very high bias in the (PET-unique) lesions, it is likely that they produce a PET output based mostly upon the MR input. This would explain the low standard deviation (as the MR does not change across PET noise realisations) and high PET-unique RoI bias.

It is difficult to interpret the outputs of individual kernels and/or layers and the effect they have on the overall network. However, in the case of *CNN-1*, there is only one kernel, which acts as a learned post-processing filter. This is shown in Figure 5.12 below. Unsurprisingly, the network has learned to apply a Gaussian-like post-smoothing to the low count PET input image. Interestingly, there also appears to be a slight MR-guided sharpening.

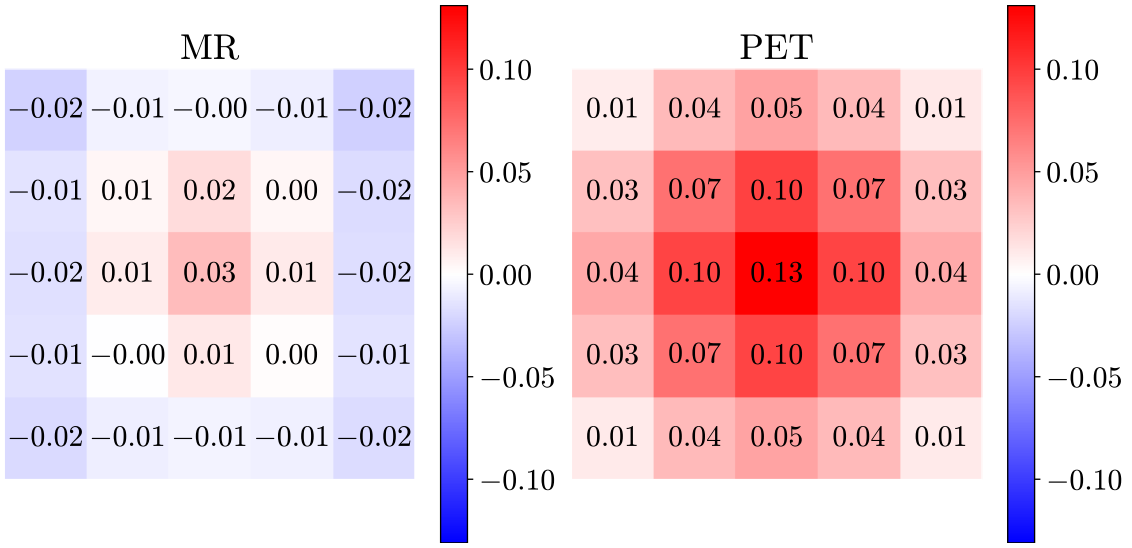


Figure 5.12: Learned 2-channel 2D convolutional kernel for *CNN-1*. The MR channel corresponds to a slight sharpening operation, while the PET channel performs Gaussian-like smoothing.

Quantitative metrics (averaged across 9 training, 1 validation, and 10 test subjects) for bias and standard deviation (calculated across 3 realisations per subject) are given in Figure 5.13. The *LA-GAN-AC* has virtually no standard deviation, meaning the output is the same when given a different input noise realisation. This indicates that the network has memorised that three different PET noise realisations (and the same single MR input for all three) should always produce the same single output. Such marked memorisation is a strong indicator of the PET input being ignored, and overfitting to the MR. By comparison, the far simpler *CNN-2* has lower bias in all cases. Compared to the low count input, all methods apart from the *LA-GAN-AC* and *CNN-1* reduce bias.

Apart from the *LA-GAN-AC*, training and validation metrics are similar, implying negligible overfitting in most cases. Note that validation bias and standard deviation are only marginally lower than test metrics, meaning that the validation dataset is reasonably sized.

The bias and standard deviation can be summarised by their Euclidean sum in the NRMSE as per Equation (3.4). The trade-off between foreground whole-brain (the ground truth brain mask extended by 2 surrounding voxels) and lesion RoI (the lesion mask extended by 2 surrounding voxels) NRMSE is shown for all endpoint volumes in Figure 5.14. Curves are also depicted for increasing PS and NLM-guided filtering of the OSEM reconstructions.

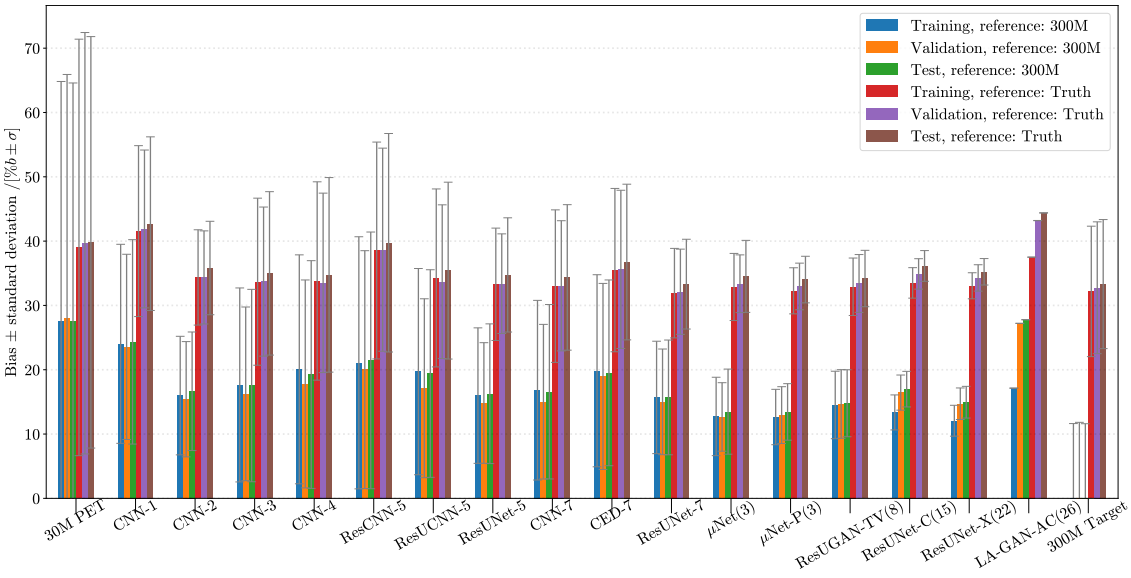


Figure 5.13: Training, validation and test whole-brain metrics (bias b and standard deviation σ across 3 noise realisations as per Equations (3.2) and (3.3)) for each method. The bar plots show b, while the error bars show σ for training (blue), validation (orange) and test (green) data using the 300 M count reconstruction as a reference. If the ground truth is used as a reference instead, the results are shown in red, magenta, and brown, respectively.

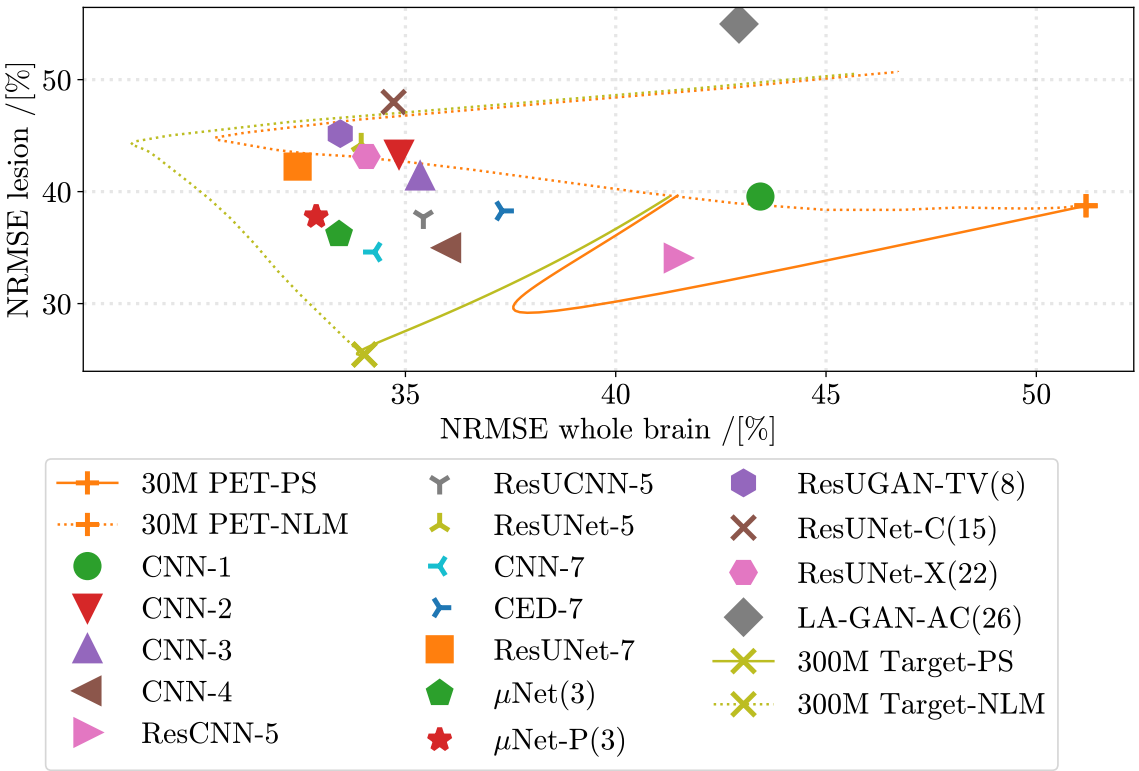


Figure 5.14: Whole-brain versus lesion test NRMSE trade-off for various methods. Curves are also shown for increasing Gaussian post-smoothing (PS of 0 to 100 mm FWHM) and NLM-guided filtering ($\Omega \in [10^{-5}, 10^{5}]$) of the MLEM reconstructions. Excluding the target 300 M reconstruction, the lowest lesion NRMSE is obtained by simple PS followed by *ResCNN-5*. Most networks with 3 to 7 convolutional layers decrease lesion NRMSE compared to the unsmoothed input, and all smaller/larger networks perform worse. Meanwhile, for the whole-brain, the lowest NRMSE is obtained by the *ResUNet-7* followed by *μNet-P*.

It is difficult to see a clear trend in performance. The best performing networks have between 3 and 7 convolutional layers depending on the desired trade-off between whole-brain and lesion NRMSE. To gain more insight, an interesting consideration is the ratio of network parameters to training data size. Such a ratio can be considered an indicator of network complexity relative to the training data. This is given in Figures 5.15 and 5.16 for whole-brain and lesion RoIs, respectively. Some grid search architectures are also selected to be completely retrained on reduced datasets to trace out curves (halved training data each time until just one 2D slice of one patient is used in training). Note that given the large amount of time required to retrain, curves are not shown for all architectures. Each network is also retrained multiple times on the same data in order to compute standard error (y-error bars). Interesting, the standard errors are mostly too small to discern, indicating robustness to randomised parameter initialisation in most cases.

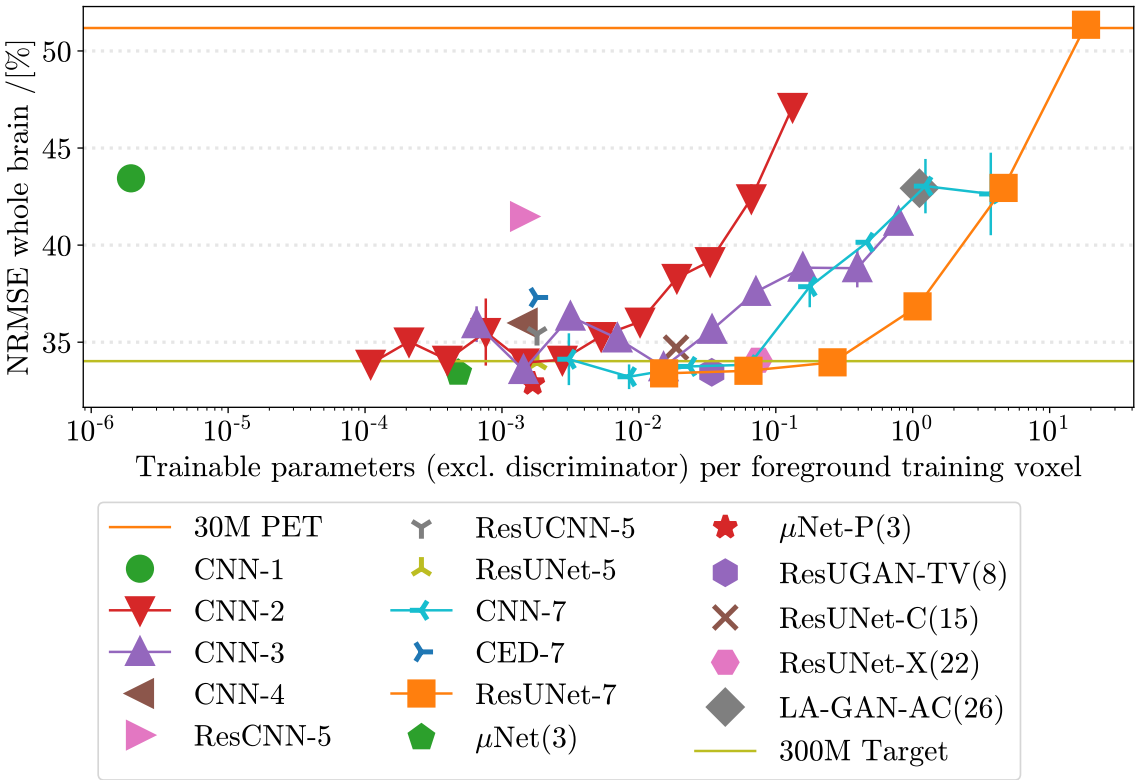


Figure 5.15: Test whole-brain NRMSE against network complexity. For some networks, the training dataset is sucessively halved in size and complete retraining is done in order to trace out the curves. For each dataset size, the network is also retrained multiple times in order to produce standard errors (y-error bars that are often too small to see).

It is clear that reducing training data eventually harms performance. However, a small parameter-to-data ratio is not a guarantee of good performance. For instance, *CNN-3* performs poorly compared to *μNet-P* despite having the same number of convolutional layers. Evidently, more subtle design choices such as number of kernels per layer and pre-processing inputs (e.g. with NLM-guided filtering) are crucial to achieving good performance.

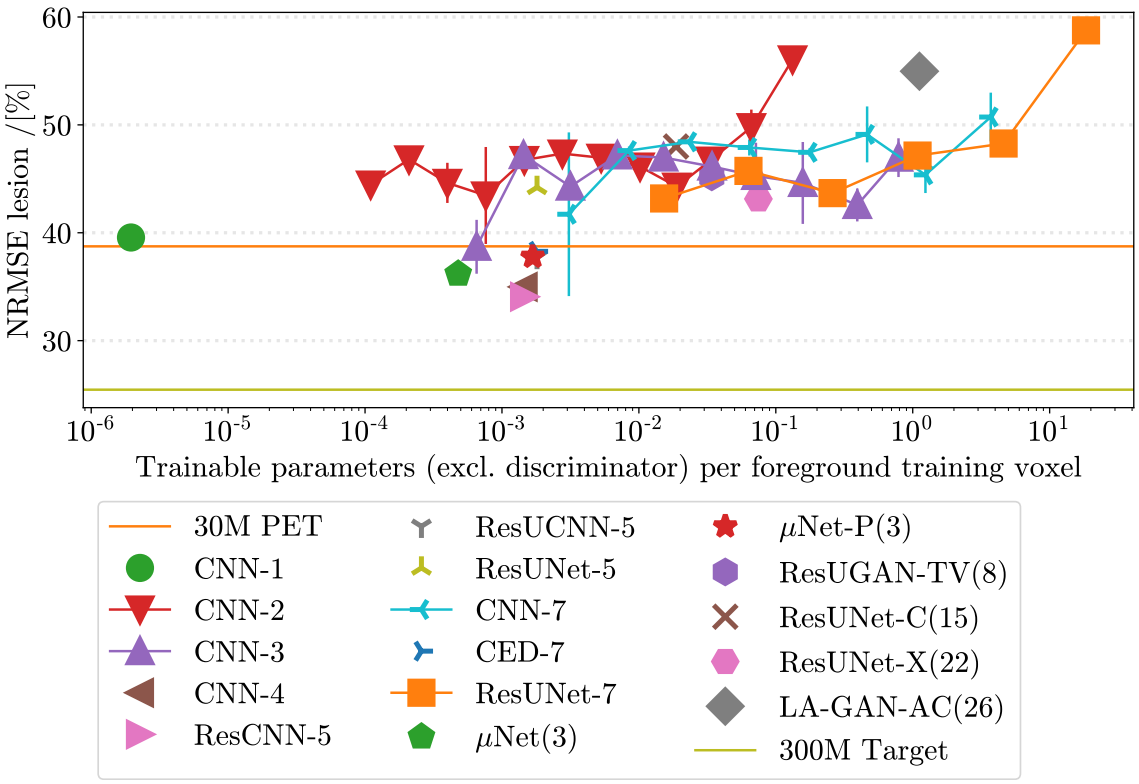


Figure 5.16: Test **lesion** NRMSE against network complexity (where complexity is estimated as the ratio of trainable parameters to training data). Most networks degrade lesions. Only some networks with 3 to 7 convolutional layers potentially decrease NRMSE.

As training data is halved, test NRMSE eventually starts to increase. This increase occurs sooner for larger networks (which are more likely to begin overfitting). The *CNN-1* result in Figure 5.16 is particularly concerning for many of the proposals from the current literature. The results demonstrate that an optimal post-smoothing filter is far less likely to produce misleading results around rare yet crucial lesions than overly complex networks.

As an alternative to NRMSE, whole-brain to lesion performance trade-offs are shown using MSSIM in Figure 5.17. All networks with more than 7 convolutional layers degrade lesion MSSIM. Unlike the case with NRMSE, the *CNN-1* causes a slight improvement rather than harm to the lesion. Neverthless, simple PS is still capable of producing slightly better lesion MSSIM than the best networks (μ*Net* and *CNN-7*).

It should be noted that most networks perform quite well on whole-brain MSSIM, with only *LA-GAN-AC*, *CNN-1*, and *ResCNN-5* failing to outperform PS. Based on these results, none of the CNNs considered here seem appropriate for clinical oncology.

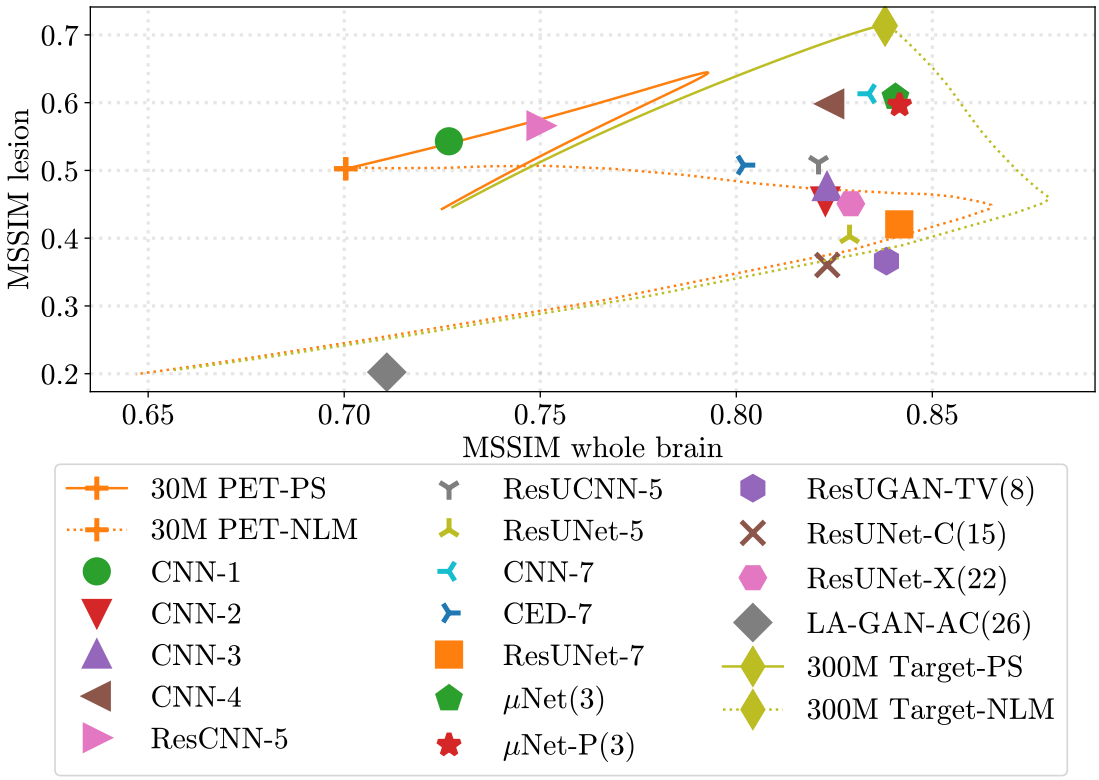


Figure 5.17: Whole-brain versus lesion MSSIM trade-off for various methods (similar to NRMSE results in Figure 5.14).

Trade-offs against network complexity for whole-brain and lesion MSSIM included in Appendix A.2, showing very similar results to that of NRMSE (Figures 5.15 and 5.16).

In lieu of network complexity or size, it is also interesting to see MSSIM against amount of training data (Figure 5.18). The curves indicate that – provided a network isn't overly small or big – performance is mainly controlled by the amount of available data, with a maximum performance threshold achieved with $\mathcal{O}(10^6)$ foreground (whole-brain) training voxels.

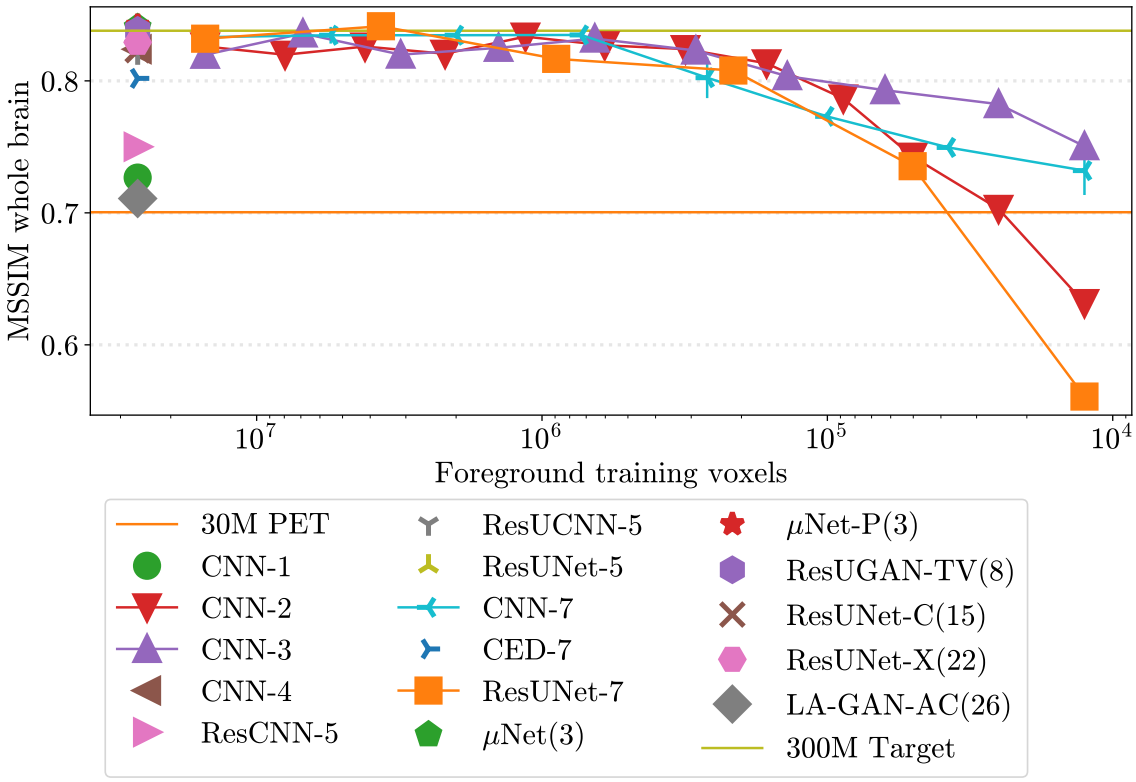


Figure 5.18: Whole-brain test MSSIM versus amount of training data.

All networks use the Adam optimiser, apart from *ResUNet-X*, which uses RMSProp. As discussed in Section 2.2, the Adam optimiser works well with a large range of learning rates, so bespoke optimisation for the data used here should not be required. However, it could still nevertheless be possible that the results above may be affected by suboptimal learning rates. In particular, the hyperparameter grid search networks have not been proposed in literature and thus have no suggested learning rate. Figure 5.19 below shows the result of re-training some of these networks with different learning rates. In particular a learning rate $\in [10^{-2}, 10^{-3}]$ works well for complex networks such as *ResUNet-7* as well as simpler ones such as *CNN-3*. This implies that the learning rate of 10^{-2} used for all grid search networks is appropriate and does not negatively impact performance.

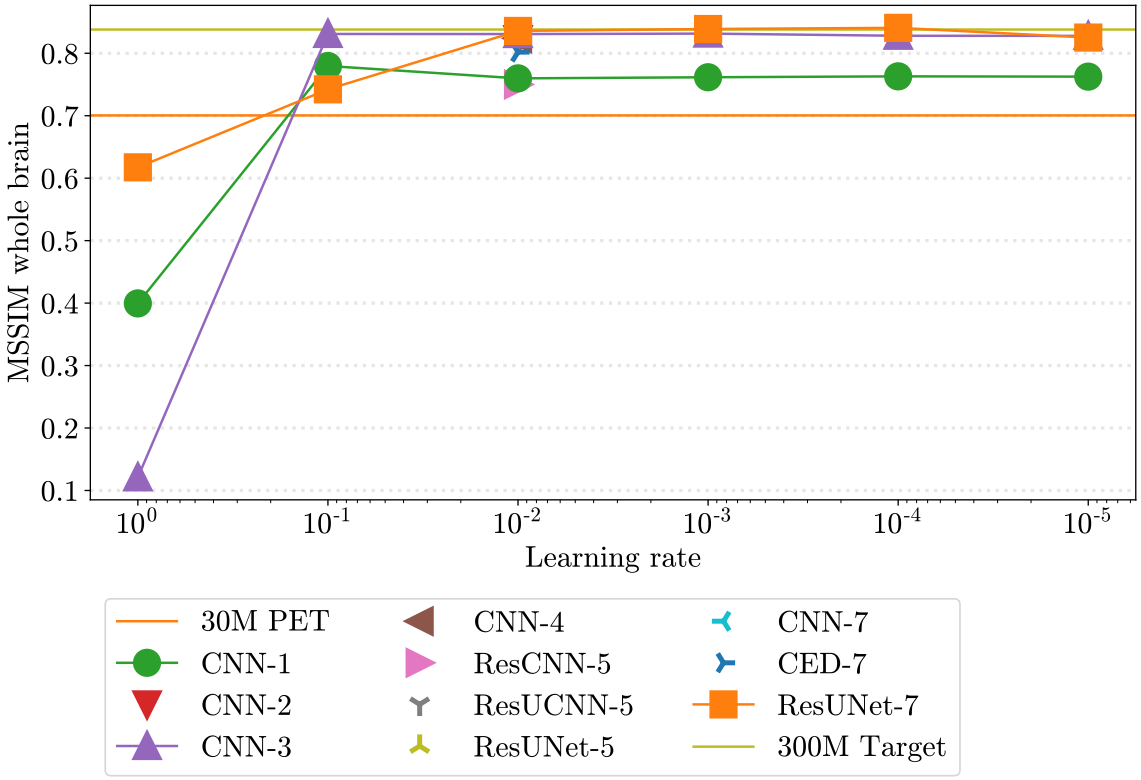


Figure 5.19: Validation loss with varying Adam learning rate.

5.3.2 Patient Data

All networks are re-trained on real patient datasets of precisely the same size (number of patients/subjects and noise realisations) and count levels as used in the simulations above. The significant change is that noise realisations are generated by bootstrap sampling (with replacement) from the raw listmode (averaging 430 M counts per acquisition) data. Reconstructions of the raw data are used in lieu of the ground truth for the purposes of calculating evaluation metrics after training.

The whole-volume (including voxels outside the brain) training and validation NRMSE evolution with epochs is shown in Figure 5.20. Once again, upon training termination, each network's parameters are restored to their values corresponding to the minimum validation loss.

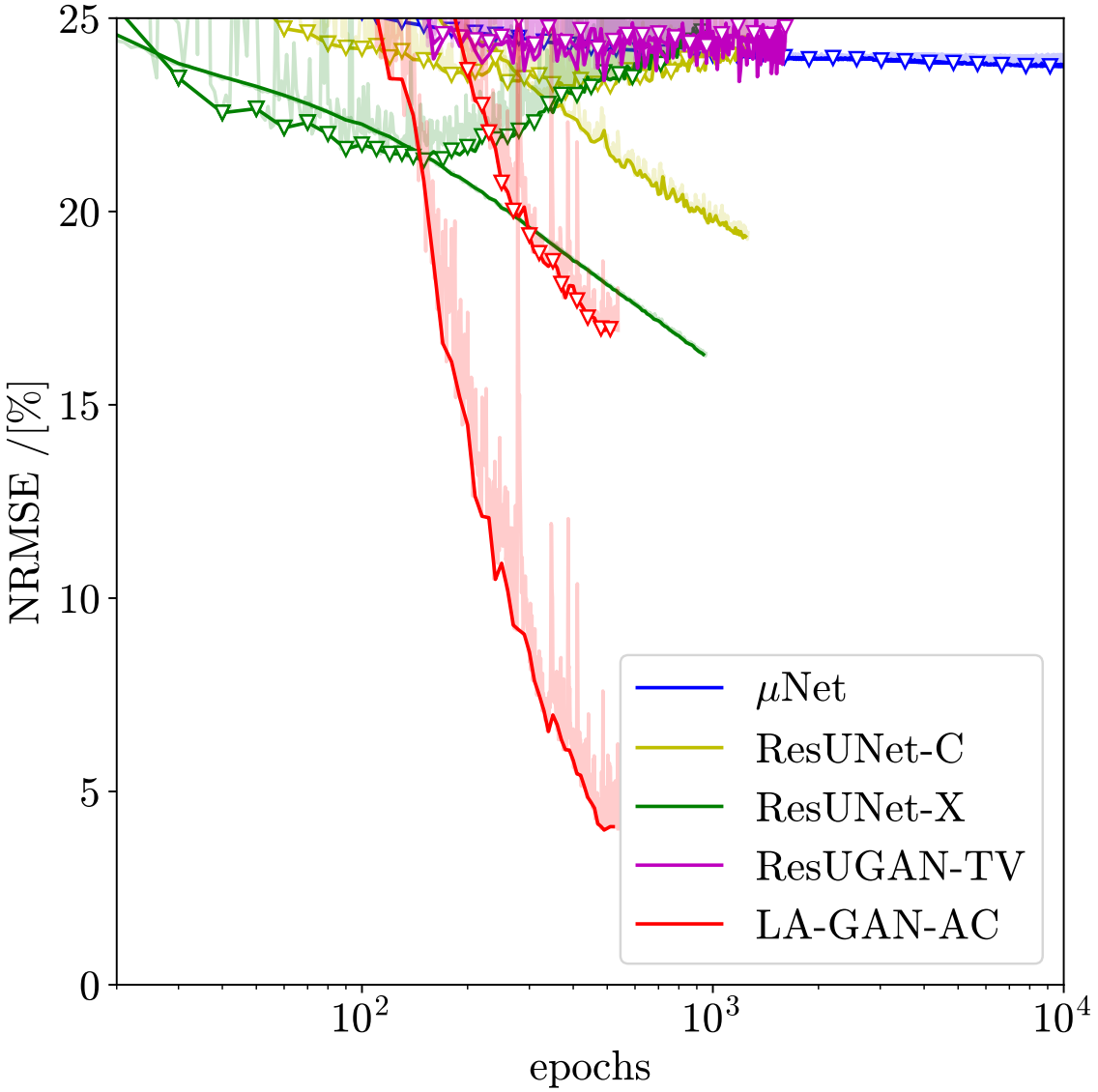
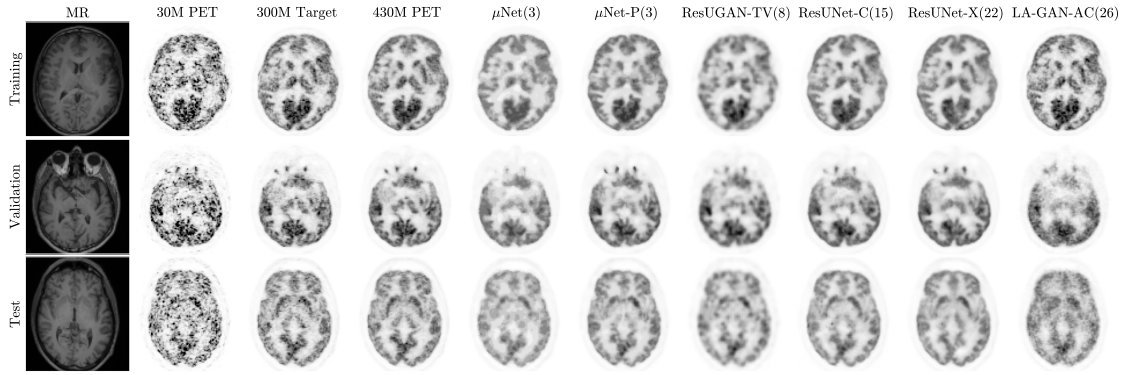


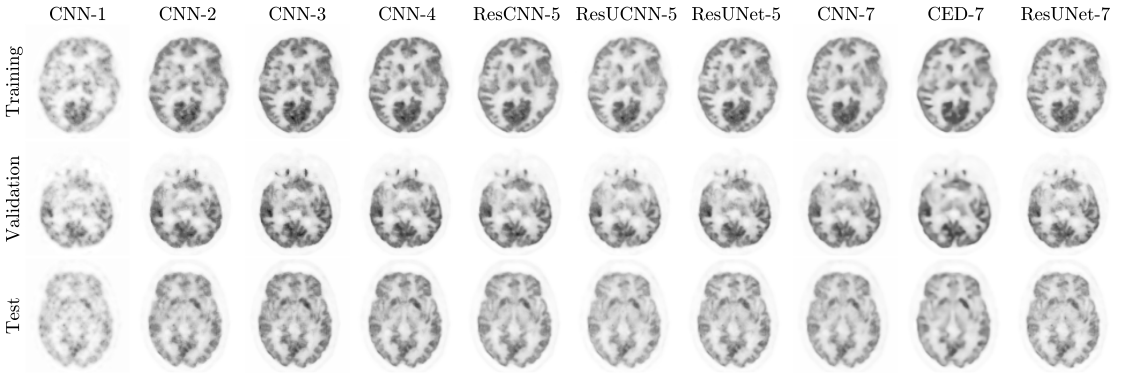
Figure 5.20: Whole-volume training (no markers) and validation (triangles) NRMSE against training epochs for real patient data (compare with simulation results in Figure 5.8).

The NRMSE values tend to be higher than compared to the simulation results from Figure 5.8, but otherwise follow the same general shape and relative trends.

Central slices of a training, validation and test patient are shown in Figure 5.21 (comparable to Figure 5.9). There are clear cases of lower resolution with reduced contrast – especially *CNN-1*, *ResUCNN-5*, and *μNet*. Meanwhile the *LA-GAN-AC* appears to have generated an image with noise properties similar to that of the target, but is otherwise inaccurate. The best contrast and edge accuracy seems to be achieved by *μNet-P*, *ResUNet-C*, *ResUNet-X*, and *CNN-3*.



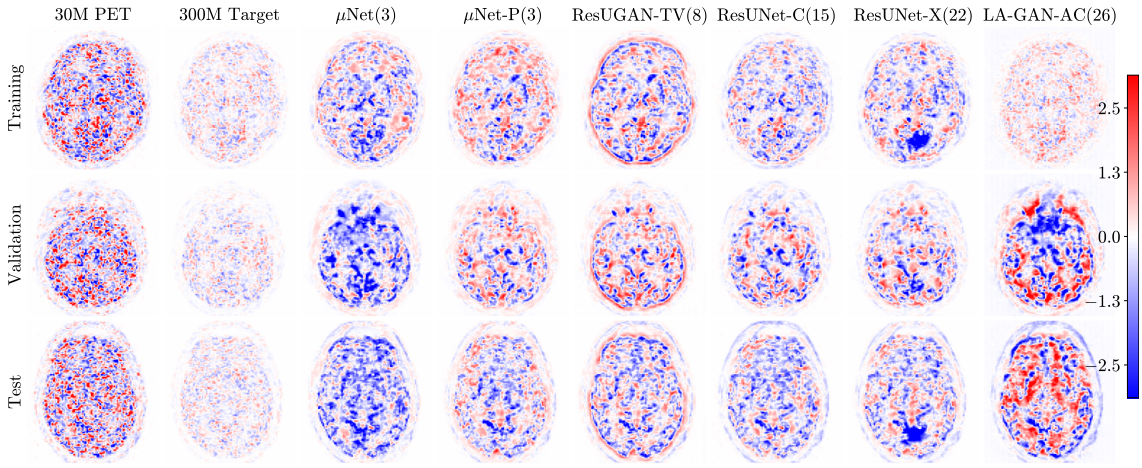
(a) The first four columns show input (MR and 30 M count PET), target (300 M count PET), and reference (unsampled, circa 430 M count PET) images. The remaining columns show output (networks proposed in literature) images. The number of convolutional layers in each network are parenthesised in the column headings.



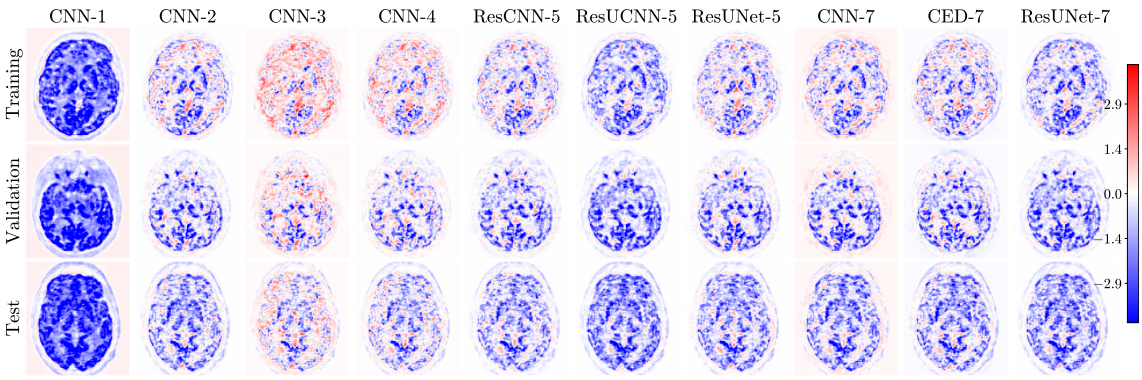
(b) Hyperparamter grid search network output images. The number of convolutional layers in each network are given in the column headings.

Figure 5.21: Training, validation and test simulation datasets. For each dataset, a central slice of one noise realisation of one patient is shown.

Corresponding bias and standard deviation PET images are shown in Figure 5.22 and Figure 5.23, respectively.



(a) Bias images corresponding to Figure 5.21(a).

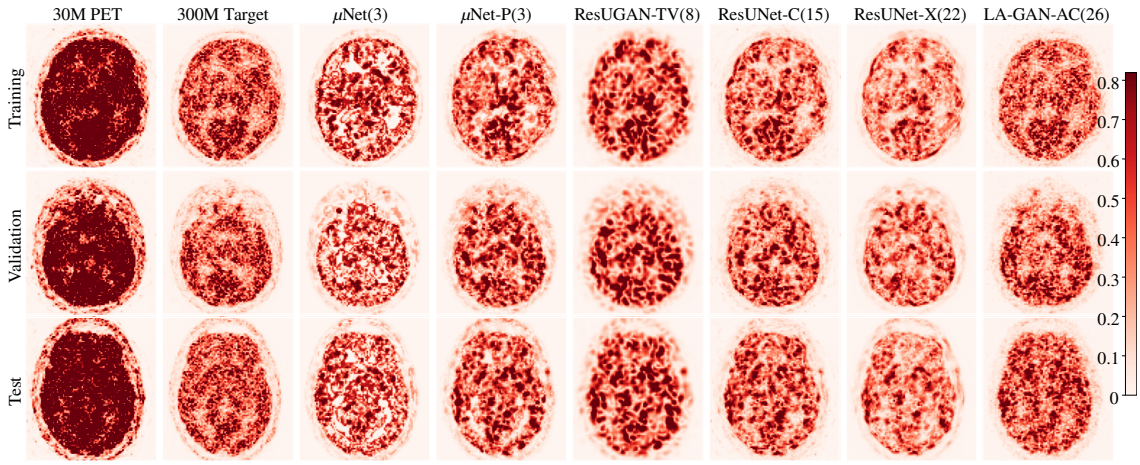


(b) Bias images corresponding to Figure 5.21(b).

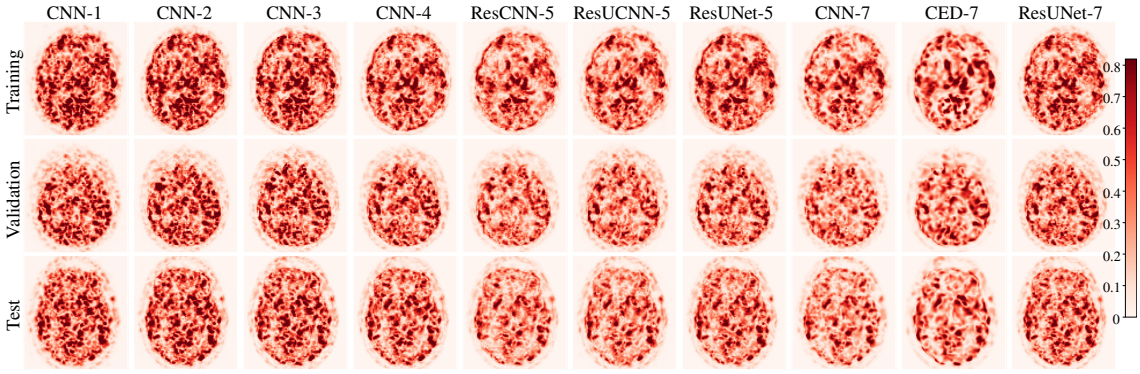
Figure 5.22: Bias images corresponding to Figure 5.21. All images use a common colourscale so are immediately comparable.

For the test results (Figure 5.22, bottom row) there are clear cases of regions of large bias such as in the occipital lobe for *ResUNet-X*. It is clearer that *μNet-P* and *CNN-3* generally reduce bias, while all other methods perform comparatively poorly.

All methods reduce standard deviation, though this effect is less pronounced than with the simulations from Figure 5.11.



(a) Standard deviation images corresponding to Figure 5.21(a).



(b) Standard deviation images corresponding to Figure 5.21(b).

Figure 5.23: Standard deviation images corresponding to Figure 5.21. All images use a common colourscale so are immediately comparable.

Quantitative metrics (averaged across 9 training, 1 validation, and 10 test patients) for bias and standard deviation (calculated across 3 bootstrap realisations per patient) are given in Figure 5.24. Note that metrics are calculated against the raw (circa 430 M count) reconstructions in lieu of the ground truth.

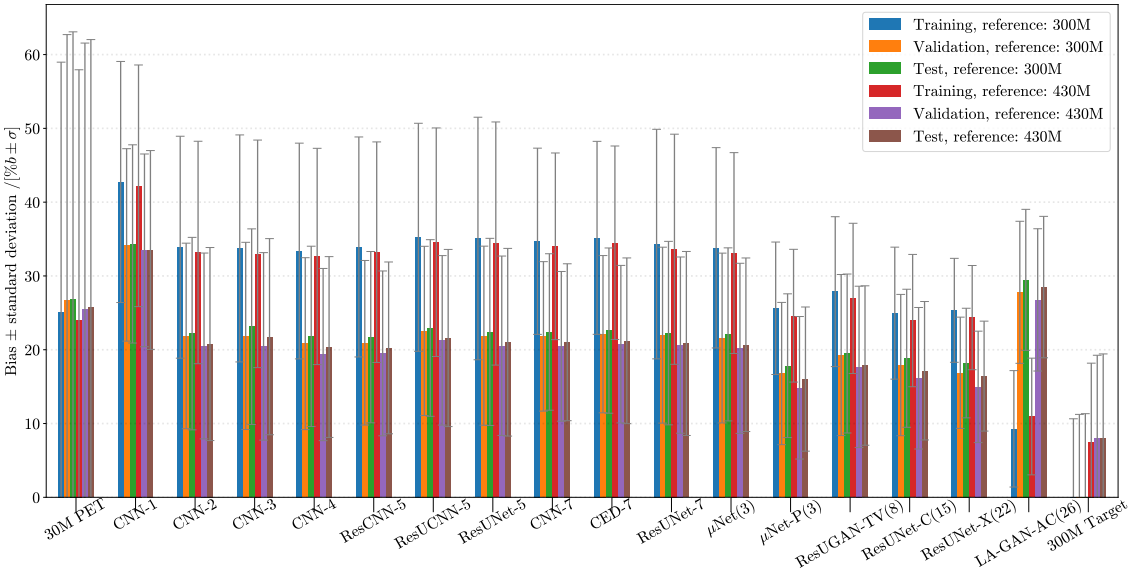


Figure 5.24: Training, validation and test whole-brain metrics (bias and standard deviation across 3 noise realisations) for each method (compare to simulation results from Figure 5.13).

Unlike the simulation studies from Figure 5.13, the *LA-GAN-AC* now has a reasonable standard deviation. Once again, in all cases the marginally higher test errors (compared to validation) mean that the validation data is likely sufficient in size.

The trade-off between foreground whole-brain and "lesion" (high intensity regions) NRMSE is shown for all endpoint volumes in Figure 5.25. Curves are also depicted for increasing PS and NLM-guided filtering of the OSEM reconstructions.

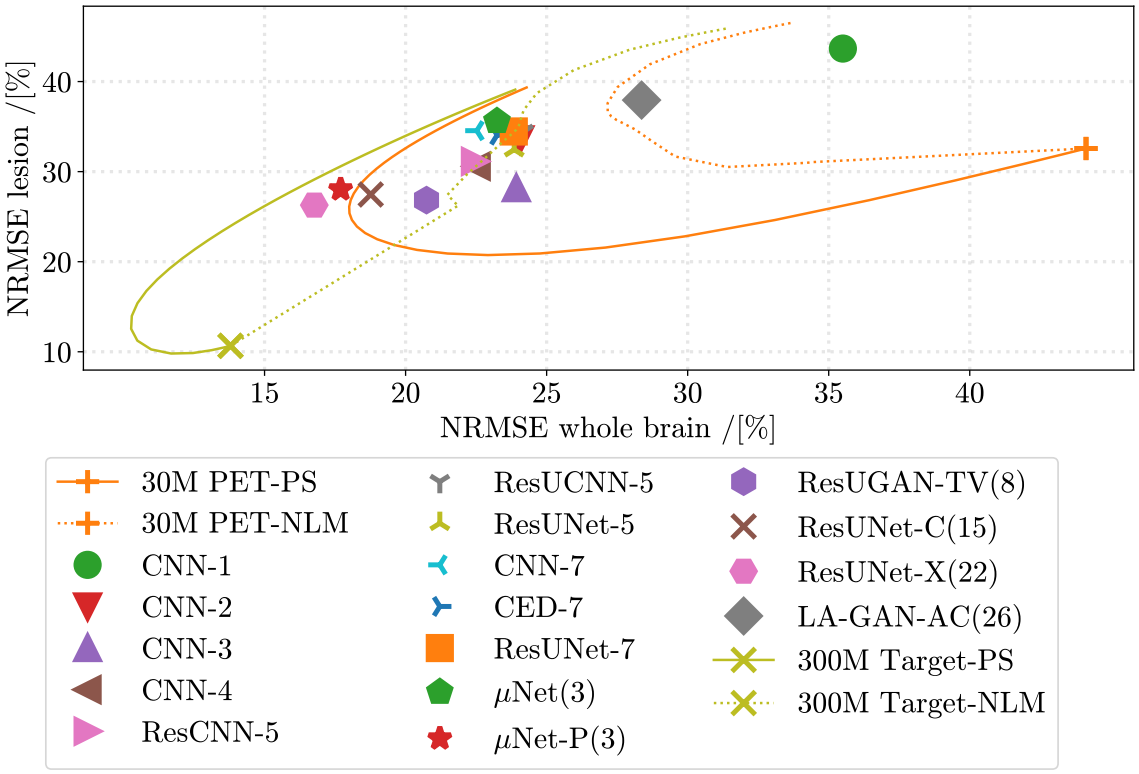


Figure 5.25: Whole-brain versus "lesion" (high-intensity) NRMSE trade-off for various methods. Curves are also shown for increasing Gaussian post-smoothing and NLM-guided filtering (compare to simulation results from Figure 5.14).

In this case, the best performing networks are all proposals from the current literature: *ResUNet-X*, *μNet-P*, *ResUNet-C*, and *ResUGAN-TV*. As with the simulation results from Figure 5.14, simple PS is capable of outperforming all methods for lesion NRMSE. However it should be noted that the metrics are calculated in this case against raw (unsampled) reconstructions rather than a ground truth, which may make the results somewhat misleading (discussed in the section below).

The effect of network complexity on whole-brain and high-intensity RoI performance is shown in Figures 5.26 and 5.27, respectively.

Unlike the case with simulations (Figure 5.15), none of the networks outperform the target for whole-brain NRMSE. Additionally, some of the larger networks now seem to perform well – *ResUNet-X* with 22 convolutional layers producing the lowest NRMSE.

As an alternative to NRMSE, whole-brain to lesion performance trade-offs are shown using MSSIM in Figure 5.28. The top performing networks are much easier to see, with *ResUNet-X* continuing to outperform rivals, closely followed by *ResUGAN-TV*, *ResUNet-C*, and *μNet-P*.

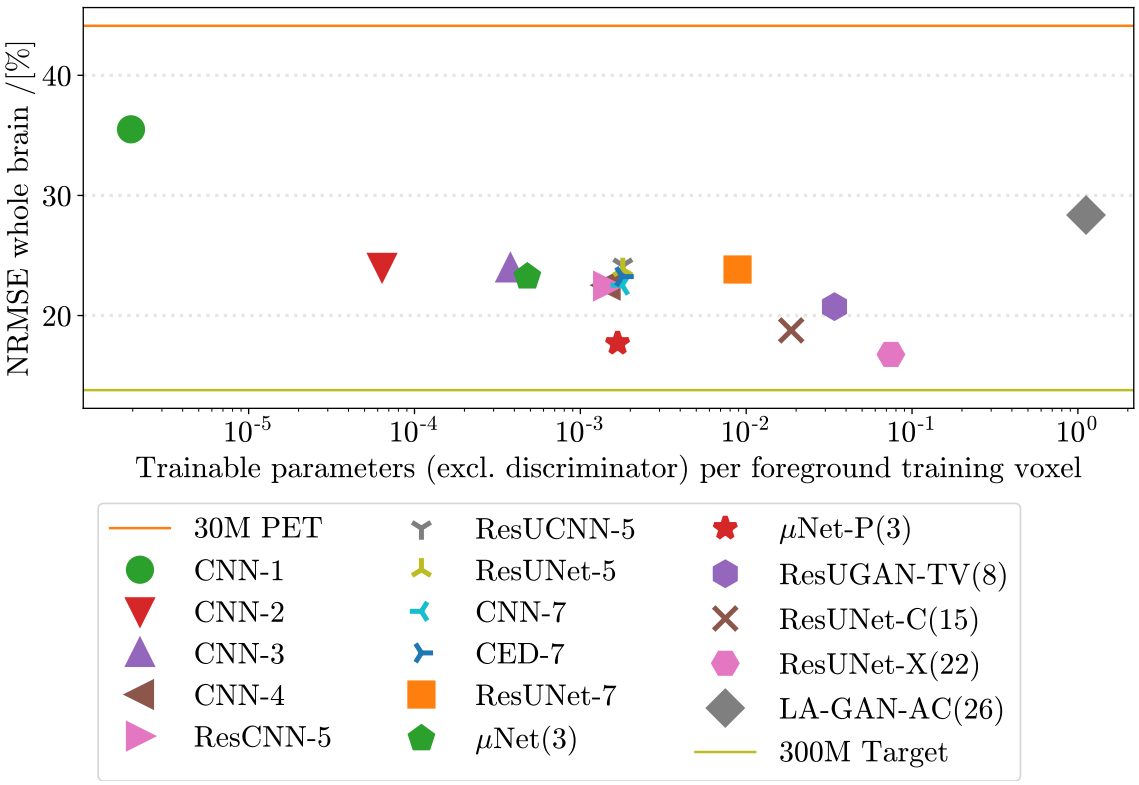


Figure 5.26: Test whole-brain NRMSE against network complexity. Unlike the simulation results in Figure 5.15, there was insufficient time to retrain the networks here on sucessively smaller datasets.

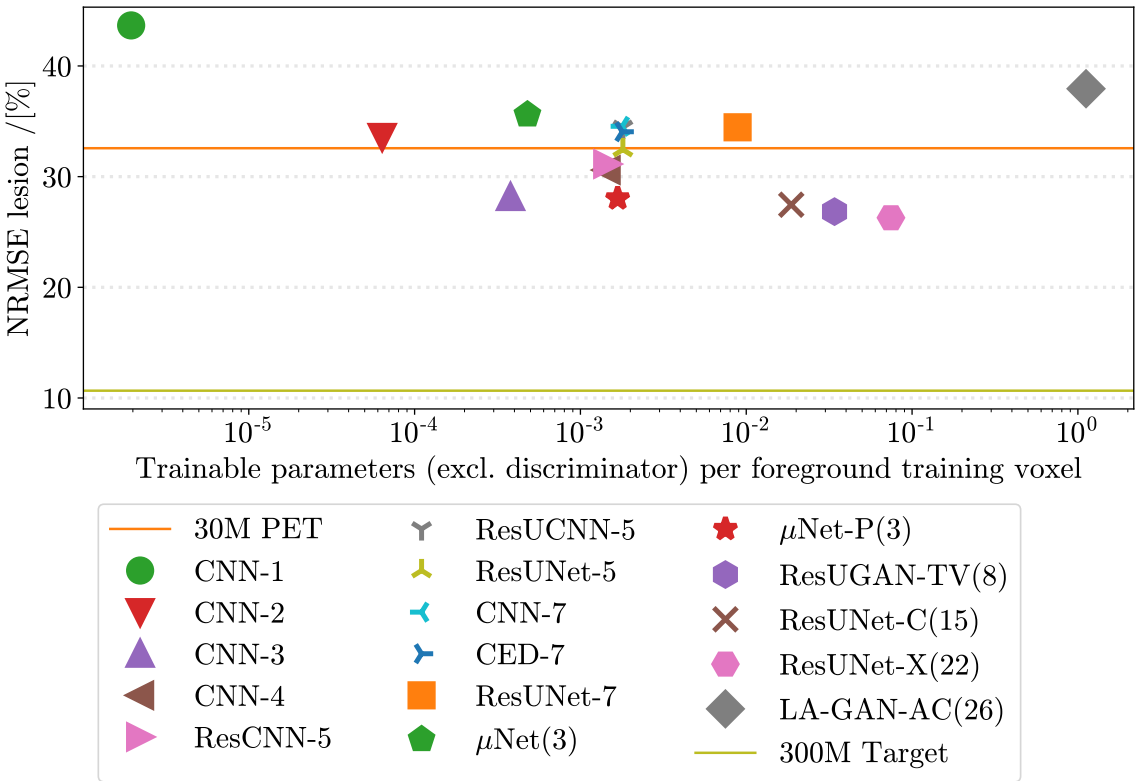


Figure 5.27: Test "lesion" (high-intensity) NRMSE against network complexity.

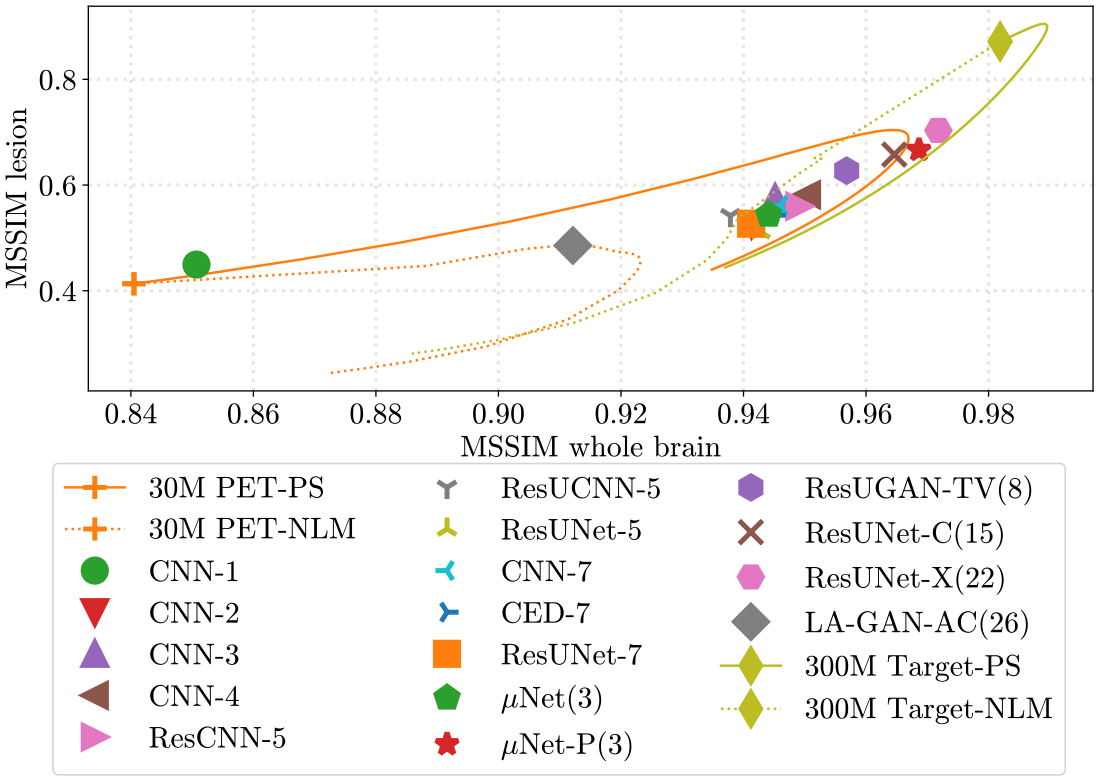


Figure 5.28: Whole-brain versus "lesion" (high-intensity) MSSIM trade-off for various methods (similar to NRMSE results in Figure 5.25).

Trade-offs against network complexity for whole-brain MSSIM are included in Appendix A.3, with results closes matching that of NRMSE from Figure 5.26.

The trade-off for lesions when using MSSIM as a metric (Figure 5.29 below) is also similar to the NRMSE results of Figure 5.26. One notable difference is that all methods improve the lesion MSSIM over the low count input (wheres only some improve lesion NRMSE).

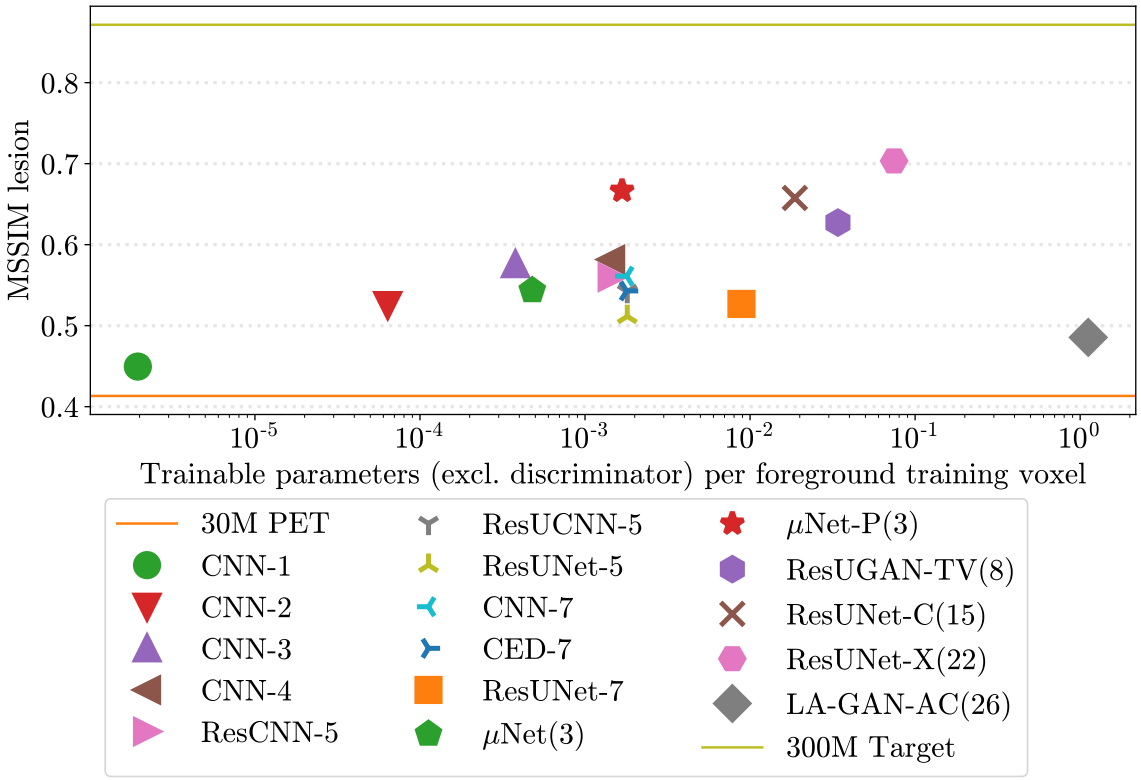


Figure 5.29: Test "lesion" (high-intensity) MSSIM against network complexity (similar to similar to NRMSE results in Figure 5.26).

5.4 Discussion

In all cases, MSSIM appears to correspond very well to visual assessment of the images. Meanwhile, NRMSE values are comparably more tightly distributed, making it harder to use to rank methods. Where time permitted, networks were retrained multiple times (with different randomised initialisation of parameters). The resultant small standard errors (e.g. mostly small or invisible y-errors in Figure 5.15) indicate that statistical flukes in the training process are unlikely to have affected the findings above. However, significantly different datasets (from different anatomical regions and acquired using different scanner manufacturers) would merit further investigation in future to eliminate all potential sources of statistical flukes.

There is a notable difference between the simulation and real patient relative method performance. In particular, the *μNet*, *CNN-7*, *μNet-P*, and *CNN-4* clearly produce the best lesion MSSIM for the simulations (Figure 5.17), while *ResUNet-X* (followed by *μNet-P*, appropriate PS, and *ResUNet-C*) are clearly the best for real patient data (Figure 5.28). However, as mentioned this discrepancy may be due to the fact that real data metrics are evaluated against raw (circa 430 M count) reconstructions rather than a known ground truth. Consider for instance the following hypotheses about how networks can denoise:

1. Autoencoding: a bottleneck (relatively small intermediate "latent space" layer output) forces the network to discard information which is likely of high spatial frequency, which in turn is likely noise.
2. Regularisation: for example, early termination or any other method described in Section 2.2.2. Networks tend to learn low frequency mappings first (resulting in a low-pass filter effect as with autoencoding).
3. Learning the mean: this is a proposed more intuitive explanation; namely learning mean-to-mean mappings. This is why unsupervised or self-supervised approaches such as Noise2Noise [126] can work despite having no higher count training target.

Regarding learning the mean, a useful thought experiment would be to consider repeatedly rolling three dice. The rolls of the first die corresponds to an input signal, while the sum of the other two dice corresponds to an output signal. The analogy of the low-pass filter effect (points 1 and 2 above) is to predict the output is twice the input. Meanwhile learning the mean mapping (point 3) is to predict that the output will be exactly 7 provided that the inputs are in the range $[1, 6]$ (otherwise the output is undefined).

In this extreme example of independent dice rolls, there is no meaningful difference between these viewpoints as the input and output signals are uncorrelated. However when mapping low to full count PET images, there will be a strong correlation and thus a set of mean mappings will be far better at suppressing noise.

This is consistent with the fact that – when the ground truth is used as a reference, as with the simulations – some of the networks results actually produce a whole-brain MSSIM marginally superior to that of the target (Figure 5.17). Therefore, it is possible that the relative network performance rankings using simulated data are more representative of true performance on real clinical data.

Despite the possibility of designing layers to perform specific mathematical operations and algorithms [181], the results here indicate that – for a fixed amount of training data – the specifics of an architecture are not as important as the number of optimisation parameters (i.e. network size, given by the number of weights and biases). This is exemplified by Figure 5.18, where most networks converge to a similar MSSIM given enough training data. Since network size dictates the learning capacity, it would appear that once a certain minimum capacity is achieved to extract as much useful information from the given training data as possible, then architecture ceases to have a significant impact. This finding is consistent with that of Chapter 4, where networks designed and constrained to solving a particular problem are shown to be comparable or even inferior to unconstrained equivalents.

5.5 Summary and Future Work

Overall, for oncological tasks it would appear that Gaussian smoothing remains superior to any of the CNNs investigated here. Meanwhile for whole-brain imaging, it seems likely that low count PET post-processing is best achieved with a relatively shallow *μ-Net*, rather than a deep U-Net. While the 22-layer *ResUNet-X* proved slightly superior with clinical data, this result may be misleading due to a lack of a ground truth as outlined above. Other results from the current literature confirm that skip/concatenation connections in U-nets can be detrimental in certain circumstances [135]. In related work, Klyuzhin et al. attempt to address the problem of voxel-level kinetic modelling in dynamic PET using a CAE, and claim better performance than U-nets [63]. Similarly, Peng et al. [189] empirically find that for multilevel wavelet networks (MWNs), increasing wavelet decomposition levels or increasing convolution layers within each level does not increase denoising ability. Instead, "progressive training" of a multilevel wavelet residual network (MWRN) with "scale-specific loss" is better.

With ever-increasing computational power, however, networks proposed in the literature are likely to become larger. A recent proposal re-purposes a 30-layer CT-denoising cycle-consistent residual GANs (from [190]) to denoise low dose PET [191], and Domingues et al. have reviewed other PET-CT denoising methods [192]. An even larger proposal uses parameterised *Inception*-inspired blocks called "MultiRes" [193]. A very recent proposal by Chen et al. [194] builds upon their earlier proposal (Section 5.2.3 based on [186]) by inserting one more Conv layer at each U-net level and fine-tuning the last residual layer. In light of the results above, it is unlikely that this makes much difference. It should be noted that in theory a very complex network should always outperform a smaller network – if provided sufficient data or regularisation. The additional training time and data requirements for such networks will become more feasible with time.

Future work may analyse the effect of 3D and other input modalities. However, this should help all methods and is unlikely to affect relative rankings.

Low count PET is nevertheless an active area of research, and future work should consider more recently proposed methods [195]. For example, Schramm et al. [144] approximate an MR-guided (asymmetric Bowsher prior [196], [197]) PET reconstruction using a PET (unguided) reconstruction and corresponding MR as input channels to a post-processing CNN (fully 3D, with 9 convolutional layers). The primary demonstrated benefit is a speedup compared to guided reconstruction while maintaining similar recovery coefficients (RCs) and SSIM. However, predicted images remain comparable to full count (20 min scan) guided reconstructions even when the count level of the input PET image is reduced (to a 1 min scan equivalent). Meanwhile, supplementary figures in [144] indicate that MR artefacts and misregistration cause similar errors in both the Bowsher and CNN methods. Interestingly, the additional figures also show that the CNN performs some deblurring (similar to iterative Richardson-Lucy, resulting in Gibbs ringing) if the MR input channel is replaced with a constant (all voxels the same value) volume.

It would also be interesting to incorporate aspects of unsupervised PET denoising methods (such as proposed by Cui et al. [198]), especially as the results in this chapter indicate that better-than-target performance is indeed achievable.

Other important areas for thorough investigation are choice of activation functions and optimisation strategy (including stochastic gradient descent variants as well as loss functions and adversarial training with discriminators). Traditionally, such hyperparameters and design choices are often made empirically after iterative trial-and-error on available datasets. Comparison is also usually only done with other non-DL methods, or with DL proposals not designed with the same specific dataset in mind. For example, Gong et al. [199] propose a perceptual loss as superior to MSE. It would be insightful to include more loss functions in the comparison.

The methods considered in this chapter all perform post-processing in image-space. However, there are also claims that projection-space denoising is superior [200]. While correlations between distant sinogram bins may require a network with large kernels or great depth and width to deal with, such a problem may become more tractable with improving GPGPU hardware in future.

There are many architectural considerations and proposals when designing a CNN. Some of these are likely to work well in a broad range of scenarios. Proposals can thus broadly be split into two categories: firstly, a data-dependant and task-dependant set of choices, and secondly a set of guidelines which work well for a wide variety of data and tasks.

In related work in unsupervised representation learning, Radford et al. [93] propose the following architecture guidelines for CNNs and GANs:

- use strided and fractional strided (also called transposed) convolutions in lieu of pooling;
- use batch normalisation;
- avoid fully connected layers;

- use ReLU for generators layer activations (apart from the output layer, which should use hyperbolic tangent (tanh)), and
- use LReLU for discriminator layer activations.

It would be interesting to investigate whether these guidelines generalise to PET denoising and artefact suppression, and if not, what modifications are required.

Chapter 6

Discussion and Conclusions

6.1 Summary of Key Findings and Contributions

Chapters 1 and 2 serve as introductions for the PET and ML techniques used in the later chapters. Chapter 1 focuses in particular on MLEM reconstruction of sinogram data and post-processing techniques used in PET. It sets out three major goals which all result in lowering acquisition counts: lower radiotracer doses; shorter scan durations, and high frame rates for dynamic PET (effectively many short scans). Lowering counts results in lower SNR, but post-processing methods can help compensate for this. ML techniques suited to image processing tasks – in particular CNNs – are proposed in the current literature for the task of PET post-processing. Chapter 2 covers fundamental ML concepts including backpropagation (Section 2.2), regularisation (Section 2.2.2), CNNs (Section 2.3) and specific applications in PET (Section 2.4).

Chapter 3 describes a novel proposal for post-processing called a micro-network or μ-net due to its comparatively small size versus most other architectures in the current literature. Starting with the smallest possible network (single-layer, single-kernel), more feed-forward layers and kernels are appended until the network's performance is maximised for the given simulated phantom and real patient datasets. Activation functions which have continuous derivatives (Sigmoids and ELUs) are used in lieu of batch normalisation and regularisation to help prevent vanishing gradients, while the test metric (NRMSE) is directly used in the loss function. The proposed μ-net architecture has clearly impressive noise-suppression abilities, reducing NRMSE by up to 89 % compared to MLEM and 72 % compared to RM+PS (Figure 3.23). This network outperforms many far more complex U-nets (as seen in Chapter 5) and the proposed reason is that it is designed to perform PET post-processing on these datasets (unlike U-nets, which were originally designed to perform segmentation of electron microscopy images [123]).

While smaller networks such as μ-nets are demonstrably robust to small amounts of training data and unseen test data (i.e. less likely to overfit), they still nevertheless consist of thousands of empirically-chosen hyperparameters. In order to increase confidence in the methods used, in Chapter 4 the possibility of enforcing sensible (i.e. informed by the system model) constraints on a post-processing CNN is investigated. Since PET acquisitions are subject to (Poisson) noise, full consistency with the data is not actually desirable.

Nevertheless, a null-space network formulation is proposed which enforces consistency with one component of the model – namely, the resolution-modelling image-space PSF. The task of the proposed null-net is to transform MLEM (in this case just the RM component) reconstructions into ground truth predictions – free of ringing artefacts – in a robust manner consistent with the forward model. Consistency in this context means the forward model (smoothing with the system's PSF) should produce identical results whether applied to the input or to the output predictions of the null-net. The iterative R-L reconstruction is actually incorporated into the training of a residual micro-net. While these null-nets do not quite achieve the same level of performance (in terms of NRMSE and MSSIM) as their unconstrained counterparts, marked improvement over traditional post-processing methods is demonstrated. The demonstrated key advantage of null-nets however is the consistency guarantee – smoothing the null-net's inputs and outputs results in nearly identical volumes. This is demonstrated even when tested on patient reconstructions from noisy data (despite training on noise-free simulations and therefore degrading the noisy test data, the degradation is limited since the forward model invariance is maintained). In clinical practice, where robustness and performance guarantees are of paramount importance, null-nets may prove essential for ML-informed post-processing to gain acceptance.

Finally, Chapter 5 compares some of the most promising methods for MR-assisted PET denoising in the current literature for DL up to the year 2020. While there have been more proposals in recent months which have not been included in the comparison, there still remains a lack of a thorough investigation between methods in the current literature. Most publications tend to introduce novel methods without a fair comparison to other state-of-the-art methods, and use significantly different datasets – making relative performance against competing methods impossible unless further independent research is conducted. When trained on the same 2D datasets (which is at least as large as the datasets used in any of the original proposals), it is found that network architecture is surprisingly insignificant. Gradually reducing the amount of training data initially has no effect, but after a point eventually leads to dropping performance (see Figure 5.18 for example). This indicates that in all cases investigated, performance is not hindered if all of the available training data (9 patient volumes, 3 noise realisations each) are used.

A completely novel architectural or approach is needed to increase network performance, if at all possible, since the architectures considered are consistently underwhelming – and especially poor for oncological tasks – when compared to straightforward Gaussian smoothing. Before use in the clinic is possible, more validation studies on a larger data corpus are required for whole-brain imaging tasks, and more work is required to improve performance for oncology.

6.2 Limitations

A major limitation of this work is the data available for the experiments conducted here. At most, data from 20 different patient scans and 21 phantom simulations are used. Using bootstrap sampling or simulations, at most 10 noise realisations are generated for each. Since bias, standard deviation, and NRMSE are each averaged over all voxels (Equations (3.2) to (3.4)), the metrics are robust and unlikely to change if more data is used. In Chapter 5, the performance of all compared methods is found to reach a plateau as the amount of training data is increased, further implying that additional data is not required. Nevertheless, all patient scans and phantom simulations are performed using a Siemens Biograph mMR scanner and geometry, and focus on [^{18}F]FDG brain PET. It is possible that there may be some differences in experimental findings if other scanner geometries, tracers, and anatomical parts are included.

When enforcing model consistency with null-nets (Chapter 4), only a relatively small part of the model (the PSF) is addressed. Future work should consider incorporating the PET projectors into the null-net to potentially obviate this issue. Unlike unrolled methods [133], [138], [139], [175] which use DL for regularisation – thereby subtly altering the model – the null-net approach would remain consistent with the original model. An alternative avenue for progress would be to replace the network ($M(\hat{\boldsymbol{\theta}})$ from Equation (4.9)) with any other classical regularised reconstruction.

This work also focuses on post-processing (with Chapter 4 only incorporating R-L rather than the full MLEM reconstruction into the training process). End-to-end deep learning has not been considered. The latter would involve mapping directly from listmode data or sinograms to image space (as discussed in Section 2.4.1). While the current results are not yet suitable for clinical practice [140], [141], it remains a very worthy area for future investigation. Allowing a network to learn and approximate the entire system matrix could potentially compensate for limitations of MLEM such as bias and "zero trapping" (see Section 1.3.1).

6.3 Unexplored Methods

There are a number of topics in the current literature which have not been fully explored in the chapters above. These topics are either relatively novel developments and/or there was insufficient time to investigate them all here. It should also be noted that some details in proposals from current literature exist as work-arounds. For example, techniques to increase training speed at the slight cost of accuracy in order to avoid lengthy delays in producing results. While these are useful practical techniques, for the purposes of the investigations in Chapters 3 to 5 they are considered to be relatively minor implementation details and therefore not explored. Nevertheless, the following should be examined in future work:

Number of Dimensions In particular, 2D networks are often applied after slicing up 3D medical imaging data (as done in Chapter 5 to conduct a fair comparison). This decreases the computational cost and thus increases training speed. Note that prediction (inference) speeds – while also increased – are relatively insignificant in all cases (effectively the time required for the forward pass, i.e. without backpropagation, of a single training epoch). More importantly, 2D networks have a significantly lower memory cost, thus facilitating very deep designs within the constraints of available GPGPU random access memory (RAM). Training networks involves backpropagation which (apart from exceptional cases such as reversible layers [201]) requires temporary storage of each layer's outputs. Downsampling inputs (or, in the extreme case, dropping a dimension) would thus decrease the memory cost of every subsequent layer. However, this also reduces the amount of potential contextual information available to a network.

Patching Patch-based methods [137], [165], [166] are often proposed for the same memory constraint-related reasons. Depending on the specific method proposed, using patches could be mathematically equivalent to operating on whole images with a specific convolutional stride for the first layer and a specific mini-batch size. It should be noted that other arguments for patch-based methods exist. Class imbalance – where there may be a large number of background voxels in raw images – can be avoided by masking out background patches or weighting foreground patches differently [202]. However, given an appropriate optimisation strategy (for example, only including contributions of foreground voxels when evaluating the loss function) this bias can be avoided. Furthermore, patch-based methods prevent models from over-fitting to features which are larger than the patch size (conversely, they also prevent higher level reasoning using these larger features). The same effect, however, could be achieved by ensuring a similarly limited receptive field in a non-patch-based model.

Data augmentation It should be noted that the number of images/volumes in medical imaging datasets is comparatively small. This is usually due to a combination of ethical and privacy-related reasons as well as cost. Within medical imaging, PET datasets in particular are small. This is because the required ionising radiation is both harmful and expensive, making it nearly impossible to acquire copious amounts of data from healthy volunteers. MRI, by comparison, is a non-ionising modality and thus safe for use in studies involving healthy volunteers. Overall this means DL architectures need to be able to cope with relatively limited amounts of training data. This can be problematic since robustness to unseen (test) data is often achievable by simply using more training data (which may not be available in the case of PET). Augmentation techniques are often work-arounds which do not genuinely add more information during the training phase. For example, generating more data by performing rotations and mirroring could result in a network with essentially symmetric kernels (symmetric kernels are the easiest way to ensure that rotating the input produces a rotated output). While there are scenarios where enforcing symmetry is desirable [203], this is not in general true for PET reconstructions. Test data – as with the pre-augmented training data – is unlikely to be symmetric (for example, scans typically have the patient face-up, meaning certain features would only appear in a certain range of orientations) and thus the test performance will suffer. If suitable for the task, an alternative strategy could be elastic deformations. In any case, augmentation is a form of regularisation which may both improve and degrade performance, and more work is needed to uncover the conditions required for augmentation to be of benefit.

Dropout At each epoch or mini-batch, individual nodes are "dropped" (input and output values ignored, effectively set to zero) with a probability d. Sparsity is thus enforced on layer outputs. During backpropagation (when updating weights), "dropped" connections are not touched [92]. *Dropout* and related regularisation methods (e.g. *DropConnect*, where only outputs are dropped and sparsity is thus enforced on layer weights [204]) have been shown to reduce the chance of overfitting [106]. However, how to choose d is an open question. Arguably, d should be smaller for earlier layers closer to the input – discarding raw input data is unlikely advantageous.

Maxout By taking a maximum across the feature dimension, *maxout* is proposed as a way of combining (rather than omitting like *dropout* does) the models contained in a network [205].

Stochastic pooling This interesting replacement for deterministic pooling (such as e.g. MaxPool) is well-suited to CNNs, and has the same effect as data augmentation using small local perturbations. This is equivalent to building in tolerance for small local deformations, and has been shown to improve performance in classification tasks [206].

Multiple pathways When multi-modal information is available (such as with PET-MR), multi-pathway models process each modality separately through its own pipeline of layers before being merged. Research indicates that equivalent (in terms of number of parameters and layer output sizes) single-pathway networks (where modalities are presented as multi-channel network inputs) are superior [207]. This is likely due to the fact that such single-pathway models have increased joint processing abilities.

Layer density The density of a layer is given by its receptive field (i.e. kernel spatial dimensions). Dense (also called fully connected (FC) or perceptron) layers are designed so that each output element is a function of all input elements. Usually this means having $\mathcal{O}(N^2)$ weights (assuming N input and N output elements), where each output element is a weighted sum of all inputs. Dense layers are thus very costly both in terms of memory usage as well as training time. The large number or parameters also increases the ability of the network to perfectly memorise the training data, thereby overfitting and reducing generalisability to unseen test data. Large amounts of training data as well as GPU memory and time are usually required when using dense layers. Convolutional layers were proposed as low-parameter alternatives to dense layers to avoid these issues. A sufficient number of sequentially-applied convolutional layers are capable of achieving a whole-image receptive field with far fewer parameters; effectively equivalent to a single dense layer parameterised with just $\mathcal{O}(N)$ variables. It should be noted, however, that the increased learning capacity of dense layers may be advantageous in certain scenarios. If the overall network is required to have a certain receptive field, there will be a trade-off between network depth (number of layers) and width (size of kernels). In general, the formula for overall receptive field of a CNN is related to the kernel widths of each layer $(1 + \sum_j s_j - 1$, where s_j is the kernel width for layer j. The receptive field of a single layer is the same as its kernels' width. Subsequent layers expand this receptive field further. However, the central value of each subsequent kernel does not contribute to this expansion, hence the -1 term). Increasing layer density is a way to increase receptive field and perform large matrix operations without introducing more layers, matrix parameterisation/factorisation, and non-linearities. More investigation is needed to determine if there are scenarios where denser layers are advantageous.

Activation functions Some activation functions such as parametric rectified linear unit (PReLU) have not been considered in this work.

Losses Some alternatives such as perceptual loss [199] and norms greater than ℓ_2 have not been considered in this work.

Parameter regularisation While used in some of the methods considered here, a thorough investigation is not conducted on the effect of ℓ_1 and ℓ_2 parameter regularisation. This should provide useful guidelines which can help reduce optimisation difficulty.

Spatial compression While U-nets may spatially compress with depth, the overall concatenation (skip) and/or residual connections mean that such compression is not forced. It would be interesting to include architectures which enforce a bottleneck for comparison.

Some of the methods outlined above are work-arounds which address current practical and computational limitations. Given sufficient time, memory and minor architectural redesign, many proposed work-arounds would not be required. Due to the ever-increasing power of GPGPUs, such work-arounds (especially restricting spatial dimensions) are gradually becoming more and more unnecessary. There are however completely new techniques which have not been explored here. Promising candidates for further investigation are listed below:

Transformer networks These networks have delivered impressive results in the field of natural language processing (NLP), and have been proposed as a replacement for CNNs [208]. More recently, transformers have been proposed for natural image denoising and super-resolution tasks [209], [210]. The literature suggests that CNN performance is limited by the fact that kernels have small local receptive fields, and transformer networks perform better as they have a receptive field which covers the entire input. In some ways this appears to be a surprising conceptual backtrack to fully connected MLP layers. The real reason why transformers may indeed work well in image processing tasks is that – given sufficient training data and GPGPUs memory and time such that overfitting is prevented – a large FC network should be able to outperform a CNN. Indeed, the study in [210] required tens of thousands of days worth of GPGPU training time despite operating on 2D patches of dimensions 16×16 at most, and concludes that large scale training (on datasets of up to 300 M images) "trumps inductive bias." Such scale cannot be achieved in medical imaging currently without heavy use of simulations and augmentations.

Automatic architecture engineering Hyperparameter selection can be at least partially automated using a self-configuring nnU-net [211], the neural architecture search (NAS) framework [147], or a variety of AutoML techniques [212].

6.4 Future Work

DL is clearly a large and active topic of research with many applications and much untapped potential, both in general and in the field of medical imaging. Future work should address the limitations and unexplored methods outlined in this chapter, with the goal of truly automating hyperparameter selection and architecture design. It is also exciting that robustness guarantees can be achieved by building in model consistency, but much work remains to be done in this new approach to ML in PET reconstruction.

Glossary

[^{18}F]FBB florbetaben 19, 130

[^{18}F]FBP florbetapir 19

[^{18}F]FDG fluorodeoxyglucose 14, 17, 19, 20, 22, 61, 66, 68, 112, 123, 130, 165

AD Alzheimer's disease 17, 19, 61

Adam adaptive moment estimation 49, 52, 71, 73, 74, 79, 127, 128, 130, 133, 135, 137, 149

ADMM alternating direction method of multipliers 62

AI artificial intelligence 39

ALARA as low as reasonably achieveable 24, 36

ANN artificial neural network 41, 53, 56, 119

APD avalanche photodiode 21

ARSAC Administration of Radioactive Substances Advisory Committee 20

BCD block coordinate descent 62

BCE binary cross entropy 59, 89

BM3D block-matching in 3D 35

BN batch normalisation 51, 57, 129, 130, 134

CAE convolutional autoencoder 61, 123, 160

CED convolutional encoder-decoder 59, 60

CI catastrophic interference 53, 72

CNN convolutional neural network 35, 40, 53, 54, 55, 56, 58, 60, 61, 62, 63, 64, 65, 66, 71, 74, 75, 99, 100, 102, 103, 105, 107, 108, 109, 110, 111, 112, 114, 117, 118, 119, 120, 121, 135, 147, 160, 161, 163, 164, 167, 168, 169

CNR contrast to noise ratio 61

Conv convolution 55, 56, 71, 73, 129, 130, 132, 134, 160

CRLB Cramer-Rao lower bound 28

CT computed tomography 16, 17, 23, 24, 31, 32, 61, 65, 121, 122, 123, 135, 160

CUDA compute unified device architecture 175

DIP deep image prior 52, 60, 62, 124

DL deep learning 105, 118, 119, 120, 121, 122, 124, 125, 161, 164, 165, 167, 169

DNN deep neural network 58

ED effective dose 20

ELU exponential linear unit 57, 82, 84, 95, 98, 101, 109, 132, 136, 163

EM expectation maximisation 28

EMA exponential moving average 46

FBP filtered backprojection 17, 21, 25, 105

FC fully connected 56, 168, 169

FoV field of view 21, 22, 23, 25, 26

FWHM full width at half maximum 21, 22, 68, 72, 75, 110, 112

GAN generative adversarial network 35, 59, 89, 98, 102, 105, 123, 160, 161

GPGPU general-purpose graphics processing unit 41, 71, 77, 161, 166, 169

ID3 iterative dichotomiser 3 66

KEM kernel expecation maximisation 33

KL Kullback-Leibler 107

L-BFGS limited-memory Broyden-Fletcher-Goldfarb-Shanno 52

lerp linear interpolation 57, 129

LLF local linear fitting 62

LMS least mean square 43

LoR line of response 20, 21, 23, 24, 25, 26, 68

LReLU leaky rectified linear unit 57, 130, 134, 162

LS least squares 107

LSO lutetium oxyorthosilicate 21

MAP maximum a-posteriori 27

MaxPool maximum-value pooling 129, 130, 167

MCI mild cognitive impairment 17

MeanPool mean-value pooling 130, 131

ML machine learning 35, 37, 38, 39, 40, 50, 61, 62, 65, 105, 119, 163, 164, 169

MLAA maximum likelihood activity and attenuation 61, 124

MLE maximu likelihood estimation 28

MLEM maximum likelihood expectation maximisation 17, 25, 28, 29, 33, 38, 39, 43, 49, 68, 73, 87, 89, 98, 100, 102, 104, 106, 107, 108, 113, 117, 118, 120, 163, 164, 165

MLP multilayer perceptron 42, 46, 47, 51, 53, 56, 169

MoDL model-based deep learning 106

MPRAGE magnetisation-prepared rapid acquisition with gradient echo 17, 68, 69

MR magnetic resonance 17, 22, 28, 32, 60, 61, 63, 64, 65, 66, 67, 68, 69, 70, 71, 73, 74, 75, 78, 82, 89, 99, 102, 103, 106, 119, 121, 122, 124, 125, 126, 128, 130, 131, 134, 140, 142, 143, 161, 164, 168

MRI magnetic resonance imaging 17, 32, 61, 68, 106, 167

MSE mean square error 31, 43, 46, 73, 79, 123, 161

MSSIM mean structural similarity index 110, 111, 114, 115, 117, 118, 123, 147, 148, 155, 157, 158, 159, 164, 178, 179

MWN multilevel wavelet network 160

MWRN multilevel wavelet residual network 160

NAG Nesterov accelerated gradient 49, 52

NAS neural architecture search 169

NLM non-local means 9, 32, 33, 35, 37, 62, 70, 71, 75, 77, 82, 84, 85, 87, 89, 91, 93, 95, 100, 102, 120, 125, 128, 143, 155, 175

NLP natural language processing 169

NMI normalised mutual information 35

NN neural network 39, 53

NRMSE normalised root mean square error 9, 69, 70, 73, 74, 75, 83, 84, 87, 89, 90, 93, 94, 95, 98, 100, 101, 102, 108, 109, 110, 111, 114, 115, 117, 118, 123, 127, 128, 130, 137, 138, 143, 145, 146, 147, 150, 155, 157, 158, 163, 164, 165

ODE ordinary differential equation 121

OSEM ordered subsets expectation maximisation 28, 61, 62, 127, 128, 129, 131, 132, 135, 138, 143, 155

PAT photoacoustic tomography 105

PET positron emission tomography 14, 16, 17, 19, 20, 21, 22, 23, 24, 25, 26, 29, 31, 32, 33, 35, 36, 37, 39, 40, 47, 60, 61, 63, 64, 65, 66, 67, 68, 69, 70, 71, 73, 74, 75, 77, 78, 82, 87, 99, 100, 101, 102, 103, 104, 105, 106, 107, 109, 110, 112, 113, 118, 119, 120, 121, 122, 123, 126, 128, 129, 130, 131, 134, 136, 138, 140, 142, 143, 152, 159, 160, 161, 162, 163, 164, 165, 167, 168, 169

PReLU parametric rectified linear unit 168

PS Gaussian post-smoothing 30, 64, 70, 71, 75, 82, 87, 93, 98, 100, 102, 105, 125, 143, 147, 155, 158, 163

PSF point spread function 26, 29, 30, 75, 104, 105, 106, 107, 112, 118, 164, 165

PSMA prostate-specific membrane antigen 19

PSNR peak signal to noise ratio 37, 61, 69, 122, 123, 130

PVE partial volume effect 74, 98, 101

RAM random access memory 166

RC recovery coefficient 161

R-L Richardson-Lucy 106, 107, 108, 117, 118, 164, 165

RECIST response evaluation criteria in solid tumors 16

ReLU rectified linear unit 45, 52, 57, 84, 101, 109, 129, 130, 162

ResNet residual network 57

RM resolution modelling 29, 30, 75, 78, 89, 98, 100, 104, 105, 106, 110, 113, 114, 118, 163, 164

RMSE root mean square error 61

RMSProp root mean square propagation 130, 149

RNN recurrent neural network 53

RoI region of interest 31, 95, 142, 143, 145, 155

SGD stochastic gradient descent 44, 48, 51, 52, 53

SISR single-image super-resolution 35

SNR signal to noise ratio 29, 36, 121, 123, 163

SPECT single photon emission computed tomography 20

SPGR spoiled gradient-recalled sequence 17, 68

SPM12 Statistical Parametric Mapping version 12 35, 69

SR super-resolution 35

SSIM structural similarity index 61, 130, 135, 161

SUV standardised uptake value 123

SUV_{max} maximum standardised uptake value 17, 29, 142

SVD singular value decomposition 105

tanh hyperbolic tangent 57, 162

ToF time of flight 21, 61

TV total variation 6, 31, 32, 52, 62, 64, 89, 90, 95, 98, 101, 120

UAT universal approximation theorem 45, 56

VSM voxel-driven scatter model 68

WRN wide residual network 66

Appendix A

Algorithm Implementations (Source Code)

A.1 Non-local means (NLM) guided filtering

The following are 3D and 2D compute unified device architecture (CUDA) based implementations of NLM as described in Equation (1.17). PyCUDA [213] is used to expose these definitions for use in Python.

```
#define DTYPE float
#define SIZE_T int64
#define B_DIM_X 16
#define B_DIM_Y 16
#define B_DIM_Z 4
#define N_DIM_X 344
#define N_DIM_Y 344
#define N_DIM_Z 127


/**
 * dst         : destination (output) volume
 * img         : input volume
 * ref         : reference (guidance) volume
 * exp_norm    : hyperparameter = -1.0 / (2.0 * sigma * sigma)
 * HALF_WIDTH  : half the neighbourhood width (excluding central voxel)
 */
__global__ void nlm3d(DTYPE dst[N_DIM_X][N_DIM_Y][N_DIM_Z],
```

```
                    const DTYPE img[N_DIM_X][N_DIM_Y][N_DIM_Z],
                    const DTYPE ref[N_DIM_X][N_DIM_Y][N_DIM_Z],
                    const DTYPE exp_norm, const SIZE_T HALF_WIDTH) {
  SIZE_T x = blockIdx.x * B_DIM_X + threadIdx.x; if (x >= N_DIM_X) return;
  SIZE_T y = blockIdx.y * B_DIM_Y + threadIdx.y; if (y >= N_DIM_Y) return;
  SIZE_T z = blockIdx.z * B_DIM_Z + threadIdx.z; if (z >= N_DIM_Z) return;

  DTYPE result = 0;
  DTYPE norm = 0;
  DTYPE weight;
  for (SIZE_T i = max(0, x - HALF_WIDTH); i < min(N_DIM_X, x + HALF_WIDTH); ++i) {
    for (SIZE_T j = max(0, y - HALF_WIDTH); j < min(N_DIM_Y, y + HALF_WIDTH); ++j) {
      for (SIZE_T k = max(0, z - HALF_WIDTH); k < min(N_DIM_Z, z + HALF_WIDTH); ++k) {
        weight = exp(pow(ref[i][j][k] - ref[x][y][z], 2) * exp_norm);
        // distance weight: doesn't work as well for minimising NRMSE vs Truth
        // weight /= (0.5 + sqrt(pow((i - x), 2) + pow((j - y), 2) + pow((k - z), 2)));
        result += weight * img[i][j][k];
        norm += weight;
      }
    }
  }

  dst[x][y][z] = norm == 0 ? img[x][y][z] : result / norm;
}

#define B_DIM_X 32
#define B_DIM_Y 32

/// 2D version of nlm3d
__global__ void
nlm2d(DTYPE dst[N_DIM_X][N_DIM_Y], const DTYPE img[N_DIM_X][N_DIM_Y],
      const DTYPE ref[N_DIM_X][N_DIM_Y],
      const DTYPE exp_norm,
      const SIZE_T HALF_WIDTH) {
  SIZE_T x = blockIdx.x * B_DIM_X + threadIdx.x; if (x >= N_DIM_X) return;
  SIZE_T y = blockIdx.y * B_DIM_Y + threadIdx.y; if (y >= N_DIM_Y) return;
```

```
  DTYPE result = 0;
  DTYPE norm = 0;
  DTYPE weight;
  for (SIZE_T i = max(0, x - HALF_WIDTH); i < min(N_DIM_X, x + HALF_WIDTH); ++i) {
    for (SIZE_T j = max(0, y - HALF_WIDTH); j < min(N_DIM_Y, y + HALF_WIDTH); ++j) {
      weight = exp(pow(ref[i][j] - ref[x][y], 2) * exp_norm);
      result += weight * img[i][j];
      norm += weight;
    }
  }

  dst[x][y] = norm == 0 ? img[x][y] : result / norm;
}
```

A.2 Simulations comparison: network complexity for whole-brain and lesion mean structural similarity

Trade-offs against network complexity for whole-brain and lesion MSSIM are show in Figures A.1 and A.2, respectively.

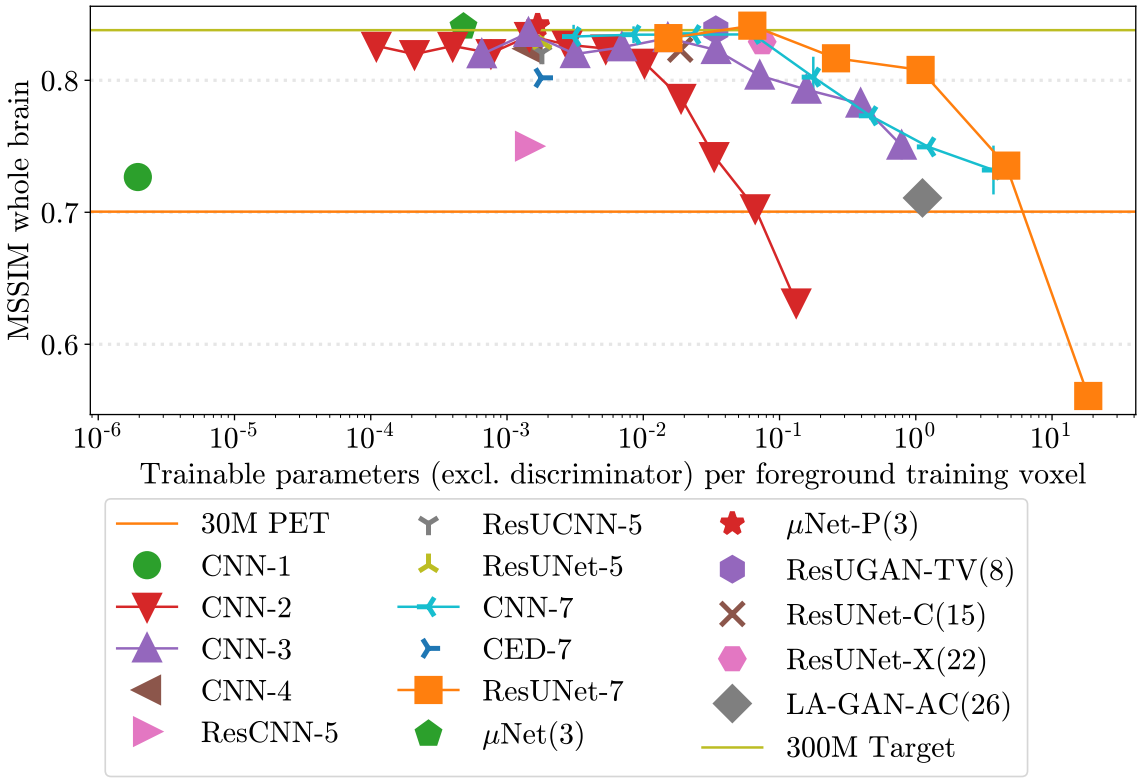


Figure A.1: Test whole-brain MSSIM against network complexity (similar to NRMSE results in Figure 5.15).

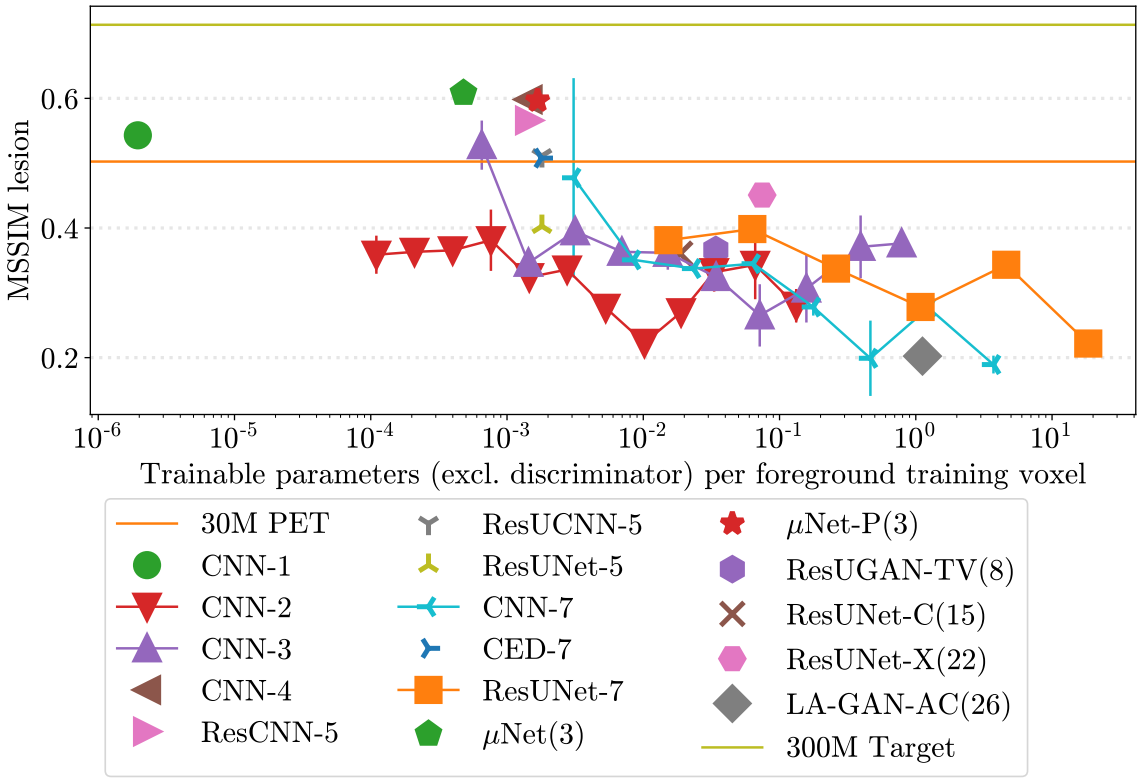


Figure A.2: Test **lesion** MSSIM against network complexity (similar to NRMSE results in Figure 5.16).

A.3 Patient data comparison: network complexity for whole-brain mean structural similarity

Trade-offs against network complexity for whole-brain MSSIM are show in Figure A.3.

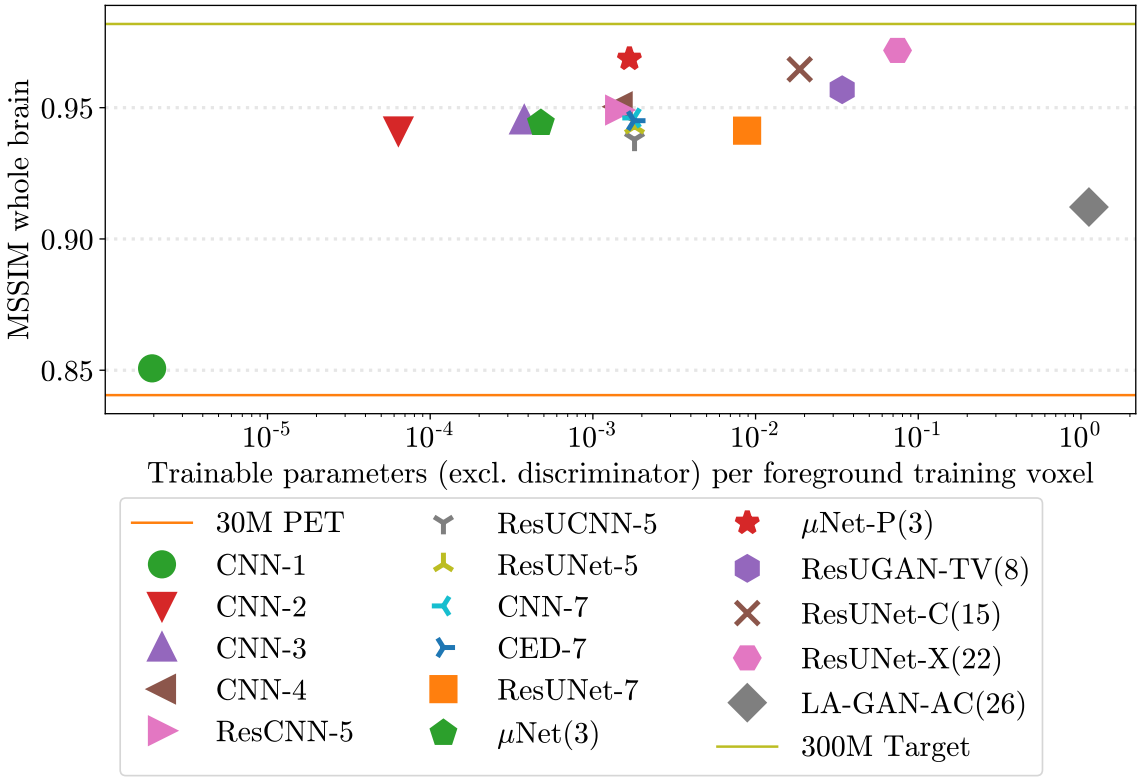


Figure A.3: Test whole-brain MSSIM against network complexity (similar to NRMSE results in Figure 5.26).

Publications

The following works have been produced as part of or in support of the research presented in this thesis.

Journal Articles

P. J. Markiewicz *et al.*, "Uncertainty analysis of MR-PET image registration for precision neuro-PET imaging," *Neuroimage*, vol. 232, p. 117821, May 2021, doi: 10.1016/j.neuroimage.2021.117821

C. O. da Costa-Luis and A. J. Reader, "Micro-Networks for Robust MR-Guided Low Count PET Imaging," *IEEE Trans. Radiat. Plasma Med. Sci.*, vol. 5, no. 2, pp. 202–212, Mar. 2021, doi: 10.1109/TRPMS.2020.2986414

A. J. Reader, G. Corda, A. Mehranian, C. da Costa-Luis, S. Ellis, and J. A. Schnabel, "Deep Learning for PET Image Reconstruction," *IEEE Trans. Radiat. Plasma Med. Sci.*, vol. 5, no. 1, pp. 1–25, Jan. 2021, doi: 10.1109/TRPMS.2020.3014786

E. Ovtchinnikov *et al.*, "SIRF: Synergistic Image Reconstruction Framework," *Comput. Phys. Commun.*, vol. 249, 2020, doi: 10.1016/j.cpc.2019.107087

J. Bland *et al.*, "Intercomparison of MR-informed PET image reconstruction methods," *Med. Phys.*, vol. 46, no. 11, 2019, doi: 10.1002/mp.13812

Conference Proceedings

C. O. da Costa-Luis and A. J. Reader, "3D Convolutional Adversarial Micro-networks for Low Count PET-MR Post-processing," in *2019 IEEE Nuclear Science Symposium and Medical Imaging Conference (NSS/MIC)*, 2019, doi: 10.1109/NSS/MIC42101.2019.9059905

C. O. da Costa-Luis and A. J. Reader, "Convolutional micro-networks for MR-guided low-count PET image processing," in *2018 IEEE Nuclear Science Symposium and Medical Imaging Conference Proceedings (NSS/MIC)*, 2018, doi: 10.1109/NSSMIC.2018.8824373

J. Bland *et al.*, "Intercomparison of MR-Informed Methods for PET Image Reconstruction," in *2018 IEEE Nuclear Science Symposium and Medical Imaging Conference Proceedings (NSS/MIC)*, 2018, doi: 10.1109/NSSMIC.2018.8824610

C. O. da Costa-Luis and A. J. Reader, "Deep Learning for Suppression of Resolution-Recovery Artefacts in MLEM PET Image Reconstruction," in *2017 IEEE Nuclear Science Symposium and Medical Imaging Conference (NSS/MIC)*, 2017, doi: 10.1109/NSSMIC.2017.8532624

E. Ovtchinnikov *et al.*, "SIRF: Synergistic Image Reconstruction Framework," in *2017 IEEE Nuclear Science Symposium and Medical Imaging Conference (NSS/MIC)*, 2017, doi: 10.1109/NSSMIC.2017.8532815

Software

C. O. da Costa-Luis, "`NiftyML`: PET-MR Machine Learning," *Zenodo/NiftyPET* v0.0.0, 2021, doi: 10.5281/zenodo.4654096

C. O. da Costa-Luis, "`CuVec`: Unifying Python/C++/CUDA memory," *Zenodo/AMYPAD* v2.7.2, 2021, doi: 10.5281/zenodo.4446211

C. O. da Costa-Luis, "`SPM12`: Python wrapper around MATLAB and SPM12," *Zenodo/AMYPAD* v1.0.6, 2021, doi: 10.5281/zenodo.4272003

C. O. da Costa-Luis, "`miutil`: Medical imaging utilities," *Zenodo/AMYPAD* v0.7.2, 2021, doi: 10.5281/zenodo.4281542

P. J. Markiewicz and C. O. da Costa-Luis, "`NIMPA`: High-throughput neuroimage processing and analysis," *Zenodo/NiftyPET* v2.1.0, 2021, doi: 10.5281/zenodo.4417633

C. O. da Costa-Luis, P. J. Markiewicz, "`NIPET`: High-throughput neuroimage PET reconstruction," *Zenodo/NiftyPET* v2.0.0, 2021, doi: 10.5281/zenodo.4417679

E. Ovtchinnikov *et al.*, "`SIRF`: Synergistic Image Reconstruction Framework," *Zenodo/CCPPETMR* v2.2.0, 2020, doi: 10.5281/zenodo.2707911

E. Pasca *et al.*, "`SIRF-SuperBuild`," *Zenodo/SyneRBI* v2.2.0, 2020, doi: 10.5281/zenodo.4408776

C. O. da Costa-Luis, "BrainWeb-based multimodal models of 20 normal brains," *Zenodo* v1.6.0, 2020, doi: 10.5281/zenodo.3269888

C. O. da Costa-Luis, "`brain_phantom`," *Zenodo* v0.1.0-rc3, 2017, doi: 10.5281/zenodo.596266

References

[1] P. Therasse *et al.*, "New Guidelines to Evaluate the Response to Treatment in Solid Tumors," *JNCI: Journal of the National Cancer Institute*, vol. 92, no. 3, pp. 205–216, Feb. 2000, doi: 10.1093/jnci/92.3.205.

[2] E. A. Eisenhauer *et al.*, "New response evaluation criteria in solid tumours: Revised RECIST guideline (version 1.1)," *European Journal of Cancer*, vol. 45, no. 2, pp. 228–247, Jan. 2009, doi: 10.1016/j.ejca.2008.10.026.

[3] R. Boellaard *et al.*, "FDG PET and PET/CT: EANM procedure guidelines for tumour PET imaging: version 1.0," *European Journal of Nuclear Medicine and Molecular Imaging*, vol. 37, no. 1, pp. 181–200, Jan. 2010, doi: 10.1007/s00259-009-1297-4.

[4] D. R. Baldwin, B. White, M. Schmidt-Hansen, A. R. Champion, and A. M. Melder, "Diagnosis and treatment of lung cancer: summary of updated NICE guidance," *BMJ*, vol. 342, no. 1, pp. d2110–d2110, Apr. 2011, doi: 10.1136/bmj.d2110.

[5] National Institute for Health and Care Excellence, "Lung cancer: diagnosis and management." 2019 [Online]. Available: https://www.nice.org.uk/guidance/ng122

[6] Y. F. Tai, "Applications of positron emission tomography (PET) in neurology," *J. Neurol. Neurosurg. Psychiatry*, vol. 75, no. 5, pp. 669–676, May 2004, doi: 10.1136/jnnp.2003.028175.

[7] J. P. Mugler and J. R. Brookeman, "Three-dimensional magnetization-prepared rapid gradient-echo imaging (3D MP RAGE)," *Magnetic Resonance in Medicine*, vol. 15, no. 1, pp. 152–157, Jul. 1990, doi: 10.1002/mrm.1910150117.

[8] A. Jacoby, D. Snape, and G. A. Baker, "Epilepsy and social identity: the stigma of a chronic neurological disorder," *The Lancet Neurology*, vol. 4, no. 3, pp. 171–178, Mar. 2005, doi: 10.1016/S1474-4422(05)01014-8.

[9] J. W. Sander, "The epidemiology of epilepsy revisited," *Current Opinion in Neurology*, vol. 16, no. 2, pp. 165–170, Apr. 2003, doi: 10.1097/01.wco.0000063766.15877.8e.

[10] I. Boscolo Galazzo *et al.*, "Cerebral metabolism and perfusion in MR-negative individuals with refractory focal epilepsy assessed by simultaneous acquisition of 18 F-FDG PET and arterial spin labeling," *NeuroImage: Clinical*, vol. 11, pp. 648–657, 2016, doi: 10.1016/j.nicl.2016.04.005.

[11] T. Jones and D. Townsend, "History and future technical innovation in positron emission tomography," *Journal of Medical Imaging*, vol. 4, no. 1, p. 011013, Mar. 2017, doi: 10.1117/1.JMI.4.1.011013.

[12] R. Nutt, "The History of Positron Emission Tomography," *Molecular Imaging & Biology*, vol. 4, no. 1, pp. 11–26, Feb. 2002, doi: 10.1016/S1095-0397(00)00051-0.

[13] G. L. Brownell, "A History of Positron Imaging," MIT, 1999.

[14] A. P. Dempster, N. M. Laird, and D. B. Rubin, "Maximum Likelihood from Incomplete Data Via the EM Algorithm," *Journal of the Royal Statistical Society: Series B (Methodological)*, vol. 39, no. 1, pp. 1–22, Sep. 1977, doi: 10.1111/j.2517-6161.1977.tb01600.x.

[15] L. A. Shepp and Y. Vardi, "Maximum Likelihood Reconstruction for Emission Tomography," *IEEE Trans Med. Imaging*, vol. 1, no. 2, pp. 113–122, 1982, doi: 10.1109/TMI.1982.4307558.

[16] E. Ü. Mumcuoglu, R. M. Leahy, S. R. Cherry, and E. Hoffman, "Accurate geometric and physical response modelling for statistical image reconstruction in high resolution PET," in *1996 IEEE NSS Conf. Rec.*, 1996, vol. 3, pp. 1569–1573, doi: 10.1109/NSSMIC.1996.587924.

[17] J. Nuyts, "Unconstrained image reconstruction with resolution modelling does not have a unique solution," *EJNMMI Physics*, vol. 1, no. 1, p. 98, 2014, doi: 10.1186/s40658-014-0098-4.

[18] E. Picano *et al.*, "The appropriate and justified use of medical radiation in cardiovascular imaging: a position document of the ESC Associations of Cardiovascular Imaging, Percutaneous Cardiovascular Interventions and Electrophysiology," *European Heart Journal*, vol. 35, no. 10, pp. 665–672, Mar. 2014, doi: 10.1093/eurheartj/eht394.

[19] V. Cunningham *et al.*, "A method of studying pharmacokinetics in man at picomolar drug concentrations." *British Journal of Clinical Pharmacology*, vol. 32, no. 2, pp. 167–172, Aug. 1991, doi: 10.1111/j.1365-2125.1991.tb03877.x.

[20] M. E. Phelps, S. C. Huang, E. J. Hoffman, C. Selin, L. Sokoloff, and D. E. Kuhl, "Tomographic measurement of local cerebral glucose metabolic rate in humans with (F-18)2-fluoro-2-deoxy-D-glucose: Validation of method," *Annals of Neurology*, vol. 6, no. 5, pp. 371–388, Nov. 1979, doi: 10.1002/ana.410060502.

[21] C. C. Rowe *et al.*, "18F-Florbetaben PET beta-amyloid binding expressed in Centiloids," *European Journal of Nuclear Medicine and Molecular Imaging*, vol. 44, no. 12, pp. 2053–2059, Nov. 2017, doi: 10.1007/s00259-017-3749-6.

[22] Administration of Radioactive Substances Advisory Committee, "Notes for guidance on the clinical administration of radiopharmaceuticals and use of sealed radioactive sources." Public Health England, 2021 [Online]. Available: www.gov.uk/arsac

[23] R. Boellaard *et al.*, "FDG PET/CT: EANM procedure guidelines for tumour imaging: version 2.0," *European Journal of Nuclear Medicine and Molecular Imaging*, vol. 42, no. 2, pp. 328–354, Feb. 2015, doi: 10.1007/s00259-014-2961-x.

[24] G. Delso *et al.*, "Performance Measurements of the Siemens mMR Integrated Whole-Body PET/MR Scanner," *Journal of Nuclear Medicine*, vol. 52, no. 12, pp. 1914–1922, Dec. 2011, doi: 10.2967/jnumed.111.092726.

[25] P. Lecoq, "Pushing the Limits in Time-of-Flight PET Imaging," *IEEE Transactions on Radiation and Plasma Medical Sciences*, vol. 1, no. 6, pp. 473–485, Nov. 2017, doi: 10.1109/TRPMS.2017.2756674.

[26] R. M. Mersereau and A. V. Oppenheim, "Digital reconstruction of multidimensional signals from their projections," *Proceedings of the IEEE*, vol. 62, no. 10, pp. 1319–1338, 1974, doi: 10.1109/PROC.1974.9625.

[27] B. D. Man and S. Basu, "Distance-driven projection and backprojection in three dimensions," *Phys. Medicine Biol.*, vol. 49, no. 11, pp. 2463–2475, Jun. 2004, doi: 10.1088/0031-9155/49/11/024.

[28] L. A. Shepp and B. F. Logan, "The Fourier reconstruction of a head section," *IEEE Transactions on Nuclear Science*, vol. 21, no. 3, pp. 21–43, Jun. 1974, doi: 10.1109/TNS.1974.6499235.

[29] W. W. Moses, "Fundamental limits of spatial resolution in PET," *Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment*, vol. 648, pp. S236–S240, Aug. 2011, doi: 10.1016/j.nima.2010.11.092.

[30] A. Sánchez-Crespo, P. Andreo, and S. A. Larsson, "Positron flight in human tissues and its influence on PET image spatial resolution," *European Journal of Nuclear Medicine and Molecular Imaging*, vol. 31, no. 1, pp. 44–51, Jan. 2004, doi: 10.1007/s00259-003-1330-y.

[31] K. Shibuya *et al.*, "Annihilation photon acollinearity in PET: volunteer and phantom FDG studies," *Physics in Medicine and Biology*, vol. 52, no. 17, pp. 5249–5261, Sep. 2007, doi: 10.1088/0031-9155/52/17/010.

[32] J. M. Anton-Rodriguez *et al.*, "Experimental validation of estimated spatially variant radioisotope-specific point spread functions using published positron range simulations and fluorine-18 measurements," *Physics in Medicine and Biology*, vol. 63, no. 24, 2018, doi: 10.1088/1361-6560/aaecb6.

[33] C. S. Levin and E. J. Hoffman, "Calculation of positron range and its effect on the fundamental limit of positron emission tomography system spatial resolution," *Physics in Medicine and Biology*, vol. 45, no. 2, p. 559, Feb. 2000, doi: 10.1088/0031-9155/45/2/501. [Online]. Available: https://iopscience.iop.org/article/10.1088/0031-9155/45/2/501

[34] C. Catana, Y. Wu, M. S. Judenhofer, J. Qi, B. J. Pichler, and S. R. Cherry, "Simultaneous acquisition of multislice PET and MR images: initial results with a MR-compatible PET scanner." *Journal of nuclear medicine : official publication, Society of Nuclear Medicine*, vol. 47, no. 12, pp. 1968–76, Dec. 2006.

[35] B. E. Hammer, N. L. Christensen, and B. G. Heil, "Use of a magnetic field to increase the spatial resolution of positron emission tomography," *Medical Physics*, vol. 21, no. 12, pp. 1917–1920, Dec. 1994, doi: 10.1118/1.597178.

[36] P. J. Markiewicz *et al.*, "Rapid processing of PET list-mode data for efficient uncertainty estimation and data analysis," *Phys Med. Biol.*, vol. 61, no. 13, pp. N322–N336, Jul. 2016, doi: 10.1088/0031-9155/61/13/N322.

[37] T. Yamaya *et al.*, "A proposal of an open PET geometry," *Physics in Medicine and Biology*, vol. 53, no. 3, pp. 757–773, Feb. 2008, doi: 10.1088/0031-9155/53/3/015.

[38] P. E. Kinahan, D. W. Townsend, T. Beyer, and D. Sashin, "Attenuation correction for a combined 3D PET/CT scanner," *Medical Physics*, vol. 25, no. 10, pp. 2046–2053, Oct. 1998, doi: 10.1118/1.598392.

[39] E. Z. Xu, N. A. Mullani, K. L. Gould, and W. L. Anderson, "A segmented attenuation correction for PET," *Journal of Nuclear Medicine*, vol. 32, no. 1, pp. 161–165, 1991.

[40] L. Venneri *et al.*, "Cancer risk from professional exposure in staff working in cardiac catheterization laboratory: Insights from the National Research Council's Biological Effects of Ionizing Radiation VII Report," *American Heart Journal*, vol. 157, no. 1, pp. 118–124, Jan. 2009, doi: 10.1016/j.ahj.2008.08.009.

[41] B. L. Cohen, "Catalog of Risks Extended and Updated," *Health Physics*, vol. 61, no. 3, pp. 317–335, Sep. 1991, doi: 10.1097/00004032-199109000-00002.

[42] T. Murano *et al.*, "Evaluation of the risk of radiation exposure from new 18FDG PET/CT plans versus conventional X-ray plans in patients with pediatric cancers," *Annals of Nuclear Medicine*, vol. 24, no. 4, pp. 261–267, 2010, doi: 10.1007/s12149-010-0342-5.

[43] C. Catana, "The Dawn of a New Era in Low-Dose PET Imaging," *Radiology*, vol. 290, no. 3, pp. 657–658, Mar. 2019, doi: 10.1148/radiol.2018182573.

[44] D. Bailey, B. Bendriem, and D. W. Townsend, "The theory and practice of 3D PET," in *Dordrecht*, Kluwer Academic Publishers, 1998, pp. 55–109.

[45] M. A. Belzunce and A. J. Reader, "Assessment of the impact of modeling axial compression on PET image reconstruction," *Med. Phys*, vol. 44, no. 10, pp. 5172–5186, Oct. 2017, doi: 10.1002/mp.12454.

[46] F. J. López-González *et al.*, "Intensity normalization methods in brain FDG-PET quantification," *NeuroImage*, vol. 222, no. August, p. 117229, Nov. 2020, doi: 10.1016/j.neuroimage.2020.117229.

[47] R. M. Leahy and J. Qi, "Statistical approaches in quantitative positron emission tomography," *Statistics and Computing*, vol. 10, pp. 147–165, 2000, doi: 10.1023/A:1008946426658.

[48] T. F. Budinger, "PET instrumentation: What are the limits?" *Seminars in Nuclear Medicine*, vol. 28, no. 3, pp. 247–267, Jul. 1998, doi: 10.1016/S0001-2998(98)80030-5.

[49] P. J. Green, "On Use of the EM Algorithm for Penalized Likelihood Estimation," *Journal of the Royal Statistical Society: Series B (Methodological)*, vol. 52, no. 3, pp. 443–452, Jul. 1990, doi: 10.1111/j.2517-6161.1990.tb01798.x.

[50] H. H. Barrett, D. W. Wilson, and B. M. W. Tsui, "Noise properties of the EM algorithm. I. Theory," *Physics in Medicine and Biology*, vol. 39, no. 5, pp. 833–846, May 1994, doi: 10.1088/0031-9155/39/5/004.

[51] J. D. Schaefferkoetter *et al.*, "Quantitative Accuracy and Lesion Detectability of Low-Dose 18 F-FDG PET for Lung Cancer Screening," *Journal of Nuclear Medicine*, vol. 58, no. 3, pp. 399–405, Mar. 2017, doi: 10.2967/jnumed.116.177592.

[52] P. E. B. Vaissier, M. C. Goorden, A. B. Taylor, and F. J. Beekman, "Fast Count-Regulated OSEM Reconstruction With Adaptive Resolution Recovery," *IEEE Transactions on Medical Imaging*, vol. 32, no. 12, pp. 2250–2261, Dec. 2013, doi: 10.1109/TMI.2013.2279851.

[53] K. Van Slambrouck *et al.*, "Bias Reduction for Low-Statistics PET: Maximum Likelihood Reconstruction With a Modified Poisson Distribution," *IEEE Transactions on Medical Imaging*, vol. 34, no. 1, pp. 126–136, Jan. 2015, doi: 10.1109/TMI.2014.2347810.

[54] K. Lange, "Convergence of EM image reconstruction algorithms with Gibbs smoothing," *IEEE Transactions on Medical Imaging*, vol. 9, no. 4, pp. 439–446, 1990, doi: 10.1109/42.61759.

[55] C. Cloquet and M. Defrise, "MLEM and OSEM deviate from the cramer-rao bound at low counts," *IEEE Trans Nucl. Sci.*, vol. 60, no. 1, pp. 134–143, 2013, doi: 10.1109/TNS.2012.2217988.

[56] A. Rahmim, J. Qi, and V. Sossi, "Resolution modeling in PET imaging: Theory, practice, benefits, and pitfalls," *Med. Phys*, vol. 40, no. 6, p. 064301, 2013, doi: 10.1118/1.4800806.

[57] S. Tong, a. M. Alessio, and P. E. Kinahan, "Noise and signal properties in PSF-based fully 3D PET image reconstruction: an experimental evaluation." *Phys Med. Biol.*, vol. 55, no. 5, pp. 1453–1473, 2010, doi: 10.1088/0031-9155/55/5/013.

[58] F. L. Andersen, T. L. Klausen, A. Loft, T. Beyer, and S. Holm, "Clinical evaluation of PET image reconstruction using a spatial resolution model," *Eur. J. Radiol.*, vol. 82, no. 5, pp. 862–869, 2013, doi: 10.1016/j.ejrad.2012.11.015.

[59] A. M. Alessio, A. Rahmim, and C. G. Orton, "Resolution modeling enhances PET imaging," *Med. Phys*, vol. 40, no. 12, p. 120601, Oct. 2013, doi: 10.1118/1.4821088.

[60] S.-L. Hu, Z.-Y. Yang, Z.-R. Zhou, X.-J. Yu, B. Ping, and Y.-J. Zhang, "Role of SUVmax obtained by 18F-FDG PET/CT in patients with a solitary pancreatic lesion," *Nucl. Med. Commun.*, vol. 34, no. 6, pp. 533–539, 2013, doi: 10.1097/MNM.0b013e328360668a.

[61] O. L. Munk, L. P. Tolbod, S. B. Hansen, and T. V. Bogsrud, "Point-spread function reconstructed PET images of sub-centimeter lesions are not quantitative," *EJNMMI Phys*, vol. 4, no. 1, p. 5, 2017, doi: 10.1186/s40658-016-0169-9.

[62] T. Chang, G. Chang, J. W. Clark, R. H. Diab, E. Rohren, and O. R. Mawlawi, "Reliability of predicting image signal-to-noise ratio using noise equivalent count rate in PET imaging," *Med. Phys*, vol. 39, no. 10, pp. 5891–5900, Sep. 2012, doi: 10.1118/1.4750053.

[63] I. S. Klyuzhin, J.-C. Cheng, C. Bevington, and V. Sossi, "Use of a Tracer-Specific Deep Artificial Neural Net to Denoise Dynamic PET Images," *IEEE Trans. Medical Imaging*, vol. 39, no. 2, pp. 366–376, Feb. 2020, doi: 10.1109/TMI.2019.2927199.

[64] D. L. Snyder and M. I. Miller, "The Use of Sieves to Stabilize Images Produced with the EM Algorithm for Emission Tomography," *IEEE Transactions on Nuclear Science*, vol. 32, no. 5, pp. 3864–3872, Oct. 1985, doi: 10.1109/TNS.1985.4334521.

[65] S. Stute and C. Comtat, "Practical considerations for image-based PSF and blobs reconstruction in PET," *Phys Med. Bio.*, vol. 58, no. 11, p. 3849, 2013.

[66] L. I. Rudin, S. Osher, and E. Fatemi, "Nonlinear total variation based noise removal algorithms," *Physica D: Nonlinear Phenomena*, vol. 60, no. 1–4, pp. 259–268, Nov. 1992, doi: 10.1016/0167-2789(92)90242-F.

[67] L. B. Chavez-Rivera, L. Ortega-Maynez, J. Mejia, and B. Mederos, "ML-EM reconstruction model including total variation for low dose PET high resolution data," in *2015 IEEE nuclear science symposium and medical imaging conference (NSS/MIC)*, 2015, pp. 1–5, doi: 10.1109/NSSMIC.2015.7582221.

[68] T. Alexeev, B. Kavanagh, M. Miften, and C. Altunbas, "A novel total variation based ring artifact suppression method for CBCT imaging with two-dimensional antiscatter grids," *Medical Phys.*, vol. 46, no. 5, pp. 2181–2193, May 2019, doi: 10.1002/mp.13456.

[69] A. Mikhno, E. D. Angelini, B. Bai, and A. F. Laine, "Locally weighted total variation denoising for ringing artifact suppression in pet reconstruction using PSF modeling," in *Proc. Int. Symp. Biomed. Imaging*, 2013, pp. 1252–1255, doi: 10.1109/ISBI.2013.6556758.

[70] B. Bai, Q. Li, and R. M. Leahy, "Magnetic Resonance-Guided Positron Emission Tomography Image Reconstruction," *Seminars in Nuclear Medicine*, vol. 43, no. 1, pp. 30–44, Jan. 2013, doi: 10.1053/j.semnuclmed.2012.08.006.

[71] M. S. Tahaei, A. J. Reader, and D. L. Collins, "MR-guided PET image denoising," in *2016 IEEE Nucl. Sci. Symp. Med. Imaging Conf. Proc. (NSS/MIC)*, 2016, pp. 1–3, doi: 10.1109/NSSMIC.2016.8069564.

[72] J. Bland *et al.*, "MR-Guided Kernel EM Reconstruction for Reduced Dose PET Imaging," *IEEE Trans Rad. Plasma Med. Sci.*, vol. 2, no. 3, pp. 235–243, May 2018, doi: 10.1109/TRPMS.2017.2771490.

[73] A. Klein *et al.*, "Evaluation of 14 nonlinear deformation algorithms applied to human brain MRI registration," *NeuroImage*, vol. 46, no. 3, pp. 786–802, Jul. 2009, doi: 10.1016/j.neuroimage.2008.12.037.

[74] J. Ashburner and K. J. Friston, "Diffeomorphic registration using geodesic shooting and Gauss–Newton optimisation," *NeuroImage*, vol. 55, no. 3, pp. 954–967, Apr. 2011, doi: 10.1016/j.neuroimage.2010.12.049.

[75] A. Collignon, F. Maes, D. Delaere, D. Vandermeulen, P. Suetens, and G. Marchal, "Automated multi-modality image registration based on information theory," in *Information processing in medical imaging*, 1995, vol. 3, pp. 263–274.

[76] C. Studholme, D. L. G. Hill, and D. J. Hawkes, "An overlap invariant entropy measure of 3D medical image alignment," *Pattern Recognition*, vol. 32, no. 1, pp. 71–86, Jan. 1999, doi: 10.1016/S0031-3203(98)00091-0.

[77] C. O. da Costa-Luis, *AMYPAD/SPM12*. 2020 [Online]. Available: https://doi.org/10.5281/zenodo.4272003

[78] K. Dabov, A. Foi, V. Katkovnik, and K. Egiazarian, "Image denoising with block-matching and 3D filtering," in *Image processing: Algorithms and systems, neural networks, and machine learning*, 2006, pp. 354–365, doi: 10.1117/12.643267.

[79] B. Ahn and N. I. Cho, "Block-Matching Convolutional Neural Network for Image Denoising," vol. 6, no. 1, pp. 1–12, Apr. 2017 [Online]. Available: https://arxiv.org/abs/1704.00524

[80] X. Hong, Y. Zan, F. Weng, W. Tao, Q. Peng, and Q. Huang, "Enhancing the Image Quality via Transferred Deep Residual Learning of Coarse PET Sinograms," *IEEE Trans. Medical Imaging*, vol. 37, no. 10, pp. 2322–2332, Oct. 2018, doi: 10.1109/TMI.2018.2830381.

[81] C. Dong, C. C. Loy, K. He, and X. Tang, "Image Super-Resolution Using Deep Convolutional Networks," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 38, no. 2, pp. 295–307, Feb. 2016, doi: 10.1109/TPAMI.2015.2439281. [Online]. Available: https://arxiv.org/abs/1501.00092

[82] C. Ledig *et al.*, "Photo-Realistic Single Image Super-Resolution Using a Generative Adversarial Network," *Proceedings - 30th IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017*, vol. 2017–Janua, pp. 105–114, Sep. 2016, doi: 10.1109/CVPR.2017.19. [Online]. Available: https://arxiv.org/abs/1609.04802

[83] G. Wang, A. Rahmim, and R. N. Gunn, "PET Parametric Imaging: Past, Present, and Future," *IEEE Transactions on Radiation and Plasma Medical Sciences*, vol. 7311, no. c, pp. 1–1, 2020, doi: 10.1109/TRPMS.2020.3025086.

[84] A. Bal, M. Banerjee, R. Chaki, and P. Sharma, "An efficient method for PET image denoising by combining multi-scale transform and non-local means," *Multimedia Tools and Applications*, Aug. 2020, doi: 10.1007/s11042-020-08936-0.

[85] C. Shen, D. Nguyen, Z. Zhou, S. B. Jiang, B. Dong, and X. Jia, "An introduction to deep learning in medical physics: advantages, potential, and challenges," *Physics in Medicine & Biology*, vol. 65, no. 5, p. 05TR01, Mar. 2020, doi: 10.1088/1361-6560/ab6f51.

[86] I. Sturm, S. Lapuschkin, W. Samek, and K.-R. Müller, "Interpretable deep neural networks for single-trial EEG classification," *Journal of Neuroscience Methods*, vol. 274, pp. 141–145, Dec. 2016, doi: 10.1016/j.jneumeth.2016.10.008. [Online]. Available: https://arxiv.org/abs/1604.08201

[87] T. M. Mitchell, *Machine Learning*. McGraw-Hill, 1997.

[88] D. Bau, J.-Y. Zhu, H. Strobelt, A. Lapedriza, B. Zhou, and A. Torralba, "Understanding the Role of Individual Units in a Deep Neural Network," *Proceedings of the National Academy of Sciences*, no. September, p. 201907375, Sep. 2020, doi: 10.1073/pnas.1907375117. [Online]. Available: https://arxiv.org/abs/2009.05041

[89] M. Abadi *et al.*, "TensorFlow: Large-scale machine learning on heterogeneous systems." 2015 [Online]. Available: https://www.tensorflow.org/

[90] M. Minsky and S. A. Papert, *Perceptrons: An introduction to computational geometry*. MIT press, 2017.

[91] J. Hertz, A. Krogh, and R. Palmer, "Introduction to the theory of neural computation." Addison-Wesley, MA, 1991.

[92] I. Goodfellow, Y. Bengio, and A. Courville, *Deep learning*. MIT Press, 2016.

[93] A. Radford, L. Metz, and S. Chintala, "Unsupervised Representation Learning with Deep Convolutional Generative Adversarial Networks," *4th International Conference on Learning Representations, ICLR 2016 - Conference Track Proceedings*, pp. 1–16, Nov. 2015 [Online]. Available: http://arxiv.org/abs/1511.06434

[94] K. Hornik, "Approximation capabilities of multilayer feedforward networks," *Neural Networks*, vol. 4, no. 2, pp. 251–257, 1991, doi: 10.1016/0893-6080(91)90009-T.

[95] A. Kratsios, "Characterizing the Universal Approximation Property," pp. 1–27, Oct. 2019 [Online]. Available: http://arxiv.org/abs/1910.03344

[96] P. Kidger and T. Lyons, "Universal Approximation with Deep Narrow Networks," no. 2017, pp. 1–22, May 2019 [Online]. Available: http://arxiv.org/abs/1905.08539

[97] B. D. Ripley, *Pattern recognition and neural networks*. Cambridge university press, 2007.

[98] B. T. Polyak, "Some methods of speeding up the convergence of iteration methods," *USSR Computational Mathematics and Mathematical Physics*, vol. 4, no. 5, pp. 1–17, 1964.

[99] Y. Bengio, N. Boulanger-Lewandowski, and R. Pascanu, "Advances in Optimizing Recurrent Networks," Dec. 2012 [Online]. Available: http://arxiv.org/abs/1212.0901

[100] Y. Nesterov, "A method of solving a convex programming problem with convergence rate $O(1/k^2)$," in *Soviet mathematics doklady*, 1983, vol. 27, pp. 372–376.

[101] I. Sutskever, J. Martens, G. Dahl, and G. Hinton, "On the importance of initialization and momentum in deep learning," in *Proceedings of machine learning research*, 2013, pp. 1139–1147.

[102] D. P. Kingma and J. Ba, "Adam: A Method for Stochastic Optimization," Dec. 2014 [Online]. Available: http://arxiv.org/abs/1412.6980

[103] IEEE, "IEEE Standard for Floating-Point Arithmetic," *IEEE Std 754-2019 (Revision of IEEE 754-2008)*, pp. 1–84, 2019, doi: 10.1109/IEEESTD.2019.8766229.

[104] M. Y. Park and T. Hastie, "L_1-regularization path algorithm for generalized linear models," *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, vol. 69, no. 4, pp. 659–677, Sep. 2007, doi: 10.1111/j.1467-9868.2007.00607.x.

[105] S. Ioffe and C. Szegedy, "Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift," Feb. 2015 [Online]. Available: http://arxiv.org/abs/1502.03167

[106] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: A Simple Way to Prevent Neural Networks from Overfitting," *J. Mach. Learn. Res.*, vol. 15, no. 1, pp. 1929–1958, 2014.

[107] G. Gidel, F. Bach, and S. Lacoste-Julien, "Implicit Regulari[s]ation of Discrete Gradient Dynamics in Linear Neural Networks," *arXiv*, no. NeurIPS, Apr. 2019 [Online]. Available: http://arxiv.org/abs/1904.13262

[108] C. H. Martin and M. W. Mahoney, "Implicit Self-Regularization in Deep Neural Networks: Evidence from Random Matrix Theory and Implications for Learning," *arXiv*, pp. 1–59, Oct. 2018 [Online]. Available: http://arxiv.org/abs/1810.01075

[109] G. Blanc, N. Gupta, G. Valiant, and P. Valiant, "Implicit regularization for deep neural networks driven by an Ornstein-Uhlenbeck like process," in *Proceedings of machine learning research*, 2020, vol. 125, pp. 483–513 [Online]. Available: http://arxiv.org/abs/1904.09080

[110] N. Razin and N. Cohen, "Implicit Regularization in Deep Learning May Not Be Explainable by Norms," *arXiv*, no. NeurIPS, May 2020 [Online]. Available: http://arxiv.org/abs/2005.06398

[111] S. Hochreiter, "The Vanishing Gradient Problem During Learning Recurrent Neural Nets and Problem Solutions," *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, vol. 6, no. 2, pp. 107–116, Apr. 1998, doi: 10.1142/S0218488598000094.

[112] F. Liu, T. Xiang, T. M. Hospedales, W. Yang, and C. Sun, "Semantic Regularisation for Recurrent Image Annotation," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017 [Online]. Available: https://arxiv.org/abs/1611.05490

[113] N. S. Keskar and R. Socher, "Improving Generalization Performance by Switching from Adam to SGD," Dec. 2017 [Online]. Available: http://arxiv.org/abs/1712.07628

[114] C. Zhu, R. H. Byrd, P. Lu, and J. Nocedal, "Algorithm 778: L-BFGS-B," *ACM Transactions on Mathematical Software*, vol. 23, no. 4, pp. 550–560, Dec. 1997, doi: 10.1145/279232.279236.

[115] K. Gong, C. Catana, J. Qi, and Q. Li, "PET Image Reconstruction Using Deep Image Prior," *IEEE Trans. Medical Imaging*, pp. 1–1, 2018, doi: 10.1109/TMI.2018.2888491.

[116] V. Lempitsky, A. Vedaldi, and D. Ulyanov, "Deep Image Prior," in *2018 IEEE/CVF conference on computer vision and pattern recognition*, 2018, pp. 9446–9454, doi: 10.1109/CVPR.2018.00984 [Online]. Available: https://arxiv.org/abs/1711.10925

[117] R. M. French, "Catastrophic Forgetting in Connectionist Networks," in *Encyclopedia of cognitive science*, Chichester: John Wiley & Sons, Ltd, 2006.

[118] M. McCloskey and N. J. Cohen, "Catastrophic Interference in Connectionist Networks: The Sequential Learning Problem," *Psychology of Learning and Motivation*, vol. 24, pp. 109–165, 1989, doi: 10.1016/S0079-7421(08)60536-8.

[119] K. He, X. Zhang, S. Ren, and J. Sun, "Deep Residual Learning for Image Recognition," Dec. 2015 [Online]. Available: http://arxiv.org/abs/1512.03385

[120] M. Mandal, M. Shah, P. Meena, and S. K. Vipparthi, "SSSDET: Simple Short and Shallow Network for Resource Efficient Vehicle Detection in Aerial Scenes," in *2019 IEEE international conference on image processing (ICIP)*, 2019, pp. 3098–3102, doi: 10.1109/ICIP.2019.8803262.

[121] H. Mhaskar, Q. Liao, and T. Poggio, "When and Why Are Deep Networks Better Than Shallow Ones?" in *Proceedings of the AAAI conference on artificial intelligence*, 2017, pp. 2343–2349.

[122] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, no. 7553, pp. 436–444, May 2015, doi: 10.1038/nature14539.

[123] O. Ronneberger, P. Fischer, and T. Brox, "U-Net: Convolutional Networks for Biomedical Image Segmentation," in *Medical image computing and computer-assisted intervention (MICCAI)*, 2015, pp. 234–241, doi: 10.1007/978-3-319-24574-4_28.

[124] I. Goodfellow *et al.*, "Generative Adversarial Nets," in *Advances in neural inf. Proc. sys.*, 2014, vol. 27, pp. 2672–2680.

[125] V. Antun, F. Renna, C. Poon, B. Adcock, and A. C. Hansen, "On instabilities of deep learning in image reconstruction and the potential costs of AI," *Proceedings of the National Academy of Sciences*, p. 201907377, May 2020, doi: 10.1073/pnas.1907377117. [Online]. Available: https://arxiv.org/abs/1902.05300

[126] J. Lehtinen *et al.*, "Noise2Noise: Learning Image Restoration without Clean Data," *35th International Conference on Machine Learning, ICML 2018*, vol. 7, no. 3, pp. 4620–4631, Mar. 2018 [Online]. Available: https://arxiv.org/abs/1803.04189

[127] D. Wu, K. Gong, K. Kim, X. Li, and Q. Li, "Consensus Neural Network for Medical Imaging Denoising with Only Noisy Training Samples," in *Medical image computing and computer assisted intervention – MICCAI 2019*, 2019, vol. 11767 LNCS, pp. 741–749, doi: 10.1007/978-3-030-32251-9_81 [Online]. Available: https://arxiv.org/abs/1906.03639

[128] F. Hashimoto, H. Ohba, K. Ote, A. Teramoto, and H. Tsukada, "Dynamic PET Image Denoising Using Deep Convolutional Neural Networks Without Prior Training Datasets," *IEEE Access*, vol. 7, pp. 96594–96603, 2019, doi: 10.1109/ACCESS.2019.2929230.

[129] L. Xiang, L. Wang, E. Gong, G. Zaharchuk, and T. Zhang, "Noise-Aware Standard-Dose PET Reconstruction Using General and Adaptive Robust Loss," in *Machine learning in medical imaging*, M. Liu et al., Eds. Cham: Springer International Publishing, 2020, pp. 654–662.

[130] Y. Ding *et al.*, "A Deep Learning Model to Predict a Diagnosis of Alzheimer Disease by Using 18 F-FDG PET of the Brain," *Radiology*, vol. 290, no. 2, pp. 456–464, Feb. 2019, doi: 10.1148/radiol.2018180958.

[131] T.-A. Song, S. R. Chowdhury, F. Yang, and J. Dutta, "Super-Resolution PET Imaging Using Convolutional Neural Networks," *IEEE Transactions on Computational Imaging*, vol. 6, pp. 518–528, 2020, doi: 10.1109/TCI.2020.2964229.

[132] M. J. Muckley *et al.*, "Training a neural network for Gibbs and noise removal in diffusion MRI," *Magnetic Resonance in Medicine*, p. mrm.28395, Jul. 2020, doi: 10.1002/mrm.28395. [Online]. Available: https://arxiv.org/abs/1905.04176

[133] A. J. Reader, G. Corda, A. Mehranian, C. O. da Costa-Luis, S. Ellis, and J. A. Schnabel, "Deep Learning for PET Image Reconstruction," *IEEE Transactions on Radiation and Plasma Medical Sciences*, vol. 5, no. 1, pp. 1–25, Jan. 2021, doi: 10.1109/TRPMS.2020.3014786.

[134] H.-M. Zhang and B. Dong, "A Review on Deep Learning in Medical Image Reconstruction," *Journal of the Operations Research Society of China*, vol. 8, no. 2, pp. 311–340, Jun. 2020, doi: 10.1007/s40305-019-00287-4. [Online]. Available: https://arxiv.org/abs/1906.10643

[135] D. Hwang *et al.*, "Improving the Accuracy of Simultaneously Reconstructed Activity and Attenuation Maps Using Deep Learning," *J. Nucl. Medicine*, vol. 59, no. 10, pp. 1624–1629, Oct. 2018, doi: 10.2967/jnumed.117.202317.

[136] K. Gong *et al.*, "Iterative PET Image Reconstruction Using Convolutional Neural Network Representation," *IEEE Trans. Medical Imaging*, vol. 38, no. 3, pp. 1–1, Oct. 2018, doi: 10.1109/TMI.2018.2869871. [Online]. Available: https://arxiv.org/abs/1710.03344

[137] K. Kim *et al.*, "Penalized PET Reconstruction Using Deep Learning Prior and Local Linear Fitting," *IEEE Trans. Medical Imaging*, vol. 37, no. 6, pp. 1478–1487, Jun. 2018, doi: 10.1109/TMI.2018.2832613.

[138] H. Lim, I. Y. Chun, Y. K. Dewaraja, and J. A. Fessler, "Improved Low-Count Quantitative PET Reconstruction With an Iterative Neural Network," *IEEE Transactions on Medical Imaging*, vol. 39, no. 11, pp. 3512–3522, Nov. 2020, doi: 10.1109/TMI.2020.2998480. [Online]. Available: https://arxiv.org/abs/1906.02327

[139] K. Gong *et al.*, "MAPEM-Net: an unrolled neural network for Fully 3D PET image reconstruction," in *15th int. Meet. Fully 3D image reconstr. Radiol. Nucl. medicine*, 2019, p. 102, doi: 10.1117/12.2534904.

[140] B. Zhu, J. Z. Liu, S. F. Cauley, B. R. Rosen, and M. S. Rosen, "Image reconstruction by domain-transform manifold learning," *Nature*, vol. 555, no. 7697, pp. 487–492, Mar. 2018, doi: 10.1038/nature25988. [Online]. Available: https://arxiv.org/abs/1704.08841

[141] I. Häggström, C. R. Schmidtlein, G. Campanella, and T. J. Fuchs, "DeepPET: A deep encoder–decoder network for directly solving the PET image reconstruction inverse problem," *Medical Image Anal.*, vol. 54, pp. 253–262, May 2019, doi: 10.1016/j.media.2019.03.013. [Online]. Available: https://arxiv.org/abs/1804.07851

[142] C. O. da Costa-Luis and A. J. Reader, "Convolutional micro-networks for MR-guided low-count PET image processing," in *2018 IEEE Nucl. Sci. Symp. Med. Imaging Conf. Proc. (NSS/MIC)*, 2018, pp. 1–4, doi: 10.1109/NSSMIC.2018.8824373.

[143] F. Knoll, M. Holler, T. Koesters, R. Otazo, K. Bredies, and D. K. Sodickson, "Joint MR-PET Reconstruction Using a Multi-Channel Image Regularizer," *IEEE Transactions on Medical Imaging*, vol. 36, no. 1, pp. 1–16, Jan. 2017, doi: 10.1109/TMI.2016.2564989.

[144] G. Schramm *et al.*, "Approximating anatomically-guided PET reconstruction in image space using a convolutional neural network," *NeuroImage*, vol. 224, no. February 2020, p. 117399, Jan. 2021, doi: 10.1016/j.neuroimage.2020.117399.

[145] A. Abdelhamed, S. Lin, and M. S. Brown, "A High-Quality Denoising Dataset for Smartphone Cameras," in *2018 IEEE/CVF conference on computer vision and pattern recognition*, 2018, pp. 1692–1700, doi: 10.1109/CVPR.2018.00182.

[146] W. Lu *et al.*, "An investigation of quantitative accuracy for deep learning based denoising in oncological PET," *Physics in Medicine & Biology*, vol. 64, no. 16, p. 165019, Aug. 2019, doi: 10.1088/1361-6560/ab3242.

[147] B. Zoph, V. Vasudevan, J. Shlens, and Q. V. Le, "Learning Transferable Architectures for Scalable Image Recognition," in *2018 IEEE/CVF conference on computer vision and pattern recognition*, 2018, pp. 8697–8710, doi: 10.1109/CVPR.2018.00907 [Online]. Available: https://arxiv.org/abs/1707.07012

[148] S. Zagoruyko and N. Komodakis, "Wide Residual Networks," in *Procedings of the british machine vision conference 2016*, 2016, vol. 2016–Septe, pp. 87.1–87.12, doi: 10.5244/C.30.87 [Online]. Available: http://arxiv.org/abs/1605.07146 http://www.bmva.org/bmvc/2016/papers/paper087/index.html

[149] C. A. Cocosco, V. Kollokian, R. K.-S. Kwan, and A. C. Evans, "BrainWeb: Online Interface to a 3D MRI Simulated Brain Database," *NeuroImage*, vol. 5, no. 4, pp. 2/4, S425, 1997.

[150] C. O. da Costa-Luis, "BrainWeb-based multimodal models of 20 normal brains." Jul-2019 [Online]. Available: https://doi.org/10.5281/zenodo.3269888

[151] M. A. Belzunce, C. A. Verrastro, E. Venialgo, and I. M. Cohen, "Cuda Parallel Implementation of Image Reconstruction Algorithm for Positron Emission Tomography," *Open Med. Imaging J.*, vol. 6, no. 1, pp. 108–118, Dec. 2012, doi: 10.2174/1874347101206010108.

[152] P. J. Markiewicz *et al.*, "NiftyPET: a High-throughput Software Platform for High Quantitative Accuracy and Precision PET Imaging and Analysis," *Neuroinformatics*, vol. 16, no. 1, pp. 95–115, Jan. 2018, doi: 10.1007/s12021-017-9352-y.

[153] M. A. Belzunce and A. J. Reader, "Time-invariant component-based normalization for a simultaneous PET-MR scanner," *Physics in Medicine and Biology*, vol. 61, no. 9, pp. 3554–3571, May 2016, doi: 10.1088/0031-9155/61/9/3554.

[154] B. Bai and P. D. Esser, "The effect of edge artifacts on quantification of Positron Emission Tomography," *IEEE Nucl. Sci. Symp. Conf. Rec.*, pp. 2263–2266, 2010, doi: 10.1109/NSSMIC.2010.5874186.

[155] F. C. Sureau *et al.*, "Impact of Image-Space Resolution Modeling for Studies with the High-Resolution Research Tomograph," *Journal of Nuclear Medicine*, vol. 49, no. 6, pp. 1000–1008, May 2008, doi: 10.2967/jnumed.107.045351.

[156] R. L. Siddon, "Fast calculation of the exact radiological path for a three-dimensional CT array," *Medical Physics*, vol. 12, no. 2, pp. 252–255, Mar. 1985, doi: 10.1118/1.595715.

[157] F. Jacobs, E. Sundermann, B. De Sutter, M. Christiaens, and I. Lemahieu, "A fast algorithm to calculate the exact radiological path through a pixel or voxel space," *Journal of Computing and Information Technology*, vol. 6, no. 1, pp. 89–94, 1998.

[158] V. Y. Panin, M. Chen, and C. Michel, "Simultaneous update iterative algorithm for variance reduction on random coincidences in PET," in *2007 IEEE Nucl. Sci. Symp. Conf. Rec.*, 2007, vol. 4, pp. 2807–2811, doi: 10.1109/NSSMIC.2007.4436722.

[159] G. Wang and J. Qi, "PET Image Reconstruction Using Kernel Method," *IEEE Trans Med. Imaging*, vol. 34, no. 1, pp. 61–71, Jan. 2015, doi: 10.1109/TMI.2014.2343916. [Online]. Available: https://arxiv.org/abs/NIHMS150003

[160] C. O. da Costa-Luis and A. J. Reader, "Deep Learning for Suppression of Resolution-Recovery Artefacts in MLEM PET Image Reconstruction," in *2017 IEEE Nucl. Sci. Symp. Med. Imaging Conf. Proc. (NSS/MIC)*, 2017, pp. 1–3, doi: 10.1109/NSSMIC.2017.8532624.

[161] K. He, X. Zhang, S. Ren, and J. Sun, "Delving Deep into Rectifiers: Surpassing Human-Level Performance on ImageNet Classification," *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, vol. 2015 Inter, pp. 1026–1034, Feb. 2015, doi: 10.1109/ICCV.2015.123. [Online]. Available: http://arxiv.org/abs/1502.01852

[162] Y. A. LeCun, L. Bottou, G. B. Orr, and K.-R. Müller, "Efficient BackProp," in *Neural networks: Tricks of the trade: Second edition*, G. Montavon et al., Eds. Heidelberg: Springer, 2012, pp. 9–48.

[163] D. L. Snyder, M. I. Miller, L. J. Thomas, and D. G. Politte, "Noise and edge artifacts in maximum-likelihood reconstructions for emission tomography," *IEEE Trans Med. Imaging*, vol. 6, no. 3, pp. 228–238, 1987, doi: 10.1109/TMI.1987.4307831.

[164] C. O. da Costa-Luis and A. J. Reader, "Micro-networks for robust MR-guided low count PET imaging," *IEEE Trans. on Radiat. and Plasma Medical Sci.*, pp. 1–1, 2020, doi: 10.1109/TRPMS.2020.2986414. [Online]. Available: https://ieeexplore.ieee.org/document/9061047/

[165] Y. Wang *et al.*, "3D conditional generative adversarial networks for high-quality PET image estimation at low dose," *NeuroImage*, vol. 174, no. March, pp. 550–562, Jul. 2018, doi: 10.1016/j.neuroimage.2018.03.045.

[166] S. Kaplan and Y.-M. Zhu, "Full-Dose PET Image Estimation from Low-Dose PET Image Using Deep Learning: a Pilot Study," *J. Digit. Imaging*, vol. 3, Nov. 2018, doi: 10.1007/s10278-018-0150-3.

[167] C. O. da Costa-Luis and A. J. Reader, "3D Convolutional Adversarial Micro-networks for Low Count PET-MR Post-processing," in *2019 IEEE nuclear science symposium and medical imaging conference (NSS/MIC)*, 2019, pp. 1–4, doi: 10.1109/NSS/MIC42101.2019.9059905.

[168] S. Anwar, S. Khan, and N. Barnes, "A Deep Journey into Super-resolution," *ACM Computing Surveys*, vol. 53, no. 3, May 2020, doi: 10.1145/3390462. [Online]. Available: https://arxiv.org/abs/1904.07523

[169] M. Mirza and S. Osindero, "Conditional Generative Adversarial Nets," pp. 1–7, 2014 [Online]. Available: http://arxiv.org/abs/1411.1784

[170] Y. Wang *et al.*, "3D Auto-Context-Based Locality Adaptive Multi-Modality GANs for PET Synthesis," *IEEE Trans. Medical Imaging*, vol. 38, no. 6, pp. 1328–1339, Jun. 2019, doi: 10.1109/TMI.2018.2884053.

[171] A. Lucas, M. Iliadis, R. Molina, and A. K. Katsaggelos, "Using Deep Neural Networks for Inverse Problems in Imaging: Beyond Analytical Methods," *IEEE Signal Process. Mag.*, vol. 35, no. 1, pp. 20–36, Jan. 2018, doi: 10.1109/MSP.2017.2760358.

[172] J. Schwab, S. Antholzer, and M. Haltmeier, "Deep null space learning for inverse problems: convergence analysis and rates," *Inverse Problems*, vol. 35, no. 2, p. 025008, Feb. 2019, doi: 10.1088/1361-6420/aaf14a. [Online]. Available: http://arxiv.org/abs/1806.06137 http://dx.doi.org/10.1088/1361-6420/aaf14a http://stacks.iop.org/0266-5611/35/i=2/a=025008?key=crossref.decf47e2c85fefed1713b89494df573f

[173] H. Li, J. Schwab, S. Antholzer, and M. Haltmeier, "NETT: Solving Inverse Problems with Deep Neural Networks," pp. 1–21, Jul. 2019 [Online]. Available: http://arxiv.org/abs/1803.00092v2

[174] J. Schwab, S. Antholzer, and M. Haltmeier, "Big in Japan: Regularizing Networks for Solving Inverse Problems," *Journal of Mathematical Imaging and Vision*, pp. 1–23, Oct. 2019, doi: 10.1007/s10851-019-00911-1. [Online]. Available: http://link.springer.com/10.1007/s10851-019-00911-1

[175] K. Gong *et al.*, "EMnet: an unrolled deep neural network for PET image reconstruction," in *Medical imaging 2019: Physics of medical imaging*, 2019, p. 185, doi: 10.1117/12.2513096.

[176] H. K. Aggarwal, M. P. Mani, and M. Jacob, "MoDL: Model-Based Deep Learning Architecture for Inverse Problems," *IEEE Trans. Medical Imaging*, vol. 38, no. 2, pp. 394–405, Feb. 2019, doi: 10.1109/TMI.2018.2865356. [Online]. Available: https://arxiv.org/abs/1712.02862

[177] R. Penrose, "A generalized inverse for matrices," *Mathematical Proceedings of the Cambridge Philosophical Society*, vol. 51, no. 3, pp. 406–413, Jul. 1955, doi: 10.1017/S0305004100030401.

[178] M. James, "The generalised inverse," *The Mathematical Gazette*, vol. 62, no. 420, pp. 109–114, Jun. 1978, doi: 10.1017/S0025557200086460.

[179] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli, "Image Quality Assessment: From Error Visibility to Structural Similarity," *IEEE Transactions on Image Processing*, vol. 13, no. 4, pp. 600–612, Apr. 2004, doi: 10.1109/TIP.2003.819861.

[180] M. A. Belzunce, "High-Resolution Heterogeneous Digital PET [18F]FDG Brain Phantom based on the BigBrain Atlas." Zenodo, May-2018 [Online]. Available: https://doi.org/10.5281/zenodo.1190598

[181] Y. Lu, A. Zhong, Q. Li, and B. Dong, "Beyond finite layer neural networks: Bridging deep architectures and numerical differential equations," in *Proceedings of machine learning research*, 2018, vol. 80, pp. 3276–3285.

[182] A. Krull, T.-O. Buchholz, and F. Jug, "Noise2Void - Learning Denoising From Single Noisy Images," in *2019 IEEE/CVF conference on computer vision and pattern recognition (CVPR)*, 2019, vol. 2019–June, pp. 2124–2132, doi: 10.1109/CVPR.2019.00223 [Online]. Available: https://arxiv.org/abs/1811.10980

[183] G. Huang, Z. Liu, L. van der Maaten, and K. Q. Weinberger, "Densely Connected Convolutional Networks," Aug. 2016 [Online]. Available: http://arxiv.org/abs/1608.06993

[184] J. Dolz, K. Gopinath, J. Yuan, H. Lombaert, C. Desrosiers, and I. Ben Ayed, "HyperDense-Net: A Hyper-Densely Connected CNN for Multi-Modal Image Segmentation," *IEEE Transactions on Medical Imaging*, vol. 38, no. 5, pp. 1116–1126, May 2019, doi: 10.1109/TMI.2018.2878669. [Online]. Available: https://arxiv.org/abs/1804.02967

[185] C. O. da Costa-Luis, *NiftyPET/NiftyML*. 2021 [Online]. Available: https://doi.org/10.5281/zenodo.4654096

[186] K. T. Chen *et al.*, "Ultra–low-dose ^{18}f-florbetaben amyloid PET imaging using deep learning with multi-contrast MRI inputs," *Radiol.*, vol. 290, no. 3, pp. 649–656, Mar. 2019, doi: 10.1148/radiol.2018180940.

[187] J. Xu, E. Gong, J. Ouyang, J. Pauly, G. Zaharchuk, and S. Han, "Ultra-low-dose 18F-FDG brain PET/MR denoising using deep learning and multi-contrast information," in *Medical imaging 2020: Image processing*, 2020, p. 60, doi: 10.1117/12.2548350.

[188] D. Wu, K. Kim, G. E. Fakhri, and Q. Li, "A Cascaded Convolutional Neural Network for X-ray Low-dose CT Image Denoising," May 2017 [Online]. Available: http://arxiv.org/abs/1705.04267

[189] Y. Peng, Y. Cao, S. Liu, J. Yang, and W. Zuo, "Progressive Training of Multi-level Wavelet Residual Networks for Image Denoising," vol. 14, no. 8, pp. 1–13, Oct. 2020 [Online]. Available: http://arxiv.org/abs/2010.12422

[190] E. Kang, H. J. Koo, D. H. Yang, J. B. Seo, and J. C. Ye, "Cycle-consistent adversarial denoising network for multiphase coronary CT angiography," *Medical Physics*, vol. 46, no. 2, pp. 550–562, Feb. 2019, doi: 10.1002/mp.13284. [Online]. Available: https://arxiv.org/abs/1806.09748

[191] L. Zhou, J. D. Schaefferkoetter, I. W. K. Tham, G. Huang, and J. Yan, "Supervised learning with CycleGAN for low-dose FDG PET image denoising," *Medical Image Analysis*, p. 101770, Jul. 2020, doi: 10.1016/j.media.2020.101770.

[192] I. Domingues, G. Pereira, P. Martins, H. Duarte, J. Santos, and P. H. Abreu, "Using deep learning techniques in medical imaging: a systematic review of applications on CT and PET," *Artificial Intelligence Review*, vol. 53, no. 6, pp. 4093–4160, Aug. 2020, doi: 10.1007/s10462-019-09788-3.

[193] N. Ibtehaz and M. S. Rahman, "MultiResUNet : Rethinking the U-Net architecture for multimodal biomedical image segmentation," *Neural Networks*, vol. 121, pp. 74–87, Jan. 2020, doi: 10.1016/j.neunet.2019.08.025. [Online]. Available: https://arxiv.org/abs/1902.04049

[194] K. T. Chen *et al.*, "True ultra-low-dose amyloid PET/MRI enhanced with deep learning for clinical interpretation," *European Journal of Nuclear Medicine and Molecular Imaging*, Jan. 2021, doi: 10.1007/s00259-020-05151-9.

[195] C.-C. Liu and J. Qi, "Higher SNR PET image prediction using a deep learning model and MRI image," *Phys Med. Bio.*, Mar. 2019, doi: 10.1088/1361-6560/ab0dc0.

[196] K. Vunckx and J. Nuyts, "Heuristic modification of an anatomical Markov prior improves its performance," in *IEEE nuclear science symposuim & medical imaging conference*, 2010, pp. 3262–3266, doi: 10.1109/NSSMIC.2010.5874408.

[197] J. E. Bowsher *et al.*, "Utilizing MRI Information to Estimate F18-FDG Distributions in Rat Flank Tumors," in *IEEE symposium conference record nuclear science 2004.*, 2004, vol. 4, pp. 2488–2492, doi: 10.1109/NSSMIC.2004.1462760.

[198] J. Cui *et al.*, "PET image denoising using unsupervised deep learning," *European Journal of Nuclear Medicine and Molecular Imaging*, vol. 46, no. 13, pp. 2780–2789, Dec. 2019, doi: 10.1007/s00259-019-04468-4.

[199] K. Gong, J. Guan, C.-C. Liu, and J. Qi, "PET Image Denoising Using a Deep Neural Network Through Fine Tuning," *IEEE Trans Rad. Plasma Med. Sci.*, vol. 3, no. 2, pp. 153–161, Mar. 2019, doi: 10.1109/TRPMS.2018.2877644.

[200] A. Sanaat, H. Arabi, I. Mainta, V. Garibotto, and H. Zaidi, "Projection-space implementation of deep learning-guided low-dose brain PET imaging improves performance over implementation in image-space," *Journal of Nuclear Medicine*, p. jnumed.119.239327, Jan. 2020, doi: 10.2967/jnumed.119.239327.

[201] A. N. Gomez, M. Ren, R. Urtasun, and R. B. Grosse, "The Reversible Residual Network: Backpropagation Without Storing Activations," in *Advances in neural information processing systems 30*, 2017, pp. 2215–2225 [Online]. Available: http://arxiv.org/abs/1707.04585

[202] K. Kamnitsas *et al.*, "Efficient multi-scale 3D CNN with fully connected CRF for accurate brain lesion segmentation," *Medical Image Analysis*, vol. 36, pp. 61–78, Feb. 2017, doi: 10.1016/j.media.2016.10.004.

[203] T. Cohen, M. Weiler, B. Kicanaoglu, and M. Welling, "Gauge Equivariant Convolutional Networks and the Icosahedral CNN," in *Proceedings of machine learning research*, 2019, pp. 1321–1330 [Online]. Available: http://proceedings.mlr.press/v97/cohen19d.html

[204] L. Wan, M. Zeiler, S. Zhang, Y. LeCun, and R. Fergus, "Regulari[s]ation of Neural Networks using DropConnect," in *Proceedings of machine learning research*, 2013, vol. 28, pp. 1058–1066 [Online]. Available: http://proceedings.mlr.press/v28/wan13.html

[205] I. Goodfellow, D. Warde-Farley, M. Mirza, A. Courville, and Y. Bengio, "Maxout Networks," in *Proceedings of machine learning research*, 2013, pp. 1319–1327 [Online]. Available: http://proceedings.mlr.press/v28/goodfellow13.html

[206] M. D. Zeiler and R. Fergus, "Stochastic Pooling for Regularization of Deep Convolutional Neural Networks," *1st International Conference on Learning Representations, ICLR 2013 - Conference Track Proceedings*, pp. 1–9, Jan. 2013 [Online]. Available: http://arxiv.org/abs/1301.3557

[207] Z. Guo, X. Li, H. Huang, N. Guo, and Q. Li, "Deep Learning-Based Image Segmentation on Multimodal Medical Imaging," *IEEE Trans. Radiat. Plasma Medical Sci.*, vol. 3, no. 2, pp. 162–169, Mar. 2019, doi: 10.1109/TRPMS.2018.2890359.

[208] A. Vaswani *et al.*, "Attention Is All You Need," in *Advances in neural information processing systems*, 2017, pp. 5999–6009 [Online]. Available: http://arxiv.org/abs/1706.03762

[209] N. Parmar *et al.*, "Image Transformer," *35th International Conference on Machine Learning, ICML 2018*, vol. 9, pp. 6453–6462, Feb. 2018 [Online]. Available: http://arxiv.org/abs/1802.05751

[210] A. Dosovitskiy *et al.*, "An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale," pp. 1–21, Oct. 2020 [Online]. Available: http://arxiv.org/abs/2010.11929

[211] F. Isensee, P. F. Jaeger, S. A. A. Kohl, J. Petersen, and K. H. Maier-Hein, "nnU-Net: a self-configuring method for deep learning-based biomedical image segmentation," *Nature Methods*, vol. 18, no. 2, pp. 203–211, Feb. 2021, doi: 10.1038/s41592-020-01008-z.

[212] F. Hutter, L. Kotthoff, and J. Vanschoren, Eds., *Automated Machine Learning*. Cham: Springer International Publishing, 2019.

[213] A. Klöckner, N. Pinto, Y. Lee, B. Catanzaro, P. Ivanov, and A. Fasih, "PyCUDA and PyOpenCL: A scripting-based approach to GPU run-time code generation," *Parallel Computing*, vol. 38, no. 3, pp. 157–174, Mar. 2012, doi: 10.1016/j.parco.2011.09.001. [Online]. Available: https://arxiv.org/abs/0911.3456