Deep learning in medical imaging: Techniques for image reconstruction, super-resolution and segmentation

Daniel Rueckert
Biomedical Image Analysis Group
Department of Computing
Imperial College London, UK
Machine learning in medical imaging:
There is a lot of hype

AI In Medicine: Rise Of The Machines (Forbes, 2017)

A.I. VERSUS M.D.
What happens when diagnosis is automated?

By Siddhartha Mukherjee
Machine learning in medical imaging: There is a lot of hype

“They should stop training radiologists now.”
Geoffrey Hinton (godfather of deep learning) in 2017

"To the question, will AI replace radiologists, I say the answer is no…”

“… but radiologists who do AI will replace radiologists who don’t.”
Curtis Langlotz in 2017
Machine learning for medical imaging: Opportunities

• Big data is slowly arriving in medical imaging

UK Biobank will provide large-scale imaging data from 100,000 subjects
Machine learning for medical imaging: Opportunities

• Big data is slowly arriving in medical imaging

UK Biobank will provide large-scale imaging data from 100,000 subjects
Machine learning for medical imaging: Opportunities

Value proposition

Level of diagnostic support

- Computer Aided Diagnosis
- Computer Aided Interpretation
- Quantification of Imaging Biomarkers
- Semantic Image Interpretation
- Image Enhancement
- Image Acquisition and Reconstruction

Screening
- Tumour quantification
- Radiomics
- Organ localisation
- Organ segmentation
- Super-resolution
- Automated scan planning
- Accelerated imaging
Practical challenges for ML in medical imaging

- Most ML approaches are supervised: 
  Training data is key

- How to obtain training data?

- Training data is expensive: 
  - manpower, cost, time 
  - years of training and expertise required

- Training data is not perfect: 
  - training data may be wrongly labelled

If training data is not perfect how do we validate?
Practical challenges for ML in medical imaging

- ML-based solutions often degrade when deployed in clinical scenarios

- This is caused by differences between training and test data, e.g.
  - different scanner hardware
  - scanner protocols/sequences
  - artefacts

- Manually annotating new data for each test domain is not a feasible solution
Overview

Image reconstruction

Image super-resolution

Image segmentation

5. Discussion and Conclusion

We have presented DeepMedic, a 3D CNN architecture for automatic lesion segmentation that surpasses state-of-the-art on challenging data. The proposed novel training scheme is not only computationally efficient but also offers an adaptive way of partially alleviating the inherent class-imbalance of segmentation problems. We analyzed the benefits of using small convolutions...
• Magnetic Resonance Imaging (MRI)
  – MRI acquisition is inherently a slow process
  – Slow acquisition is
    • ok for static objects (e.g. brain, bones, etc)
    • problematic for moving objects (e.g. heart, liver, fetus)
  – Options for MRI acquisition:
    • real-time MRI: fast, but 2D and relatively poor image quality
    • gated MRI: fine for period motion, e.g. respiration or cardiac motion but requires gating (ECG or navigators) leading to long acquisition times (30-90 min).
Example: Cardiac imaging
Cardiac MRI: Full acquisition is slow

- MRI acquisition is performed in k-space by sequentially traversing sampling trajectories.

Image acquisition is slow
Cardiac MRI: Full acquisition is slow

- MRI acquisition is performed in k-space by sequentially traversing sampling trajectories.
Cardiac MRI: Full acquisition is slow

- MRI acquisition is performed in k-space by sequentially traversing sampling trajectories.
Cardiac MRI: Full acquisition is slow

- MRI acquisition is performed in k-space by sequentially traversing sampling trajectories.

There is significant spatio-temporal redundancy
K-space undersampling

- Acquiring a fraction of k-space **accelerates** the process but introduces **aliasing** in signal space.
K-space undersampling

- Acquiring a fraction of k-space **accelerates** the process but introduces **aliasing** in signal space.
Image reconstruction from undersampled k-space

• One can recover full k-space through compressed sensing techniques:
  – Lustig et al., MRM 2007
  – Jung et al., MRM 2009
  – Otazo et al., MRM 2010

• More recently other techniques have shown to be powerful for this task as well:
  – Caballero et al., IEEE TMI 2014: Dictionary learning
  – Bhatia et al., MICCAI 2016: Manifold learning
  – Schlemper et al., IEEE TMI 2017: Deep learning for cardiac MRI
  – K. Hammernik et al., MRM 2017: Deep learning for knee MRI

Based on generic priors, e.g. sparsity or low-rank

Based on learnt priors
Problem formulation

- Reconstruct image $x \in \mathbb{C}^N$ given undersampled k-space measurements $y \in \mathbb{C}^M$ ($M \ll N$):

$$y = F_u x + e$$

Undersampled Fourier encoding matrix  Acquisition noise
Problem formulation

- Reconstruct image $x \in \mathbb{C}^N$ given undersampled k-space measurements $y \in \mathbb{C}^M$ ($M \ll N$):

$$y = F_{u}x + e$$

Undersampled Fourier encoding matrix

Acquisition noise

- In the case of Cartesian sampling we have $F_{u} = MF$ where $F \in \mathbb{C}^{N \times N}$ applies the 2D Fourier transform and $M \in \mathbb{C}^{M \times N}$ is the undersampling mask in k-space

In this work we consider reconstructing dynamic sequences that are becoming the state-of-the-art technique for various imaging problems including image classification [12], object localisation [13], and image segmentation [14]. Deep architectures are increasingly abstract representations of 2D cardiac MR images with Cartesian undersampling, yet often high-quality reconstructions are achieved in real-time applications. For the dynamic case, sequences can be reconstructed within 10s, which is reasonably fast for off-line libraries, images can be reconstructed efficiently using the proposed method outperforms them in terms of reconstructions. To resolve this issue, they can exploit data redundancy through learned representations of 2D cardiac MR images stacked as a column vector, where the sparse code of patches is heavily affected by aliasing from sub-Nyquist sampling, note that the sparse code of patches is acquisition noise modelled as additive white Gaussian noise $\nu$. Since it is trained to reconstruct the sequence without a-priori knowledge, the CNN formulation forces $\nu$ directly produces a reconstruction as an output. Since the CNN formulation forces $\nu$ directly produces a reconstruction as an output, since

$$F = \begin{bmatrix} F_{1} & \ldots & F_{N} \end{bmatrix}$$

is acquisition noise modelled as additive white Gaussian noise $\nu$. Since it is trained to reconstruct the sequence without a-priori knowledge, the CNN formulation forces $\nu$ directly produces a reconstruction as an output. Since
Problem formulation

- We are trying to solve the following unconstrained optimisation problem:

\[
\min_{\mathbf{x}} \mathcal{R}(\mathbf{x}) + \lambda \| \mathbf{y} - \mathbf{F}_u \mathbf{x} \|_2^2
\]

- **Regularisation term** (in CS usually the \(l_0\) or \(l_1\) norm)

- **Data fidelity term**
Problem formulation

- We are trying to solve the following unconstrained optimisation problem:

\[
\min_x \mathcal{R}(x) + \lambda \|y - F_u x\|_2^2
\]

- For CNN based reconstruction we formulate the problem as

\[
\min_x \|x - f_{\text{cnn}}(x_u | \theta)\|_2^2 + \lambda \|F_u x - y\|_2^2
\]
Data consistency layer

- To ensure data fidelity, we add a data consistency layer. For fixed network parameters we can write:

\[ s_{\text{rec}}(j) = \begin{cases} 
  s_{\text{cnn}}(j), & \text{if } j \notin \Omega \\
  s_{\text{cnn}}(j) + \frac{s_0(j)}{1 + \lambda}, & \text{if } j \in \Omega
\end{cases} \]

- Fourier-encoding of reconstructed image

\[ s_{\text{cnn}} = Fx_{\text{cnn}} = Ff_{\text{cnn}}(x_u | \theta) \]
Data consistency layer

- End-to-end training requires specification of forward and backward passes
- Forward pass:

\[ f_L(x, y; \lambda) = F^H \Lambda Fx + \frac{\lambda}{1 + \lambda} F_u^H y \]

- Backward pass:

\[ \frac{\partial f_L}{\partial x^T} = F^H \Lambda F \]

\[ \left[ \frac{\partial f_{dc}(s, s_0; \lambda)}{\partial \lambda} \right]_j = \begin{cases} 0 & \text{if } j \notin \Omega \\ \frac{s_0(j) - s_{\text{cnn}}(j)}{(1 + \lambda)^2} & \text{if } j \in \Omega \end{cases} \]

Jacobian of the DC layer with respect to the layer input \( x \)

If made trainable
Deep Cascade of CNNs for MRI Reconstruction

Schlemper et al. IEEE TMI 2017
Deep Cascade of CNNs for MRI Reconstruction

Schlemper et al. IEEE TMI 2017
Deep Cascade of CNNs for MRI Reconstruction

\[ s_{\text{rec}}(j) = \begin{cases} 
    s_{\text{cnn}}(j) & \text{if } j \notin \Omega \\
    \frac{s_{\text{cnn}}(j) + \lambda s_0(j)}{1 + \lambda} & \text{if } j \in \Omega
\end{cases} \]

Schlemper et al. IEEE TMI 2017
Magnitude reconstruction (6-fold)

(a) 6x Undersampled  (b) DLTG  (c) CNN  (d) Ground Truth

Schlemper et al. IEEE TMI 2017
Magnitude reconstruction (11-fold)

(a) 11x Undersampled  (b) DLTG  (c) CNN  (d) Ground Truth

Schlemper et al. IEEE TMI 2017
Deep Cascade of CNNs for MRI Reconstruction: Results

- Test error across 10 subjects:

<table>
<thead>
<tr>
<th>Model</th>
<th>R=4 (dB)</th>
<th>R=8 (dB)</th>
</tr>
</thead>
<tbody>
<tr>
<td>DLTG</td>
<td>27.5 (1.31)</td>
<td>22.6 (0.95)</td>
</tr>
<tr>
<td>CNN</td>
<td>31.0 (1.08)</td>
<td>25.2 (1.00)</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Model</th>
<th>Time</th>
</tr>
</thead>
<tbody>
<tr>
<td>DLMRI/DLTG</td>
<td>~6 hr (CPU)</td>
</tr>
<tr>
<td>CNN (2D)</td>
<td>0.69 s (GPU)</td>
</tr>
<tr>
<td>CNN (2D+t)</td>
<td>10 s (GPU)</td>
</tr>
</tbody>
</table>
5. Discussion and Conclusion

We have presented DeepMedic, a 3D CNN architecture for automatic lesion segmentation that surpasses state-of-the-art on challenging data. The proposed novel training scheme is not only computationally efficient but also enables an adaptive way of partially alleviating the inherent class-imbalance of segmentation problems. We analyzed the benefits of using small convolutional filters in the proposed architecture.
Convolutional Neural Networks for Medical Image Segmentation

W. Bai et. submitted to JCMR, 2018
arXiv:1710.09289v3
Image segmentation as a machine learning problem

- Manual annotations of 4,872 subjects (QMUL/Oxford) with 93,128 pixelwise annotated 2D images slices
- Divided into training/validation/test: 3,972/300/600
SA, basal
SA, mid-ventricular
SA, apical
LA, 2 chamber
LA, 4 chamber

W. Bai et. submitted to JCMR, 2018
arXiv:1710.09289v3
### Evaluation of segmentation accuracy
### Comparison to expert observers

#### Extended Data Table 4:

The difference in clinical measures between automated segmentation and manual segmentation, as well between segmentations by different human observers.

<table>
<thead>
<tr>
<th></th>
<th>Auto vs Man (n = 600)</th>
<th>O1 vs O2 (n = 50)</th>
<th>O2 vs O3 (n = 50)</th>
<th>O3 vs O1 (n = 50)</th>
</tr>
</thead>
<tbody>
<tr>
<td>LVEDV (mL)</td>
<td>6.1±5.3</td>
<td>6.1±4.4</td>
<td>8.1±4.7</td>
<td></td>
</tr>
<tr>
<td>LVESV (mL)</td>
<td>5.3±4.9</td>
<td>4.1±4.2</td>
<td>8.8±4.8</td>
<td></td>
</tr>
<tr>
<td>LVM (gram)</td>
<td>6.9±5.5</td>
<td>4.2±3.7</td>
<td>8.7±4.8</td>
<td></td>
</tr>
<tr>
<td>RVEDV (mL)</td>
<td>8.5±7.1</td>
<td>8.7±5.8</td>
<td></td>
<td></td>
</tr>
<tr>
<td>RVESV (mL)</td>
<td>7.2±6.8</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

#### Extended Data Table 5:

The Dice metric, mean contour distance (MCD) and Hausdor distance (HD) between automated segmentation and manual segmentation for long-axis images.

<table>
<thead>
<tr>
<th></th>
<th>Dice</th>
<th>MCD (mm)</th>
<th>HD (mm)</th>
</tr>
</thead>
<tbody>
<tr>
<td>LA cavity (2Ch)</td>
<td>0.93</td>
<td>1.46±1.06</td>
<td>5.76±5.85</td>
</tr>
<tr>
<td>LA cavity (4Ch)</td>
<td>0.95</td>
<td>1.04±0.38</td>
<td>4.03±2.26</td>
</tr>
<tr>
<td>RA cavity (4Ch)</td>
<td>0.96</td>
<td>0.99±0.43</td>
<td>3.89±2.39</td>
</tr>
</tbody>
</table>

Computer performs as well as different expert observers.
But: Cardiac imaging is still challenging

• Acquisition of cardiac MRI typically consists of 2D multi-slice data due to
  – constraints on SNR
  – breath-hold time
  – total acquisition time

• This leads to thick slice data (thickness 8-10 mm per slice)
But: Cardiac imaging is still challenging

• Acquisition of cardiac MRI typically consists of 2D multi-slice data due to
  – constraints on SNR
  – breath-hold time
  – total acquisition time
• This leads to thick slice data (thickness 8-10 mm per slice)
• Motion between slices can lead to artefacts
Conventional CNNs: Problem
Conventional CNNs: What we want

Super-resolution
Segmentation
Conventional CNNs: No explicit use of prior knowledge

• Standard Loss for **segmentation**: Cross-Entropy loss

\[
L_x = - \sum_{i \in S} \sum_{c=1}^{C} \log \left( \frac{e^{f(c,i)}}{\sum_j e^{f(j,i)}} \right)
\]

• Standard loss for **super-resolution**: L2 or L1 loss

\[
\sum_{i \in S} \| \Phi(x_i, \theta_r) - y_i \|^2
\]
Anatomically constrained CNNs

Low-resolution input

Segmentation & Super Resolution Network

High-resolution output

O. Oktay et al. IEEE TMI 2017
Anatomically constrained CNN: T-L networks for representing priors

Figure 6.3: Block diagram of the stacked convolutional autoencoder (AE) network (in grey), which is trained with segmentation labels. The AE model is coupled with a predictor network (in blue) to obtain a compact non-linear representation that can be extracted from both intensity and segmentation images. The whole model is named as T-L network.

6.2.2 Convolutional Autoencoder Model and ACNN-Seg

An autoencoder (AE) \cite{267} is a neural network that aims to learn an intermediate representation from which the original input can be reconstructed. Internally, it has a hidden layer $h$ whose activations represent the input image, often referred as codes. To avoid the AE to directly copy its output, the AE are often designed to be undercomplete so that the size of the code is less than the input dimension as shown in Fig. 6.3. Learning an AE forces the network to capture the most salient features of the training data. The learning procedure minimises a loss function $L_x(y_s, g(f(y_s; \theta_f)))$, where $L_x$ is penalising $g(f(y_s; \theta_f))$ being dissimilar from $y_s$. The functions $g$ and $f$ are typically Euclidean and $\ell_2$ distances, respectively. The gradients $\frac{\partial L_h}{\partial \theta_f}$ and $\frac{\partial L_h}{\partial \theta_p}$ are backpropagated through the network to update the weights $\theta_f$ and $\theta_p$, respectively.
Anatomically constrained CNN: Segmentation framework

**ACNN-Segmentation Model**
- **Input Image (x)**
- **Segmentation Φ(·)**
  - X-Entropy Loss $\frac{\partial L_x}{\partial \theta_s}$
  - Prediction $\Phi(x)$
- **Encoder f(·)**
  - Euclidean Loss $L_{he}$
  - Gradients for Global Loss
  - Gradients for Pixel-Level Loss
- **GT Labels (y_s)**
Anatomically constrained CNN: Segmentation results

Figure 6.7: Segmentation results on two different 2D stack cardiac MR images. The proposed ACNN model is insensitive to slice misalignments as it is anatomically constrained and it makes less errors in basal and apical slices compared to the 2D-FCN approach. The results generated from low resolution image is better correlated with the HR ground-truth annotations (green). Performance in basal and apical parts of the heart as shown in Fig. 6.7. Previous slice by slice segmentation approaches validated their methods on LR annotations; however, we see that the produced label maps are far from the true underlying ventricular geometry and it can be a limiting factor for the analysis of ventricle morphology. Similar results were obtained in clinical studies [66], which however required HR image acquisition techniques. (II) The results also show that introduction of shape priors in segmentation models can be useful to tackle false-positive detections and motion-artefacts. As can be seen in the bottom row of Fig. 6.7, without the learnt shape priors, label map predictions are more prone to imaging artefacts. Indeed, it is the main reason why we observe such a large difference in terms of Hausdorff distance. For endocardium labels, on the other hand, the difference in dice score metric is observed to be less due to the larger size of the LV blood pool compared to the myocardium. Lastly (III), we observe a performance difference between the cascaded AE based segmentation (AE-Seg [217]) and the proposed ACNN-Seg models: the segmentations generated with the former model are strongly regularised due to the second stage AE. It results in reduced Hausdorff distance with marginal statistical significance, but the model overlooks fine details of the myocardium surface since the segmentations are generated only from the coarse level feature-maps. More importantly, cascaded approaches add additional computational complexity.

O. Oktay et al. IEEE TMI 2017
Anatomically constrained CNN: Super-resolution framework

Figure 6.5: Training scheme of the proposed anatomically constrained convolutional neural network (ACNN) for image super-resolution task. The predictor part of the proposed T-L network is used as a regularisation model to enforce the model predictions to follow the distribution of the learned low-dimensional representations or priors.

The training objective shown above is composed of weight decay, pixel-wise and global loss terms. Here $\lambda_1$ and $\lambda_2$ determine the weight of shape priors and weight decay terms while the smooth $\ell_1$ norm loss function quantifies the reconstruction error. The global loss $L_{hp}$ is

$$L_{hp} = \frac{1}{2} \| w \|^2_2 \quad \text{◆} \quad (6.2)$$

The smooth $\ell_1$ Loss $\psi(.)$ is

$$\psi(.) = \min_{\theta_r} \mathcal{L}_{hp}(\theta_r) + \lambda_1 \cdot \|
\frac{\partial L_{hp}}{\partial \theta_r}\|_1 + \lambda_2 \cdot \|w\|^2_2$$

Training details are further discussed in Section 6.3.2. It is important to note that the T-L regulariser model is used only at training time but not during inference; in other words, the fully convolutional (FCN) segmentation and super-resolution models can still be used for applications using different image sizes. In this paper, the proposed SR model is referred to as ACNN-SR and its training scheme is shown in Figure 6.5.

O. Oktay et al. IEEE TMI 2017
Anatomically constrained CNN: Super-resolution results

In Table 6.3, SSIM and MOS scores for the standard interpolation techniques, SR-CNN, and the proposed ACNN-SR models are provided. In addition to the increased image quality, the ACNN-SR model is computationally more efficient in terms of run-time in comparison to the SR-CNN model by a factor of 5. This is due to the fact that ACNN-SR performs feature...
Anatomically constrained CNN: Super-resolution results

O. Oktay et al. IEEE TMI 2017
Challenges for medical image segmentation: Deployment in the clinic

- ML-based segmentation often degrades when deployed in clinical scenarios.

- This is caused by differences between training and test data, e.g. due to variations in:
  - scanner hardware
  - scanner protocols and sequences

- Manually annotating new data for each test domain is not a feasible solution.

Unsupervised domain adaptation using adversarial neural networks can be used to train a CNN-based segmentation:
- which is more invariant to differences in the input data
- which does not require any annotations on the test domain
Deploying machine learning into clinical practice: What is the problem?

**Source (S)**
- Domain: $D_S = \{\mathcal{X}_S, P(X_S)\}$
- Task: $T_S = \{\mathcal{Y}_S, f'_S: \mathcal{X}_S \mapsto \mathcal{Y}_S\}$
- Given: $(X_S, Y_S)$
- $X_S = \{x_{S1}, \ldots, x_{Sn}\}, x_{Si} \in \mathcal{X}_S$
- $Y_S = \{y_{S1}, \ldots, y_{Sn}\}, y_{Si} \in \mathcal{Y}_S$
- Learn: $f_S \approx f'_S$
- $f_S(x) \approx P_S(y|x)$

**Target (T)**
- Domain: $D_T = \{\mathcal{X}_T, P(X_T)\}$
- Task: $T_T = \{\mathcal{Y}_T, f'_T\}$
- Here: $\mathcal{Y}_T = \mathcal{Y}_S$
- Domain Shift: $P(X_T) \neq P(X_S)$
- $f'_T \neq f'_S$

\[ \mathcal{X} = (\mathcal{X}^1, \mathcal{X}^2) \]
\[ \mathcal{Y} = \{x, o\} \]
Solution: Unsupervised domain adaptation with adversarial networks

• Learn a domain classifier $f_D$

$X_T \neq P(X_S)$
DeepMedic: Overview

Figure 5: Multi-scale 3D CNN with two convolutional pathways. The kernels of the two pathways are here of size $5 \times 3$ (for illustration only to reduce the number of layers in the figure). The neurons of the last layers of the two pathways thus have receptive fields of size $17 \times 3$ voxels. The inputs of the two pathways are centered at the same image location, but the second segment is extracted from a down-sampled version of the image by a factor of 3. The second pathway processes context in an actual area of size $51 \times 3$ voxels.

DeepMedic, our proposed 11-layers architecture, results by replacing each layer of the depicted pathways with two that use $3 \times 3$ kernels (see Sec. 2.3). Number of FMs and their size depicted as $(\text{Number} \times \text{Size})$. 

Combining multi-scale features has been found beneficial in other recent works (Long et al. (2015); Ronneberger et al. (2015)), in which whole 2D images are processed in the network by applying a few number of convolutions and then down-sampling the FMs for further processing at various scales. Our decoupled pathways allow arbitrarily large context to be provided while avoiding the need to load large parts of the 3D volume into memory. Additionally, our architecture extracts features completely independently from the multiple resolutions. This way, the features learned by the first pathway retain finest details, as they are not involved in processing low resolution.
DeepMedic in Action

K. Kamnitsas et al. Medical Image Analysis, 2016
DeepMedic: Unsupervised domain adaptation with adversarial networks

Segmenter:
At the core of our system is a fully convolutional neural network (CNN) for image segmentation [12]. Given an input $x$ of arbitrary size, which can be a whole image or a sub-segment, this type of network predicts labels for multiple voxels in $x$, one for each stride of the network's receptive field over the input. The parameters of the network $\theta_{seg}$ are learnt by iteratively minimizing a segmentation loss $L_{seg}$ using stochastic gradient descent (SGD). The loss is commonly the cross-entropy of the predictions on a training batch $B_{seg} = \{(x_1, y_1), \ldots, (x_{N_{seg}}, y_{N_{seg}})\}$ of $N_{seg}$ samples. In our settings, $(x_i, y_i)$ are sampled from the source database $S = (X_S, Y_S)$, for which labels $Y_S$ are available. We borrowed the 3D multi-scale CNN architecture from [10], depicted in Fig 1 and adopt the same configuration for all meta-parameters.

Domain discriminator:
When processing an input $x$, the activations of any feature map (FM) in the segmenter encode a hidden representation $h(x)$. If samples come from different distributions $P(X_S) \neq P(X_T)$, e.g. due to different domains, and the filters of the segmenter are not invariant to the domain-specific variations, the distributions of the corresponding activations will differ as well, $P(h(X_S)) \neq P(h(X_T))$. This is expected when the segmenter is trained only on samples from $S$ where learnt features will be specific to the source domain. Similar to [5], we choose a certain representation $h_a(x)$ from the segmenter and use a second network as a domain-classifier that takes $h_a(x)$ as input and tries to classify whether it comes from $P(h_a(X_S))$ or $P(h_a(X_T))$. This is equivalent to classifying the domain of $x$. Classification accuracy serves as an indication of how source-specific the representation $h_a(\cdot)$ is. The architecture we use for a domain classifier is a 3D CNN with five layers. The first four have 100 kernels of size $3 \times 3$. The last classification layer uses 1 $3 \times 3$ kernels. This architecture has a receptive field of $9 \times 3$ with respect to its input $h_a(\cdot)$ and was chosen for compatibility with the size of feature maps in the 3 last layers of the segmenter.

We train this domain-discriminator simultaneously with the segmenter. For this, we form a second training batch $B_{adv} = \{(x'_1, y'_1), \ldots, (x'_{N_{adv}}, y'_{N_{adv}})\}$. Equal number of samples $x'_i$ are extracted from $X_S$ and $X_T$, so there is no bias towards $K$.
DeepMedic: Unsupervised domain adaptation with adversarial networks

Segmentation system with domain discriminator

Segmenter: At the core of our system is a fully convolutional neural network (CNN) for image segmentation \cite{12}. Given an input $x$ of arbitrary size, which can be a whole image or a sub-segment, this type of network predicts labels for multiple voxels in $x$, one for each stride of the network's receptive field over the input. The parameters of the network $\theta_{seg}$ are learnt by iteratively minimizing a segmentation loss $L_{seg}$ using stochastic gradient descent (SGD). The loss is commonly the cross-entropy of the predictions on a training batch $B_{seg} = \{(x_1, y_1), \ldots, (x_{N_{seg}}, y_{N_{seg}})\}$ of $N_{seg}$ samples. In our settings, $(x_i, y_i)$ are sampled from the source database $S = (X_S, Y_S)$, for which labels $Y_S$ are available. We borrowed the 3D multi-scale CNN architecture from \cite{10}, depicted in Fig 1 and adopt the same configuration for all meta-parameters.

Domain discriminator: When processing an input $x$, the activations of any feature map (FM) in the segmenter encode a hidden representation $h(x)$. If samples come from different distributions $P(X_S) \neq P(X_T)$, e.g. due to different domains, and the filters of the segmenter are not invariant to the domain-specific variations, the distributions of the corresponding activations will differ as well, $P(h(x)|X_S) \neq P(h(x)|X_T)$. This is expected when the segmenter is trained only on samples from $S$ where learnt features will be specific to the source domain.

Similar to \cite{5}, we choose a certain representation $h_a(x)$ from the segmenter and use a second network as a domain-classifier that takes $h_a(x)$ as input and tries to classify whether it comes from $P(h_a(X_S))$ or $P(h_a(X_T))$. This is equivalent to classifying the domain of $x$. Classification accuracy serves as an indication of how source-specific the representation $h_a(\cdot)$ is. The architecture we use for a domain classifier is a 3D CNN with five layers. The first four have 100 kernels of size $3 \times 3$. The last classification layer uses $1 \times 3$ kernels. This architecture has a receptive field of $9 \times 3$ with respect to its input $h_a(\cdot)$ and was chosen for compatibility with the size of feature maps in the 3 last layers of the segmenter.

We train this domain-discriminator simultaneously with the segmenter. For this, we form a second training batch $B_{adv} = \{(x_1, y_d_1), \ldots, (x_{N_{adv}}, y_d_{N_{adv}})\}$. Equal number of samples $x_i$ are extracted from $X_S$ and $X_T$, so there is no bias towards the source or target domain.

Segmenter: \[ L_{seg} = -\frac{1}{m} \sum_{i=1}^{m} [f(x_i) = y_i] \log(f(x_i)), (x_i, y_i) \sim (X_S, Y_S) \]

Domain Discr.: \[ L_{adv} = -\frac{1}{m} \sum_{i=1}^{m} \log(f_D(h(x_i))) - \frac{1}{m} \sum_{j=1}^{m} \log(1 - f_D(h(x_j))), x_i \sim X_S, x_j \sim X_T \]

DeepMedic: Unsupervised domain adaptation with adversarial networks

Proposed unsupervised domain adaptation:
We train the segmenter on all data of $S$ and adapt the domains using half the subjects of $T$, but no labels. GE and SWI share the same input channel. We test segmentation accuracy on the other half of $T$. The experiment is repeated for the other fold. Our method learns filters invariant to the two imaging protocols and transfers knowledge from $S$ to $T$, allowing the system to segment haemorrhages only visible on SWI without ever seeing a manual annotation from $T$ (Fig. 2). This improves by 3% DSC over the non-adapted segmenter that uses only information from $S$ and the common sequences, covering 44% of the difference between the practical lower bound and the upper bound achieved by supervised domain adaptation using labels from both domains.

Fig. 2: (top row) Example case from S. (middle/bottom row) Visual results for two examples. A model trained on $S$ fails on $T$ when GE is simply replaced by SWI (3rd col.). A model trained on $S$ using only the four common sequences misses microbleeds visible only on SWI (4th col.). Our method mitigates these problems by learning features invariant to the imaging protocol (5th col.). (T2, MPRAGE and PD of $T$ are used but not depicted.)

Challenges for medical image segmentation: DeepMedic, FCN & U-Net

• The good:
  – There are some good/promising CNN-based segmentation approaches (DeepMedic, FCN & U-Net)

• The bad:
  – A lot of meta-parameters
  – Architecture & config influence performance
  – Architecture & config influence behavior

• The ugly:
  – Chosen model & config may be suboptimal for other data/task
  – Results and conclusions of analysis are strongly biased

Ensemble of Multiple Models & Architectures (EMMA)

Performance **insensitive** to suboptimal configuration

Behaviour **unbiased** by architecture & configuration
Challenges for medical image segmentation: Behaviour and performance is variable

Fig. 1: Left to right: FLAIR; manual annotation of a BRATS'17 subject, where yellow depicts oedema surrounding tumour core; confidence of a CNN predicting oedema, trained with cross-entropy or IoU loss. Although overall performance is similar, training with IoU (or Dice, not shown) loss alters the CNN's behaviour, which tends to output only highly confident predictions, even when false.

Automatic segmentation systems aim at providing an objective and scalable solution. Representative early works are the atlas-based outlier detection method [5] and the joint segmentation-registration framework, often guided by a tumour growth model [6, 7, 8]. The past few years saw rapid developments of machine learning methods, with Random Forests being among the most successful [9, 10].

More recently, convolutional neural networks (CNN) have gained popularity by exhibiting very promising results for segmentation of brain tumours [11, 12, 13]. A variety of CNN architectures have been proposed, each presenting different strengths and weaknesses. Additionally, networks have a vast number of meta-parameters. The multiple configuration choices for a system influence not only performance but also its behaviour (Fig. 1). For instance, different models may perform better with different types of pre-processing. Consequently, when investigating their behaviour on a given task, findings can be biased. Finally, a configuration highly optimized on a given database may be an over-fit, and not generalise to other data or tasks.

In this work we push towards constructing a more reliable and objective deep learning model. We bring together a variety of CNN architectures, configured and trained in diverse ways in order to introduce high variance between them. By combining them, we construct an Ensemble of Multiple Models and Architectures (EMMA), with the aim of averaging away the variance and with it model- and configuration-specific behaviours. Our approach leads to: (1) a system robust to unpredictable failures of independent components, (2) enables objective analysis with a generic deep learning model of unbiased behaviour, (3) introduces the new perspective of ensembling for objectiveness. This is in contrast to common ensembles, where a single model is trained with small variations such as initial seeds, which renders the ensemble biased by the main architectural choices. As a first milestone in this endeavour, we evaluated EMMA in the Brain Tumour Segmentation (BRATS) challenge 2017. Our method won the first position in the final testing stage among 50+ competing teams. This indicates the reliability of the approach and paves the way for its use in further analysis.
Ensemble of Multiple Models and Architectures (EMMA)

Need to learn: $P(Y|X)$

Approximate it by model: $P(Y|X; \theta_m, m)$

with learnt parameters $\theta_m = \min_{\theta_m} d(P(Y|X; \theta_m, m), P(Y|X))$, $d$ the loss.

Model is defined by chosen meta-parameters $m$.

Commonly $m$ is neglected, but it biases the results!

We define stochastic random variable $M$, over configurations of interest.

Need to marginalise out influence of $M$:

$$P(Y|X) = \sum_{\forall m \in M} P(Y, M = m|X) = \sum_{\forall m \in M} P(Y|X, M = m) P(M = m)$$

**EMMA approximate the joint by ensembling individual models:**

$$\sim P_{EMMA}(Y|X) = \sum_{\forall m \in M} P(Y|X; \theta_m, m) \frac{1}{|M|}$$
**M**: Network architectures

**DeepMedic [Kamnitsas 2015, 2016, 2017]**

**FCN [Long 2015]:**

**U-Net [Ronneberger 2015]:**
M: Network configurations

- Architecture configuration:
  - depth, width, scales, residuals, etc.

- Training Loss:
  - Cross-Entropy, IoU, DSC, etc.

- Sampling strategy:
  - equally per class, foreground/background, etc.

- Optimisation:
  - optimizer, learning rate, momentum, regulariser…

- Data normalisation:
  - z-score, bias field correction, histogram matching
1st Place
2017 MICCAI BraTS Challenge
( Segmentation Task)
K. Kamnitsas, et al. "Ensembles of Multiple Models and Architectures for Robust Brain Tumour Segmentation"
BRATS’17 Challenge: Quantitative validation

• **EMMA: 2 x DeepMedic, 3 x FCNs, 1 x U-Net**
  – Different training losses, sampling strategies, widths, depths, configurations
  – No config was heavily optimised for the task (3/6 nets were quite suboptimal)

<table>
<thead>
<tr>
<th></th>
<th>DSC</th>
<th>Sensitivity</th>
<th>Specificity</th>
<th>Hausdorff.95</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Enh. Whole Core</td>
<td>Enh. Whole Core</td>
<td>Enh Whole Core</td>
<td>Enh Whole Core</td>
</tr>
<tr>
<td>EMMA</td>
<td>75.7 90.2</td>
<td>82.0 79.0 90.9</td>
<td>78.3 99.8 99.5</td>
<td>99.9 4.22 4.56</td>
</tr>
<tr>
<td>UCL-TIG</td>
<td>75.2 89.7</td>
<td>82.5 77.1 91.2</td>
<td>83.9 99.8 99.4</td>
<td>99.7 4.78 3.97</td>
</tr>
<tr>
<td>MIC_DKFZ</td>
<td>73.2 89.6</td>
<td>79.7 79.0 89.6</td>
<td>78.1 99.8 99.6</td>
<td>99.9 4.55 6.97</td>
</tr>
</tbody>
</table>

• **Robustness:**
  – EMMA of all 6 was better than individuals.
  – Ensemble of 3 best nets was only marginally better than EMMA of all 6 nets.
Summary and Conclusions

• Deep learning already plays a crucial role in medical imaging for:
  – Image acquisition and reconstruction
  – Image quantification and analysis

• Applications of deep learning in computer-aided decision support have been limited so far:
  – But there is some (unjustified) hype

• There is great potential for deep learning for truly intelligent computer-aided diagnosis:
  – Learning from unlabelled, large-scale population data
  – Integration of imaging and non-imaging information, e.g. clinical records and genetics

Validation is challenging

Requires collaboration between computer scientists, engineers and clinicians

Optimisation of imaging pipeline with respect to clinically useful information
Current state-of-the-art

Acquisition → Reconstruction → Analysis → Define relevant information
Future: End-to-end optimisation of entire imaging pipeline via deep learning

End-to-end optimisation of acquisition, reconstruction, analysis & interpretation via deep learning

Define relevant information
Future: End-to-end optimisation of entire imaging pipeline via deep learning
Acknowledgements

Ben Glocker
Bernhard Kainz
Wenjia Bai
Matthew Sinclair
Giacomo Tarroni
Ozan Oktay
Martin Rajchl
Aaron M. Lee
Nay Aung
Elena Lukaschuk
Mihir M. Sanghvi
Filip Zemrak
Kenneth Fung
Jose Miguel Paiva
Valentina Carapella
Young Jin Kim
Hideaki Suzuki
Paul M. Matthews
Steffen E. Petersen
Stefan K. Piechnik
Stefan Neubauer
Enzo Ferrante
Steven McDonagh
Ghislain Vaillant
Jo Hajnal
Jose Caballero
Christian Ledig
Christian Baumgartner
Kostas Kamnitsas
Tong Tong
Wenzhe Shi
Martin Rajchl
Jo Schlemper
Carlo Biffo
Nick Pawlowski
Matthew Lee

This research has been conducted mainly using the UKBB Resource under Application Number 2946. The initial stage of the research was conducted using the UKBB Resource under Application Number 18545.