To Learn or Not to Learn Features for Deformable Registration?

Raghav Mehta

Centre for Intelligent Machines (CIM),
McGill University, Canada
To Learn or Not to Learn Features for Deformable Registration?

Aabhas Majumdar, Raghav Mehta, and Jayanthi Sivaswamy

Centre for Visual Information Technology (CVIT)
IIIT Hyderabad, India
Registration problem

Fixed Image  Moving Image  Registered Image
Introduction

• Feature-based registration has been a very popular technique to solve the registration problem.

• Following features have been explored
  ○ Intensity values, edges
  ○ Geometric moment ¹
  ○ 3D Gabor attributes ²
  ○ Modality Independent Neighborhood Descriptor (MIND) ³
  ○ Self-Similarity Context (SSC) ⁴

(1) Shen et al., IEEE TMI 2002 (HAMMER)
(2) Ou et al., MedIA 2011 (DRAMMS)
(3) Heinrich et al., MedIA 2012 (MIIND)
(4) Heinrich et al., MICCAI 2013 (SSC)
Introduction

- A very natural question to feature-based registration in the current time would be “Can learning of features lead to better registration?”

- Some initial works:
  - Deep features learnt using an unsupervised method \(^5\)
  - A Co-Registration and Co-Segmentation framework \(^6\)

\(^{(5)}\) Wu et al., IEEE TBME 2016
\(^{(6)}\) Shakeri et al, MICCAI 2016
In this paper..

- We explore the pros and cons of using different DNNs to learn features in the context of registration.
- Features are learnt by training DNNs for structure segmentation task on Brain MRIs
Our Method

(7) Heinrich et al., WBIR 2014
Architecture of DNNs used

(A) U-net

(B) CAE

(C) M-net

(8) Ronneberger et al., MICCAI 2015
(9) Mehta et al, ISBI 2017
(10) Masci et al., ICANN 2011
Datasets Used

DNN Training Dataset

1. **MICCAI-2012**: 135 labels, Whole brain parcellated
2. **IBSR18**: 32 labels, Whole brain parcellated
3. **LPBA40**: 57 labels, Partial brain parcellated

(12) Rohlfing, TMI 2012
(13) Shattuck et al., NeuroImage 2008
Datasets Used

Registration Testing Dataset
Chosen based on their popularity for evaluating registration \(^{14}\).

1. **CUMC12**: 12 volumes, 130 labels
2. **MGH10**: 10 volumes, 106 labels

(14) Klein et al., NeuroImage 2009
Implementation

- DNNs trained on a NVIDIA K40 GPU with 12 GB RAM
- Training time ~ 3 days
- Code for Deep Learning in Python with Keras Library
- Code for Discrete Registration in C++
- Optimiser: Adam
- Hyper parameters: LR = 0.001, $\beta_1 = 0.9$, $\beta_2 = 0.99$ and $\epsilon = 10 \times e^{-8}$
- **Evaluation**: mean Jaccard Coefficient over all pair-wise registration

$$J(A, B) = \frac{|A \cap B|}{|A \cup B|}.$$
1. Role of complexity of learning architecture:

U-net

Encoding

Decoding

Ronneberge et al., MICCAI 2015
1. Role of complexity of learning architecture:

(9) Mehta et al., ISBI 2017
1. **Role of complexity of learning architecture:** M-net with added residual and supervision connections gives better performance than U-net

**SP\textsubscript{135}**: Segmentation Priors from M-net trained on MICCAI-2012

**USP\textsubscript{135}**: Segmentation Priors from U-net trained on MICCAI-2012
2. **Supervised vs Unsupervised Learning:**

(9) Mehta et al., ISBI 2017
2. **Supervised vs Unsupervised Learning:**

(9) Masci et al., ICANN 2011
2. **Supervised vs Unsupervised Learning**: Supervised Learning gave better performance than Unsupervised Learning.

**SP\textsubscript{135}**: Segmentation Priors from M-net trained on MICCAI-2012

**CAE**: Features from Convolutional Auto-Encoder (CAE) trained on MICCAI-2012
3. **Choice of learnt features:**

(C) M-net

(9) Mehta et al., ISBI 2017
3. Choice of learnt features:

(C) M-net

Penultimate Layer Features

(9) Mehta et al., ISBI 2017
3. **Choice of learnt features:** Both PLF and SP provided features which were comparable in performance.

SP\textsubscript{135} : Segmentation Priors from M-net trained on MICCAI-2012

PLF\textsubscript{135} : Penultimate Layer Features from M-net trained on MICCAI-2012
4. Role of the number of labeled structures in training data:

MICCAI-2012

IBSR18
4. Role of the number of labeled structures in training data: Features learned from different number of structures in training dataset appeared to be equally effective.

\[ \text{SP}_{135} \text{ and } \text{SP}_{32} \text{ : Segmentation Priors from M-net trained on MICCAI-2012 and IBSR18 respectively} \]

\[ \text{PLF}_{135} \text{ and } \text{PLF}_{32} \text{ : Penultimate Layer Features from M-net trained on MICCAI-2012 and IBSR18 respectively} \]
5. Parcellation of training dataset:

- **MICCAI-2012**
- **LPBA40**
  
  DNN Training Datasets

- **CUMC12**
- **MGH10**
  
  Registration Testing Datasets
5. **Parcellation of training dataset:** CNN trained on whole brain parcellated dataset gave better results than partial brain parcellated dataset.

**SP$_{135}$ and SP$_{57}$:** Segmentation Priors from M-net trained on MICCAI-2012 and LPBA40 respectively

**PLF$_{135}$ and PLF$_{57}$:** Penultimate Layer Features from M-net trained on MICCAI-2012 and LPBA40 respectively
6. Learnt Features vs Hand-crafted Features:

- Hand-crafted Feature:
  - Intensity
  - Edge
  - SSC\(^4\)

- Segmentation Priors from M-net trained on MICCAI-2012

---

(4) Heinrich et al., MICCAI 2013 (SSC)
6. **Learnt Features vs Hand-crafted Features:** Features learned using *Deep Learning Failed* to give better performance than SSC.

![Graphs showing Jaccard Coefficient for CUMC12 and MGH10 datasets with different feature types.]
Conclusions

Learning features was explored with different DNN architectures and training regimes.

- Learning features requires high computational resources
  - A feature which need not be learnt (SSC) is the best option in **low-resource settings** and **limited annotated data** scenario, especially if only registration is of interest.

- In a scenario where **both registration and segmentation are of interest**, learning is the better option.
Thank You!
Visualisation of Features

(a) LPBA40    (b) MICCAI-2012    (c) IBSR18
Results

**SP**\textsubscript{135}: Segmentation Priors from M-net trained on MICCAI-2012  **CAE**: Convolutional AutoEncoder

**USP**\textsubscript{135}: Segmentation Priors from U-net trained on MICCAI-2012  **PLF**: Penultimate Layer Features

**SP**\textsubscript{32}: trained on MICCAI-2012  **PLF**\textsubscript{32}: trained on IBSR18  **SP**\textsubscript{57}: trained on LPBA40  **PLF**\textsubscript{57}: trained on LPBA40

---

**Jaccard Coefficient for CUMC12 dataset**

- **SP**\textsubscript{135}: 35.05
- **USP**\textsubscript{135}: 31.73
- **CAE**: 32.37
- **PLF**\textsubscript{135}: 35.19
- **SP**\textsubscript{32}: 35.03
- **PLF**\textsubscript{32}: 34.9
- **SP**\textsubscript{57}: 33.86
- **PLF**\textsubscript{57}: 31.72
- **intensity**: 29.13
- **edge**: 31.15
- **SSC**: 35.93
Results

![Bar chart showing Jaccard Coefficient for MGH10 dataset](chart.png)

- **SP_{135}**: Segmentation Priors from M-net trained on MICCAI-2012
- **USP_{135}**: Segmentation Priors from U-net trained on MICCAI-2012
- **CAE**: Convolutional AutoEncoder
- **PLF**: Penultimate Layer Features
- **SP_{32}**: trained on MICCAI-2012
- **PLF_{32}**: trained on IBSR18
- **SP_{57}**: trained on MICCAI-2012
- **PLF_{57}**: trained on LPBA40
Computational Time

- 1 Pairwise Registration takes 2 mins of CPU and 8 min of GPU time for Registration using Learnt Features

- SSC only takes 2-3 mins of CPU time for 1 Pairwise Registration.
Discrete Registration

- The Cost function to be minimised consists of a similarity and regularisation term.
  \[ E(u) = \sum_{\Omega} S(I_f, I_m, u) + \alpha |\nabla u|^2 \]

- The deformation field is only allowed values from a quantised set of 3-D displacement.

- A 6 dimensional displacement space volume is created for storing the cost of translating a voxel \( x \) with a displacement \( d \).

  \[ DSV(x, d) = S(I_f(x), I_m(x + d)) \]

(7) Heinrich et al., WBIR 2014
Discrete Registration

- The displacement field is obtained by winner-takes-all method by selecting the field with the lowest cost for each voxel.

(7) Heinrich et al., WBIR 2014
Why SSC better than learnt features?

- SSC is a feature explicitly derived for registration whereas learnt features such as SP are optimised for good segmentation as they are trained on a segmentation dataset.
- It gives a good context of within the neighbourhood of the voxel.
  - Uses pairs of patches in six neighbourhood (with a spatial distance $\sqrt{2}$)
  - Avoids central patch for robustness against noise

$$s(I, x, y) = \exp \left( -\frac{SSD(x, y)}{\sigma^2} \right), \ x, y \in N$$