Learning-based Image Registration with Meta-Regularization

Ebrahim Al Safadi[1,2], Xubo Song[1]
[1]Oregon Health & Science University [2]Amazon
{alsafadi,songx}@ohsu.edu

Abstract

We introduce a meta-regularization framework for learning-based image registration. Current learning-based image registration methods use high-resolution architectures such as U-Nets to produce spatial transformations, and impose simple and explicit regularization on the output of the network to ensure that the estimated displacements are smooth. While this approach works well on small deformations, it has been known to struggle when the deformations are large. Our method uses a more advanced form of meta-regularization to increase the generalization ability of learned registration models. We motivate our approach based on Reproducing Kernel Hilbert Space (RKHS) theory, and approximate that framework via a meta-regularization convolutional layer with radially symmetric, positive semi-definite filters that inherent its regularization properties. We then provide a method to learn such regularization filters while also learning to register. Our experiments on synthetic and real datasets as well as ablation analysis show that our method can improve anatomical correspondence compared to competing methods, and reduce the percentage of folding and tear in the large deformation setting, reflecting better regularization and model generalization.

1. Introduction

Deformable image registration is the process of transforming images to be in anatomical correspondence with one another. It is a critical step in many applications in medical imaging including segmentation, diagnosis, and computational anatomy. In the traditional approach to image registration, the transformation is obtained by solving an optimization problem over a space of transformations based on image similarity [1]. This optimization problem is solved for a specific pair, and no generalization is obtained throughout this process. This demands repeating the optimization process for any new pair of images which is inefficient and time-consuming.

Recent advances in deep learning have enabled a paradigm shift in the field. With the introduction of differentiable Spatial Transformer layers [2] and the development of advanced architectures such as U-Nets [3] and Fully Convolutional Networks (FCNs) [4], a new level of abstraction has become achievable. Instead of optimizing for every input image pair, a deep model can be trained to *learn* how to register in an end-to-end fashion [5]. Once a model is obtained, it can be used to infer dense spatial transformations between new pairs in a single forward pass which is much faster than running an optimizer.

As variations in anatomical structures are usually local and non-rigid, the registration we seek is deformable, and the transformation model must have many degrees of freedom to capture the local changes with high resolution [6]. Therefore, one of the most challenging parts of the registration process is ensuring that the estimated spatial transformation between images is smooth and realistic, and that the high degrees of freedom don't result in overfitting. Furthermore, the choice of the regularization function has fundamental consequences on the class of transformations that the model can produce, and is therefore key to building a successful model. Most learning-based registration methods impose an explicit and generic regularization loss on the output of the registration function, which is usually a penalty on the first order derivative of the displacement field [7, 8, 9, 10, 11, 12]. We find this approach to be too limited, as it presumes the same simple smoothness properties regardless of image content and permissible deformations, which is not realistic. For example, some transformation models must preserve discontinuities in the case of sliding organs, but would not accurately apply elsewhere [13].

Furthermore, many state of the art learning-based image registration networks leverage the high-resolution properties of U-Nets for good performance [7, 14, 11, 12]. However, this type of architecture, while good in retaining spatial resolution, struggles with datasets that have large deformations [12][15][16]. We have also observed this in practice through controlled experiments, as we will show in 5.4.1. Motivated to solve these two important problems, we propose a meta-regularization framework for learning-based image registration models. Specifically, we first build on the rich framework of Reproducing Kernel Hilbert Space

(RKHS) theory. As this framework is computationally expensive, we then show how in some special cases it can be simplified to a convolution operation that can approximately retain its regularization properties. Once we establish this, we propose a simple method to train a meta-regularization convolutional layer in registration networks to learn how to regularize, thereby increasing the model's generalizability. Our main contributions are:

1. We introduce a meta-regularization framework to increase the generalization ability of learning-based deformable image registration methods.
2. We draw connections between regularizing displacement fields through (a) applying differential operators and (b) convolving with radially symmetric, rank-1, positive semi-definite (PSD) filters.
3. We propose a method to train convolution filters in a registration network to have spatial regularization properties.
4. We demonstrate that this method can both improve anatomical correspondence and reduce the folding and tear in the predicted displacement fields, especially in the large deformation setting.

2. Background

In traditional image registration, a moving image $\mathbf{M}$ is spatially transformed to be in anatomical correspondence with a fixed image $\mathbf{F}$ according to an image-similarity metric [1]. The problem is ill-posed in that many transformations exist which can achieve the correspondence, even though most are not physically realizable. A regularization term is therefore essential in such problems. If the spatial transformation is defined as a dense displacement field ϕ, then the registration can be formulated as the solution to the optimization problem

$$\phi^* = \arg\min_{\phi} \ \mathcal{S}\big(\mathrm{F}, \mathrm{M} \circ \phi\big) + \lambda \mathcal{R}(\phi), \qquad (1)$$

where $\mathcal{S}(\cdot,\cdot)$ is an image similarity loss function such as the squared Euclidean distance or normalized cross correlation, $\mathcal{R}(\cdot)$ is a regularization function, and λ is a regularization hyperparameter. The composition operation $\circ$ denotes spatially transforming and resampling the moving image according to ϕ.

This traditional framework requires solving the optimization problem in Eq. (1) for any new pair of images F and M, which is a slow process. The recent development of learning-based image registration uses a set of image pairs to train a deep model to register [5]. Subsequently, the registration of a new pair is done by a single forward pass of the network which offers dramatic computational savings in the long run [7]. This approach defines a displacement-producing function $g{:}(\mathrm{F},\mathrm{M}) \rightarrow \phi$, which takes the image pair and produces a spatial transformation ϕ that maps the moving image M to the fixed one F. This function can be represented by a regression neural network with parameters θ, which can be trained in an unsupervised fashion on an ensemble of image pairs. In this work, we not only train this network to produce spatial transformations, but also to learn convolution filters that are capable of spatial regularization.

3. Related Work

3.1. Learning-Based Image registration

Although neural networks have been used within the traditional image registration setting, for example, to learn feature representations [17], the emergence of learning-based image registration based on deep models is more recent. The introduction of differentiable spatial transformer layers [2] and fully convolutional networks [4] facilitated building end-to-end models that can not only register, but also learn how to register. DIRNet modified the affine spatial transformation model in [2] to become deformable [5], and introduced one of the first learning-based methods. However, the model only had trainable parameters at the encoding part of the network. The method then used interpolation to decode the transformation on the control points to the dense pixel level, which reduced the resolution of the model.

Fully convolutional networks were used in [18] to improve upon this shortcoming, since it used trainable layers at the decoder side. Nonetheless, the hourglass shape of this network still limits the resolution of the network to what is distilled in the bottleneck layer. Furthermore, fully trainable deconvolution layers at the decoder side make it more challenging for the network to produce smooth fields, and could introduce checkerboard artifacts [19]. With the introduction of more advanced architectures such as U-Nets [3], higher resolution registration models became possible [7, 14, 20, 9, 10, 11, 12].

As training these models requires big data sets, and obtaining anatomical labels is expensive, a lot of focus has been on unsupervised training. However, producing smooth and topologically preserving transformations becomes particularly challenging, especially in the case of large deformations [10, 21, 12]. To alleviate this, architectures that promote cycle-consistency were proposed in [12]. Cascading a sequence of incremental transformations for gradual registration was also proposed in [8] and [21]. Learning-based diffeomorphic transform methods were also recently introduced as a more advanced model than displacement fields. For example, stationary velocity fields were used in [20, 6, 9]. Nonetheless, velocity-methods remain significantly more computationally demanding than simpler displacement-based models.

In contrast to previous unsupervised learning-based approaches, our method addresses registration generaliza-

tion and regularization by introducing a meta-regularization strategy. Unlike most of these methods which only impose an explicit regularization function at the output of the network to ensure regularity, the meta-regularization layer in our network inherently learns how to regularize.

3.2. Regularization in RKHS

The theory of RKHS provides a powerful regularization model from functional analysis, with functions being confined to live in a Hilbert Space defined by a PSD reproducing kernel. It has been predominately used in the Large Deformation Diffeomorphic Metric Mapping (LDDMM) framework to regularize velocity fields [22, 23], and more recently in a diffeomorphic model that learns a spatially varying regularization function [6].

Representer Theorem lets us express a regularized function in RKHS as a linear product of a Gram matrix and a vector [24]. This property has been used in [25, 13, 26], but in the traditional (non-learning based) registration setting and without the presence of a deep model that parameterizes this vector. Furthermore, as the Gram matrix grows quadratically with the data size, it is impractical to compute or tune during training. Therefore, in our work we use Representer Theorem as a starting point, and then focus on approximating it with a trainable convolutional layer that learns spatial regularization properties.

3.3. Weight symmetry in neural networks

Weight symmetry has been explored in the context of deep learning for the purpose of model compression in [27, 28]. Such methods attempted to increase generalization of a network by forcing it to learn with fewer shared parameters. Horizontal and vertical symmetries were considered in convolutional kernels to increase flip and rotational invariance of networks in [29]. In our method, weight symmetry arises from requiring that the convolution filters of the meta-regularization layer approximate the operation of multiplying by an RKHS Gram matrix. We show that in a specific setting, this requirement imposes radial symmetry on the filter weights, and this turns out to increase model generalization as well.

4. Method

Given a set of image pairs $\{(\mathrm{F}_i, \mathrm{M}_i)\}_{i=1}^n$ drawn from a distribution $\mathcal{D}$, we take a deep unsupervised approach to learn a displacement-producing function g_θ, represented by a neural network with parameters θ. Once trained, this function maps new image pairs $(\mathrm{F}_j, \mathrm{M}_j)$ to a displacement field ϕ_j, such that when the moving image M_j is transformed by ϕ_j, it aligns with the fixed image F_j. The objective, therefore, is to minimize an expected loss

$$\theta^* = \arg\min_\theta \mathbb{E}_{\mathcal{D}} \left[\mathcal{S}\big(\mathrm{F},\, \mathrm{M} \circ g_\theta(\mathrm{F}, \mathrm{M})\big) + \lambda \mathcal{R}(g_\theta) \right] \quad (2)$$

over the entire set of image pairs. Since the composition operation $\circ$ is now part of the network, a differentiable spatial transformer layer $\Psi(\mathrm{M}, g_\theta)$ is used to carry out this operation. Typically, $\mathcal{R}(\cdot)$ takes the form

$$\mathcal{R}(\phi) = \sum_i^d \|\nabla \phi_i(x)\|^p, \quad (3)$$

which imposes different smoothness properties on the displacement field ϕ depending on the power p. When $p = 2$, (3) becomes a diffusion regularization, which penalizes the squared ℓ_2-norm of the derivatives of the components of ϕ, and is used in [9, 14]. When $p = 1$, it becomes regularization by total variation, which is popular because of its edge-preserving properties and is used in [18, 21].

This approach of explicitly imposing a specific regularization function on the output of the registration network is limiting. It requires hand-crafting suitable forms for different datasets and does not enable the registration network to adapt its regularization properties [9]. We opt for a mechanism to learn a more general regularization form.

4.1. Spatial Transformations in RKHS

Generalizing on (1), we can define image registration as a search for a spatial transformation

$$\mathcal{T}^* = \arg\min_{\mathcal{T}} \; \mathcal{S}\big(\mathrm{F}, \mathrm{M} \circ \mathcal{T}\big) + \lambda \mathcal{R}(\mathrm{L}\mathcal{T}), \quad (4)$$

while imposing a smoothness penalty according to a differential operator, L. This is equivalent to searching for a function in RKHS, i.e.,

$$\mathcal{T}^* = \arg\min_{\mathcal{T} \in \mathcal{H}} \; \mathcal{S}\big(\mathrm{F}, \mathrm{M} \circ \mathcal{T}\big) + \lambda \mathcal{R}(\mathcal{T}), \quad (5)$$

where the function space $\mathcal{H}$ is induced by a reproducing kernel $k(\cdot, x)$ which is the Green's function of $\mathrm{LL}^\dagger$ [30]. Therefore, defining any permissible kernel function automatically applies a penalty on the smoothness of the transformation $\mathcal{T}$, per the differential operator it corresponds to.

Invoking an RKHS framework permits us to use Representer Theorem, which lets us represent the transformation in the Hilbert space induced by $k(\cdot, x)$ as a linear transformation in a finite space, $\mathcal{T} = \mathbf{K}\alpha$, for some vector α to be found [24, 31]. The matrix $\mathbf{K}$ is the Gram matrix, such that $\mathbf{K_{i,j}} = k(p_i, p_j)$ for two pixels p_i and p_j. Consequently we can reduce (5) to the search over α, i.e.,

$$\alpha^* = \arg\min_{\alpha} \; \mathcal{S}\big(\mathrm{F}, \mathrm{M} \circ \mathbf{K}\alpha\big) + \lambda \mathcal{R}(\mathbf{K}\alpha). \quad (6)$$

Finally, returning to our deep learning framework, we can regard α as the output of the displacement-producing function, g_θ, so that $\alpha = g_\theta(\mathrm{F}, \mathrm{M})$ and the problem becomes

$$\theta^* = \arg\min_\theta \mathbb{E}\big[\mathcal{S}\big(\mathrm{F}, \mathrm{M} \circ \mathbf{K} g_\theta(\mathrm{F}, \mathrm{M})\big) + \lambda \mathcal{R}(\mathbf{K} g_\theta)\big]. \quad (7)$$

Comparing this to (2) we see that this setting creates a regularization layer between the output of the network g_θ and the transformed image, $\mathrm{M} \circ \mathbf{K} g_\theta(\mathrm{F}, \mathrm{M})$. This adds a powerful regularization model into the image registration network, but is not viable in practice. The matrix $\mathbf{K}$ grows quadratically with the number of pixels and becomes prohibitively complex to compute, and even more difficult to consider updating or learning within the network. Instead, we attempt to exploit the structure of $\mathbf{K}$ for a special case of kernels to reduce its operation to a convolution, while inheriting its regularization effects.

4.2. From RKHS to Convolutions that Regularize

If we restrict our reproducing kernel function to be translationally invariant, i.e., $k(p_i, p_j) = k(p_i - p_j)$, and interpret the input to the spatial transformer layer as displacements off of a uniform rectilinear grid, then we can decompose the Gram matrix along its spatial dimensions via a Kronecker product, $\mathbf{K} = \mathrm{K}^x \otimes \mathrm{K}^y$, or $\mathbf{K} = \mathrm{K}^x \otimes \mathrm{K}^x$ for square grids [32] which is what we assume for simplicity. This matrix will have the structure of a doubly-block symmetric Toeplitz matrix,

$$\mathbf{K} = \begin{pmatrix} \mathsf{toep}(\mathrm{K}^x_{:,0}) & \mathsf{toep}(\mathrm{K}^x_{:,1}) & \cdots & \mathsf{toep}(\mathrm{K}^x_{:,n-1}) \\ \mathsf{toep}(\mathrm{K}^x_{:,1}) & \mathsf{toep}(\mathrm{K}^x_{:,0}) & \cdots & \mathsf{toep}(\mathrm{K}^x_{:,n-2}) \\ \vdots & \vdots & \vdots & \vdots \\ \mathsf{toep}(\mathrm{K}^x_{:,n-1}) & \mathsf{toep}(\mathrm{K}^x_{:,n-2}) & \cdots & \mathsf{toep}(\mathrm{K}^x_{:,0}) \end{pmatrix}. \tag{8}$$

where $\mathsf{toep}(v)$ is a Toeplitz matrix constructed from a vector v. Let $\mathbf{B}_{i,j}$ denote the $(i,j)^{th}$ Toeplitz block in $\mathbf{K}$. We define a reduced matrix $\tilde{\mathrm{K}}$ by concatenating the mid-columns of the blocks $\mathbf{B}_{\frac{n}{2},0}, \mathbf{B}_{\frac{n}{2},1}, \ldots$, i.e.,

$$\tilde{\mathrm{K}} = \begin{pmatrix} \mathbf{B}^m_{\frac{n}{2},0} & \mathbf{B}^m_{\frac{n}{2},1} & \cdots & \mathbf{B}^m_{\frac{n}{2},n} \end{pmatrix}, \tag{9}$$

where $\mathbf{B}^m$ denotes the mid-column of a block matrix $\mathbf{B}$. We can now approximate the large doubly-block symmetric Toepltiz matrix $\mathbf{K}$ in terms of the smaller matrix $\tilde{\mathrm{K}}$ as

$$\mathsf{mat}(\mathbf{K} g_\theta) \approx \tilde{\mathrm{K}} * \mathsf{mat}(g_\theta), \tag{10}$$

where the mat operation returns vectorized outputs to their matrix form and $*$ denotes convolution [33]. Notice that by the structure produced by the Kronecker product, we have that $\mathrm{K}^x_{:,1} = \mathrm{K}^x_{1,0}\mathrm{K}^x_{:,0}$, and more generally that $\mathrm{K}^x_{:,i} = \mathrm{K}^x_{i,0}\mathrm{K}^x_{:,0}$, where $\mathrm{K}^x_{:,i}$ denotes the i^{th} column of K^x. This implies that $\tilde{\mathrm{K}}$ can be directly written as an outer product, i.e., $\tilde{\mathrm{K}} = \mathrm{K}^x_{:,\frac{n}{2}}(\mathrm{K}^x_{:,\frac{n}{2}})^T$, and is therefore PSD with rank-1 [34]. As such, $\mathrm{Tr}(\tilde{\mathrm{K}}) \geq 0$, because $\mathrm{Tr}(\tilde{\mathrm{K}}) = \sum_i \lambda_i(\tilde{\mathrm{K}}) = (\mathrm{K}^x_{:,\frac{n}{2}})^T \mathrm{K}^x_{:,\frac{n}{2}}$. This is a useful property for the next section when we train the registration network to learn convolution filters that have the regularization properties of $\tilde{\mathrm{K}}$. Lastly, as $\mathrm{K}^x_{:,\frac{n}{2}}$ is symmetric, the matrix $\tilde{\mathrm{K}}$ would be *radially* symmetric.

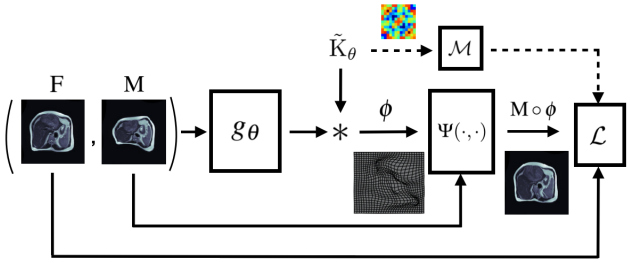

Figure 1. Overall architecture of the learning-based image registration network with meta-regularization. The loss function $\mathcal{L}$ combines the similarity, regularization, and meta-regularization losses.

In summary, we have shown that starting with a spatially invariant PSD kernel $k(\cdot, x)$ evaluated on a rectilinear grid, we can approximate the operation of its corresponding large matrix $\mathbf{K}$ by convolving with a smaller PSD matrix $\tilde{K}$ that is radially-symmetric, rank-1, and of non-negative trace.

4.3. Learning to regularize (and register)

So far, we have demonstrated how to regularize the registration network by convolving its output with the matrix $\tilde{\mathrm{K}}$ as a proxy for multiplying by the regularizing Gram matrix $\mathbf{K}$, with the premise that it will save us computational effort. However, a crucial point here is that we do not pre-define a reproducing kernel $k(\cdot, x)$, and therefore have no control over what corresponding differential operator L the network will use to penalize the displacement field ϕ. Instead, we let the network learn this kernel in its most general form (i.e., we don't simply learn parameters of a predefined reproducing kernel). To achieve this, we apply a meta-regularization term which penalizes the trainable regularization filters to inherent the properties we derived earlier.

More specifically, we denote by $\tilde{\mathrm{K}}_\theta$ a trainable convolution filter applied to g_θ, and regularize it to converge to a radially-symmetric, PSD matrix. Radial symmetry is easy to promote using a penalty on the deviation of $\tilde{\mathrm{K}}_\theta$ from symmetry about its diagonal and anti-diagonal. Promoting positiveness semi-definiteness, on the other hand, is more difficult. We can do this by learning the nonzero entries of an upper triangular matrix U_θ with a non-negativity constraint on its diagonal, and then produce $\tilde{\mathrm{K}}_\theta = \mathrm{U}_\theta \mathrm{U}_\theta^T$.

A simpler necessary but insufficient condition is to ensure that $\mathrm{Tr}(\tilde{\mathrm{K}}_\theta) \geq 0$. The ℓ_1-norm of the diagonal can also be penalized for the eigenvalues to be sparse and for $\mathrm{Tr}(\tilde{\mathrm{K}}_\theta)$ to be low-rank. However, we have noticed that the first process of promoting radial symmetry in addition to imposing a non-negativity trace penalty can approximately produce this result in practice. Wrapping this all together, we finally arrive at the learning-based registration model with meta-

regularization. We define this by the objective function

$$\theta^* = \arg\min_{\theta} \mathbb{E}_{(\mathrm{F},\mathrm{M})\sim D}\big[\mathcal{S}\big(\mathrm{F}, \mathrm{M} \circ (\tilde{\mathrm{K}}_\theta * g_\theta)\big) + \lambda\mathcal{R}(\tilde{\mathrm{K}}_\theta * g_\theta) + \mathcal{M}(\tilde{\mathrm{K}}_\theta), \tag{11}$$

where the meta-regularization function $\mathcal{M}(\cdot)$ takes the form

$$\mathcal{M}(\tilde{\mathrm{K}}_\theta) = \rho_1(\|\tilde{\mathrm{K}}_\theta - \tilde{\mathrm{K}}_\theta^T\|^2 + \|\,\tilde{\mathrm{K}}_\theta\,\mathrm{J} - \mathrm{J}\,\tilde{\mathrm{K}}_\theta\|^2) - \rho_2\,(\mathrm{Tr}(\tilde{\mathrm{K}}_\theta) + \mathrm{Tr}(\mathrm{J}\,\tilde{\mathrm{K}}_\theta)), \tag{12}$$

where J is an exchange matrix (i.e., an anti-diagonal identity matrix) and ρ_1 and ρ_2 are hyperparameters. We highlight that $\mathcal{M}$ can take many other forms, but we found this form to be stable and computationally efficient. Also note that a trivial regularization that could be learned via this mechanism occurs when the kernel filter collapses to a 2D delta function, as this satisfies the radial symmetry, positive-trace, and PSD conditions. However, such a kernel results in severe pixelation in the registered image, and we found that a reasonable image similarity loss function will safeguard against this degenerate case in practise.

4.4. End-to-End Architecture

The overall architecture includes three main parts. The first part is a fully convolutional network, g_θ, that takes an input pair of images, $(\mathrm{F}_i, \mathrm{M}_i)$, and produces a dense displacement field ϕ_i. The architecture is flexible, and we adopt a U-Net architecture because of its ability to preserve the resolution of the spatial transformation. The second part is a meta-regularization layer. This is a trainable convolutional layer equipped with its own penalty function $\mathcal{M}$ that promotes the regularization properties we derived earlier. The third part is a differentiable spatial transformer layer $\Psi(\cdot,\cdot)$ which produces the new moving image given the original moving image and the predicted regularized displacement field.

The final loss function, $\mathcal{L}$, is the accumulation of three losses: an image similarity loss $\mathcal{S}(\mathrm{F}, \mathrm{M})$, a displacement field regularization loss $\mathcal{R}$ which operates on the estimated displacement, and a meta-regularization loss, $\mathcal{M}$ which operates on the kernel filters of the meta-regularization layer. Fig. 1 illustrates this architecture.

5. Experiments

5.1. Datasets

We demonstrate our method on three datasets corresponding to different anatomical structures, namely liver MRI, echocardiography, and chest X-ray. To test the performance of our method against competing methods under a controlled severity of deformation, we first carry out experiments on a liver MRI dataset [35] with synthetic displacement fields. These fields were generated by a spatial Gaussian mixture model, and the deformation energy σ was set to two modes: $\sigma = 10$, corresponding to moderate deformations, and $\sigma = 16$, corresponding to large deformations. This enables us to investigate the performance and generalization properties of learning-based registration methods under four scenarios: Training on a dataset of pairs which have undergone moderate(large) deformations, while testing on another set of pairs which have undergone moderate(large) deformations. We refer to these four experiments as 1) Train Moderate, Test Moderate, 2) Train Moderate, Test Large, 3) Train Large, Test Moderate, and 4) Train Large, Test Large. We train on 1,000 image pairs of size 256×256 and evaluate on 80 pairs that have been anatomically annotated by a human expert and include the liver and spleen segments.

The second set of experiments falls in the large deformation category and uses cardiac motion images from the EchoNet-Dynamic database [36]. This database is composed of echocardiogram videos and human expert annotations for subjects that underwent imaging at Stanford University Hospital. For each video, the left ventricle is marked at the endocardial border at two separate time points representing end-systole and end-diastole. We use these two images corresponding to the end-systole and end-diastole phases as the fixed and moving images, respectively. Each endocardial boarder is annotated with 40 landmarks provided in [36], and we use the concave hull of this region to define the anatomical boundary of interest in each image. Overall, we use 2,090 pairs for training and 200 pairs for evaluation. The third dataset also falls in the large deformation category and corresponds to chest X-ray images of healthy patients and ones infected with Tuberculosis [37][38]. The dataset contains 662 noisy and highly variable images and we used the lung masks in [38] to validate our results on 60 holdout pairs.

5.2. Evaluation metrics

Our learning-based image registration method is unsupervised, and does not use anatomical labels for training. However, we use anatomical labels to test registration accuracy. In each test set, the segmentation mask of the moving image is transformed according to the inferred displacement field ϕ and compared to the mask of the fixed image. We use the Dice coefficient to measure anatomical correspondence before and after registration. The success of the registration model also depends on the smoothness and topology-preserving properties of the spatial transformation. We evaluate the Jacobian determinant $|J_\phi|$ at each displaced pixel, compute the percentage of the Jacobian determinants that are positive, and report the mean and standard deviation of this statistic in each experiment. We also report the statistics of the minimum determinant $|J_\phi|_{\min}$ to keep track of folding severity. Lastly, we report the runtime statistics (in seconds) to register a pair of images.

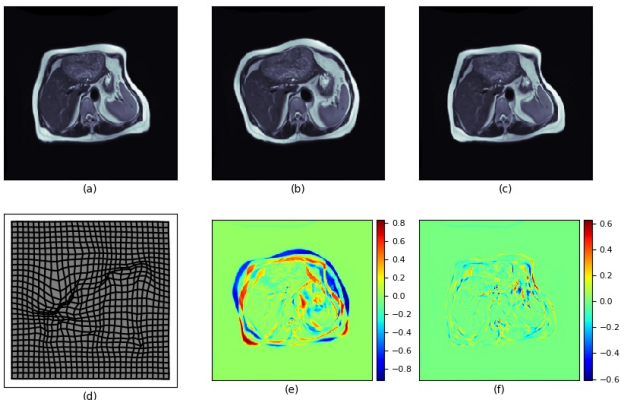

Figure 2. Sample result of registering liver images. (a) Fixed image, (b) Moving image, (c) Registered moving image, (d) Displacement field, (e) Intensity difference before registration, (f) Intensity difference after registration.

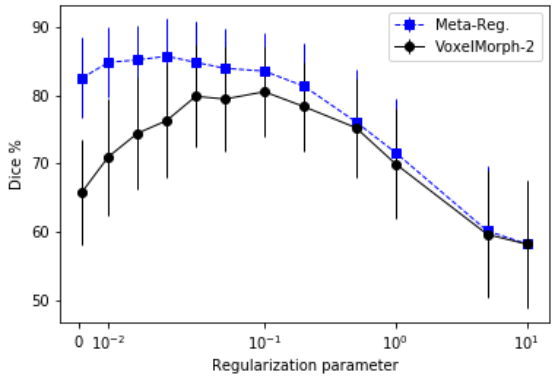

Figure 3. Comparison of Dice % across regularization parameter λ for the large deformation liver dataset ($\sigma_{train} = 16, \sigma_{test} = 16$).

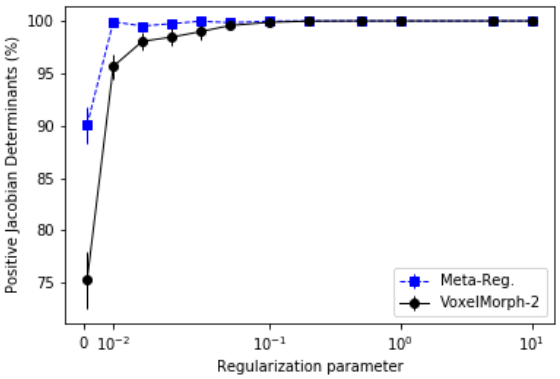

Figure 4. Comparison of percentage of positive Jacobian determinants across regularization parameter λ for the large deformation liver dataset ($\sigma_{train} = 16, \sigma_{test} = 16$).

5.3. Baseline Methods and Implementation

We compare our method to VoxelMorph [7, 14], DIRNet [5] and FCN-SS [18]. VoxelMorph is currently a state-of-the-art method for unsupervised learning-based image registration and has an architecture based on U-Nets. We compare with both VoxelMorph-1, which outputs a displacement at half the image size and then upsamples to full size, and VoxelMorph-2, which uses a full U-Net and therefore operates at a higher spatial resolution. DIRNet uses a simpler convolutional model by only encoding with a sequence of convolutional layers. It then uses interpolation for the decoder to arrive at the full image resolution of the displacement field. Lastly, FCN-SS uses a fully convolutional network with deep self-similarity so that all the encoding and decoding layers of the network are trainable.

For consistency, we used the same implementation of the spatial transformer layer on all methods, which is the one provided in [39] and that we found to be the best. We also tuned the parameters of each method beyond their default values for better performance. The registration networks were trained using an Adam optimizer with a batch size of two pairs, and with mean squared error as the similarity measure. We fixed our regularizing filters to size 13×13 and tuned the hyperparameters ρ_1 and ρ_2 in (12) over the values $\{10^i\}_i$, where $i \in (-1, 0, 1, 2, 3)$ for the former and $i \in (-5, -4, -3, -2, -1)$ for the latter. We implemented our method in Keras and TensorFlow on a 2.8 GHz Intel Core i7 CPU.

5.4. Results

5.4.1 Controlled Deformations on a Synthetic Dataset

The results of the four experiments applied to the first (liver) dataset are shown in Tables 1 and 2. Table 1 demonstrates the two cases where the train and test sets are at the same level of deformation (moderate-moderate and large-large), while Table 2 shows the result of training on moderate deformations while testing on large deformations and vice versa. FCN-SS underperforms in all four scenarios, especially in the two cases where it trains on large deformations. We attribute this to two reasons: (1) The method uses a simpler architecture than U-Nets, and (2) It includes trainable convolutional layers at the decoder side without a suitable regularization mechanism. DIRNet performs better, given its fixed interpolation function at the decoder, but this limits its registration resolution and it is consistently surpassed by VoxelMorph.

Our method outperforms VoxelMorph in the four scenarios. Moreover, it is particularly effective in the large deformation settings, ($\sigma_{train} = 16, \sigma_{test} = 16$) and ($\sigma_{train} = 16, \sigma_{test} = 10$). This demonstrates that meta-regularization can adapt and generalize better than other methods in scenarios were the deformation is large. The 4.3% increase in anatomical correspondence is also associated with a slight increase in positive Jacobian determinants, which implies that the improvement in anatomical correspondence does not come at the cost of increased folding and can actually

Method	Train Moderate, Test Moderate ($\sigma_{train} = 10, \sigma_{test} = 10$)				Train Large, Test Large ($\sigma_{train} = 16, \sigma_{test} = 16$)			
	Dice (%)	$\lvert J_\phi\rvert \geq 0$	$\lvert J_\phi\rvert_{\min}$	CPU	Dice (%)	$\lvert J_\phi\rvert \geq 0$	$\lvert J_\phi\rvert_{\min}$	CPU
DIRNet	90.53±3.27	99.78±0.37	−0.07±0.18	0.0459	76.37±6.91	99.46±0.56	−0.30±0.29	0.0440
FCN-SS	73.85±6.41	99.47±0.11	−4.84±1.15	0.0430	56.65±9.05	99.40±0.09	−6.22±1.60	0.0463
VoxelMorph-1	93.81±1.87	99.92±0.10	−0.39±0.39	0.0619	76.27±7.61	97.86±0.95	−1.87±0.52	0.0771
VoxelMorph-2	93.80±1.95	99.78±0.28	−0.76±0.66	0.1016	80.50±6.64	99.87±0.21	−0.12±0.23	0.1144
Meta-Reg. (Ours)	94.12±1.41	99.96±0.10	−0.09±0.23	0.1020	**84.80±6.09**	**99.97±0.06**	−0.05±0.14	0.1145

Table 1. Comparison of Liver results for the two cases where train and test pairs have the same level of deformation.

Method	Train Moderate, Test Large ($\sigma_{train} = 10, \sigma_{test} = 16$)				Train Large, Test Moderate ($\sigma_{train} = 16, \sigma_{test} = 10$)			
	Dice (%)	$\lvert J_\phi\rvert \geq 0$	$\lvert J_\phi\rvert_{\min}$	CPU	Dice (%)	$\lvert J_\phi\rvert \geq 0$	$\lvert J_\phi\rvert_{\min}$	CPU
DIRNet	77.60±8.39	99.12±0.51	−0.41±0.24	0.0408	87.74±3.11	99.34±0.55	−0.39±0.25	0.0397
FCN-SS	56.30±8.97	99.38±0.08	−5.88±1.27	0.0368	73.87± 6.33	99.48±0.11	−5.21±1.31	0.0357
VoxelMorph-1	84.91±6.54	99.29±0.45	−1.13 ±0.62	0.0560	88.66±3.79	98.45±0.87	−2.09±0.88	0.0610
VoxelMorph-2	84.61±6.27	98.46±0.73	−2.47±1.63	0.0912	86.59±4.99	97.61±1.03	−3.09±1.72	0.0985
Meta-Reg. (Ours)	85.56±6.14	99.15±0.59	−0.88±0.50	0.1004	**92.60±1.79**	**99.84±0.21**	−0.42±0.44	0.1064

Table 2. Comparison of Liver results for the two cases where train and test pairs have different levels of deformation.

Method	Dice (%)	$\lvert J_\phi\rvert \geq 0$	$\lvert J_\phi\rvert_{\min}$	CPU
DIRNet	84.06±6.39	96.42±2.00	−2.08±1.22	0.0212
FCN-SS	76.01±7.21	97.06±1.17	−4.85±1.64	0.0246
VM-1	84.12±5.70	97.18±1.61	−3.54±2.04	0.0416
VM-2	83.96±5.66	97.16±1.53	−3.95±2.42	0.0780
Meta-Reg.	**86.52±5.26**	**98.73±1.15**	−1.64±1.22	0.0813

Table 3. Comparison of methods on Cardio data.

Method	Dice (%)	$\lvert J_\phi\rvert \geq 0$	$\lvert J_\phi\rvert_{\min}$	CPU
DIRNet	81.40±6.92	96.01±2.66	−1.10±0.80	0.0248
FCN-SS	76.41±7.07	94.90±1.23	−6.52±2.04	0.0257
VM-1	83.86±5.85	96.42±2.39	−2.08±1.24	0.0523
VM-2	82.31±5.24	**97.74±1.89**	−1.50±0.96	0.0865
Meta-Reg.	**85.20±4.85**	97.46±2.06	−1.47±0.86	0.0902

Table 4. Comparison of methods on Chest X-ray data.

improve both metrics. Fig. 2 shows an example of a registered pair along with the estimated displacement field.

Fig. 3 shows a comparison of the Dice performance of VoxelMorph-2 and our method in the ($\sigma_{train} = 16$, $\sigma_{test} = 16$) scenario as we vary the regularization parameter λ. Fig. 4 shows the increase in the percentage of positive Jacobian determinants with λ for the same setting. Our method consistently outperforms VoxelMorph-2 across the entire range of regularization strength, and attains its best performance at $\lambda = 0.04$. It is also more robust with respect to this parameter as it maintains good performance across a wider range of its values. Overall, the runtime of our method is roughly 5-10% longer than VomelMorph-2 and 60-80% longer than VoxelMorph-1.

5.4.2 Real Deformations on Echocardiography and Chest X-Ray Images

Tables 3 and 4 show the registration results on the 200 cardiac ultrasound and 60 chest X-ray holdout pairs, respectively. In the cardiac experiment, our method outperforms the others on the average Dice metric and leads the second best method, VoxelMorph-1, by 2.4 points, while also reducing the standard deviation. It also increased the percentage of positive Jacobian determinants by an average of 1.55 points. Fig. 5 shows the registration result and estimated displacement field of a sample pair.

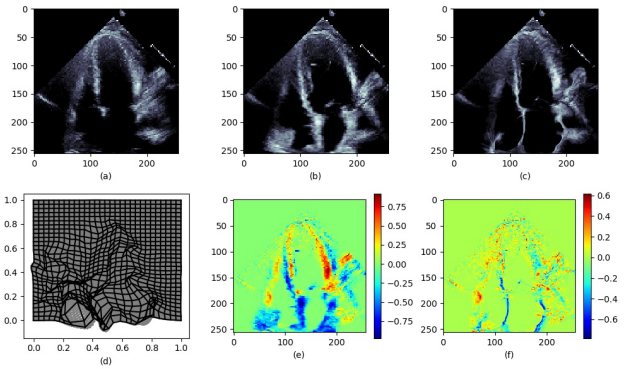

Figure 5. Sample result of registering cardiac ultrasound images. (a) Fixed image, (b) Moving image, (c) Registered moving image, (d) Displacement field, (e) Intensity difference before registration, (f) Intensity difference after registration.

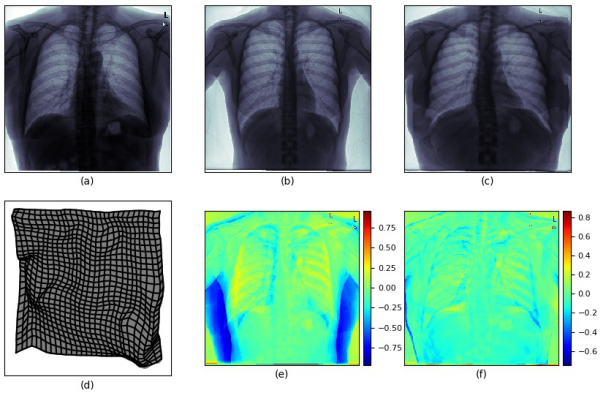

Figure 6. Sample result of registering chest X-ray images. (a) Fixed image, (b) Moving image, (c) Registered moving image, (d) Displacement field, (e) Intensity difference before registration, (f) Intensity difference after registration.

In the X-ray dataset, our method also outperformed the other methods leading the second best, VoxelMorph-1, on the Dice metric by 1.34 points, and VoxelMorph-2 by 2.89 points. However, it under-performed on the folding metric by approximately 0.28 points compared to VoxelMorph-2. Fig. 6 shows the registration result and estimated displacement field of a sample pair.

5.4.3 Ablation Study

To demonstrate the efficacy of our proposed meta-regularization strategy, we also tested out the registration model using the same end-to-end architecture, but with the meta-regularization component $\mathcal{M}$ in (12) removed. This is a network that uses a U-Net architecture with trainable transpose convolution layers in the decoder, a spatial transformer layer, and the same explicit regularization form in (3) (with p=2) applied to the output displacement field.

We applied this architecture to all three datasets. For the liver dataset which had different deformation modes, we tested the two modes ($\sigma_{train} = 10, \sigma_{test} = 10$) and ($\sigma_{train} = 16, \sigma_{test} = 16$). Table 5 compares the results of this ablated architecture with those of the meta-regularization framework. Meta-regularization consistently outperformed the ablated architecture on all settings that have large deformations. The average gain in the Dice metric is 4.71 points for liver MRI with large deformations, 2.52 points for echocardiography, and 3.64 for chest X-ray. It did not improve the performance in the moderate Liver_{10} setting.

5.4.4 Statistical Analysis

We conducted a series of unpaired t-tests of statistical significance on all experiments. At a 95% confidence level, let H_0 in each experiment be the null hypothesis that our method has the same mean Dice score with the next best method. Then H_0 is rejected, with p-value < 0.05 for $\text{liver}_{(16,16)}$, $\text{liver}_{(16,10)}$, and cardiac experiments. It is also rejected for ablation experiments for $\text{liver}_{(16,16)}$, cardiac, and chest experiments with p-values 0.0296, 8.59e-06, and 0.0002, respectively. Only for the chest experiment do we not achieve significance over VoxelMorph-1, but still do so over the second best results of VoxelMorph-2 with p-value=0.002. These results demonstrate that incorporating a meta-regularization strategy can indeed increase the generalization ability of registration networks to within statistical significance.

Dataset	Ablated Network		Meta-Reg.	
	Dice (%)	$\lvert J_\phi\rvert \geq 0$	Dice (%)	$\lvert J_\phi\rvert \geq 0$
Liver_{10}	94.36±1.40	99.94±0.09	94.12±1.41	99.96±0.10
Liver_{16}	80.08±7.08	98.29±0.71	**84.80±6.09**	99.97±0.06
Cardiac	84.00±5.90	97.04±1.68	**86.52±5.26**	98.73±1.15
Chest	81.56±5.55	98.49±1.61	**85.20±4.85**	97.46±2.06

Table 5. Ablation comparison: lift in Dice and positive Jacobian determinants we gain over an ablated network by imposing meta-regularization.

6. Conclusion

In this work we introduced a new meta-regularization framework to learning-based image registration. We built on RKHS theory and established that, for spatially invariant reproducing kernels evaluated on a grid, RKHS regularization effects can be approximately attained in convolution filters. We showed that these filters would be radially symmetric, PSD, rank-1, and with non-negative trace. We then proposed a method to learn such regularizing filters while also training the network to register. Through experimentation with controlled synthetic data and two real datasets, we demonstrated the efficacy of our proposed method in increasing the generalization ability of the registration network, especially in the large deformation regime. Compared with three other methods, our method was able to both enhance the anatomical correspondence as well as reduce the folding frequency in the predicted displacement fields. An ablation study was also conducted by maintaining the same architecture while removing the meta-regularization component, and this showed that our method improved anatomical correspondence in the large deformation sets by an average of 3.6 Dice points, reflecting better model generalization and topology preservation. Overall, this is a promising framework that can be used with many learning-based image registration architectures, and we are currently exploring its potential value in other applications such as optical flow and motion correction.

References

[1] A. Sotiras, C. Davatzikos, and N. Paragios. Deformable medical image registration: A survey. *IEEE Trans. on Medical Imaging*, 32(7), July 2013. 1, 2

[2] M. Jaderberg, K. Simonyan, A. Zisserman, and K. Kavukcuoglu. Spatial transformer networks. In *NIPS 28*, pages 2017–2025, 2015. 1, 2

[3] T. Brox O. Ronneberger, P.Fischer. U-net: Convolutional networks for biomedical image segmentation. In *MICCAI*, volume 9351 of *LNCS*, pages 234–241. Springer, 2015. 1, 2

[4] E. Shelhamer, J. Long, and T. Darrell. Fully convolutional networks for semantic segmentation. *IEEE Trans. Pattern Anal. Mach. Intell.*, 39(4):640–651, April 2017. 1, 2

[5] B. D. de Vos, F. F. Berendsen, M. A. Viergever, M. Staring, and I. Išgum. End-to-end unsupervised deformable image registration with a convolutional neural network. pages 204–212, Sep. 2017. 1, 2, 6

[6] F.-X. Vialard M Niethammer, R. Kwitt. Metric learning for image registration. In *CVPR*, pages 8463–8472, 2019. 1, 2, 3

[7] G. Balakrishnan, A. Zhao, M. R. Sabunc, J. V. Guttag, and A. V. Dalca. An unsupervised learning model for deformable medical image registration. In *CVPR 2018, Salt Lake City, UT*, pages 9252–9260, June 2018. 1, 2, 6

[8] S. Zhao, Y. Dong, E. Chang, and Y. Xu. Recursive cascaded networks for unsupervised medical image registration. In *ICCV*, pages 10599–10609, 2019. 1, 2

[9] T. C.W. Mok and A. C.S. Chung. Fast symmetric diffeomorphic image registration with convolutional neural networks. In *CVPR*, pages 4644–4653, 2020. 1, 2, 3

[10] H. Yipeng *et al.* Weakly-supervised convolutional neural networks for multimodal image registration. In *Med. Image Anal.*, volume 49, pages 1–13, Oct 2018. 1, 2

[11] A. Hering, B. van Ginneken, and S. Heldmann. mlvirnet: Multilevel variational image registration network. In *MICCAI*, volume 11769, pages 257–265, 2019. 1, 2

[12] B. Kim *et al.* Unsupervised deformable image registration using cycle-consistent cnn. In *MICCAI*, volume 11769, pages 166–174, 2019. 1, 2

[13] C. Jud, N. Möri, B. Bitterli, and P. C. Cattin. Bilateral regularization in reproducing kernel hilbert spaces for discontinuity preserving image registration. In *MLMI 2016, Athens, Greece, Oct. 2016*, pages 10–17, 2016. 1, 3

[14] M. R. Sabuncu J. Guttag A. V. Dalca G. Balakrishnan, A. Zhao. Voxelmorph: A learning framework for deformable medical image registration. *IEEE Trans. on Med. Imaging*, 38(8):1788–1800, 2019. 1, 2, 3, 6

[15] M. P. Heinrich L. Hansen. Tackling the problem of large deformations in deep learning based medical image registration using displacement embeddings. In *Medical Imaging with Deep Learning*, 2020. 1

[16] T. Ortmaier M. H. Laves, S. Ihler. Deformable medical image registration using a randomly-initialized cnn as regularization prior. In *Med. Imaging with Deep Learning*, 2019. 1

[17] G. Wu, M. Kim, Q. Wang, B. C. Munsell, and D. Shen. Scalable high-performance image registration framework by unsupervised deep feature representations learning. *IEEE Trans. on Biomed. Eng.*, 63(7):1505–1516, 2016. 2

[18] H. Li and Y. Fan. Non-rigid image registration using self-supervised fully convolutional networks without training data. In *ISBI*, pages 1075–1078, 2018). 2, 3, 6

[19] Z. Wojna *et al.* The devil is in the decoder: Classification, regression and gans. *Int. J. Comput. Vis.* 2

[20] Z. Shen, X. Han, Z. Xu, and M. Niethammer. Networks for joint affine and non-parametric image registration. In *CVPR*, pages 4224–4233, 2019. 2

[21] S. Zhao, T. Lau, J. Luo, E. I. Chang, and Y. Xu. Unsupervised 3d end-to-end medical image registration with volume tweening network. In *IEEE J. Biomed. Health Inform.*, volume 24, pages 1394–1404, May 2020. 2, 3

[22] A. Arnaudon, D. D. Holm, A. Pai, and S. Sommer. A stochastic large deformation model for computational anatomy. In *IPMI*, pages 571–582. 3

[23] X. Yang, R. Kwitt, M. Styner, and M. Niethammer. Quicksilver: Fast predictive image registration - a deep learning approach, Sep 2017. 3

[24] C. E. Rasmussen and C. K. Williams. *Gaussian Processes for Machine Learning*. The MIT Press, Cambridge, MA, 2006. 3

[25] C. Jud, N. Möri, and P. C. Cattin. Sparse kernel machines for discontinuous registration and nonstationary regularization. In *CVPR, Las Vegas, NV*, pages 449–456, June 2016. 3

[26] E. A. Safadi and X. Song. Spatial transformer spectral kernels for deformable image registration. In *BMVC 2019, Cardiff, UK, September 9-12, 2019*, page 291, 2019. 3

[27] X. S. Hu, S. Zagoruyko, and N. Komodakis. Exploring weight symmetry in deep neural networks. In *CVIU*, volume 187, Aug 2019. 3

[28] G. Dzhezyan and H. Cecotti. Symnet: Symmetrical filters in convolutional neural networks. June 2019. https://arxiv.org/pdf/1906.04252.pdf. 3

[29] V. Dudar and V. Semenov. Use of symmetric kernels for convolutional neural networks. In *ICDSIAI*, pages 3–10, 2018. 3

[30] J. O. Ramsay and B. W. Silverman. *Functional Data Analysis*. Springer, New York, NY, 2005. 3

[31] B. Schölkopf and A. J. Smola. *Learning with Kernels: Support Vector Machines, Regularization, Optimization, and Beyond*. The MIT Press, Cambridge, MA, 2001. 3

[32] A. G. Wilson and H. Nickisch. Kernel interpolation for scalable structured gaussian processes. In *ICML 2015*, volume 37, pages 1775–1784, July 2015. 4

[33] H. Sedghi, V. Gupta, and P. M. Long. The singular values of convolutional layers. In *ICLR*, May 2019. 4

[34] E. de Klerk. *Aspects of Semidefinite Programming*. Springer, US, 2002. 4

[35] I. Bickle. Fibrolamellar hepatocellular carcinoma: case study, Dec 2013. https://radiopaedia.org/cases/fibrolamellar-hepatocellular-carcinoma-1. 5

[36] A. Ghorbani M. P. Lungren E. A. Ashley D. H. Liang J. Y. Zou D. Ouyang, B. He. Echonet-dynamic: a large new cardiac motion video data resource for medical machine learning. In *NeurIPS, ML4H Workshop*, 2019. 5

[37] S. Jaeger, S. Candemir, S. Antani, Y.-X. Wang, P.-X. Lu, and G. Thoma. Two public chest x-ray datasets for computer-aided screening of pulmonary diseases. In *Quant Imaging Med Surg.*, volume 4, pages 475–477, 2014. https://lhncbc.nlm.nih.gov/publication/pub9931. 5

[38] S. Stirenko *et al*. Chest x-ray analysis of tuberculosis by deep learning with segmentation and augmentation. In *Int'l Conf. on Elect. and Nano.*, pages 422–428, 2018. 5

[39] G. Balakrishnan *et al*. Voxelmorph github repository. https://github.com/voxelmorph/voxelmorph. 6