Supervised learning in medical image registration

Sokooti, H.

5

Hierarchical Prediction of Registration Misalignment using a Convolutional LSTM: Application to Chest CT Scans

This chapter was adapted from:

H Sokooti, S Yousefi, M Elmahdy, BP Lelieveldt, and M Staring. **Hierarchical Prediction of Registration Misalignment using a Convolutional LSTM: Application to Chest CT Scans**, *IEEE Access*, 2021.

Abstract

In this paper we propose a supervised method to predict registration misalignment using convolutional neural networks (CNNs). This task is casted to a classification problem with multiple classes of misalignment: “correct” 0-3 mm, “poor” 3-6 mm and “wrong” over 6 mm. Rather than a direct prediction, we propose a hierarchical approach, where the prediction is gradually refined from coarse to fine. Our solution is based on a convolutional Long Short-Term Memory (LSTM), using hierarchical misalignment predictions on three resolutions of the image pair, leveraging the intrinsic strengths of an LSTM for this problem. The convolutional LSTM is trained on a set of artificially generated image pairs obtained from artificial displacement vector fields (DVFs). Results on chest CT scans show that incorporating multi-resolution information, and the hierarchical use via an LSTM for this, leads to overall better F1 scores, with fewer misclassifications in a well-tuned registration setup. The final system yields an accuracy of 87.1%, and an average F1 score of 66.4% aggregated in two independent chest CT scan studies.

5.1 Introduction

Most image registration techniques do not provide insight in the local misalignment after registration. It is common to manually inspect the registration quality afterwards, which is time-consuming and prone to inter-observer errors as well as human fatigue. A fast automatic dense map indicating the misalignment locally has quite a few applications in medical imaging. This dense misalignment map can be utilized in radiation dosimetry [108], image-guided interventions [31], for improving the registration quality automatically [32] or semi-automatically [84]. Moreover, a fast automatic prediction of registration misalignment could substantially reduce the manual assessment time.

Several intensity-based and registration-based features were proposed as a surrogate for registration misalignment. Park et al. [37] proposed normalized local mutual information (NMI) and Rohde et al. [38] utilized the local gradient of the NMI as a surrogate for misregistration. Schlachter et al. [85] reported that the histogram intersection, which is a distance measure between the histogram of intensities of a pair of images [109], performs well as a visual assistant to a human expert in detecting local registration quality. Although the mentioned metrics can represent the registration error, it has been shown by Rohlfing [110] that image similarities cannot necessarily distinguish accurate from inaccurate registrations. Hub et al. [35] proposed performing multiple registrations with perturbations in the B-spline grid [6] as a measure of registration uncertainty. Kybic [94] proposed bootstrapping over pixels in the cost functions. Other approaches like block matching [111] and polynomial chaos expansions [112] are utilized in the context of detecting registration misalignment. However, these algorithms are very time-consuming.

In probabilistic image registration, an uncertainty map can be provided after the registration [34, 96, 106]. This uncertainty map commonly is counted as a surrogate for image registration error. However, Luo et al. [113] reported that the uncertainty derived from probabilistic image registrations might not necessarily correlate with the registration error.

Several machine learning approaches have been used in assessing the registration quality. Muenzing et al. [39] cast the problem to a classification task. They extracted several intensity-based features around a number of distinctive landmarks in chest CT images. Sokooti et al. [55, 33] extracted both intensity and registration-based features around a dilated region of landmarks and trained a regression forest to predict the registration error. Drawbacks of these methods are that training is based on a limited number of manual landmarks, and/or can only be applied to nonrigid registration.

Deep learning-based methods have been presented recently and achieved promising results for medical image registration [17, 23, 114]. Predicting the registration error with a CNN-based approach was recently proposed by Eppenhof et al. [40]. They used a single scale method and predicted registration misalignment smaller than 4 mm. Senneville et al. [115] proposed a deep learning method to classify brain MR registrations as usable or non-usable. This method cannot predict misalignment locally, for nonrigid image registration.

Hierarchical approaches have been used in many tasks in the field of image classification. Salakhutdinov et al. [116] proposed a hierarchical classification model, in which objects with fewer occurrences can borrow statistical strength from related objects that have many training examples. Ristin et al. [117] reported that taking into account the hierarchical relations between categories and subcategories can improve the performance of classification. Such an approach has also been used in recent deep learning methods. Redmon et al. [118] in their proposed method for object detection, YOLO9000, predict labels in a hierarchical approach using conditional probability. Chen et al. [119] predict abnormality labels in chest X-ray images using a similar hierarchical approach with conditional probability. They added another stage with unconditional probabilities and reported better performance in comparison with only a single stage with conditional probability. Taherkhani et al. [120] reported that utilizing coarse images can improve weakly supervised fine image classification performance. Guo et al. [121] reported that utilizing a convolutional LSTM [122] and predicting the labels from coarse to fine, can improve the accuracy of the classification of both coarse and fine labels. In their method, the CNN and LSTM extract discriminative features and jointly optimize the fine and coarse labels classification. A similar hierarchical LSTM approach has been utilized in music genre classification [123]. In the aforementioned methods, the hierarchical approach is only applied on the network outputs (coarse and fine labels), while the inputs are kept similar in all steps of the hierarchy.

In this work, inspired by the hierarchical classification idea of [121], we propose a hierarchical convolutional LSTM approach to densely predict the registration misalignment. Moreover, we incorporate multi-resolution information for the inputs as well as the outputs. This way, the LSTM takes input images from coarse to fine resolution and progressively predicts output labels from coarse to fine. We propose to use a pre-trained registration network to encode the input image pair in a latent space, and utilize an LSTM decoder to predict the final labels from this latent space. We trained our deep learning model on image pairs artificially generated from real data, as a data augmentation step. In this way, in contrast to [39] and [33], we have access to many training samples instead of a small number of manually annotated landmarks. Different from earlier deep learning methods, the proposed method can be used to predict the registration error for any registration paradigm, including rigid and nonrigid registration. Different from [40], the proposed method is capable of detecting relatively large registration misalignments. The inference time of the

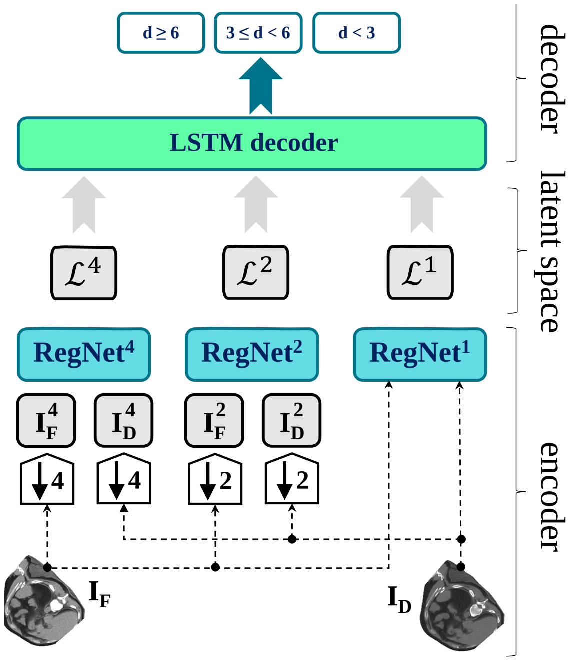


Figure 5.1: Block diagram of the proposed system. In the encoder, a pair of images is given as the input. Three RegNet architectures [18] process the input images over three resolutions (↓4, ↓2, 1) and generate a latent representation (the encoded feature maps $\mathscr{L}^i$) for each resolution. All RegNet blocks are architecturally identical, but are initialized with weights from pre-trained networks on different resolutions. In the LSTM decoder, the latent representations $\mathscr{L}^i$ are decoded to labels corresponding to the local misalignment class d.

proposed method is approximately 2.8 seconds on a 3D patch of size 205 × 205 × 205, which is substantially faster than methods involving multiple registrations like [94, 35, 33].

In Section 5.2, we introduce the network architectures (5.2.1) and explain the training data generation process (5.2.2). In Section 5.3, we describe the data sets used in this study (5.3.1), the detailed setup of the experiments (5.3.2), and the evaluation measures (5.3.3). The tuning of hyper-parameters (5.3.4) and the results 5.3.5, 5.3.6) are reported afterwards. Finally, the Discussion (Section 5.4) and Conclusion (Section 5.5) are presented.

5.2 Methods

A general block diagram of the proposed method is shown in Fig. 5.1. The input of the network is a pair of images consisting of a fixed image $\mathrm{I_F}$ and a deformed moving image I_D, resulting from an arbitrary registration method. The input image pair is then downsampled and encoded by a deep learning registration network at three resolutions. The latent representations $\mathscr{L}^i$ are subsequently fed to a decoder (an LSTM), where the decoder predicts misregistration labels d for each voxel, corresponding to the local misalignment. The LSTM not only considers the encodings at the three resolutions, but also considers these in a coarse-to-fine, hierarchical manner.

5.2.1 Network architectures

5.2.1.1 Encoder

In the encoder, an image pair (I_F, I_D) is encoded to create a latent representation of the input pair and their spatial relation. Such an encoder may be trained from scratch, or a pre-trained architecture can be chosen. Popular examples of the latter is to use a VGG or a ResNet network trained on large-scale natural images [124, 125], sometimes also used to compute a perceptual loss in a downstream task [126]. A downside of such an approach is that each of the input images is encoded separately, and subsequently the spatial relation between the input images is not represented. In addition, as reported by Raghu et al. [127], for medical imaging tasks a network trained on similar data is favored over a network trained on natural images. Instead, we therefore propose to encode the input pair by a pre-trained medical image registration network, thus allowing the direct encoding of a pair of images, while also representing the spatial relation between them.

Any registration network from the literature can be used here, and we opt for the RegNet architecture [18, 17], which we previously proposed for the registration of chest CT scans. Since this network achieved promising results, it is potentially a good candidate for the task of predicting registration misalignment as well. The RegNet architecture is given in Fig. 5.2. This design is identical to the U-Net-advanced (Uadv) design proposed in [18]. The last three layers from the original design are excluded here, and the high dimensional feature maps from the now last layer are used as a latent representation of the input pair, and thus as input for the decoder. As illustrated in Fig. 5.1, we utilize three separate encoders, each receives an input image pair at a different resolution, using a down-sampling factor of four (↓4), two (↓2) and 1 (i.e. the original resolution). This way latent representations are built at three different scales.

The RegNet architecture is a patch-based design where the size of the inputs and output are $101 \times 101 \times 101$ and $25 \times 25 \times 25$, respectively. All convolutional layers use batch normalization [70] and ReLu activation [71], except for the trilinear upsampling layer, in which a constant trilinear kernel is used. The total number of parameters in this design is 737,430.

The weights of the three encoders are initialized with the pre-trained RegNet[i]

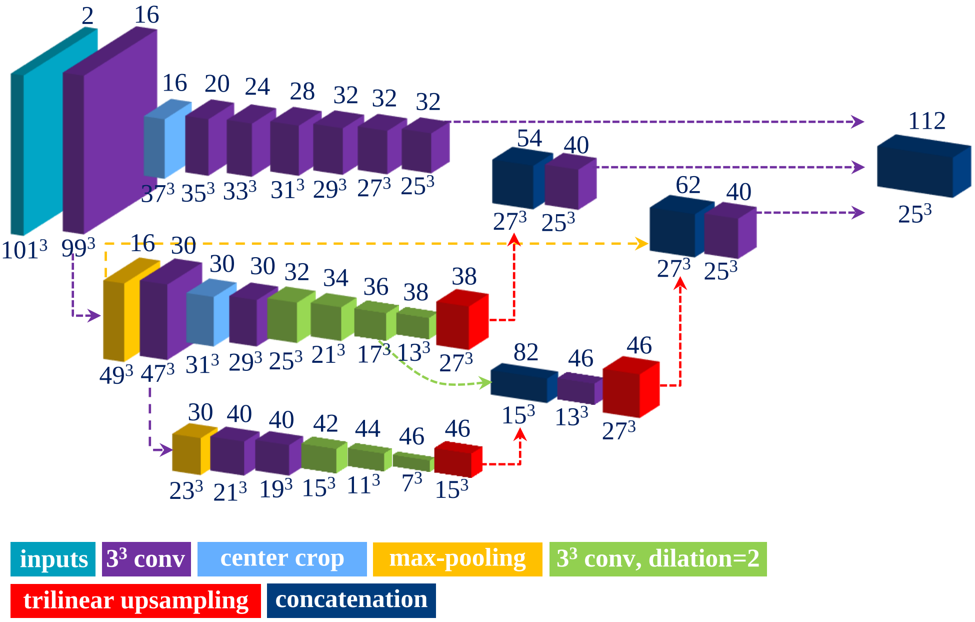
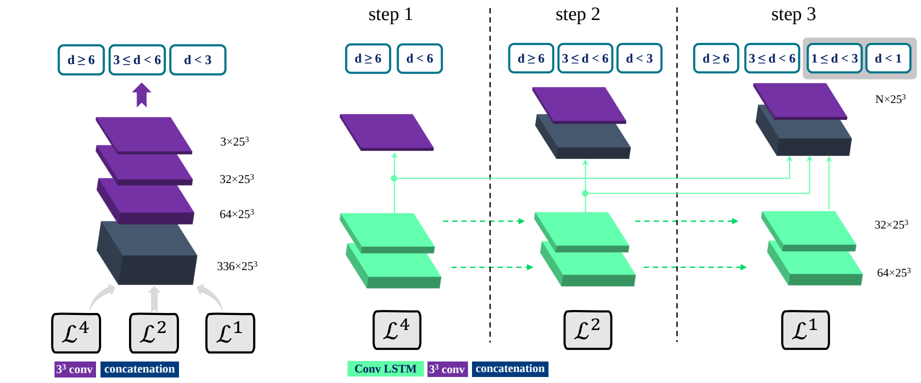


Figure 5.2: The RegNet architecture used for encoding the input image pair. This architecture is identical to the U-Net-advanced (Uadv) design proposed in [18], with the last three layers excluded. The number of feature maps and the spatial size are shown on top and bottom of each layer, respectively.

networks (see Fig. 5.1), that were previously trained for image registration [18]. Below, we report experiments both with freezing these weights and with keeping them trainable. When keeping them trainable, all layers are kept trainable, as recommended by Tajbakhsh et al. [128].

5.2.1.2 Decoder

In the decoder, the latent representations at each of the three resolutions $\mathscr{L}^i$ are considered to predict three output labels corresponding to registration misalignment: correct [0,3) mm, poor [3,6) mm and wrong [6, ∞) mm [33]. A straightforward choice for the decoder is to concatenate the latent feature maps and feed them to a convolutional neural network to predict the final labels. This approach is illustrated in Fig. 5.3a and is named multi-scale CNN. Instead, we propose a hierarchical approach using convolutional LSTM (Long Short-Term Memory) layers similar to [121] as they reported that predicting the labels from coarse to fine can improve the overall accuracy of the classification of fine labels in natural images. The coarse labels usually share a set of global features and for the fine labels more distinctive local properties are



(a) Multi-scale CNN decoder

(b) Hierarchical LSTM decoder

Figure 5.3: The decoder. The latent representations $\mathscr{L}^i$ of the three resolutions $\downarrow$4, $\downarrow$2 and 1 are merged and the final output predicts three misalignment labels: correct [0,3) mm, poor [3,6) mm and wrong [6, ∞) mm. In the CNN decoder (a), merging is done using concatenation. In the LSTM decoder (b), the latent representations $\mathscr{L}^i$ are given in sequence and the misalignment labels are gradually refined in a hierarchical manner. The labels inside the shaded boxes in the top-right of the figure represent the auxiliary labels.

extracted.

The LSTM unit was first proposed for machine translation where the input, output, and hidden states are all modeled as temporal sequences using fully connected units [129]. As this approach does not capture the spatial relations in the data, Shi et al. [122] proposed a convolutional LSTM unit, where the fully connected (FC) layers are replaced by convolutional layers. This way the unit is capable of capturing and encoding spatio-temporal information for visual series. We can imagine inputs and state as vectors standing on a spatial grid. The future state of a cell in the grid is calculated by the inputs and past states of its neighbors.

In the proposed LSTM decoder (Fig. 5.3b), rather than supplying the three latent representations $\mathscr{L}^i$ all at once, they are provided in sequence. Starting with $\mathscr{L}^4$, a coarse prediction of the registration error is first made, predicting only two labels: ‘good‘ registration with an error in the range $[0, \theta_1)$ mm, and ‘bad‘ registration with an error higher than that i.e. $[\theta_1, \infty)$ mm. In the experiments for example we have used $\theta_1 = 6$ mm. In the next time step of the convolutional LSTM, the $\mathscr{L}^2$ features are additionally considered, combining them with the hidden state of the previous time step. Now the output predictions are refined into three classes $[0, \theta_2)$ mm, $[\theta_2, \theta_1)$ mm and $[\theta_1, \infty)$ mm. We keep all the output probabilities unconditional similar to [121]. In the last time step, the latent representation $\mathscr{L}^1$ is used and combined with the hidden state, further refining the output prediction with splitting the previous smallest class to $[0, \theta_3)$ mm and $[\theta_3, \theta_2)$ mm. This way the predictions are built up in a hierarchical manner, step-by-step incorporating the multi-resolution embeddings of the input pair and step-by-step refining the registration error prediction.

In the final convolutional layers of both decoder designs, the softmax activation is used. For other convolutional layers in the CNN-based decoder, batch normalization and ReLu activation are utilized. In the LSTM design, cell outputs, hidden states, and gates (input, forget, output) have similar settings as in [122]. An additional output is allocated for each coarse label. For instance, in Fig. 5.3b, six outputs are available, four of them for fine labels and two for coarse labels. We perform experiments for various values of θ_i, where $i \in \{1,2,3\}$ and $\theta_1 \geq \theta_2 \geq \theta_3$.

5.2.2 Training data generation

In order to train the networks, we propose to artificially generate image pairs from the available real data. The main advantage of artificial generation is that numerous number of training samples can be obtained in an inexpensive way. Moreover, a dense ground truth is made, which is not achievable with other forms of ground truth such as manual landmarks or segmentation maps.

We use a similar approach as in [18] to artificially generate the DVFs and deformed image. Four types of artificial deformation are applied:

single frequency: This type of DVF is generated by perturbing B-spline grids. Since the grid knots are uniformly spaced, the generated DVF has only one random spatial frequency.

mixed frequency: A combination of the single frequency DVF filtered by a Gaussian kernel with a smaller sigma.

respiratory motion: Simulating the respiratory motion by expansion of the chest in the transversal plane, transition of the diaphragm in craniocaudal direction [35]. Finally, a random "single frequency" deformation is added.

identity transform: This type represents no misalignment between the images.

After creating the deformed images with the generated DVFs, to make the deformed images more realistic, several intensity augmentations are performed:

Gaussian noise: Gaussian noise with a standard deviation of $\sigma_N = 5$ is added to the deformed image.

Sponge model: Multiplying the intensity of the deformed moving image by the inverse of the determinant of the Jacobian of the transformation. This is an approximation based on the theory of mass preservation in the lung during breathing [73].

By applying the proposed artificial DVF generations, many image pairs can be generated for each image, by varying the hyper-parameters corresponding to each category.

5.3 Experiments and Results

5.3.1 Data

Experiments are performed using three chest CT studies: The DIR-Lab-COPDgene [75], the DIR-Lab-4DCT [74] and the SPREAD [47] studies.

In the DIR-Lab-COPDgene study, ten cases are available in inhale and exhale phases. The average image size and the average voxel size are $512 \times 512 \times 120$ and $0.64 \times 0.64 \times 2.50$ mm, respectively. 300 corresponding landmarks are manually annotated in each case.

In the DIR-Lab-4DCT study, ten cases with varying respiratory phases are available. We selected the maximum inhalation and maximum exhalation phases, as more manual landmarks are available in these phases (300 landmarks). The size of the images is approximately $256 \times 256 \times 103$ with an average voxel size of $1.10 \times 1.10 \times 2.50$ mm.

In the SPREAD study, 21 cases are available. Each case consists of a baseline and a follow-up image, in which the follow-up is taken after about 30 months. Both baseline and follow-up are acquired in the maximum inhale phase. The size of the images is about $446 \times 315 \times 129$ with a mean voxel size of $0.78 \times 0.78 \times 2.50$ mm. About 100 well-distributed corresponding landmarks were previously selected [73] semi-automatically on distinctive locations [48]. Two cases (12 and 19) are excluded because of the high uncertainty in the landmark annotations [73].

5.3.2 Experimental setup

5.3.2.1 Training data

In the SPREAD study, 10 , 1, and 8 cases are used for the training, validation, and test sets, respectively. The DIR-Lab-COPD study is used for training and validation only, where 9 cases are used for training and the remaining case for validation. The entire DIR-Lab-4DCT database (10 cases) is used as an independent test set. The validation set is mainly used for tuning the hyper-parameters and selecting the best approach. Since we initialized the weights of RegNet from the study of [18], we kept the training, validation, and test sets identical to that study, to avoid data leakage.

To generate training pairs, we use the artificial generations introduced in 5.2.2. The maximum magnitude of the DVF in each axis is set to 10 mm, so the maximum vector magnitude is about 17 mm. For each single image, 28 artificial DVFs and deformed images are generated by assigning random values to the variables of the single frequency, the mixed frequency and the respiratory motion deformations. Thus, in the training phase, a total number of 1064 artificially generated image pairs are used. All images are resampled to an isotropic voxel size of $1.0 \times 1.0 \times 1.0$ mm.

In the training phase, the patches are balanced based on the magnitude of the artificial DVFs. The probabilities of selecting patches in the range [0, 3), [3, 6) and 6, ∞) mm are 60%, 20% and 20%, respectively. This balancing is performed to make the training set more similar to the real world scenarios as the distribution of landmarks in the first range is usually higher.

5.3.2.2 Real image pairs

In this experiment, we estimate the registration error after registration in cases from the test set and compare it with the ground truth landmarks. Both fixed and moving images are taken from the same patient at different time points. In order to create a generic evaluation study, we collect samples by performing affine and four various conventional nonrigid registrations using 20, 100, 500, and 2000 iterations corresponding to overall poor registration quality to overall high quality registration. The common registration settings are: metric: mutual information, optimizer: adaptive stochastic gradient descent, transform: B-spline ([6]), number of resolutions: 3. After performing registration on the original fixed and moving images, the fixed and the deformed moving image after the registration are given as inputs to the proposed misalignment estimation method.

We define the target registration error (TRE) as the Euclidean distance after registration between the corresponding ith landmarks:

$$\mathrm{TRE}^{\mathrm{i}} = \|\boldsymbol{x}_F^{\boldsymbol{i}} - \boldsymbol{x}_D^{\boldsymbol{i}}\|_2, \tag{5.1}$$

where $\boldsymbol{x}_F$ and $\boldsymbol{x}_D$ are the corresponding landmark locations on the fixed and deformed moving images, respectively. A misalignment label is then assigned to each landmark, based on the magnitude of the TRE. The misalignment labels are defined based on the TRE value.

5.3.2.3 Network optimization

Optimizing the neural networks is done by the Adam optimizer [130] with a constant learning rate of 0.001. A stochastic mini-batch method is used with a batch size of 10. The cross-entropy loss is used for all experiments. In the LSTM design, the cross-entropy loss is applied to unconditional probabilities for all steps similar to [121]. The loss function is defined as follows:

$$\mathrm{loss} = -\frac{1}{N}\sum_{i=1}^{N}\Big(\sum_{s=1}^{S}\sum_{c\in C^s} 1\{x_i^s = c\}\log p_c\Big), \tag{5.2}$$

where N is the total number of voxels in a mini-batch, S denotes the number of steps, C^s represents the classes at step s, and p_c is the probability of class c in the output. The training is performed for 30 epochs by an NVidia RTX6000 with 24GB memory.

5.3.2.4 Software

The convolutional neural networks are implemented in Tensorflow [76], and image handling and artificial training data generation is implemented with SimpleITK [51]. `elastix` [52] is used to perform the conventional image registrations.

5.3.2.5 Additional methods

For further comparisons, two additional CNN methods are added: single-scale CNN and RegNet-t. In the single-scale CNN, only the encoded feature maps of the original resolution $\mathscr{L}^1$ is used. The weights of the encoder are kept trainable similar to the multi-scale CNN. In the RegNet-t experiment, first a three-resolution registration is performed by RegNet over the input pair [18]. The registration is performed over scales four, two and one in sequence, in which the input of each resolution is the fixed and deformed moving image of the previous resolution. Then, the magnitude of the predicted displacement vector field (DVF) is calculated and thresholded in the following ranges: [0,3), [3,6) and [6, ∞) mm. Finally, the labels “correct”, “poor” and “wrong” are assigned to them, respectively.

In addition, the proposed multi-stage hierarchical LSTM design is compared to a conventional learning-based method using random forests (RF), published earlier [33]. The random forests were trained on several hand-crafted intensity-based and registration-based features extracted from landmark neighborhoods. The output of the random forests predicted the registration error in mm. Three classes were generated by quantizing the regression results within the ranges [0,3), [3,6), and [6, ∞) mm, similar to the current study.

5.3.3 Evaluation measures

All evaluations are computed only from the landmark locations to maximize the quality of the ground truth. The misalignment labels are defined as correct, poor and wrong, when the TRE is in range [0,3), [3,6) and [6, ∞) mm, respectively, similar to [33]. We report the following statistics: overall accuracy, F1 score for each label separately, the average $\overline{\text{F1}}$ of the separate F1 scores, the number of misclassifications between the wrong and the correct label (two categories apart called cw misclassification), and finally Cohen’s kappa coefficient (κ) of the confusion matrix. The accuracy may be biased to the labels with a higher number of samples, whereas the $\overline{\text{F1}}$ and κ coefficient are more robust for imbalanced distributions.

Table 5.1: Landmark-based results on the training and validation set for tuning hyper-parameters. We report the mean values over all five registration settings: affine and B-spline registration after affine with 20, 100, 500, and 2000 iterations. The sub-indices c, p, and w correspond to the correct [0,3), poor [3,6), and wrong [6, ∞) mm classes. The best method is shown in bold and the second best method is shown in green. Total number of landmarks for all five registrations in SPREAD (cases 1 to 11) and DIR-Lab COPDgene studies are 5455 and 15000, respectively

		SPREAD (case 1 to 11)							DIR-Lab COPDgene (case 1 to 10)						
encoder	decoder	$\mathrm{F1_c}$	$\mathrm{F1_p}$	$\mathrm{F1_w}$	$\overline{\mathrm{F1}}$	Acc	κ	cw misclass	$\mathrm{F1_c}$	$\mathrm{F1_p}$	$\mathrm{F1_w}$	$\overline{\mathrm{F1}}$	Acc	κ	cw misclass
frozen	multi-scale CNN	85.8	52.5	83.5	73.9	78.1	0.58	39	**77.6**	52.5	85.8	72.0	**77.0**	0.60	387
trainable	multi-scale CNN	90.0	62.5	82.3	78.3	83.1	0.66	32	72.2	**61.4**	85.1	72.9	75.8	0.60	209
frozen	LSTM 6-3-1	92.4	54.9	83.3	76.9	85.5	0.68	52	76.9	38.5	**86.9**	67.4	76.2	0.59	391
trainable	LSTM 6-3-1	**93.0**	**63.6**	82.3	**79.6**	**86.3**	**0.71**	25	74.6	59.4	85.6	**73.2**	76.1	**0.61**	**148**
trainable	LSTM 12-6-3	83.0	54.2	**84.5**	73.9	75.6	0.56	**15**	56.7	56.3	84.6	65.8	71.9	0.53	368
trainable	LSTM 6-3-3	88.7	58.9	83.6	77.1	81.5	0.64	28	60.6	56.4	84.2	67.1	71.9	0.53	253

5.3.4 Results on the validation set

This experiment is mainly designed for tuning the hyper-parameters, i.e. the splitting values for the LSTM and to choose between the trainable and the frozen weights approach. We experiment with the two decoder architectures introduced in Section 5.2.1.2: the multi-scale CNN decoder and the hierarchical LSTM decoder. The encoding architecture is kept identical in all experiments and all weights are initialized from the pre-trained RegNet [18]. The results are reported for both frozen and trainable encoder weights. In the trainable experiment, the weights of all layers are kept trainable. Additionally, three different splitting values for the LSTM designs are tested as well.

Table 5.1 gives the results on the training and validation sets for the decoders with similar encoder design with frozen and trainable approaches. Please note that the training was performed on the artificial image pairs. However, these results are reported over real images pairs on the landmark locations. Total number of landmarks for all five registrations in SPREAD (cases 1 to 11) and DIR-Lab COPDgene studies are 5455 and 15000, respectively.

First, we compare the encoding parts between frozen and trainable approaches. In this evaluation, the splitting values of the LSTM design are set to 6, 3, 1 for θ_1, θ_2 and θ_3, respectively. As is shown in the top four rows of Table 5.1, based on $\overline{\mathrm{F1}}$, κ coefficient and the number of misclassifications between the wrong and the correct label (cw misclass), a consistent improvement can be achieved by utilizing a trainable encoder. The improvement of $\overline{\mathrm{F1}}$ in the SPREAD study is from 73.9% to 78.3% and 76.9% to 79.6%, and in the DIR-Lab COPDgene study from 72.0% to 72.9% and 67.4% to 73.2% for the multi-scale CNN and the hierarchical LSTM architecture, respectively. Accuracy (Acc) is more biased towards category c, as the number of samples for this label is much higher than for the other labels. In the SPREAD dataset, $\mathrm{F1_c}$ and the accuracy of the trainable encoders are better. However, in the DIR-Lab COPDgene set, $\mathrm{F1_c}$ and the accuracy of the frozen encoders are slightly better. On the other hand, the number of outliers significantly decreases in the DIR-Lab COPDgene study. All in all, we select the trainable approach for the encoder in the remainder of the paper.

Comparing the two decoders (with trainable encoder), the LSTM design obtained better performance in terms of $\overline{\mathrm{F1}}$, κ coefficient, the number of outliers, and accuracy, compared to the CNN, on both datasets. We keep both designs for further experiments on the independent test data.

We additionally experiment with the hierarchical splitting approach of the LSTM design, using various splitting values θ_i: 6-3-1, 12-6-3 and 6-3-3. We keep the misalignment labels of the last step equal to [0, 3), [3, 6) and [6, ∞) mm by merging the auxiliary labels. Therefore, in the LSTM design with the 6-3-1 splitting approach, labels [0, 1), [1, 3) are merged into a single label [0, 3), and in the LSTM design with the 12-6-3 splitting approach, labels [6, 12), [12, ∞) are merged into a single label [6, ∞). The results are given in the bottom two rows in Table 5.1. Based on the $\overline{\mathrm{F1}}$, κ coefficient and the number of cw misclassifications, the hierarchical splitting with values 6-3-1 achieved better performance. The $\mathrm{F1_w}$ score of LSTM 12-6-3 in the SPREAD study are relatively high. On the other hand, the $\mathrm{F1_c}$ of LSTM 6-3-1 is higher than the other LSTM designs. This indicates that utilizing an auxiliary label in a specific range can improve the performance in that range. All in all, we select the LSTM with 6-3-1 splitting values for the remainder of the paper.

5.3.5 Results on the independent test set

In this section, we investigate the performance of the proposed decoders in unseen test sets, i.e. the SPREAD study cases 13 to 21 and the DIR-Lab 4DCT cases 1 to 10. The total number of landmarks for each registration in SPREAD (case 13 to 21) and DIR-Lab 4DCT studies are 783 and 3000, respectively. For further comparisons, two additional methods are added in this experiment: single-scale CNN and RegNet-t (see Section 5.3.2.5). The landmark-based results are reported in Table 5.2 within five various registration settings (similar to the validation experiment): affine transformation, B-spline transformation with 20, 100, 500, and 2000 iterations. The B-spline registrations are performed after the initial affine transformation. The aggregation of all five registrations are presented in the “total” row.

As seen in Table 5.2, among the classification networks, in the “total” row, the multi-scale CNN and LSTM 6-3-1 achieved better results in terms of $\overline{\mathrm{F1}}$ score and the number of cw misclassifications. This demonstrates that utilizing information from different scales can improve the performance. The LSTM design performed better in the SPREAD study based on all of the measures in this table $\mathrm{F1_c}$, $\mathrm{F1_p}$, $\mathrm{F1_w}$, $\overline{\mathrm{F1}}$, accuracy (Acc), κ coefficient and the number of cw misclassifications. In the same evaluation in the DIR-Lab 4DCT study, there is no consistent superiority among the multi-scale classification networks. In terms of $\overline{\text{F1}}$, the multi-scale CNN gained slightly better results i.e. 75.9% in comparison with single-scale CNN (73.9%) and LSTM (73.1%). All in all, based on the number of cw misclassifications, the multi-scale CNN and the LSTM design performs better than the single-scale CNN.

Strikingly, direct quantization of the RegNet encoder (method RegNet-t) performs quite well for affine registration and for coarse B-spline registration with a small number of iterations (20 and 100), leading to improved kappa values compared to the other three classification networks. For instance, for affine registration, RegNet-t achieved the highest $\overline{\text{F1}}$ score of 78.2% and 83.4% for SPREAD and DIR-Lab 4DCT, respectively. However, for more realistic B-spline registration with a larger number of iterations, the LSTM and the multi-scale CNN methods perform better. For example for B-spline registration with 2000 iterations, a $\overline{\text{F1}}$ score of 68.9% and 63.9% were obtained for the LSTM on the SPREAD and DIR-Lab 4DCT datasets, respectively. Notably, the LSTM decoder performs much better in terms of the number of cw misclassifications compared to RegNet-t, especially for the DIR-Lab 4DCT dataset where this number decreases from 197 to 77 in the "total" row. The inference time on a 3D patch of size $205 \times 205 \times 205$ was approximately 2.4, 0.7, 1.3, and 2.8 seconds for RegNet-t, single-scale CNN, multi-scale CNN, and LSTM, respectively.

Detailed results for the LSTM 6-3-1 decoder are reported in Tables 5.3 and 5.4. Table 5.3 shows the confusion matrix for the three classes correct, poor, and wrong, for the results aggregated over all registration settings (the "total" row in Table 5.2). The vast majority of misclassifications is one category off, with only 0.23% (9/3915) and 0.51% (77/15000) of the misclassifications two categories off, for the SPREAD (case 13 to 21) and DIR-Lab 4DCT studies, respectively. The intermediate hierarchical prediction results for each of the LSTM time steps are given in Table 5.4. Such results are not available for the CNN-based decoder, as that architecture lacks the possibility for gradual refinement. In step 1, only low resolution latent representations are available ($\mathscr{L}^4$), with a prediction in two classes only: [0, 6) mm and above 6 mm. This results in F1 scores of 92.4% and 60.1% for these two classes, for the SPREAD data. The results are gradually refined, by adding higher resolution representations and by predicting more fine-grained registration error classes, see Table 5.4. It can be seen that as the LSTM refines its results, the F1_p and F1_w scores are gradually improved in both studies. From step2 to step3-merged all F1 measures improve, in particular for the DIR-Lab 4DCT study.

Visual examples of the predictions for LSTM 6-3-1, single CNN, multi CNN, and RegNet-t are illustrated in Fig. 5.4. The ground truth misalignment on the landmark locations are dilated for better visualization. The color bar in the top center image indicates the target registration error. For all predictions, a three-label output is

Table 5.2: Landmark-based results on the test set. We report metrics over all five registration settings: affine and B-spline registration after affine with 20, 100, 500, and 2000 iterations. The sub-indices c, p and w correspond to the correct [0,3), poor [3,6) and wrong [6, ∞) mm classes. The best method is shown in bold and the second best method is shown in green. Total number of landmarks for each registration in SPREAD (cases 13 to 21) and DIR-Lab 4DCT studies are 783 and 3000, respectively.

		SPREAD (case 13 to 21)							DIR-Lab 4DCT (case 1 to 10)						
registration	decoder	$F1_c$	$F1_p$	$F1_w$	$\overline{F1}$	Acc	κ	*cw* misclass	$F1_c$	$F1_p$	$F1_w$	$\overline{F1}$	Acc	κ	*cw* misclass
Affine	RegNet-t	**70.5**	**72.1**	**92.1**	**78.2**	**83.9**	**0.70**	**0**	88.4	**70.2**	**91.6**	**83.4**	**85.7**	**0.77**	12
	single CNN	41.3	51.5	87.8	60.2	75.1	0.49	7	86.1	59.8	90.7	78.9	83.1	0.72	9
	multi CNN	47.8	71.2	92.1	70.3	81.6	0.66	**0**	88.5	66.6	85.6	80.2	81.0	0.71	**2**
	LSTM 6-3-1	65.5	67.1	91.0	74.5	81.5	0.66	**0**	**88.6**	58.4	79.3	75.4	76.1	0.64	9
B-spline 20	RegNet-t	**89.6**	**67.7**	**82.8**	**80.0**	**83.0**	**0.69**	**2**	**92.0**	**67.3**	**88.7**	**82.7**	**85.5**	**0.77**	15
	single CNN	77.1	47.1	65.5	63.2	66.3	0.47	37	89.7	56.7	87.4	77.9	82.4	0.73	29
	multi CNN	83.7	64.4	82.1	76.7	77.4	0.62	**2**	90.2	64.0	82.2	78.8	80.5	0.71	**6**
	LSTM 6-3-1	88.4	65.6	82.1	78.7	81.2	0.67	**2**	91.2	57.3	77.8	75.4	78.5	0.67	**6**
B-spline 100	RegNet-t	95.0	51.4	75.3	73.9	90.0	0.60	8	**92.7**	**61.7**	**84.8**	**79.8**	**85.0**	**0.74**	25
	single CNN	84.6	30.6	53.0	56.0	73.2	0.36	42	88.6	47.4	83.9	73.3	80.1	0.67	55
	multi CNN	91.8	48.8	**76.4**	72.3	85.1	0.55	8	91.0	57.3	73.7	74.0	78.9	0.66	**9**
	LSTM 6-3-1	**95.6**	**56.1**	75.6	**75.8**	**90.4**	**0.65**	**3**	92.3	54.0	71.1	72.5	79.2	0.65	17
B-spline 500	RegNet-t	**96.7**	**48.5**	68.2	71.1	**92.7**	**0.58**	4	93.3	**55.5**	65.7	**71.5**	**82.8**	**0.64**	56
	single CNN	86.5	25.1	43.0	51.5	76.0	0.30	51	88.7	36.4	**75.4**	66.8	77.6	0.59	81
	multi CNN	93.7	43.8	73.8	70.5	88.3	0.52	10	91.4	53.3	62.8	69.2	79.0	0.61	**17**
	LSTM 6-3-1	95.7	44.4	**83.0**	**74.4**	91.4	0.57	**2**	**93.3**	50.8	60.8	68.3	81.1	0.61	23
B-spline 2000	RegNet-t	**96.7**	27.3	56.2	60.1	**93.0**	0.41	7	93.2	46.3	43.6	61.0	81.7	0.54	89
	single CNN	86.9	16.4	41.1	48.1	76.6	0.25	50	89.3	35.6	**71.7**	65.5	79.0	0.57	127
	multi CNN	93.6	24.1	72.7	63.5	87.7	0.39	8	92.8	**50.1**	57.6	**66.8**	81.2	**0.59**	41
	LSTM 6-3-1	96.2	**30.6**	**80.0**	**68.9**	92.2	**0.50**	**2**	**93.6**	42.9	55.3	63.9	**81.9**	0.56	**22**
total	RegNet-t	**94.2**	**63.4**	**87.9**	**81.8**	**88.5**	**0.76**	21	**92.4**	**61.6**	83.2	**79.1**	**84.1**	**0.73**	197
	single CNN	83.3	39.1	74.6	65.7	73.4	0.54	187	88.7	48.7	**84.4**	73.9	80.4	0.68	301
	multi CNN	90.3	59.2	87.7	79.1	84.0	0.70	28	91.2	59.2	77.3	75.9	80.1	0.68	**75**
	LSTM 6-3-1	93.6	60.4	87.8	80.6	87.4	0.75	**9**	92.3	53.8	73.2	73.1	79.4	0.66	77

Table 5.3: Confusion matrix of the landmark-based results on the test set, for the trainable LSTM 6-3-1 decoder. We report the aggregated values over all five registration settings: affine and B-spline registration after affine with 20, 100, 500, and 2000 iterations. The sub-indices c, p and w correspond to correct [0,3), poor [3,6) and wrong [6, ∞) mm classes. P and A refer to the predicted and actual labels for each class. Total number of landmarks for all five registrations in SPREAD (case 13 to 21) and DIR-Lab 4DCT studies are 3915 and 15000, respectively.

	SPREAD (case 13 to 21)				DIR-Lab 4DCT (case 1 to 10)		
	A_c	A_p	A_w		A_c	A_p	A_w
P_c	2441	117	3	P_c	7526	680	70
P_p	209	371	72	P_p	492	1757	1656
P_w	6	88	608	P_w	7	188	2624

illustrated i.e. correct [0,3) (green), poor [3,6) (yellow) and wrong [6, ∞) mm (red). An example of registration with affine and B-spline with 2000 iterations is given in Fig. 5.4a. LSTM 6-3-1 achieved the best performance among the others with only one misclassification out of 5 landmarks in this slice, where it incorrectly predicted

Table 5.4: Detailed hierarchical results of the landmark-based results on the test set, for the trainable LSTM 6-3-1 decoder. We report the aggregated values over all five registration settings: affine and B-spline registration after affine with 20, 100, 500, and 2000 iterations. The sub-indices c, p and w correspond to correct [0,3), poor [3,6) and wrong [6, ∞) mm classes. The shaded cells represent a combination of several fine-grained labels, as in earlier steps more coarse classes are predicted.

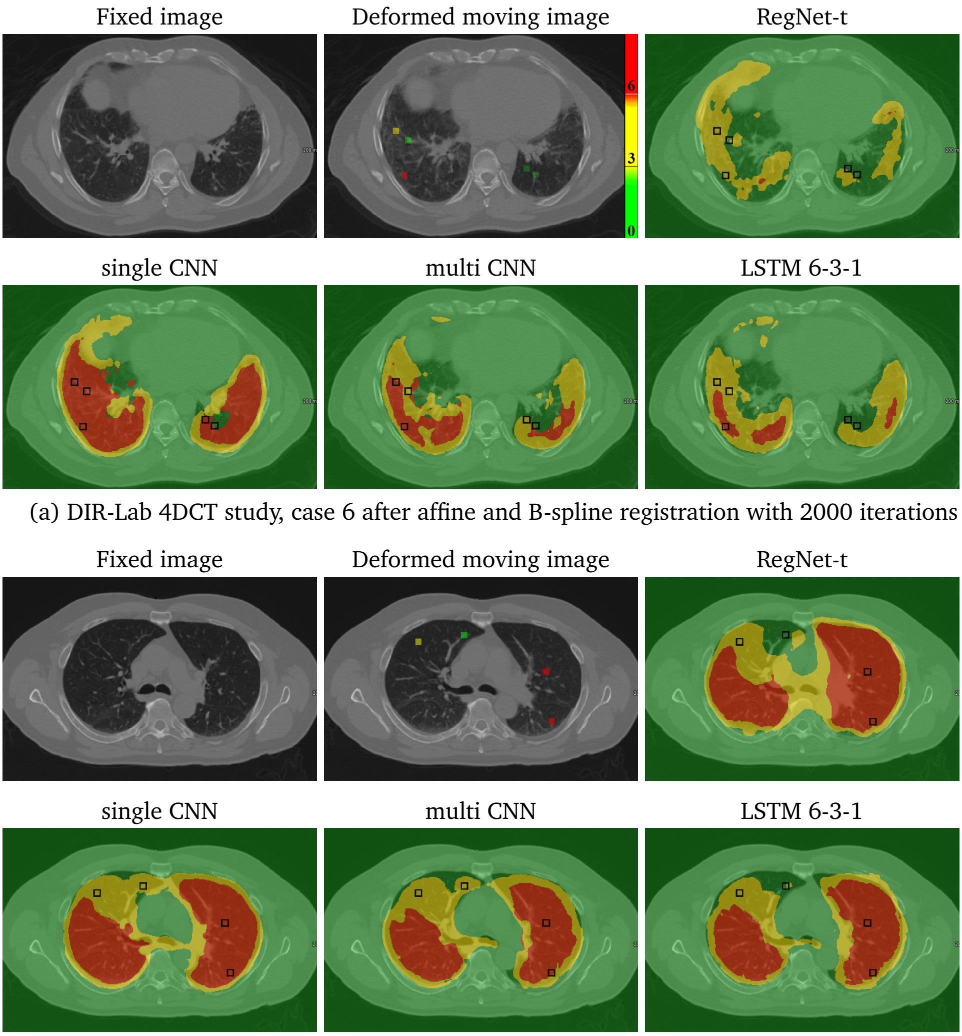
time	$F1_{c\ 0\text{-}1}$	$F1_{c\ 1\text{-}3}$	$F1_p$	$F1_w$	$\overline{F1}$	Acc	κ
SPREAD (case 13 to 21)							
step 1	92.4			60.1	77.1	89.9	0.55
step 2	94.3		53.0	68.9	72.1	83.9	0.66
step 3	23.2	64.6	60.4	87.8	59.0	60.6	0.44
step 3-merged	93.6		60.4	87.8	80.6	87.4	0.75
DIR-Lab 4DCT (case 1 to 10)							
step 1	84.2			14.9	49.6	73.3	0.11
step 2	83.6		28.0	22.9	44.8	61.6	0.32
step 3	53.8	67.2	53.8	73.2	62.0	63.3	0.50
step 3-merged	92.3		53.8	73.2	73.1	79.4	0.66

poor (yellow) label for the correct (green) landmark in the right lung (left side of this image). RegNet-t underpredicted in this slice and misclassified in the wrong (red) regions. Another example with only affine registration is given in Fig. 5.4b. In this slice LSTM 6-3-1 and RegNet-t predicted all four landmarks correctly.

5.3.6 Comparison with Random Forest method

The proposed multi-stage hierarchical LSTM design is compared to a conventional learning-based method using random forests (see Section 5.3.2.5 for details). We compare this method on the SPREAD (cases 13 - 21) and DIR-Lab 4DCT (cases 1 to 5) studies, i.e. we excluded cases 6 to 10 from DIR-Lab 4DCT as these cases were not present in the test set of [33]. Since the random forest method was designed to only predict nonrigid registration error, in this experiment we only included B-spline registrations with 20, 100, 500, and 2000 iterations, thus excluding the affine registration.

The results are reported in Table 5.5. In terms of $\overline{F1}$, the proposed LSTM design achieved significantly better results in both studies. On all F1 measures on both datasets, the LSTM method outperforms the random forest method, except for the $F1_c$ score on the SPREAD study, which were 93.6% vs 96.9% for LSTM vs RF. A compelling advantage of the LSTM method is that it can be applied to affine registrations as well



Fixed image

Deformed moving image

RegNet-t

single CNN

multi CNN

LSTM 6-3-1

(a) DIR-Lab 4DCT study, case 6 after affine and B-spline registration with 2000 iterations

Fixed image

Deformed moving image

RegNet-t

single CNN

multi CNN

LSTM 6-3-1

(b) DIR-Lab 4DCT study, case 7 after affine registration

Figure 5.4: Examples of the prediction output on entire image pairs registered using conventional registration techniques. The ground truth misalignment on the landmark locations are overlaid in the deformed moving images. These landmarks are dilated in this figure for a better visualization. The color bar indicates the target registration error, which is added on the top center image. For all predictions, a three-label output is illustrated i.e. correct [0,3) (green), poor [3,6) (yellow) and wrong [6, ∞) mm (red). (a) Results on the case 6 from the DIR-Lab 4DCT study. The deformed moving image is obtained after an affine and a B-spline registration with 2000 iterations. (b) Results on the case 7 from the DIR-Lab 4DCT study. The deformed moving image is obtained after an affine transformation.

Table 5.5: Landmark-based results on the overlapping part of the test set, comparing LSTM to the random forests method (RF) [33]. The results include B-spline registration with 20, 100, 500, and 2000 iterations. The sub-indices c, p and w correspond to correct [0,3), poor [3,6) and wrong [6, ∞) mm classes.

method	$F1_c$	$F1_p$	$F1_w$	$\overline{F1}$	Acc
SPREAD (case 13 to 21)					
RF	96.9	40.0	62.4	66.4	92.7
LSTM	93.6	60.4	87.8	80.6	87.4
DIR-Lab 4DCT (case 1 to 5)					
RF	88.2	42.3	34.7	55.1	77.3
LSTM	94.0	56.4	66.7	72.4	84.2

as nonrigid registrations. Another major advantage of the LSTM method is that the inference time is about 22 seconds (for an image size of 410 × 410 × 410 mm) compared to 3 hours for the random forests, where a lot of the time is spent in the feature calculation (registration and local normalized mutual information).

5.4 Discussion

We proposed a deep learning-based method to predict registration misalignment, using a hierarchical LSTM approach with gradual refinements. We performed a wide range of quantitative evaluations on multiple chest CT databases.

The performance of the compared decoders in Table 5.2 are not consistent in all registration settings. The B-spline registration with 2000 iterations represents the most common setting, as this represents an accurate registration. In this case the proposed hierarchical LSTM method achieved the best result in terms of $\overline{F1}$, κ coefficient and the number of cw misclassifications. In the "total" row, the number cw misclassifications of the LSTM method is much smaller than that of the RegNet-t. In the validation set in Table 5.1, the LSTM design achieved slightly better results in comparison to the multi-scale CNN design based on the $\overline{F1}$, κ coefficient and the number of cw misclassifications, showing that utilizing both the multi-resolution approach and hierarchical refinements can improve the misalignment predictions.

The proposed encoding mechanism using RegNet showed to be effective, as it achieved promising results even with a simple thresholding 'decoder' as used in RegNet-t. In predicting the misalignment of the affine registration, RegNet-t outperformed all other decoders. Since RegNet-t resamples images after each stage, potentially it can capture larger registration misalignment. We experimented with a similar setup using the LSTM approach, resampling after each step. However, the results of this experiment were not promising on the validation set. Another difference is that the

Table 5.6: A summary of some of the earlier approaches for estimating registration misalignment. For simplification, results are averaged over all reported test data. RF refers to a random forest and NA refers to “not available”.

article	output	method	data	training	testing	result
Hub et al. [35], 2009	continuous, local	perturbing input	chest CT, in-house	NA	artificial DVF	NA
Muenzing et al. [39], 2012	classification, local	cascade classifiers with intensity based features	chest CT, in-house	landmarks in real data	landmarks in real data	$\overline{\text{F1}}$ 85.2%
Sokooti et al. [33], 2019	regression, local	RF using intensity and registration based features	chest CT, in-house + public	landmarks in real data	landmarks in real data	MAE 1.42 mm, $\overline{\text{F1}}$ 60.75%
Saygili [111], 2020	regression, local	block matching + RF	chest CT, public	landmarks in real data	landmarks in real data	MAE 2.0 mm, Acc 81.8%
Eppenhof et al. [40], 2018	regression, local	CNN	chest CT, public	artificial DVF under 4 mm	landmarks in real data	RMSD 0.66 mm
Senneville et al. [115], 2020	classification, global	CNN + linear regression (classifier)	brain MR, public	artificial affine DVF	real data	Binary Acc 96.0%
Proposed method	classification, local	ConvLSTM	chest CT, in-house + public	artificial DVF under 17 mm	landmarks in real data	$\overline{\text{F1}}$ 76.5%

RegNet was trained on artificial data with a maximum deformation of 20 mm in each direction for the course resolution (RegNet^4), whereas the the maximum deformation in this study is set to 10 mm in each direction (about 17 mm in vector magnitude). It should be noted that in terms of the total number of cw misclassifications, the LSTM and CNN designs are still more in favor, which are reported as 9, 2, and 12 for the LSTM, multi-scale CNN and RegNet-t, in order (see the first four rows in Table 5.2).

The distribution of the labels “correct”, “poor” and “wrong” are highly imbalanced in image registration. For instance, in the test set within five registration settings, the distribution of samples are 67.8%, 14.7%, 17.5% in the SPREAD study and 53.5%, 17.5%, 29.0% in the DIR-Lab 4DCT for the labels correct, poor and wrong, respectively. In order to mimic the same distribution during training, the probability of selecting patches in the range [0,3), [3,6) and [6, ∞) mm are set to 60%, 20% and 20%, respectively (see Section 5.3.2.1). However, this can influence the first step of the LSTM training as the sampling becomes imbalanced again in this step.

A comparison to previous methods for predicting registration misalignment is not trivial due to differences in approach (classification, regression) as well as the use of different test datasets. Table 5.6 gives an overview of several methods from the literature. A classification-based approach to estimate registration misalignment was also presented in [39]. They proposed a classical learning-based approach using several hand-crafted features. Muenzing et al. [39] reported F1 scores of 95.3%, 73.8% and 86.6% in the labels [0,2), [2,5) and [5, ∞) mm. It is not trivial to compare our results to this method because the evaluation is done on different data and using different thresholds for labels. When it comes to the dense prediction for an entire image, calculating those hand-crafted features become quite time-consuming. In the CNN-based approaches, Eppenhof et al. [40] proposed a regression network to predict registration misalignment. They trained on the odd-numbered images from the DIR-Lab-4DCT and the COPDgene data sets and tested on the even-numbered scans, and on two additional chest CT studies. They reported a root-mean-square deviation (RMSD) of 0.66 mm between the ground truth TRE and the predicted one for landmarks with ground truth TRE below 4 mm. The main limitation is that the method predicts registration misalignment smaller than 4 mm only. Since our proposed method has one label corresponding to misalignment in the range $[6, \infty)$ mm, a quantitative comparison is not feasible. In Section 5.3.5, we drew a comparison between the proposed LSTM method and a random forests regression method [33]. We kept the experiment settings as similar as possible. However, some minor differences still exist. For instance, the voxel size in the LSTM method is resampled to an isotropic size of [1, 1, 1] mm, whereas in the random forests method, resampling is not applied. Since one of the proposed features in [33] was the variation of the transformations with respect to the initial states of the B-spline grid, it is not possible to use this approach for affine registration.

In this study, we proposed to use RegNet [18] to encode a pair of images using a multi-resolution approach to high-dimensional feature maps. Although the experiment with a simple decoder as RegNet-t reveals that encoding with RegNet is quite powerful, potentially, any registration network can be used instead of RegNet. It could therefore be interesting to perform a comparison between different network architectures. The proposed method is designed with three resolutions of the input given in three steps to the LSTM block. At the third resolution, the receptive field of the network is usually larger than an entire chest CT image (with a spacing of 1 mm). Thus, potentially no further contextual information can be achieved by increasing the number of resolutions. However, varying the number of steps in the LSTM block can be an interesting experiment. We experimented with three steps, but with various splitting values in Section 5.3.4. The number of steps of the LSTM can be increased even with identical inputs, similar to [121].

The proposed method is expected to be sensitive to anatomical changes like tumor growth. Thus, it may detect those regions as a suboptimal local registration. This limitation may potentially be addressed by adding a new type of deformation to the artificial training data strategy, which mimics such anatomical changes. For example, in this study we modelled respiratory motion specifically designed for lungs (see Section 5.2.2), as we performed all experiments on chest CT scans. This may be extended with additional realistic artificial data generation types, for other use cases.

However, the proposed training and prediction methods are generic and independent of the image type. In future work, the proposed method could be evaluated on other modalities and anatomical sites as well. Although all nonrigid experiments in this study are performed using B-spline registration, potentially, the proposed method is independent of the registration paradigm and can be applied to other nonrigid registration methods.

5.5 Conclusion

We proposed a framework for classifying registration misalignment using deep learning, consisting of encoding relevant features in a latent space and a hierarchical and gradually refining LSTM decoder for the prediction. Multi-resolution contextual information is incorporated in the design. The network is fully trained over artificially generated images, while the evaluation is performed over realistic chest CT scans. The proposed decoder is compared with two other CNN-based decoders and a method based on the output of a deep learning based registration RegNet-t. A comprehensive study is performed on two independent test sets (SPREAD case 13 to 21, and DIR-Lab 4DCT) with various registration settings. In the B-spline registration with 2000 iterations, the proposed method achieved an $\overline{\mathrm{F1}}$ and number of cw misclassifications of 68.9%, 2 and 63.9%, 22 in the SPREAD and the DIR-LAB 4DCT studies, respectively. In the aggregation of all registration settings, the proposed LSTM design obtained the least number of cw misclassifications. At the inference time, the proposed method can predict a dense map in about 22 seconds.