A Differentiable Convolutional Distance Transform Layer for Improved Image Segmentation

Duc Duy Pham(B), Gurbandurdy Dovletov, and Josef Pauli

Intelligent Systems, Faculty of Engineering, University of Duisburg-Essen, Duisburg, Germany
duc.duy.pham@uni-due.de

Abstract. In this paper we propose using a novel differentiable convolutional distance transform layer for segmentation networks such as U-Net to regularize the training process. In contrast to related work, we do not need to learn the distance transform, but use an approximation, which can be achieved by means of the convolutional operation. Therefore, the distance transform is directly applicable without previous training and it is also differentiable to ensure the gradient flow during backpropagation. First, we present the derivation of the convolutional distance transform by Karam et al. [6]. Then we address the problem of numerical instability for large images by presenting a cascaded procedure with locally restricted convolutional distance transforms. Afterwards, we discuss the issue of non-binary segmentation outputs for the convolutional distance transform and present our solution attempt for the incorporation into deep segmentation networks. We then demonstrate the feasibility of our proposal in an ablation study on the publicly available SegTHOR data set.

1 Introduction

In medical image computing, semantic segmentation of anatomical structures from various imaging modalities is a crucial task to aid in image based diagnostics. Therefore, research on automated segmentation methods is a major topic in the medical computing domain, since manual segmentation is expensive and time consuming. Especially deep learning strategies have become popular approaches to achieve state of the art results. In supervised settings, these usually require a desired ground truth segmentation, used to calculate a loss function, which is minimized during training. Mean squared error, categorical cross entropy and dice loss are common error functions, which directly make use of the ground truth and are applied for segmentation tasks.

These kind of error functions, however, usually employ a pixel-to-pixel comparison and therefore reduce the segmentation task to a pixel-wise classification task. They do not directly leverage information about higher order features, such as shape or texture. Specifically, pixels are considered independent of each other, thus, error correction for one pixel does not influence the error of another.
pixel. Consequently, recent research aims at incorporating shape priors into the segmentation process by means of an additional regularization term within the loss function, that also captures higher order information.

One possible option is to infer shape information by means of a learned latent representation of the ground truth segmentation. Oktay et al. [9] utilize a pre-trained autoencoder for shape preservation. The autoencoder’s encoding component is used to regularize the weight adaptation process of a generic segmentation network during training. This is motivated by Girdhar et al.’s work on establishing 3D representations of objects from 2D images [5]. Pham et al. [11] present a 2D end-to-end architecture, in which an autoencoder, trained for shape representation, is imitated in latent space by a separate encoder, to directly leverage the autoencoder’s decoder for shape consistent segmentation.

Another strategy is to include a well-established shape representation, particularly the distance transform, into the learning process. Comparing a one-hot encoded segmentation mask with its corresponding distance transform in each channel, the latter contains distance information about the closest object boundary in every pixel, whereas a binary mask only holds binary information of whether the structure of interest is present or not. In Fig. 1 the differences in binary images and in (Manhatten) distance transforms are illustrated on a simple toy example, in which two pixel values are swapped. For the binary representation, one can notice that only the affected pixels yield a difference, whereas in the corresponding distance transforms, the simple swap has a larger impact on the distance transform’s landscape.

Rousson et al. [13] leverage the idea of shape representation by means of signed distance transforms, proposed by Paragios et al. [10], to incorporate shape priors into the level set framework. Cremers et al. [3] also base their work on the distance transform’s shape representation to enforce shape priors. Naturally, incorporating the distance transform into deep neural networks is a plausible step to model inter-pixel relationships, as also noted in Ma et al.’s work [7]. Dangi et al. [4] apply distance map regression in a multi-task learning setting for cardiac MR image segmentation. They propose a regularization framework by formulating an Euclidean distance map regression objective, that is pursued by a sub-network of their segmentation architecture. In a related fashion Bai and Urtasun [1] incorporate the watershed transform by fundamentally learning the distance transform within image objects for instance segmentation. Bui et al. [2] propose a similar multi-task approach, in which the geodesic distance is approximated as a learning task for neonatal brain segmentation. Similarly, Navarro et al. [8] also include the learning task of distance transform approximation in their multi-task segmentation approach. In these contributions, however, the distance transform needs to be learned, since the implementation of the distance transform is often not differentiable.

In this work we propose using Karam et al.’s [6] derivation of a convolutional distance transform approximation for the application in a deep learning context. To the best of our knowledge, this is the first time an adhoc differentiable convolutional distance transform layer is proposed for deep segmentation networks. In Sect. 2 we will discuss the underlying methods, starting with Karam et al.’s [6]
Convolutional distance Transform, subsequently proposing a cascaded variant for large images, and posing an embedding suggestion into deep learning frameworks. In Sect. 3 we present our experimental setting and discuss our results in Sect. 4 before concluding with Sect. 5.

2 Methods

![Fig. 1. Differences in binary images (a) and (b) are only visible in the affected pixels, highlighted in (b). For the corresponding (Manhatten) distance transforms (c) and (d), differences propagate to further pixels, emphasized in (d), as these change the foreground shape and thus the distance landscape.](image)

A distance map of a binary image yields the distance of each pixel to its closest foreground boundary pixel. Common applicable distances are the Manhatten and the Euclidean distance. A major advantage of this type of image representation is the provision of information about boundary, shape, and location of the object of interest. We denote the distance between two pixel positions \( p_i \) and \( p_j \) in an image as \( d(p_i, p_j) \). Then a distance transform \( D_I : M \times N \rightarrow \mathbb{R}^+ \) for a binary image \( I \) of resolution \( M \times N \) can be defined pixel-wise as:

\[
D_I(p_i) = \min_{p_j : I(p_j) = 1} \{d(p_i, p_j)\}
\]  

(1)

To incorporate an adhoc distance transform into the deep learning setting, we follow Karam et al.’s [6] derivation and only consider translation invariant distances, i.e.

\[
d(p_i, p_j) = d(p_i + p_k, p_j + p_k)
\]

(2)

for any image positions \( p_i, p_j \in M \times N \) and any translation \( p_k \in \mathbb{R} \times \mathbb{R} \). Although most distances are translation invariant, this restriction needs to be mentioned, as there are also counter examples, such as the SNCF distance.

2.1 Convolutional Distance Transform (CDT)

For the computation of the distance transform, Eq. (1) shows, that we need to find the minimal distance to a boundary pixel from a given point. The minimum
Differentiable Convolutional Distance Transform Layer

function can be approximated by a log-sum-exponential. Let \(d_1, \ldots, d_n\) denote \(n\) distances, then the minimum function can be reformulated as

\[
\min\{d_1, \ldots, d_n\} = \lim_{\lambda \to 0} -\lambda \log \left( \sum_{i=1}^{n} \exp \left( -\frac{d_i}{\lambda} \right) \right),
\]

for \(1 > \lambda > 0\). The idea is, that the exponential yields very small values, the larger the distances are, as these are artificially increased by \(\lambda\) and negated in the argument. Therefore, larger distances have a significantly smaller impact on the sum than small distances. In the extreme case the exponential of large distances seek zero, leaving only the exponential of the smallest distance in the sum. The subsequent logarithmic function then reverts the exponential operation, leaving an approximation of the minimum function. With Eq. (3), it is possible to reformulate the distance transform of Eq. (1) to

\[
D_I(p_i) = \lim_{\lambda \to 0} -\lambda \log \left( \sum_{p_j: I(p_j) = 1} \exp \left( -\frac{d(p_i, p_j)}{\lambda} \right) \right)
\]

Since a translation invariant distance is assumed, the distance between two points can be rewritten to

\[
d(p_i, p_j) = d(p_i - p_j, p_j - p_j) = d(p_i - p_j, 0)
\]

Therefore, the distance transform can be formulated as

\[
D_I(p_i) = \lim_{\lambda \to 0} -\lambda \log \left( \sum_{p_j} I(p_j) \exp \left( -\frac{d(p_i - p_j, 0)}{\lambda} \right) \right),
\]

which is the definition of a convolution. Thus, for a small \(\lambda > 0\), the distance transform can be approximated by means of a convolution of the binary image \(I\) with a kernel \(\exp \left( -\frac{d(\cdot, 0)}{\lambda} \right)\), i.e.:

\[
D_I \approx -\lambda \log \left( I \ast \exp \left( -\frac{d(\cdot, 0)}{\lambda} \right) \right),
\]

where \(\ast\) is the convolutional operator. Since all operations are differentiable, this approximation may be integrated as a differentiable convolutional distance transform layer into current deep learning frameworks. It shall be noted, that Karam et al.’s work \[6\] also proposes variants of this convolutional distance transform. Initial experiments however showed most promising transforms for the presented formulation.

2.2 Cascaded Distance Transform for Large Images

A major drawback of the convolutional design of the distance transform (in all variants) is that the kernel size theoretically needs to be as large as the diagonal
of the input image. This is to ensure that even very sparse binary images can be distance transformed by the proposed method. Otherwise background pixels that are not within the kernel size reach of a foreground pixel would be assigned a distance of 0. This circumstance, however, yields the following two issues:

- The large kernel size leads to an increased computational complexity for the convolutional operation.
- For very large distances the exponential term for the kernel design in Eq. (7) may approach zero, decreasing the numeric stability of the logarithmic expression within the convolutional distance transform (CDT). This issue particularly arises for large images with only few foreground pixels. Figure 2(c) shows the CDT of a toy example image (Fig. 2(a)). It is clearly visible, that the CDT was only capable to calculate the distances for a specific range, before becoming unstable, in comparison with a standard Manhatten distance transform implementation in Fig. 2(b).

We address these issues by proposing a cascade of local distance transforms to reduce the computational complexity and overcome the numerical instability. Instead of directly computing the distance transform with a large kernel, we suggest cascading distance transforms with smaller kernels to approximate the actual transform. Since the kernel size determines the maximal distance that can be measured, it is necessary to accumulate the calculated distances to form the final distance transform approximation.

Let \( k \) denote the kernel size. Then the maximal distance to a foreground point that can be captured by the CDT is limited to a range of \( \lfloor \frac{k}{2} \rfloor \). For all background points that are further away than \( \lfloor \frac{k}{2} \rfloor \) from a foreground point, Eq. (7) yields a distance of 0, as within the kernel range, there are only background points. The idea is to iteratively extend the binary input image by the area, for which a distance calculation was possible by the locally restricted CDT, i.e. by all points which fulfill the condition that the calculated distance is greater than zero. This extended binary image can then be used to compute a new locally restricted distance transform by means of the small kernel. The calculated distances can then be utilized with the distances of the previous iterations to form the final distance transform. For the \( i \)-th iteration, let \( I^{(i)} \) denote the extended binary image, and let \( D^{(i)}_I \) denote the local CDT of \( I^{(i)} \). For the \( i \)-th iteration we assume that the original foreground area has been extended by a margin of \( i \cdot \lfloor \frac{k}{2} \rfloor \). Therefore this offset distance is additionally added to the current distances to compensate the lower kernel size. Thus, the cascaded distance transform \( D^*_I \) is updated by the current distances by adding \( i \cdot \lfloor \frac{k}{2} \rfloor + D^{(i)}_I \), wherever \( D^{(i)}_I(p) > 0 \) holds. Let \( diag \) denote the diagonal of the input image \( I \), then at most \( \lceil \frac{diag}{\frac{k}{2}} \rceil \) of such local distance transforms are necessary to cover the whole image. Algorithm 1 summarizes this suggested procedure. Let \( w, h \) denote width and height of the input image and \( k \) the kernel size used to compute the CDT. Then in general its computational complexity is given by \( \mathcal{O}(w \cdot h \cdot k^2) \) operations. Our proposed procedure can drastically reduce the number of operations from initial \( \mathcal{O}(w \cdot h \cdot diag^2) \) (for a naive implementation without using separable kernels).
Fig. 2. Toy example. By means of a toy example with a resolution of 512 × 512 (a) it becomes apparent, that compared to a standard Manhatten distance transform (b) the original global CDT (c) becomes numerically unstable for background points with large distances. (d)–(f) show the resulting cascaded CDTs with $k = 3, 5, 7$, respectively.

Since the maximally possible measured distance of $d(\cdot, 0)$ in Eq. (7) is restricted by the kernel size, a small kernel size additionally yields a more stable
Algorithm 1. Cascaded Convolutional Distance Transform

1: function CASCaded_CDT($I, \text{diag}, k$)
2: \hspace{1em} $s \leftarrow \lceil \frac{\text{diag}}{k^2} \rceil$
3: \hspace{1em} $I^{(0)} \leftarrow I$
4: \hspace{1em} $D^*_I \leftarrow I \cdot 0$
5: \hspace{1em} for $i=0$ to $s$ do
6: \hspace{2em} $D^{(i)}_I \leftarrow \text{CDT}(I^{(i)}, k)$
7: \hspace{2em} $I^{(i+1)} \leftarrow I^{(i)}$
8: \hspace{1em} for all $p : D^{(i)}_I(p) > 0$ do
9: \hspace{2em} $D^*_I(p) \leftarrow D^*_I(p) + i \cdot \lfloor \frac{k}{2} \rfloor + D^{(i)}_I(p)$
10: \hspace{2em} $I^{(i+1)}(p) \leftarrow 1$
11: return $D^*_I$

computation of the logarithmic term as the exponential does not tend to approach zero. Figures 2(d)–(f) show the cascaded CDTs with kernel sizes of 3, 5, 7, respectively. In comparison to the standard Manhatten distance transform in Fig. 2(b), it becomes apparent that the offset assumption after each iteration yields an error that is propagated to points with further distances. This error decreases with increasing kernel size. Thus, with our proposed procedure there is a trade-off between numerical stability by means of smaller kernel sizes and accuracy through larger kernel sizes that needs to be considered. We argue that for the purpose of considering inter-pixel relationships in the weight optimization process of training a convolutional neural network this approximation of the distance transform suffices. Karam et al. [6] also address this issue by multiplexing multiple $\lambda$ values, however initial experiments showed that the choice of these values heavily influence.

2.3 Convolutional Distance Transform for Deep Learning

The previous section describes an adhoc cascaded convolutional approximation method of the distance transform for binary images. We propose using this approximation to extend common segmentation networks, such as Ronneberger et al.’s U-Net [12], in order to equip the segmentation loss with an additional regression loss, which compares the distance transform of the network’s prediction with the distance transform of the ground truth. Since distance transforms are particularly sensible to distortions, the comparison of distance transformed ground truth to the distance transformed prediction may lead to less noisy segmentation results. Figure 3 shows the general idea, of how to extend the U-Net segmentation network with the proposed distance transform layer. In addition to the usual segmentation loss, e.g. the Dice loss, the predicted segmentation and the ground truth segmentation are both passed through the cascaded CDT layer to achieve the distance transforms of prediction and ground truth, respectively. These distance transforms contribute to a regression loss, e.g. the mean squared error, that considers inter-pixel relationships through the distance transforms.
Fig. 3. Network Architecture. The Convolutional Distance Transform Layer can be attached to arbitrary segmentation networks. In addition to the segmentation loss, a regression loss of distance maps is calculated.

However, it needs to be noted that the segmentation’s output is usually not binary. Assuming a final softmax or sigmoid layer, the output values for each channel vary between 0 and 1. It was necessary to assume a binary image to be able to restructure Eqs. (4) to (6). A major disadvantage of Eq. (6) is that for gray scale images, \( I(p_i) \) may be a lot larger than the exponential for large distances, even if \( I(p_i) \) is rather small. Therefore, even small probabilities of the segmentation output are considered in the sum and may be depicted as foreground pixels, distorting the actual distance map, as can be seen in Fig. 4. Here a toy example of a gray scale image is shown (Fig. 4(a)), in which the lower left square is set to a very low intensity of 0.001. The computed CDT (Fig. 4(c)), however, appears nearly identical to the CDT of the corresponding binary toy example (Fig. 4(b)). As can be observed in Fig. 4(d), the only differences of the computed distance transforms occur within the low intensity pixels, whereas the remaining distance transform’s landscape does not show any changes. This is very problematic, as in the segmentation prediction even pixels with very low probabilities would then be considered as foreground pixels in the CDT. We address this problem by proposing the following soft-threshold work around. Let \( C \) denote the number of classes, and let \( y_c(p_i) \) be the prediction for class \( c \) at position \( p_i \). Then we can soft-threshold the prediction by

\[
\tilde{y}_c(p_i) := \text{ReLU} \left( y_c(p_i) - \frac{C - 1}{C} \right).
\]  

(8)

This soft-threshold sets any prediction score below \( \frac{C - 1}{C} \) to zero. Therefore, we enforce strong and correct predictions, as weak correct prediction scores are not registered for the distance transform and negatively impact the regression loss. Figure 4(e) shows the resulting CDT after applying the proposed soft-threshold,
(a) Toy example: gray scale image  
(b) Cascaded CDT of Fig.2(a)  

(c) Cascaded CDT of (a)  
(d) Difference of (c) and (b)  

(e) Cascaded CDT of (a) with soft threshold  
(f) Difference of (e) and (b)  

Fig. 4. Toy example. (a) shows a gray scale image, in which the intensity within the lower left gray square is set to 0.001. (b) shows the cascaded CDT of Fig.2(a) as reference. Although the intensity of the gray area is very low, the cascaded CDT (c) shows a nearly identical landscape to (b). The only difference can be found within the lower left square (d). Application of the soft-threshold (assuming 2 classes) diminishes the low intensity area (e)–(f).

assuming two classes. Low intensity pixels are considered as background, so that a significant change in the distance transform landscape compared to the reference can be observed in Fig. 4(f).
3 Experiments

3.1 Data

We conducted ablation studies on the example of thoracic segmentation from CT scans of the thorax, using the publicly available SegTHOR CT data set [14] consisting of 40 CT volumes with corresponding ground truths of esophagus, heart, trachea, and aorta, for training and 20 CT volumes for testing. The goal of our ablation study is to investigate the influence of our proposed distance transform layer on the segmentation output. Therefore, we did not aim to outperform optimized ensemble methods within the SegTHOR challenge, but set value on a valid comparison. Thus, we trained a U-Net architecture with and without our proposed layer on the same training and validation set to ensure fair comparability. In a hold-out validation manner, we trained both models on the 40 available training volumes, and submitted the predictions. For evaluation the Dice Similarity Coefficient (DSC) and the Hausdorff Distance (HDD) were considered as evaluation metrics, which were both provided by the challenge’s submission platform.

3.2 Implementation Details

We implemented a 2D U-Net and the convolutional distance transform layer in Tensorflow 1.12 with an input size of 256 $\times$ 256. Thus, we resized the CT slices to the corresponding image size. Our U-Net implementation yields 5 image size levels with 2 convolutional layers, batch normalization and a max-pooling layer in each level. Starting with 32 kernels for each convolutional layer in the first size level of each contracting path, we doubled the number of kernels for each size level on the contracting side and halved the number of kernels on the expansive side. We used a kernel size of 3 $\times$ 3 for every convolutional layer and 2 $\times$ 2 max-pooling in each architecture. We used the standard dice loss $L_{dice}$ as loss function for U-Net and used mean squared error for the regression loss $L_{dist}$ of the distance transforms. We constructed a total loss function $L_{total} := L_{dice} + w_{dist} L_{dist}$ with weight $w_{dist} := 0.5$ to train the U-Net, equipped with our additional distance transform layer. The optimization was performed with an Adam Optimizer with an initial learning rate of 0.001. The training slices were augmented by means of random translation, rotation and zooming. With a batch size of 4, we trained both models for 200 epochs and chose the model with best validation loss for evaluation. For the distance transform layer, we chose $\lambda := 0.35$, as suggested by Karam et al. [6]. The experiments were conducted on a GTX 1080 TI GPU.

4 Results

It should be noted that the main focus is the ablation study and the image resizing to 256 $\times$ 256 canonically decreases the segmentation quality in comparison to methods that use the full image size. Table 1 shows the achieved DSC scores
Table 1. Achieved DSCs for each organ.

|            | Esophagus | Heart       | Trachea    | Aorta     |
|------------|-----------|-------------|------------|-----------|
| U-Net      | 0.726312  | 0.809626    | 0.770571   | 0.853953  |
| U-Net with distance transform | **0.739028** | **0.822095** | **0.785847** | 0.853011  |

of the trained models for esophagus, heart, trachea, and aorta. It is observable, that with the extension of the convolutional distance transform the scores increase for all organs, except for the aorta. In this case the standard U-Net yields marginally better results. While the DSC score improves by more than 1% for the other organs, for the aorta the additional layer does not seem to bring any benefit. We applied Wilcoxon significance tests with a significance level of 5% on our evaluation results. We found that the improvements in DSCs for Trachea and Esophagus are significant with p-values of 0.0169 and 0.0040, respectively. For the DSC improvement in heart segmentation and the DSC difference in aorta segmentation, we could, however, not find any significance.

The improvements can also be noticed in Table 2, in which the Hausdorff distances are depicted. For esophagus, heart, and trachea the distances decrease with our proposed layer, showing an improvement by approximately 25–30%. However, for the aorta a slightly worse mean distance is observed. This may be due to the fact, that the aorta seems to be a rather simple structure, that U-Net can already easily extract. The improvements in Hausdorff distance are especially noteworthy, as we use a distance based regularization technique to improve the segmentation. Regarding the Wilcoxon significance test we could observe significant improvements for Trachea, Esophagus and Heart with p-values of 0.0072, 0.0008 and 0.0400, respectively. This underlines our assumption that our proposed layer adds significant value to more complex shapes, whereas simple structures as the almost circular aorta and heart slices are already well extracted by a standard U-Net. Figure 5 shows exemplary segmentations of both models on test data slices. The top images indicate better performance for esophagus segmentation with the proposed layer, while the bottom images show superior segmentation results with our layer for the trachea. In both top and bottom row, the segmentations of the aorta do not show much difference.

Table 2. Achieved Hausdorff Distances (HDD) for each organ.

|            | Esophagus | Heart       | Trachea    | Aorta     |
|------------|-----------|-------------|------------|-----------|
| U-Net      | 1.511865  | 1.949374    | 2.137093   | **1.900747** |
| U-Net with distance transform | **1.113825** | **1.533211** | **1.649077** | 2.004237 |
Fig. 5. Exemplary segmentations on the test set from both models. Top images indicate better performance for esophagus segmentation, bottom images for trachea.

5 Conclusion

In this paper we propose a novel differentiable convolutional distance transform layer for segmentation networks, that can be used adhoc without prior training. We present a cascaded procedure to reduce the computational complexity and to overcome the numerical instability. Additionally we suggest a soft-threshold work around to address the demonstrated issue regarding non-binary segmentation outputs. The proposed layer is used to regularize the training process. We conducted ablation studies on the example of the segmentation of thoracic organs and used the SegTHOR data set for training and evaluation. The experiments show promising results, as compared to an equally trained U-Net our extension yields significant improvements for most organs, particularly regarding Hausdorff distance. We demonstrated on this example, that a combination of proven non-deep learning concepts, such as the distance transform, with deep learning methods may yield great potential. In the future, we aim at extending our proposed layer for 3D segmentation networks.
References

1. Bai, M., Urtasun, R.: Deep watershed transform for instance segmentation. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 5212–5220 (2017)

2. Bui, T.D., Wang, L., Chen, J., Lin, W., Li, G., Shen, D.: Multi-task learning for neonatal brain segmentation using 3D dense-Unet with dense attention guided by geodesic distance. In: Wang, Q., et al. (eds.) DART/MIL3ID 2019. LNCS, vol. 11795, pp. 243–251. Springer, Cham (2019). https://doi.org/10.1007/978-3-030-33391-1_28

3. Cremers, D., Sochen, N., Schnör, C.: Towards recognition-based variational segmentation using shape priors and dynamic labeling. In: Griffin, L.D., Lilhollm, M. (eds.) Scale-Space 2003. LNCS, vol. 2695, pp. 388–400. Springer, Heidelberg (2003). https://doi.org/10.1007/3-540-44935-3_27

4. Dangi, S., Linte, C.A., Yaniv, Z.: A distance map regularized CNN for cardiac cine MR image segmentation. Med. Phys. 46(12), 5637–5651 (2019)

5. Girdhar, R., Fouhey, D.F., Rodriguez, M., Gupta, A.: Learning a predictable and generative vector representation for objects. In: Leibe, B., Matas, J., Sebe, N., Welling, M. (eds.) ECCV 2016. LNCS, vol. 9910, pp. 484–499. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-46466-4_29

6. Karam, C., Sugimoto, K., Hirakawa, K.: Fast convolutional distance transform. IEEE Signal Process. Lett. 26(6), 853–857 (2019)

7. Ma, J., et al.: How distance transform maps boost segmentation CNNs: an empirical study. In: Medical Imaging with Deep Learning (2020)

8. Navarro, F., et al.: Shape-aware complementary-task learning for multi-organ segmentation. In: Suk, H.-I., Liu, M., Yan, P., Lian, C. (eds.) MLMI 2019. LNCS, vol. 11861, pp. 620–627. Springer, Cham (2019). https://doi.org/10.1007/978-3-030-32692-0_78

9. Oktay, O., Ferrante, E., et al.: Anatomically Constrained Neural Networks (ACNNs): application to cardiac image enhancement and segmentation. IEEE Trans. Med. Imaging 37(2), 384–395 (2018)

10. Paragios, N., Rousson, M., Ramesh, V.: Matching distance functions: a shape-to-area variational approach for global-to-local registration. In: Heyden, A., Sparr, G., Nielsen, M., Johansen, P. (eds.) ECCV 2002. LNCS, vol. 2351, pp. 775–789. Springer, Heidelberg (2002). https://doi.org/10.1007/3-540-47967-8_52

11. Pham, D.D., Dovletov, G., Warwas, S., Landgraebber, S., Jäger, M., Pauli, J.: Deep learning with anatomical priors: imitating enhanced autoencoders in latent space for improved pelvic bone segmentation in MRI. In: 2019 IEEE 16th International Symposium on Biomedical Imaging (ISBI 2019), pp. 1166–1169. IEEE (2019)

12. Ronneberger, O., Fischer, P., Brox, T.: U-net: convolutional networks for biomedical image segmentation. In: Navab, N., Hornegger, J., Wells, W.M., Frangi, A.F. (eds.) MICCAI 2015. LNCS, vol. 9351, pp. 234–241. Springer, Cham (2015). https://doi.org/10.1007/978-3-319-24574-4_28

13. Rousson, M., Paragios, N.: Shape priors for level set representations. In: Heyden, A., Sparr, G., Nielsen, M., Johansen, P. (eds.) ECCV 2002. LNCS, vol. 2351, pp. 78–92. Springer, Heidelberg (2002). https://doi.org/10.1007/3-540-47967-8_6

14. Trullo, R., Petitjean, C., Dubray, B., Ruan, S.: Multiorgan segmentation using distance-aware adversarial networks. J. Med. Imaging 6(1), 014001 (2019)