



AlignShift: Bridging the Gap of Imaging Thickness in 3D Anisotropic Volumes

Jiancheng Yang^{1,2,3}, Yi He³, Xiaoyang Huang^{1,2}, Jingwei Xu^{1,2}, Xiaodan Ye⁴,
Guangyu Tao⁴, and Bingbing Ni^{1,2,5}(✉)

¹ Shanghai Jiao Tong University, Shanghai, China
nibingbing@sjtu.edu.cn

² MoE Key Lab of Artificial Intelligence, AI Institute, Shanghai Jiao Tong University, Shanghai, China

³ Diance Technology, Shanghai, China

⁴ Shanghai Chest Hospital, Shanghai Jiao Tong University School of Medicine, Shanghai, China

⁵ Huawei Hisilicon, Shanghai, China

Abstract. This paper addresses a fundamental challenge in 3D medical image processing: how to deal with imaging thickness. For anisotropic medical volumes, there is a significant performance gap between thin-slice (mostly 1 mm) and thick-slice (mostly 5 mm) volumes. Prior arts tend to use 3D approaches for the thin-slice and 2D approaches for the thick-slice, respectively. We aim at a unified approach for both thin- and thick-slice medical volumes. Inspired by recent advances in video analysis, we propose *AlignShift*, a novel parameter-free operator to convert theoretically any 2D pretrained network into thickness-aware 3D network. Remarkably, the converted networks behave like 3D for the thin-slice, nevertheless degenerate to 2D for the thick-slice adaptively. The unified thickness-aware representation learning is achieved by shifting and fusing aligned “virtual slices” as per the input imaging thickness. Extensive experiments on public large-scale DeepLesion benchmark, consisting of 32 K lesions for universal lesion detection, validate the effectiveness of our method, which outperforms previous *state of the art* by considerable margins without whistles and bells. More importantly, to our knowledge, this is the first method that bridges the performance gap between thin- and thick-slice volumes by a unified framework. To improve research reproducibility, our code in PyTorch is open source at <https://github.com/M3DV/AlignShift>.

Keywords: Imaging thickness · Anisotropy · DeepLesion.

1 Introduction

Deep learning has been dominating medical image analysis research in a wide range of tasks (*e.g.*, classification [6,33,34], segmentation [11,21], detection

J. Yang and Y. He—These authors have contributed equally.

© Springer Nature Switzerland AG 2020

A. L. Martel et al. (Eds.): MICCAI 2020, LNCS 12264, pp. 562–572, 2020.

https://doi.org/10.1007/978-3-030-59719-1_55

[22, 29], registration [1, 4]). However, deployment of the medical image AI systems is still challenging due to numerous difficulties, *e.g.*, open set scenarios [19], calibration and uncertainty quantification [7, 10] in real-world distribution, label ambiguity in clinical annotations [12, 30]. In this study, we focus on a fundamental issue in 3D medical image analysis: how to deal with the imaging thickness, which denotes the physical distance between axial slices. In practice, there exist both thin-slice (mostly 1 mm) and thick-slice (mostly 5 mm) for a same task, *e.g.*, lesion detection [28], organ and tumor segmentation [20]. Standard procedure treats this issue as pre-processing; spatial normalization is commonly applied to normalize the dataset into a same reference thickness (*e.g.*, 2 mm). However, the spatial normalization may amplify unwanted noises in medical images [5]. Figure 1 depict spatially normalized thin- and thick-slice computed tomography (CT) scans of a same subject. As illustrated, spatial normalization introduces significant artifacts to the thick-slice (note the sagittal and coronal views). If the spatially normalized thin- and thick-slice volumes are processed by a same CNN with standard convolutions, it will lead to domain shift. We conjecture that it is the reason why 3D approaches are preferred for thin-slice volumes, while 2D approaches tend to be superior for thick-slice/anisotropic volumes [11]. Spatial normalization for thick-slice data leads to larger information loss compared to that for thin-slice data. For this reason, we challenge the spatial normalization as a standard pre-processing procedure for 3D medical image processing.

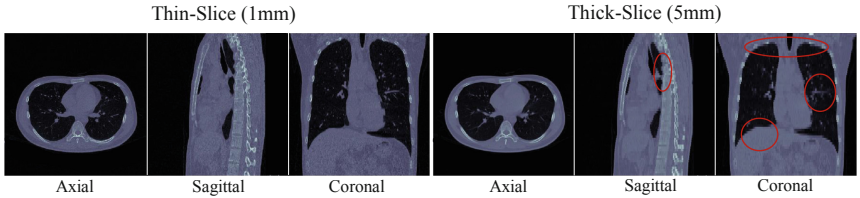


Fig. 1. Illustration of spatially normalized thin- and thick-slice computed tomography (CT) scans of a same subject. Data is from our custom dataset. Note that even though there is no significant difference between the axial slices, artifacts (the highlights on the plots) are not neglectable for the sagittal and coronal views in the thick-slice data, which results in domain shift for 3D approaches.

To address the thickness issue, we propose a novel parameter-free operator, *AlignShift*, to convert theoretically any 2D pretrained network into thickness-aware 3D network. The proposed *AlignShift* operator is inspired by Temporal Shift Module (TSM) in video analysis [14], which enables temporal (or 3D) information fusion by shifting adjacent slices (details in Sect. 2.1). Notably, TSM enables 2D-to-3D transfer learning, *i.e.*, pretrained 3D networks on 2D datasets, which is also highly related to previous studies [16, 31]. Although superior to 2.5D approaches [13, 27, 35], TSM does not bridge the performance gap between thin- and thick-slice volumes (see Sect. 3.3). As a comparison, the *AlignShift* operator

shifts and fuses aligned “virtual slices” as per the input imaging thickness, which results in unified thickness-aware representation learning (details in Sect. 2.2). Remarkably, the *AlignShift*-converted networks adaptively behave like 3D for the thin-slice, and like 2D for the thick-slice.

We validate the effectiveness of the proposed method on large-scale DeepLesion benchmark [28], a universal lesion detection dataset with 3D inputs and key-slice annotations of 32 K lesions. Without whistles and bells, the proposed methods outperform previous *state of the art* [13, 27, 35] by considerable margins. More importantly, our method closes the performance gap between thin- and thick-slice volumes compared to both 2.5D and TSM approaches; to our knowledge, we are the first to achieve this by a unified framework.

2 Methods

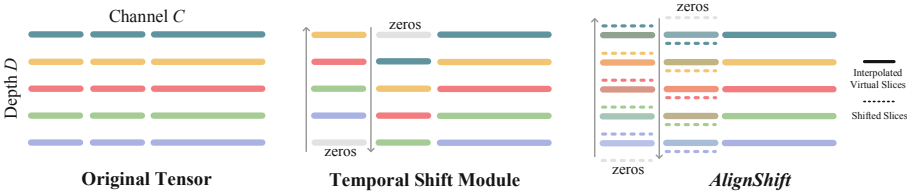


Fig. 2. Illustration of Temporal Shift Module (TSM) [14] and the proposed *AlignShift*. **Left:** A 3D tensor ($C \times D \times H \times W$) with channel C , depth D , height H , width W , H and W are not depicted for simplicity. **Middle:** Temporal Shift Module (TSM). The channels are split into three parts for shifting up, shifting down and keeping original. Border slices are padded with zeros. **Right:** *AlignShift*. Instead of shifting physical slices in TSM, we shift “virtual” slices (solid lines in different colors) in *AlignShift*. The virtual slices are interpolated from shifted slices (dash lines) by a reference thickness r .

2.1 Preliminary: Temporal Shift Module (TSM)

Prior arts in 3D images processing utilize pure 2D network to leverage the 2D pretrained weights, while 3D randomly-initialized network is necessarily adopted to fuse the feature into 3D representations. 2.5D representation, *i.e.*, several slices as channels for 2D networks, is insufficient to capture 3D contexts. It is thus meaningful to directly convert a 2D pretrained network into a 3D counterpart. To this end, we introduce Temporal Shift Module [14] (TSM) from the field of video understanding, which enables 2D-to-3D network conversion. To our knowledge, this paper is the first to introduce TSM into medical images with proven effectiveness on DeepLesion [28] benchmark. TSM tries to leverage data shift operation [25] to capture 3D semantics under 2D CNN framework, in which data shift indicates shifting data along a dimension by a certain number of slices.

Algorithm 1: In-place *AlignShift* for 3D volumes (zero padding)

Input: input 3D feature $\mathbf{X} \in \mathbb{R}^{C \times D \times H \times W}$, input actual thickness $s \in \mathbb{R}^+$, shift channels C^+, C^- , ($C^+ + C^- < C$), reference thickness $r \in \mathbb{R}^+$.

Output: $\mathbf{X} \in \mathbb{R}^{C \times D \times H \times W}$ (in-place assignment).

- 1 Compute align factor: $\alpha = r/s$;
- 2 Shift up: $\mathbf{X}^+ = [\mathbf{X}[:, C^+], \mathbf{0}] \in \mathbb{R}^{C^+ \times (D+1) \times H \times W}$;
- 3 Obtain virtual slices: $\mathbf{X}[:, C^+] = \alpha \cdot \mathbf{X}^+[:, 1:] + (1 - \alpha) \cdot \mathbf{X}^+[:, :-1]$;
- 4 Shift down: $\mathbf{X}^- = [\mathbf{0}, \mathbf{X}[C^+ : (C^+ + C^-)]] \in \mathbb{R}^{C^- \times (D+1) \times H \times W}$;
- 5 Obtain virtual slices: $\mathbf{X}[C^+ : (C^+ + C^-)] = \alpha \cdot \mathbf{X}^-[:, :-1] + (1 - \alpha) \cdot \mathbf{X}^-[:, 1:]$.

In practice, it is inserted as an operator before a $1 \times K \times K$ 3D convolution with kernel size K . Given a 3D tensor ($C \times D \times H \times W$) with channel C , depth D , height H , width W , TSM shifts the slices in the depth dimension by $+1$ slice in one part of the channels, and by -1 slice in another part of the channels, while the rest part of channels remain static (see Fig. 2 Middle). Information between slices is fused by channel in this way. To some extent, TSM imitates 3D approaches by slices shifting. It is capable of processing 3D data in an efficient way. However, for medical images, TSM faces the issue of various imaging thickness, which is widespread in many medical image datasets such as DeepLesion [28]. TSM itself does not deal with this issue. Hence all the volumes are supposed to be normalized to the same thickness via interpolation before fed into a TSM model, no matter how large the volume imaging thickness is. To those volumes with thick-slice, the interpolation process could cause large distortion of information, as the highlighted artifacts showed in Fig. 1. According to the extensive experiments in Sect. 3, TSM has good performance over thin-slice volumes, while it declines enormously for thick-slice volumes.

2.2 *AlignShift*

We believe that spatial normalization by interpolation induces the performance gap between thin- and thick-slices volumes (see results in Table 3). Domain shift takes place when normalizing thick-slices volumes, which damages the performance. To address the thickness issue, we introduce virtual slices and propose *AlignShift* which enables adaptive data shift operation based on the given imaging thickness. *AlignShift* avoids spatial normalization to thick-slice volumes by treating thin- and thick-slice volumes separately. Without loss of generality, we define thick volumes as volumes that have a thickness larger than a reference thickness r , and vice versa. For thin-slice volumes, we normalize the thickness to the reference thickness r by interpolation as usual. For thick-slice volumes which could be easily skewed by interpolation, their original thicknesses s are kept. Given a 3D feature tensor \mathbf{X} of shape $C \times D \times H \times W$ with channel C , depth D , height H , width W and physical thickness s on the depth dimension, *AlignShift* shifts part of channels up (denoted as C_+), and another part of channels down (denoted as C_-) along the depth dimension, while the rest channels

remain static. In order to maintain a consistent “receptive field” in physical sense along the depth dimension, it shifts the data by a continuous step, whose step size (align factor) $\alpha = r/s$ depends on the reference thickness r and the volume’s actual thickness s . As illustrated in Fig. 2, the shifted slice, called “virtual slice”, is obtained by interpolation between the adjacent two slices. See Algorithm 1 for the mathematical formulation. Compared to our method, TSM discretely shifts the data by one full slice. The data shift strategy of TSM results in an inconsistent “receptive field” along the depth dimension in convolution, given the non-unified thickness. Thereby the data shift operation of TSM is so-called “unaligned” under this situation. Contrarily, our method aligns the 3D features of various thickness, allowing the network to learn thickness-aware representations with the same kernels. It is guaranteed by *AlignShift* that the physical distance between shifted and un-shifted channels is always consistent among volumes with different thickness. *AlignShift* bridges the performance gap between thin-slice and thick-slice CTs theoretically and empirically.

In practice, *AlignShift* is simple to use and implement. Similar to TSM, it is inserted as an operator before a $1 \times K \times K$ 3D convolution. No additional spatial normalization is needed. The original thickness is sent to the network to allow adaptive data shift operation. Compared to TSM, only modest modification is needed to gain a great performance boost. Note that *AlignShift* is able to capture 3D semantics like TSM, while it could degenerate to 2D for data with extremely large thickness, as a result of the align factor close to zero.

AlignShift serves as a parameter-free operator enabling to convert theoretically any 2D pretrained network into thickness-aware 3D network. The conversion process is straight-forward. Table 1 lists how main operators in 2D CNNs are converted to the counterpart in 3D CNNs.

Table 1. Convert a pretrained 2D backbone into 3D. We use DenseNet-121 [9]; *AlignShift* is only applied in the dense blocks. K denotes the kernel size.

| | | | |
|-------------|--|------------------------------|--------|
| 2D Backbone | Conv2D $K \times K$ | Pool2D $K \times K$ | Norm2D |
| 3D Backbone | (<i>AlignShift</i> +)Conv3D $1 \times K \times K$ | Pool3D $1 \times K \times K$ | Norm3D |

2.3 3D Network for Universal Lesion Detection on Key Slices

We experiment with the proposed method on DeepLesion benchmark [28], which is a large-scale dataset on universal lesion detection. The inputs are 3D slices whereas only 2D key-slice annotations are available. We develop a 3D network with 2D detection heads, based on Mask RCNN [8]. As illustrated in Fig. 3, the 3D backbone is converted from the truncated DenseNet-121 [9] with *AlignShift*, serving as a 3D feature encoder. All the $K \times K$ 2D convolutions in the dense blocks are converted to *AlignShift*+ $1 \times K \times K$ 3D convolutions. The encoder takes a grey-scale 3D tensor of $1 \times D \times 512 \times 512$ as input, where D is the length of

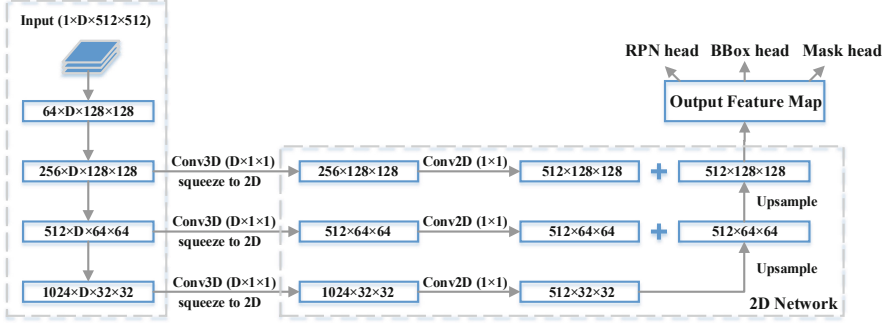
3D Backbone Converted from
Truncated DenseNet-121


Fig. 3. The 3D network for universal lesion detection on DeepLesion [28]. The 3D backbone converted by a truncated DenseNet-121 [9, 27] processes a grey-scale 3D input of $D \times 512 \times 512$, where D is the length of slices ($D = 3, 7$ in this study). 2D features of key slices are extracted, and then upsampled in a feature pyramid [15]. Detection is based on instance segmentation by Mask-RCNN [8]. Weak segmentation “ground truth” are generated from weak RECIST labels [35].

key slices ($D = 3, 7$ in this study), and extracts 3D features through three dense blocks. Each dense block increases the feature channels, downsamples the feature in the height and width dimension while maintaining the scale in the depth dimension. The feature output of each dense blocks is processed by a $D \times 1 \times 1$ 3D convolution and then squeezed into a 2D shape. A 2D decoder combines these features under different resolutions and upsamples the features step by step. The final feature map is fed into RPN head, BBox head and Mask head for detection and instance segmentation supervised by weak RECIST labels.

We implement the Mask-RCNN with PyTorch [17] and MMDetection [3]. As counterparts of the proposed *AlignShift*, we also implement 1) 2.5D Mask-RCNN, which stacks the slices as input channels for standard 2D networks, and 2) TSM-converted Mask-RCNN, which uses TSM instead of *AlignShift*. Note that the model sizes of the TSM models are strictly the same as the *AlignShift* models, and the 2.5 models are slightly smaller because of the Conv3D ($D \times 1 \times 1$) layer.

3 Experiments

3.1 Dataset and Experiment Setting

DeepLesion dataset [28] consists of 32,120 axial CT slices from 10,594 studies of unique patients. the thickness of the dataset is almost 1 mm (48.66%) and 5 mm (50.26%), which is appropriate to develop and evaluate the proposed method. There are 1 to 3 lesions in each slice, with totally 32,735 lesions from several organs, whose sizes vary from 0.21 to 342.5 mm. RECIST diameter (critical for tumor staging and prognosis assessment [32]) coordinates and bounding boxes

were labeled on the key slices, with adjacent slices (above and below 30 mm) provided as contextual information. We use GrabCut [18] to generate weak segmentation “ground truth” from weak RECIST labels [2, 35]. Hounsfield units of the input are clipped into $[-1024, 2050]$ and normalized. In *AlignShift* experiments, $r = 2$ mm is regarded as the reference thickness; we normalize the thickness into 2 mm for thin-slice data with thickness ≤ 2 mm; for thick-slice data with thickness > 2 mm, we keep the original thickness, since the spatial normalization for thick-slice data leads to larger information loss compared to that for thin-slice data. For 2.5D and TSM counterparts, we adapt standard strategy to normalize the thickness of all images into 2 mm. Data augmentation including horizontal flip, shift, rescaling and rotation is applied during training stage, no test-time augmentation (TTA) is applied. We resize each input slice to 512×512 before feeding into the networks. We use official data split (training/validation/test: 70%/15%/15%); following prior studies [13, 27, 35], sensitivity at various false positives levels (*i.e.*, FROC analysis) is evaluated on the test set.

Table 2. Sensitivity (%) at various false positives (FPs) per image of previous *state-of-the-art* and the proposed methods, on the large-scale DeepLesion benchmark [28]. Note that MULAN [27] uses extra tag supervision and an addition Score Refinement Layer (SRL) with tag inputs; We report the performance of MULAN under 171-tag supervision as well as that without SRL.

| Methods | Slices | 0.5 | 1 | 2 | 4 | 8 | 16 | Avg. [0.5, 1, 2, 4] |
|------------------------|-------------|--------------|--------------|--------------|--------------|--------------|--------------|---------------------|
| 3DCE [26] MICCAI’18 | $\times 27$ | 62.48 | 73.37 | 80.70 | 85.65 | 89.09 | 91.06 | 75.55 |
| ULDor [23] ISBI’19 | $\times 1$ | 52.86 | 64.8 | 74.84 | 84.38 | 87.17 | 91.8 | 69.22 |
| V.Attn [24] MICCAI’19 | $\times 3$ | 69.10 | 77.90 | 83.80 | — | — | — | — |
| Retina. [35] MICCAI’19 | $\times 3$ | 72.15 | 80.07 | 86.40 | 90.77 | 94.09 | 96.32 | 82.35 |
| MVP [13] MICCAI’19 | $\times 3$ | 70.01 | 78.77 | 84.71 | 89.03 | — | — | 80.63 |
| MVP [13] MICCAI’19 | $\times 9$ | 73.83 | 81.82 | 87.60 | 91.30 | — | — | 83.64 |
| MULAN [27] MICCAI’19 | $\times 9$ | 76.12 | 83.69 | 88.76 | 92.30 | 94.71 | 95.64 | 85.22 |
| w/o SRL [27] MICCAI’19 | $\times 9$ | — | — | — | — | — | — | 84.22 |
| Ours 2.5D | $\times 3$ | 71.27 | 79.82 | 86.30 | 90.61 | 93.75 | 95.70 | 82.00 |
| Ours 2.5D | $\times 7$ | 72.66 | 81.45 | 87.07 | 90.98 | 93.40 | 95.30 | 83.04 |
| Ours TSM | $\times 3$ | 70.24 | 79.52 | 86.28 | 90.90 | 94.06 | 96.09 | 81.73 |
| Ours TSM | $\times 7$ | 75.98 | 83.65 | 88.44 | 92.14 | 94.89 | 96.50 | 85.05 |
| Ours <i>AlignShift</i> | $\times 3$ | 72.90 | 80.74 | 87.15 | 91.92 | 94.85 | 96.48 | 83.18 |
| Ours <i>AlignShift</i> | $\times 7$ | 79.40 | 85.50 | 90.09 | 93.26 | 95.24 | 96.66 | 87.06 |

3.2 Performance Compared with *State of the Art*

In Table 2, we depict sensitivity at various false positives per image (FPs), which shows that our proposed methods significantly outperform the previous *state-of-the-art* MULAN [27]. Notably, it is achieved without additional information beyond the CT images such as tags from medical reports and demographic information. The performance of our 2.5D counterparts is comparable to Improved RetinaNet [35] and MVP-Net [13]. The proposed TSM-converted networks outperform these studies [13, 35], and even MULAN without tag supervision, which

validates the superiority of pretrained 3D backbones over 2.5D. *AlignShift* further boosts performance and surpasses our TSM and previous *state-of-the-art* MULAN [27]. Due to memory constraints, we report the performance of maximum 7 slices, whereas more slices are expected with better performance.

Table 3. Detection performance analysis of our methods on all, thin-slice (mostly 1 mm) and thick-slice (mostly 5 mm) data. “diff.” denotes the average sensitivity (Avg. [0.5, 1, 2, 4]) difference between thin/thick-slice data and that of all data.

| Methods | Thickness | 0.5 | 1 | 2 | 4 | 8 | 16 | Avg. [0.5, 1, 2, 4] | diff |
|------------------------------|-----------|-------|-------|-------|-------|-------|-------|---------------------|-------|
| 2.5D $\times 3$ | All | 71.27 | 79.82 | 86.30 | 90.61 | 93.75 | 95.70 | 82.00 | – |
| | Thin | 72.78 | 80.65 | 87.21 | 90.94 | 93.97 | 95.86 | 82.89 | +0.89 |
| | Thick | 69.88 | 79.16 | 85.51 | 90.48 | 93.65 | 95.65 | 81.26 | –0.74 |
| 2.5D $\times 7$ | All | 72.66 | 81.45 | 87.07 | 90.98 | 93.40 | 95.30 | 83.04 | – |
| | Thin | 75.77 | 83.93 | 88.85 | 92.37 | 94.26 | 95.78 | 85.23 | +2.19 |
| | Thick | 69.76 | 78.96 | 85.75 | 90.03 | 92.67 | 94.99 | 81.13 | –1.91 |
| TSM $\times 3$ | All | 70.24 | 79.52 | 86.28 | 90.90 | 94.06 | 96.09 | 81.73 | – |
| | Thin | 73.74 | 81.84 | 87.54 | 92.21 | 94.92 | 96.72 | 83.83 | +2.10 |
| | Thick | 67.03 | 77.37 | 84.98 | 89.95 | 93.16 | 95.44 | 79.83 | –1.90 |
| TSM $\times 7$ | All | 75.98 | 83.65 | 88.44 | 92.14 | 94.89 | 96.50 | 85.05 | – |
| | Thin | 78.76 | 85.53 | 89.67 | 93.48 | 95.61 | 96.68 | 86.86 | +1.81 |
| | Thick | 73.26 | 81.97 | 87.10 | 90.96 | 94.14 | 96.42 | 83.32 | –1.73 |
| <i>AlignShift</i> $\times 3$ | All | 72.90 | 80.74 | 87.15 | 91.92 | 94.85 | 96.48 | 83.18 | – |
| | Thin | 73.51 | 81.59 | 87.62 | 92.37 | 94.87 | 96.51 | 83.77 | +0.61 |
| | Thick | 72.85 | 80.18 | 87.10 | 91.94 | 94.91 | 96.54 | 83.02 | –0.16 |
| <i>AlignShift</i> $\times 7$ | All | 79.40 | 85.50 | 90.09 | 93.26 | 95.24 | 96.66 | 87.06 | – |
| | Thin | 80.73 | 86.43 | 91.02 | 93.97 | 95.78 | 96.97 | 88.04 | +0.98 |
| | Thick | 78.27 | 84.74 | 88.89 | 92.47 | 94.67 | 96.38 | 86.09 | –0.97 |

3.3 Performance Analysis on Thin-Slice and Thick-Slice Data

To demonstrate the benefits of *AlignShift* on bridging the performance gap of thin- and thick-slice data, we conducted a performance analysis of our 2.5D, TSM and *AlignShift* models on the thin- and thick-slice data separately. As there is no open trained models available for prior *state of the art* [13, 27, 35], we believe our 2.5D models represent the performance of these studies, considering these studies follow 2.5D fashion. As illustrated in Table 3, there is a significant performance gap between thin- and thick-slice CTs for 2.5D and TSM approaches, which validates our argument (Sect. 1) that there is a significant domain shift if the thin- and thick-slice data are processed by a same CNN with standard convolutions. Note that the gap becomes larger when more slices are used, since the 3D information is less valuable for thick-slice data than the thin-slice. In comparison, by learning unified thickness-aware representation, the proposed *AlignShift* reduces the performance gap to neglectable for 3-slice setting; even for the 7-slice setting, the gap is closed compared to 2.5D and TSM approaches.

4 Conclusion

In this study, we challenge spatial normalization as a standard pre-processing approach to solve thickness issue in 3D medical images. Our experiment results indicate that both 2.5D and 3D (*i.e.*, TSM in this study) approaches do not fundamentally address the domain shift issue introduced by spatial normalization, which results in a significant performance gap between thin- and thick-slice data. In this regard, we propose a novel parameter-free operator *Align-Shift*, which enables us to convert theoretically any 2D pretrained network into thickness-aware 3D network. Extensive experiments on DeepLesion benchmark empirically validate that our methods bridge the gap between thin- and thick-slice data. Without whistles and bells, we establish a new *state of the art* on DeepLesion, which surpasses prior arts by considerable margins.

Acknowledgment. This work was supported by National Science Foundation of China (61976137, U1611461). This work was also supported by Interdisciplinary Program of Shanghai Jiao Tong University (YG2017QN661). Authors appreciate the Student Innovation Center of SJTU for providing GPUs.

References

1. Balakrishnan, G., Zhao, A., Sabuncu, M.R., Guttag, J., Dalca, A.V.: VoxelMorph: a learning framework for deformable medical image registration. *IEEE Trans. Med. Imaging* **38**, 1788–1800 (2019)
2. Cai, J., et al.: Accurate weakly-supervised deep lesion segmentation using large-scale clinical annotations: slice-propagated 3D mask generation from 2D RECIST. In: Frangi, A.F., Schnabel, J.A., Davatzikos, C., Alberola-López, C., Fichtinger, G. (eds.) MICCAI 2018. LNCS, vol. 11073, pp. 396–404. Springer, Cham (2018). https://doi.org/10.1007/978-3-030-00937-3_46
3. Chen, K., Wang, J., Pang, J., Cao, Y., et al.: MMDetection: open MMLab detection toolbox and benchmark. arXiv preprint [arXiv:1906.07155](https://arxiv.org/abs/1906.07155) (2019)
4. Dalca, A.V., Balakrishnan, G., Guttag, J., Sabuncu, M.R.: Unsupervised learning for fast probabilistic diffeomorphic registration. In: Frangi, A., Schnabel, J., Davatzikos, C., Alberola-López, C., Fichtinger, G. (eds.) MICCAI 2018. LNCS, vol. 11070, pp. 729–738. Springer, Heidelberg (2018). https://doi.org/10.1007/978-3-030-00928-1_82
5. Glocker, B., Robinson, R., de Castro, D.C., Dou, Q., Konukoglu, E.: Machine learning with multi-site imaging data: an empirical study on the impact of scanner effects. In: Medical Imaging meets NeurIPS Workshop (2019)
6. Gulshan, V., et al.: Development and validation of a deep learning algorithm for detection of diabetic retinopathy in retinal fundus photographs. *JAMA* **316**(22), 2402–2410 (2016)
7. Guo, C., Pleiss, G., Sun, Y., Weinberger, K.Q.: On calibration of modern neural networks. In: ICML (2017)
8. He, K., Gkioxari, G., Dollár, P., Girshick, R.B.: Mask R-CNN. In: ICCV, pp. 2980–2988 (2017)
9. Huang, G., Liu, Z., Van Der Maaten, L., Weinberger, K.Q.: Densely connected convolutional networks. In: CVPR, vol. 1, p. 3 (2017)

10. Huang, X., Yang, J., et al.: Evaluating and boosting uncertainty quantification in classification. arXiv preprint [arXiv:1909.06030](https://arxiv.org/abs/1909.06030) (2019)
11. Isensee, F., et al.: NNU-net: self-adapting framework for U-net-based medical image segmentation. arXiv preprint [arXiv:1809.10486](https://arxiv.org/abs/1809.10486) (2018)
12. Kohl, S., et al.: A probabilistic u-net for segmentation of ambiguous images. In: NIPS, pp. 6965–6975 (2018)
13. Li, Z., Zhang, S., Zhang, J., Huang, K., Wang, Y., Yu, Y.: Mvp-net: Multi-view fpn with position-aware attention for deep universal lesion detection. In: Shen, D., et al. (eds.) MICCAI 2019. LNCS, vol. 11769, pp. 13–21. Springer, Heidelberg (2019). https://doi.org/10.1007/978-3-030-32226-7_2
14. Lin, J., Gan, C., Han, S.: TSM: temporal shift module for efficient video understanding. In: ICCV, pp. 7082–7092 (2019)
15. Lin, T.Y., Dollár, P., Girshick, R.B., He, K., Hariharan, B., Belongie, S.J.: Feature pyramid networks for object detection. In: CVPR, pp. 936–944 (2016)
16. Liu, S., et al.: 3D anisotropic hybrid network: transferring convolutional features from 2D images to 3D anisotropic volumeSs. In: Frangi, A.F., Schnabel, J.A., Davatzikos, C., Alberola-López, C., Fichtinger, G. (eds.) MICCAI 2018. LNCS, vol. 11071, pp. 851–858. Springer, Cham (2018). https://doi.org/10.1007/978-3-030-00934-2_94
17. Paszke, A., Gross, S., Chintala, S., Chanan, G., Yang, E., et al.: Automatic differentiation in pytorch (2017)
18. Rother, C., Kolmogorov, V., Blake, A.: “grabcut” interactive foreground extraction using iterated graph cuts. ACM TOG **23**(3), 309–314 (2004)
19. Scheirer, W.J., Rocha, A., Sapkota, A., Boult, T.E.: Toward open set recognition. IEEE T-PAMI **35**, 1757–1772 (2013)
20. Simpson, A.L., et al.: A large annotated medical image dataset for the development and evaluation of segmentation algorithms. arXiv preprint [arXiv:1902.09063](https://arxiv.org/abs/1902.09063) (2019)
21. Tang, H., et al.: Clinically applicable deep learning framework for organs at risk delineation in CT images. Nat. Mach. Intell. 1–12 (2019)
22. Tang, H., Zhang, C., Xie, X.: Nodulenet: Decoupled false positive reduction for pulmonary nodule detection and segmentation. In: Shen, D., et al. (eds.) MICCAI 2019. LNCS, vol. 11769, pp. 266–274. Springer, Cham (2019). https://doi.org/10.1007/978-3-030-32226-7_30
23. Tang, Y.B., Yan, K., Tang, Y.X., Liu, J., Xiao, J., Summers, R.M.: ULDor: a universal lesion detector for CT scans with pseudo masks and hard negative example mining. In: ISBI, pp. 833–836. IEEE (2019)
24. Wang, X., Han, S., Chen, Y., Gao, D., Vasconcelos, N.: Volumetric attention for 3D medical image segmentation and detection. In: Shen, D., et al. (eds.) MICCAI 2019. LNCS, vol. 11769, pp. 175–184. Springer, Cham (2019). https://doi.org/10.1007/978-3-030-32226-7_20
25. Wu, B., et al.: SHIFT: a zero flop, zero parameter alternative to spatial convolutions. In: CVPR, pp. 9127–9135 (2017)
26. Yan, K., Bagheri, M., Summers, R.M.: 3D context enhanced region-based convolutional neural network for end-to-end lesion detection. In: Frangi, A.F., Schnabel, J.A., Davatzikos, C., Alberola-López, C., Fichtinger, G. (eds.) MICCAI 2018. LNCS, vol. 11070, pp. 511–519. Springer, Cham (2018). https://doi.org/10.1007/978-3-030-00928-1_58
27. Yan, K., et al.: Mulan: Multitask universal lesion analysis network for joint lesion detection, tagging, and segmentation. In: Shen, D., et al. (eds.) MICCAI 2019. LNCS, vol. 11769, pp. 194–202. Springer, Heidelberg (2019). https://doi.org/10.1007/978-3-030-32226-7_22

28. Yan, K., et al.: Deep lesion graphs in the wild: relationship learning and organization of significant radiology image findings in a diverse large-scale lesion database. In: CVPR, pp. 9261–9270 (2018)
29. Yang, J., Deng, H., Huang, X., Ni, B., Xu, Y.: Relational learning between multiple pulmonary nodules via deep set attention transformers. In: ISBI (2020)
30. Yang, J., Fang, R., Ni, B., et al.: Probabilistic radiomics: ambiguous diagnosis with controllable shape analysis. In: Shen, D., et al. (eds.) MICCAI 2019. LNCS, vol. 11769, pp. 658–666. Springer, Cham (2019). https://doi.org/10.1007/978-3-030-32226-7_73
31. Yang, J., Huang, X., Ni, B., Xu, J., Yang, C., Xu, G.: Reinventing 2D convolutions for 3D medical images. arXiv preprint [arXiv:1911.10477](https://arxiv.org/abs/1911.10477) (2019)
32. Yang, Y., Yang, J., Ye, Y., Xia, T., Lu, S.: Development and validation of a deep learning model to assess tumor progression to immunotherapy. In: ASCO, vol. 37, pp. e20601–e20601 (2019)
33. Zhao, W., et al.: Toward automatic prediction of EGFR mutation status in pulmonary adenocarcinoma with 3D deep learning. *Cancer Med.* **8**, 3532–3543 (2019)
34. Zhao, W., et al.: 3D deep learning from CT scans predicts tumor invasiveness of subcentimeter pulmonary adenocarcinomas. *Cancer Res.* **78**(24), 6881–6889 (2018)
35. Zlocha, M., Dou, Q., Glocker, B.: Improving retinanet for CT lesion detection with dense masks from weak recist labels. In: Shen, D., et al. (eds.) MICCAI 2019. LNCS, vol. 11769, pp. 402–410. Springer, Cham (2019). https://doi.org/10.1007/978-3-030-32226-7_45