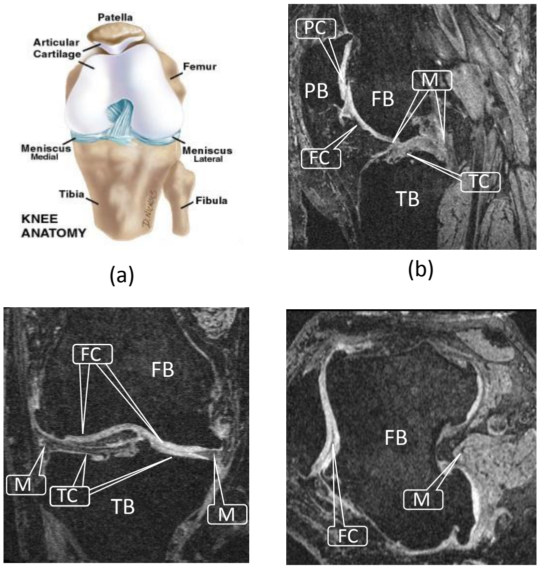
KCB-Net: A 3D Knee Cartilage and Bone Segmentation Network via Sparse Annotation

Yaopeng Peng, Hao Zheng, Fahim Zaman, Lichun Zhang, Xiaodong Wu, *Senior Member, IEEE*, Milan Sonka, *Fellow, IEEE*, Danny Z. Chen, *Fellow, IEEE*

***Abstract*— Knee cartilage and bone segmentation is critical for physicians to analyze and diagnose articular damage and knee osteoarthritis (OA). Deep learning (DL) methods for medical image segmentation have largely outperformed traditional methods, but they often need large amounts of annotated data for model training, which is very costly and time-consuming for medical experts, especially on 3D images. In this paper, we report a new knee cartilage and bone segmentation framework, KCB-Net, for 3D MR images based on sparse annotation. KCB-Net selects a small subset of slices from 3D images for annotation, and seeks to bridge the performance gap between sparse annotation and full annotation. Specifically, it first identifies a subset of the most effective and representative slices with an unsupervised scheme; it then trains an ensemble model using the annotated slices; next, it self-trains the model using 3D images containing pseudo-labels generated by the ensemble method and improved by a bi-directional hierarchical earth mover's distance (bi-HEMD) algorithm; finally, it fine-tunes the segmentation results using the primal-dual Internal Point Method (IPM). Experiments on two 3D MR knee joint datasets (the Iowa dataset and iMorphics dataset) show that our new framework outperforms state-of-the-art methods on full annotation, and yields high quality results even for annotation ratios as low as 5%.**

***Index Terms*— Knee cartilage and bone segmentation; Sparse annotation; Ensemble learning; 3D MR images.**



(a) (b)

(c) (d)

Fig. 1: Knee joint. (a) Anatomy of the knee joint (adopted from [1]). (b)-(d) Sagittal, coronal, and transverse MR image planes, showing the femur bone (FB), femoral cartilage (FC), tibia bone (TB), tibial cartilage (TC), patella bone (PB), patellar cartilage (PC), and meniscus (M).

I. INTRODUCTION

Osteoarthritis (OA) is a prevalent chronic disease caused by the damage and degeneration of cartilages. It is estimated that 20% of Americans may suffer from various levels of OA by 2030. Magnetic resonance imaging (MRI) has become a common technique for studying and assessing changes within the knee joint, including cartilages and bones. Fig. 1 illustrates the anatomical structure of the knee joint.

Considering the knee joint anatomy, the femoral cartilage (FC), tibial cartilage (TC), patellar cartilage (PC), and menisci (M) are the main tissues affecting the knee joint health. To quantitatively measure the thickness of the knee cartilages and identify the bone-cartilage interface, accurate cartilage and bone segmentation is needed.

To capture the detailed structure of the knee anatomy, 3D MR images are commonly scanned at high in-plane resolution. However, labeling 3D MR images is very time-consuming.

The best current methods for knee-joint segmentation, as to be discussed in Section II, depend on large-sized training data to learn segmentation parameters. But, forming large enough annotated datasets is difficult in medical image analysis. In this paper, we propose a new framework, KCB-Net, for 3D cartilage and bone segmentation with sparse annotation, and demonstrate its performance on a knee-joint segmentation task.

II. RELATED WORK

Automated and semi-automated methods for knee joint segmentation have been investigated for several decades. Shape models, graph optimization approaches, and deep learning (DL) methods exhibited high performance in recent years.

Yaopeng Peng, Hao Zheng, and Danny Z. Chen are with the Dept. of Computer Science and Engineering, University of Notre Dame, Notre Dame, IN 46556, USA (e-mails: {ypeng4, hzheng3, dchen}@nd.edu).

Fahim Zaman, Lichun Zhang, Xiaodong Wu, and Milan Sonka are with the Department of Electrical and Computer Engineering, University of Iowa, Iowa City, IA 52242, USA (e-mails: {fahim-zaman, lichun-zhang, xiaodong-wu, milan-sonka}@uiowa.edu).

This research was supported in part by NIH NIBIB Grant R01-EB004640 and NSF Grant CCF-1617735.

3D graph based methods are well suited for knee cartilage segmentation. Yin et al. [2] proposed a layered optimal graph image segmentation for multiple objects and surfaces (LOGISMOS) framework to simultaneously segment multiple interacting surfaces of objects by incorporating multiple spatial interrelationships of surfaces in a D-dimensional graph. Kashyap et al. [3] extended the LOGISMOS framework to simultaneously segment 3D knee objects for multiple follow-up visits of the same patient – effectively performing optimal 4D (3D+time) segmentation. Xie et al. [4] proposed a primal-dual Internal Point Method (IPM) to first learn the parameters of the surface cost functions for the LOGISMOS algorithm and then solve an optimization problem for the final segmentation.

Several deep convolutional neural network (CNN) approaches showed close-to-human level performance. Liu et al. [5] proposed a fully automatic musculoskeletal tissue segmentation method that integrates CNN and 3D simplex deformable approaches to improve the accuracy and efficiency. Ambellan et al. [6] combined the strengths of statistical shape models and CNN to successfully segment knee bones/cartilages. Tan et al. [7] proposed a method to first extract the regions of interest (ROIs) for three cartilage areas and then fuse the three ROIs to generate fine-grained segmentation results.

Zheng et al. [8] proposed a 3D segmentation method that ensembles three 2D models and one 3D model (called base-learners). It first trains the base-learners using labeled data, and ensembles the base-learners by training a meta-learner [9]. It then re-trains the base-learners and meta-learner with pseudo-labels to obtain a 3D segmentation model. However, such base-learners still rely on fully annotated 3D data. In [10], Zheng et al. proposed a sparse annotation strategy to select the most representative 2D slices for annotation. It first encodes each slice into a low-dimensional vector, and prioritizes the slices based on their representativeness in a set of 3D images. Next, three 2D modules and one 3D module (3D FCN [11]) are trained, and pseudo-labels of the unlabeled data are generated using the base-learners. A Y-shape DenseVoxNet [9] is used to train a meta-learner, which ensembles the 2D and 3D modules. Zheng et al. [12] further extended this sparse annotation strategy, and designed a K-head FCN to compute the pseudo-label uncertainty of each slice and rule out highly uncertain pixels in the subsequent training process.

III. METHOD

A. Overview

Our KCB-Net combines and extends previously reported ensemble learning [8] and sparse annotation [10] methods for 3D segmentation. Fig. 2 shows its main steps. (1) *Representative slice selection*: As in [10], each 2D slice in every major xy, yz, or xz orientation in the entire set W of 3D training images is encoded as a low-dimensional latent vector, and all slices are prioritized by their representativeness. The top-ranked k slices are selected as the ones, in which to perform expert annotations. (2) *Base-learner training and pseudo-label generation*: As in [8], three 2D modules, one for each xy, yz, or xz orientations, are trained on the selected and annotated slices. Once 2D modules are trained, pseudo-labels are assigned to all remaining un-annotated slices in W and a 3D module is trained. KU-Net mechanism [13] is newly used to extract multi-scale features. Each module extracts information across different scales to support fine-scale feature extraction. Instead of using a sparse 3D FCN [11] as in [10], a DenseVoxNet [9] uses labels of the expert-annotated slices and pseudo-labels of the un-annotated slices. As in [14], an edge-aware branch is added to the 3D module to increase the weights of cartilage and bone surface locations. To explore the appearance consistency among consecutive slices and further improve the quality of the pseudo-labels generated, the H-EMD method [15] is newly enhanced by incorporating a bi-directional hierarchical earth mover's distance (bi-HEMD) when generating pseudo-labels of the un-annotated slices. Our bi-HEMD method first produces object candidates by applying multiple threshold values on the probability maps, and then selects object instances by minimizing the earth mover's distance based on a reference set of the object instances. (3) *Ensembling and self-training*: Following the pseudo-label generation, 2D and 3D modules are ensembled by training a 3D Y-shape DenseVoxNet [8] as a meta-learner using the original input images and pseudo-labels, which learns the target object segmentation from the labels/pseudo-labels. The output of the ensemble model is utilized to iteratively re-train the modules in Step (2) and the ensemble model in Step (3), repeated until convergence. (4) *Post-processing*: We newly add a post-processing step exploiting the task-specific characteristics that knee bones and cartilages are anatomically adjacent with one other. A fine-tuning network [4] that incorporates the surface interrelationships between adjacent bones and cartilages is trained by taking the probability maps generated in Step (3) as input and the pseudo-labels as the learning targets. The fine-tuning network is optimized using the IPM algorithm [4].

B. Representative Slice Selection

Identifying a small-enough set of the most representative 2D slices for annotation that subsequently facilitates the segmentation method training is critical for the success of our proposed approach. This section presents our slice selection scheme, called representative annotation (RA).

Medical experts often annotate a 3D image by choosing one orthogonal plane (xy, yz, or xz) and labeling the corresponding slices one by one. It may, however, be beneficial to annotate 2D slices along each of the three orthogonal planes. Fig. 3 illustrates the slice selection method.

1) Slice Representation: For a specified annotation ratio (e.g., 10% of all slices), to select the most representative slices to label, we first need to efficiently represent the slices. Medical image slices can commonly be represented as latent feature vectors of a much smaller size compared to the original 2D image matrix. By comparing slices using their latent vectors, not only can we reduce the computation cost but also extract their most useful information.

We utilize an auto-encoder as the representation extractor for the slices in our 3D training image set W, which learns efficient features in an unsupervised manner and conducts a lossy compression in the encoding process. It learns to store relevant information and disregard noise. This auto-encoder

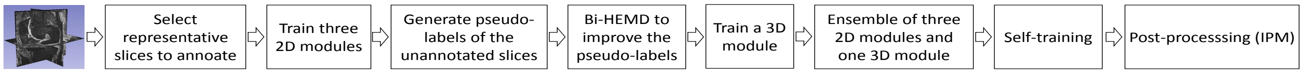

Select representative slices to annoate
Train three 2D modules
Generate pseudo-labels of the unannotated slices
Bi-HEMD to improve the pseudo-labels
Train a 3D module
Ensemble of three 2D modules and one 3D module
Self-training
Post-processsing (IPM)

Fig. 2: The pipeline of our proposed KCB-Net framework.

consists of two parts: An encoder produces a compressed knowledge representation x for an input image (or slice) I; a decoder takes the representation x as input and outputs $\hat{x}$ as a reconstruction of the original image. The entire auto-encoder model is optimized by minimizing the sum of the reconstruction error $\mathcal{L}(x, \hat{x})$, which measures the differences between the original image and the reconstruction produced, and a regularization term for alleviating overfitting. This can be formulated as:

$$\phi^*, \psi^* = \arg\min_{\phi,\psi} (\mathcal{L}(x, \hat{x}) + \lambda_1 \times \sum_{i=1}^{M} w_i^2), \qquad (1)$$

where $\mathcal{L}$ is the reconstruction loss between x and $\hat{x}$, λ_1 is a scaling parameter for the regularization term $\sum_{i=1}^{M} w_i^2$ to adjust the trade-off between the sensitivity to the input and overfitting, w_i is the i-th parameter of the auto-encoder, and ϕ and ψ are the parameters of the encoder and decoder, respectively.

To facilitate a fast training and convergence of the auto-encoder, we use a ResNet-101 [16] pre-trained on ImageNet [17] as the encoder backbone. A light-weight decoder (ResNet-50 [16]) is added to map the latent vectors to the original input space. Since slices along each orthogonal plane will be selected, we train the auto-encoder using all the slices of the 3D training set W along the three orthogonal planes.

2) Prioritizing the Slices: After training the auto-encoder, we measure the representativeness of each slice in the 3D training image set W as in [10]. First, we feed a 2D slice I to the encoder, and take the generated latent vector f as the representation of the slice I. Second, we define and compute the similarity between two slices I_i and I_j as $Sim(I_i, I_j) = cosine(f_i, f_j)$, where f_i and f_j are the latent vectors of I_i and I_j respectively, and $cosine$ denotes cosine similarity.

Next, a subset S of slices is selected from all the slices $S(W)$ of the set W (for an annotation ratio or a given size of S). The representativeness of S with respect to W is defined as:

$$F(S, W) = \sum_{I \in S(W)} \max_{I_s \in S}(Sim(I_s, I)). \qquad (2)$$

Finding an optimal slice subset S was formulated as a maximum cover problem in [10], which is NP-hard, and a polynomial time approximation solution was obtained using a greedy method. Suppose a subset S' is the most representative for the images in W. The next choice (if needed) is a slice I^* in the remaining slice set $S(W) - S'$ that maximally increases the representativeness of the new subset $S' \cup \{I^*\}$, i.e.,

$$I^* = \arg\max_{I \in (S(W) - S')} (F(S' \cup \{I\}, W) - F(S', W)). \qquad (3)$$

This selection process puts all the slices in W in decreasing order based on their representativeness. The slices with better representativeness have higher priorities for annotation.

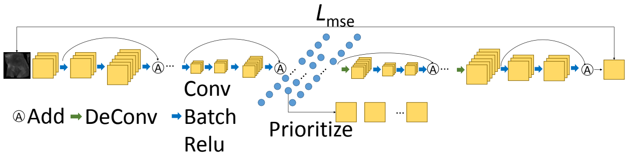


Fig. 3: Illustrating the representative slice selection method. L_{mse} denotes the mean square error.

C. Base-learner Training and Pseudo-label Generation

After the representative slice selection, the selected slices are labeled by experts, which we denote as $S_L = \{S_{l_1}, S_{l_2}, \ldots, S_{l_N}\}$, where l_N is the number of slices selected. Due to the limited training data, we apply the bottleneck structure in [18], which can achieve better performance compared to a common U-Net since it has fewer parameters and thus can alleviate overfitting.

To better exploit multi-scale features of the objects in our 3D knee images, we apply the KU-Net design in [13] to our backbone network to build a K-FCN network, which consists of K FCN submodules connected sequentially as in [13]. K-FCN first extracts information at different scales sequentially and then feeds the extracted information to the subsequent FCN submodules to assist feature extraction in finer scales. We apply the FCN structure in [18] as the backbone (with fewer parameters than U-Net). The first submodule of K-FCN is used to extract coarser-scale features, which are fed to the next submodule to extract features in a finer scale. The structure of our K-FCN is shown in Fig. 4 (with $K = 2$).

A 2D segmentation model can use a relatively large receptive field, but it does not utilize the interactions between consecutive slices well, which may result in spatial slice-to-slice inconsistency. Hence, we follow the ensemble method in [8] and train a 3D module, which produces smoother 3D results. We choose DenseVoxNet [9] as the backbone for our 3D module, since it has better parameter efficiency and thus a smaller chance to incur overfitting, especially with limited training data. Likewise, we use the KU-Net design and build a K-DenseVoxNet to exploit 3D multi-scale features. The coarse features extracted by the first DenseVoxNet submodule are fed to the second submodule to obtain fine-grained features.

For knee joint segmentation, the bone and cartilage boundaries are more important than other areas, since they usually serve as the main criteria to measure whether a cartilage is damaged. Hence, adding an edge-aware regulation can force the network to focus more on the boundary areas. Fig. 5 shows the structure of our edge-aware K-DenseVoxNet. The edge gate $F_{L\rho G}$ is defined as:

$$F_{L\rho G}(I) = k_G * \rho(k_L * I), \qquad (4)$$

where k_G and k_L represent the Gaussian smoothing kernel and Laplacian kernel respectively, $*$ denotes convolution, and

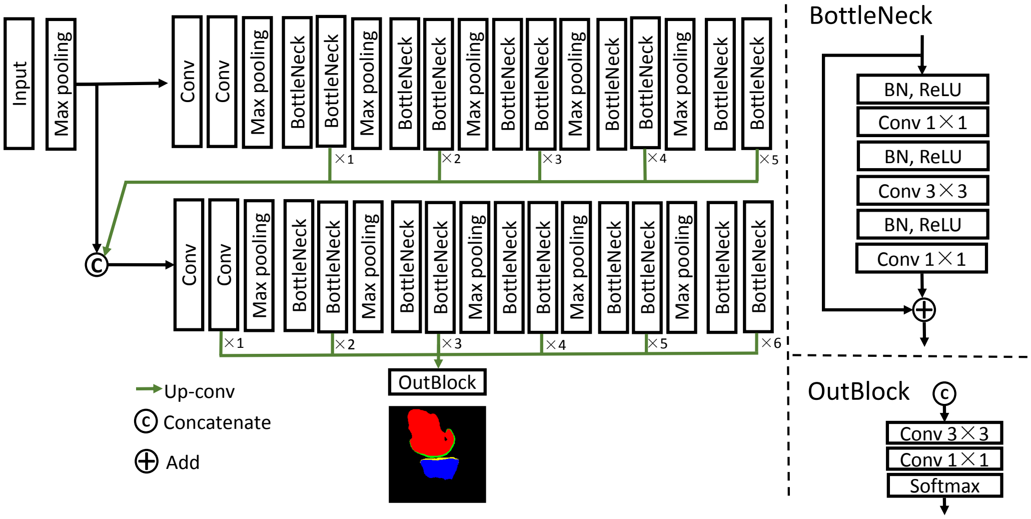


Fig. 4: The structure of our K-FCN ($K = 2$).

ρ is an activation function.

The loss function of our 3D module is defined as:

$$\mathcal{L} = L_{region} + \lambda_2 L_{edge}, \tag{5}$$

where L_{region} and L_{edge} are the cross entropy losses of the region branch and edge branch respectively, and λ_2 is a scaling parameter to regularize the edge branch.

We first train our three 2D segmentation modules using the selected labeled slices for each of the three orthogonal planes, and generate the probability maps of the unlabeled slices using the three trained 2D modules. We then train our 3D edge-aware K-DenseVoxNet using the 3D images in W that contain both the labeled slices and unlabeled slices that are now "labeled". Specifically, the pseudo-labels produced by the three 2D modules are first improved by the bi-HEMD algorithm in Section III-D. Then, the probability maps attained by the three 2D modules are averaged to generate the pseudo-labels used for training our 3D module. These four trained segmentation modules generate their pseudo-labels respectively for all the unlabeled slices. For simplicity, we average the results of these four modules as the probability map for each 3D image in W.

D. Bi-directional Hierarchical Earth Mover's Distance

After training our three 2D modules, probability maps of all the unlabeled slices in W are obtained. One observation on the 3D knee images is that the appearances of bones and cartilages between consecutive slices are often similar in size and shape. Exploring such appearance similarity can help improve the pseudo-label quality. Hence, we apply the hierarchical earth mover's distance (H-EMD) method [15] that uses many threshold values of the probability map for each unannotated slice and exploits the appearance consistency between consecutive slices to optimize the pseudo-labels.

The H-EMD method [15] takes two key steps. (i) Candidate instance generation: For a set of v threshold values, $\{t_h\}_{h=1}^{v}$, from the probability map of a slice S_i in a 3D image, produce a set IC_i of possible object instance candidates. These object candidates can be organized into a forest structure F_i. Also, a reference set R_{i-1} of object instances is built on the slice S_{i-1} (obtained iteratively). (ii) Candidate instance selection: For each pair of an instance candidate in F_i and a reference instance in R_{i-1}, compute their matching score as the cosine distance between their instance feature vectors. The goal is to maximize the sum of the weighted matching scores between the candidate set IC_i and reference set R_{i-1} to select the "best" object instances for the slice S_i. This can be solved by integer linear programming. For a dataset with n different classes, a feature vector for each instance candidate is defined as $(x, y, z, v_1, \ldots, v_n)$, whose first three items are the coordinates of its center pixel and the last n items are for an n-D one-hot vector denoting the category of the instance.

Rather than using the Euclidean distance as in [15], our method applies cosine distance, since our vectors contain two different types of information, which make the L_2 distance unsuitable to measure the differences between these vectors. Similar to bi-directional RNN [13], we perform the H-EMD process in two opposite directions (bi-HEMD). That is, for any two labeled slices S_i and S_j in a 3D image, $i < j$, we apply H-EMD along the direction of $S_{i+1}, S_{i+2}, \ldots, S_{j-1}$, and along $S_{j-1}, S_{j-2}, \ldots, S_{i+1}$. With the bi-HEMD process, the pseudo-labels generated by the 2D modules are improved, which are then used to train the 3D module in Section III-C.

E. Tuning the Final 3D Model Using Pseudo-labels

We now have three 2D K-FCNs and one 3D K-FCN trained with labeled or pseudo-labeled slices along the xy, yz, and xz planes. Next, we produce the probability maps of each 3D image M in W using these four FCN modules, denoted as m_{xy}, m_{yz}, m_{xz}, and m_{3D}, respectively. These probability maps are averaged, and the results are used to train our 3D meta-learner. This meta-learner is a Y-shaped K-DenseVoxNet that is aware of the raw images and their pseudo-labels so as to ease overfitting. Fig. 6 shows our meta-learner.

After training our 3D meta-learner, we apply the self-training strategy in [8] to further improve the model perfor-

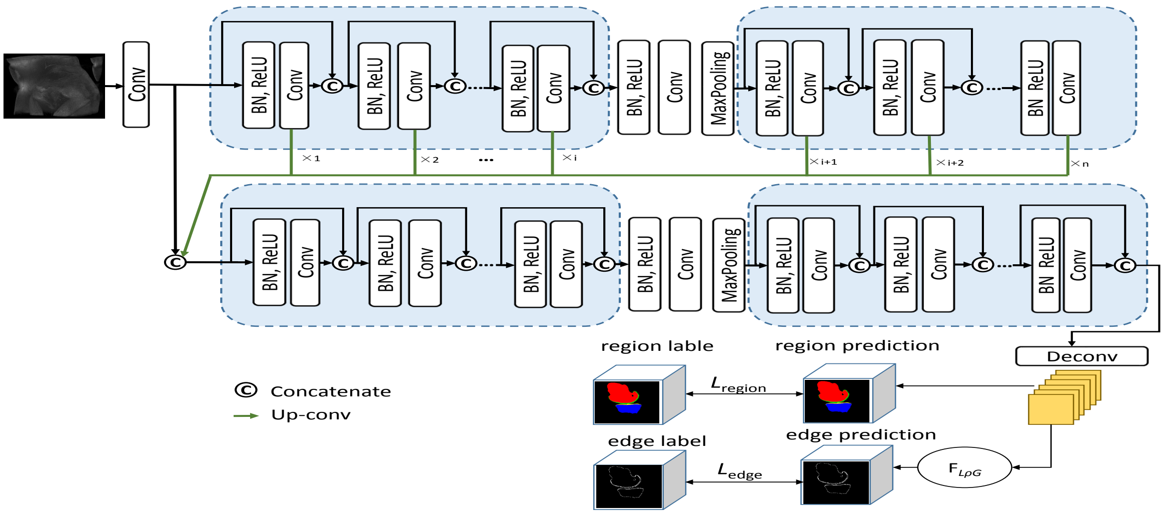


Fig. 5: The structure of our K-DenseVoxNet with edge-aware branches ($K = 2$).

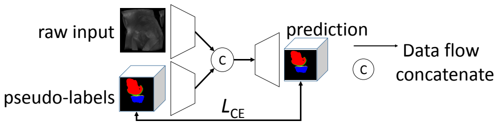


Fig. 6: The structure of our meta-learner.

mance. In this self-training process, the segmentation results of the meta-learner are regarded as pseudo “ground truth” of the unlabeled slices, which are used to re-train the 2D/3D base-learners (the three 2D base-learners are re-trained with the “labeled” slices along the three orthogonal planes). Note that the base-learners are first trained in the step of Section III-C. Here, we apply the SGD optimizer and a smaller learning rate to ensure the robustness and convergence of the entire training process. The loss function L_{CE} of the 3D meta-learner (see Fig. 6) is defined as the cross-entropy between the predictions and input pseudo-labels. The base-learners are re-trained, and generate four versions of pseudo-labels for each 3D image in W, which are averaged and used to train the meta-learner again. We repeat this self-training process for a few iterations, until the meta-learner performance no longer improves, giving rise to our final 3D model.

F. Post-processing Using IPM

Instead of applying the softmax function to the final probability maps, we further perform some post-processing to fine-tune the probability maps. One observation is that the surfaces of bones and cartilages are mutually “coupled” in some areas, within which the topology and relative positions of the bones and cartilages are known and the distances between them are within specific ranges. Furthermore, physicians care more about the “coupled” areas since osteoarthritis is usually caused by damages of the knee cartilages in such areas. Thus, we apply the IPM method [4] by incorporating the surface interrelationships between the bones and cartilages into the segmentation process to further improve the segmentation performance. An advantage of the IPM method over traditional graph based methods is that it parameterizes the surface cost functions in the graph model and leverages DL to learn the parameters rather than relying on hand-crafted features.

Instead of using ground truth to train the surface segmentation network of IPM [4], we use the pseudo-labels generated by our meta-learner to optimize this network in the first iteration. Afterwards, the pseudo-labels are updated by IPM and used to re-train the network. Such operations are repeated several times until convergence. The details of the above training process are shown in Fig. 8 [4].

Since the bone and cartilage surfaces are not terrain-like, we need to first unfold the knee joint into seven parts following the practice in [19], i.e., the front, back, top, center, bottom, left and right parts, respectively, as shown in Fig. 7.

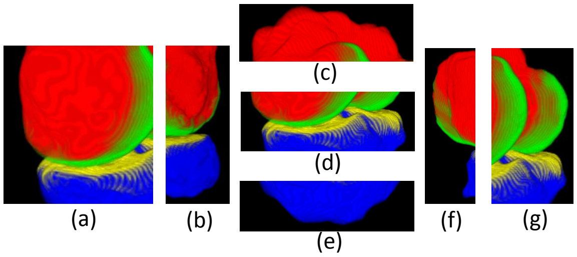


Fig. 7: Illustrating the seven unfolded parts of the knee joint. The corresponding parts in the sagittal view are: (a) front; (b) back; (c) top; (d) center; (e) bottom; (f) left; (g) right.

Specifically, for the center part (see Fig. 7(d)), we replace U-Net used in the original IPM method [4] with the probability maps generated by our final fine-tuned ensemble model. Finally, we patch its 6 junction areas (i.e., the junction areas between center and front, center and back, center and top, center and bottom, center and left, and center and right), and average the center area and its corresponding junction areas processed by IPM to smooth the final results.

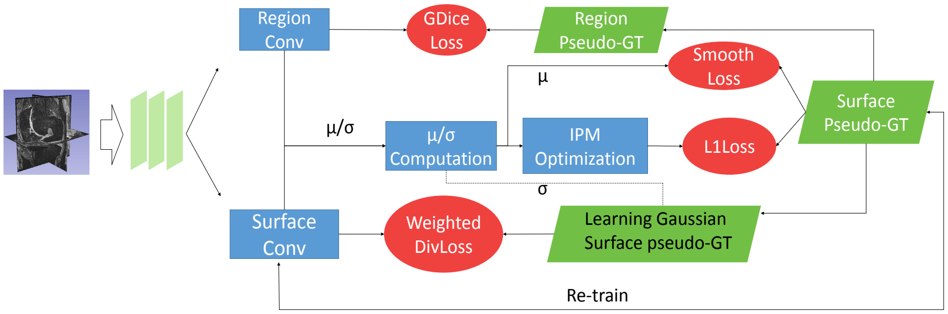


Fig. 8: The process of the post-processing step [4].

IV. EXPERIMENTS AND ANALYSIS

To demonstrate the capabilities of our KCB-Net approach, its performance was compared with state-of-the-art knee segmentation methods using full annotations as well as compared with two state-of-the-art slice selection strategies: equal-interval annotation (EIA) and random slice selection (RSS). Furthermore, the effect of each component in our KCB-Net framework was assessed and the robustness of the method was quantified for different sparse annotation ratios.

A. Datasets and Implementation Details

The performance of our KCB-Net model was evaluated on knee joint images from the Osteoarthritis Initiative database (OAI, http://www.oai.ucsf.edu/). The image size is $384\times384\times160$, with voxel size of $0.36mm \times 0.36mm \times 0.7mm$. Two subsets with ground truth are available: (1) A University of Iowa annotated portion of the OAI that was first segmented by the LOGISMOS method [3] and the automated segmentations then corrected by the just-enough-interaction (JEI) approach in 4D (3D+time) [20]. This Iowa dataset consists of 1462 double echo steady state (DESS) 3D MR images from 248 subjects. Four compartments are annotated: femur bone (FB), femoral cartilage (FC), tibia bone (TB), and tibial cartilage (TC). (2) The iMorphics dataset, available directly from the OAI database, includes 176 3D MR knee images acquired with 3T Siemens MAGNETOM Trio scanners and quadrature transmit-receive knee coils (USA Instruments, Aurora, OH, USA). The annotated compartments are femoral cartilage (FC), tibia cartilage (TC), patellar cartilage (PC), and menisci (M).

We implemented all the networks using PyTorch [21]. For our auto-encoder, ResNet-101 [16] is used as the backbone of its encoder and ResNet-50 [16] as the backbone of its decoder. The encoder is initialized with a model pre-trained on ImageNet [17]. All the other parameters are initialized as in [16], and λ_1 in Eq. (1) is set to $5e$-5. The network was optimized using the Adam optimizer (learning rate = $1e-4$, $\beta_1 = 0.9$, $\beta_2 = 0.999$). The 3D images were first cropped so as to remove the background clearly outside of the knee area. Each slice or 3D image was normalized to zero mean and unit standard variance. In the data augmentation for 3D model training, starting points are randomly selected in a 3D image, and a patch of size $80\times192\times160$ is cropped at each starting point. Afterwards, common spatial transforms (e.g., rotation, scaling, and mirroring) are applied. In 2D model training, each slice is augmented with common spatial transforms.

We set $K = 2$ for the K-FCNs and K-DenseVoxNet with edge-aware branches (for larger K, the model costs increase largely but the accuracy improves little [13]). We use *mean square error* as the auto-encoder's loss. We set the parameter of the edge regularizer in the edge-aware K-DenseVoxNet as $\lambda_2 = 1e-4$ (see Eq. (5)).

B. Evaluation Metrics

Dice similarity coefficient (DSC) and average symmetric surface distance (ASSD, in mm) between the labeled and segmented surfaces are used as our evaluation metrics.

1) Dice Similarity Coefficient: Dice similarity coefficient (DSC) is calculated as:

$$DSC = \frac{2\times V(GT\cap Pred)}{V(GT)+V(Pred)}, \tag{6}$$

where GT is the ground truth, $Pred$ is the prediction, and $V(X)$ denotes the volume of a 3D object X.

2) Average Symmetric Surface Distance: Average symmetric surface distance (ASSD) focuses on the absolute distances between surfaces of the segmented objects and their ground truths, calculated as:

$$ASSD = \frac{1}{n\partial A + n\partial B}\Big(\sum_{a\in\partial A} d(a,\partial B) + \sum_{b\in\partial B} d(b,\partial A)\Big), \tag{7}$$

where ∂A and ∂B denote the surfaces of objects A and B respectively, $n\partial A$ and $n\partial B$ denote the numbers of voxels on ∂A and ∂B respectively, and $d(x,\partial S)$ denotes the nearest Euclidean distance of a point x to a surface ∂S.

C. Experimental Results on Full Annotation

To obtain robust results, we conduct 5-fold cross validation on the Iowa and iMorphics knee datasets. For the Iowa dataset, 1170 3D images (out of 1462) are for training and 292 3D images for testing in each fold. For the iMorphics dataset, 140 3D images are for training and 36 3D images for testing.

Table I shows the performance comparison of KCB-Net and other methods trained on fully annotated Iowa dataset. The Iowa dataset was also used for other comparisons as follows: (i) 4D LOGISMOS [3]: utilizing a hierarchical set of random forest classifiers to learn the cartilage appearance and simultaneously segment multiple interacting surfaces of objects based on an algorithmic incorporation of multiple spatial interrelationships in an n-dimensional graph. (ii) The ensemble learning method [10]: Ensembling four 2D/3D FCNs and self-training with fully labeled 3D data.

From Table I, one can see that our KCB-Net outperforms LOGISMOS-4D on both the bone and cartilage segmentations. KCB-Net also outperforms the ensemble method [10], which demonstrates that the KU-Net design, edge-aware DenseVoxNet, bi-HEMD method, and IPM post-processing method that we use help improve the segmentation performance.

Table II presents the results achieved on the fully annotated iMorphics dataset. We compare with three recent methods: (i) UDA [22]: utilizing mixup and adversarial unsupervised domain adaptation to improve the robustness of DL-based knee cartilage segmentation in new MRI acquisition settings; (ii) CML [7]: detecting the regions of interest and fusing the cartilages by a fusion layer; (iii) the ensemble method [10]. Our method attains better DSC scores on FC, TC, PC, and M compared to the UDA method. We also outperform the CML and ensemble methods in both DSC and surface errors of FC, TC, and PC, suggesting that our method can obtain more quantitatively accurate knee cartilage/bone segmentation.

Performance improvement of our new method over the original ensemble method [10] was evaluated on the Iowa and iMorphics datasets, using paired t-tests. Tables I and II show that in most compared cases, our new approach significantly outperforms the earlier approach [10] (with $p < 0.05$).

D. Experimental Results on Sparse Annotation

To evaluate the performance of our method on sparsely annotated data, we compare its performances on data with changing sparse annotation ratios vs. those achieved using different slice selection schemes. Specifically, we compare the representative annotation (RA) scheme used in our KCB-Net pipeline with two common slice selection schemes: equal-interval annotation (EIA) and random slice selection (RSS). Suppose for an annotation ratio, S_k slices are to be selected. The EIA scheme selects $S_k/3$ slices at equal distance along each axis, and the RSS scheme randomly selects $S_k/3$ slices along each axis. We repeat the RSS process 10 times, and take the average of the results as the RSS base performance. Figs. 9 and 10 show the performance comparison with various annotation ratios on the Iowa and iMorphics datasets, respectively.

From Figs. 9 and 10, one can see that our RA outperforms the EIA and RSS schemes on both the cartilage and bone segmentations. Our method can notably alleviate performance degradation, especially for annotation ratios $\leq 5\%$. This is because EIA selects the locationally same slice indices in each 3D image, which might make the trained model overfit on the selected slices and cause segmentation errors on the remaining slices. RSS performs better than EIA in very sparse annotation ratios ($< \%10$) but worse than EIA in less sparse annotation ratios ($> \%40$), since RSS can select different slices in different 3D images, likely incurring less overfitting.

Another observation from Figs. 9 and 10 is that the performance drops drastically when the annotation ratios are $< 5\%$, suggesting that this annotation ratio may be the “lower limit” for a satisfactory performance on knee segmentation.

To examine the statistical significance of the improvements of RA over EIA and RA over RSS, we computed the p-values for RA over EIA, and RA over RSS at different annotation ratios. We observed that the improvements of RA over EIA and RA over RSS are typically statistically significant (p-values < 0.05) when the annotation ratios are quite small ($\leq 20\%$); for larger annotation ratios ($> 20\%$), the p-values tend to be ≥ 0.05. We think the reason for this trend is that for dense annotations, the chance of selecting the same or similar slices by different selection schemes increases quickly. Figs. 11 and 12 illustrate this trend on the Iowa and iMorphics datasets using the range 0%–30% of annotation ratios.

E. Ablation Study

To examine the contribution of each component in our KCB-Net, we conducted the ablation study to compare the performances of its components, denoted as follows. (1) S1: 2D xy module; (2) S2: 2D yz module; (3) S3: 2D xz module; (4) S4: 3D module; (5) S5: ensembling of the three 2D modules and the 3D module; (6) S6: bi-HEMD; (7) S7: self-training; (8) S8: IPM post-processing.

Performance of each individual component in S1, S2, S3, and S4 is given first, followed by the ensemble performance (S5) that combines all these four components. For S6–S8, components were repeatedly added to the framework each time; the more the performance increases, the more important the corresponding component (in S6–S8) is. Thus, note that S8 actually reflects the performance of the entire framework including all its components.

Tables III and IV present the ablation study results on the Iowa and iMorphics datasets, respectively. We observe that the ensemble of the 2D and 3D modules can substantially improve the performance over the individual modules. The 3D module often attains better performance than the 2D modules since it exploits the inter-relations among consecutive slices. The ensemble strategy can benefit from both the 2D modules (with a large receptive field) and the 3D module (exploiting the interactions among consecutive slices). Since some cartilages are very thin along the sagittal plane, it is quite difficult for DL models to detect them along such a plane, especially with very sparse annotation. Utilizing other 2D modules can help address this issue. Both Table III and Table IV show that the ensemble strategy and the self-training mechanism play more important roles than the other components. Figs. 13 and 14 qualitatively compare results in the sagittal view on the Iowa and iMorphics datasets.

F. Discussion

From Figs. 9 and 10, one can see that our representative annotation (RA) scheme substantially reduces the performance gap between different annotation ratios, meaning that our framework can achieve comparatively good results while using much less annotated data than required for full annotation. Our ensemble method and the self-training using pseudo-labels improved by the bi-HEMD method largely improve the

TABLE I: Comparison with state-of-the-art methods using full annotation on the Iowa dataset. Here, “–” denotes that the corresponding results were not reported in the original paper. Paired t-test values indicate the significance status of the improved performance of our method vs. the ensemble method [10] (statistically significant improvements are in bold).

	Femur Bone		Femoral Cartilage		Tibia Bone		Tibial Cartilage	
	DSC	ASSD	DSC	ASSD	DSC	ASSD	DSC	ASSD
LOGISMOS-4D [3]	–	–	–	0.55±0.11	–	–	–	0.60±0.14
Ensemble method [10]	0.940±0.011	0.551±0.017	0.830±0.020	0.541±0.010	0.930±0.131	0.557±0.156	0.812±0.034	0.590±0.177
Our method	**0.961±0.006**	**0.515±0.020**	**0.835±0.027**	0.522±0.009	0.957±0.102	**0.521±0.143**	**0.817±0.039**	**0.565±0.132**
p-value	0.043	0.001	0.047	0.066	0.071	0.009	0.032	0.044

TABLE II: Comparison with state-of-the-art methods using full annotation on the iMorphics dataset. Here, “–” denotes that the corresponding results were not reported in the original paper. Paired t-test values indicate the significance status of the improved performance of our method vs. the ensemble method [10] (statistically significant improvements are in bold).

	Femoral Cartilage		Tibial Cartilage		Patellar Cartilage		Menisci	
	DSC	ASSD	DSC	ASSD	DSC	ASSD	DSC	ASSD
UDA [22]	0.907±0.019	–	0.897±0.028	–	0.871±0.046	–	0.863±0.034	–
CML [7]	0.900±0.037	–	0.889±0.038	–	0.880±0.043	–	–	–
Ensemble method [10]	0.908±0.019	0.218±0.054	0.903±0.030	0.187±0.065	0.887±0.018	0.360±0.422	0.880±0.021	0.305±0.221
Our method	**0.919±0.020**	0.212±0.096	**0.909±0.025**	**0.184±0.068**	**0.900±0.026**	0.348±0.409	**0.889±0.024**	**0.295±0.210**
p-value	$\ll$ 0.001	0.233	0.001	0.002	$\ll$ 0.001	0.312	$\ll$ 0.001	0.002

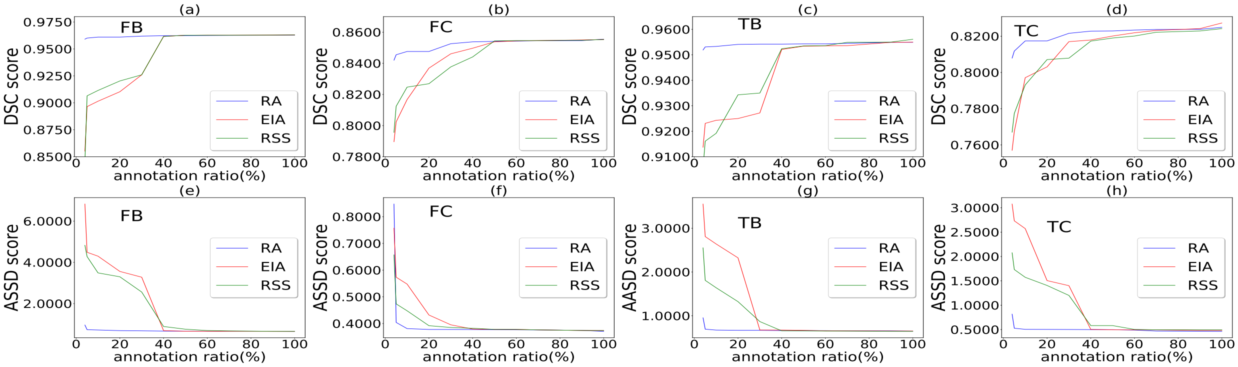


Fig. 9: Comparison of three slice selection schemes (RA, EIA, RSS) on the Iowa dataset.

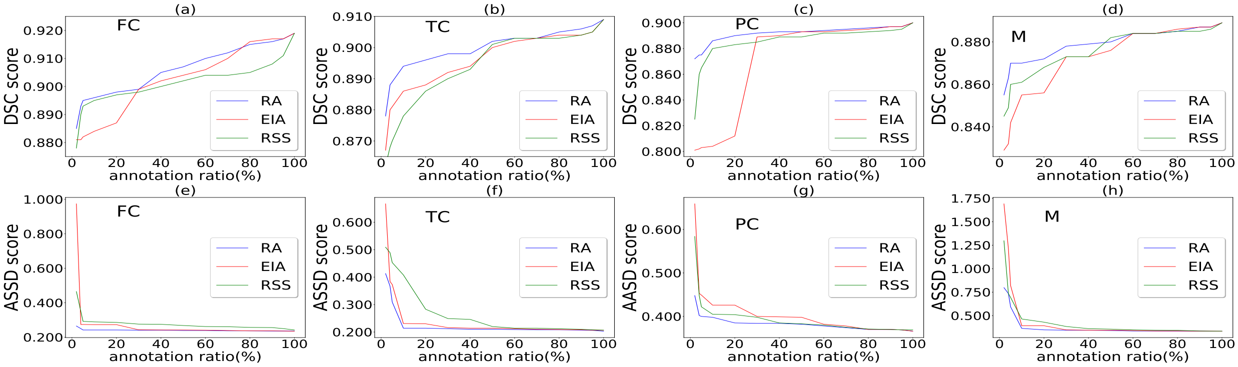


Fig. 10: Comparison of three slice selection schemes (RA, EIA, RSS) on the iMorphics dataset.

TABLE III: Ablation study of our method on the Iowa dataset.

	Femur Bone		Femoral Cartilage		Tibia Bone		Tibial Cartilage	
	DSC	ASSD	DSC	ASSD	DSC	ASSD	DSC	ASSD
S1 (xy)	0.938±0.025	0.550±0.027	0.815±0.037	0.557±0.015	0.924±0.187	0.562±0.109	0.796±0.040	0.603±0.192
S2 (yz)	0.931±0.016	0.562±0.020	0.811±0.025	0.566±0.016	0.916±0.129	0.573±0.217	0.790±0.086	0.599±0.210
S3 (xz)	0.936±0.019	0.558±0.023	0.812±0.024	0.564±0.009	0.918±0.163	0.571±0.156	0.792±0.069	0.602±0.191
S4 (3D)	0.940±0.023	0.550±0.016	0.817±0.012	0.556±0.011	0.926±0.125	0.560±0.147	0.796±0.094	0.601±0.221
S5 (ensemble)	0.947±0.007	0.540±0.014	0.820±0.039	0.552±0.019	0.933±0.109	0.552±0.233	0.804±0.033	0.613±0.219
S6 (bi-HEMD)	0.949±0.006	0.545±0.018	0.822±0.021	0.548±0.015	0.937±0.133	0.553±0.164	0.805±0.086	0.609±0.126
S7 (self-training)	0.957±0.007	0.517±0.012	0.831±0.035	0.528±0.010	0.950±0.082	0.524±0.143	0.814±0.036	0.572±0.138
S8 (IPM)	0.961±0.006	0.515±0.020	0.835±0.027	0.522±0.009	0.957±0.102	0.521±0.143	0.817±0.039	0.565±0.132

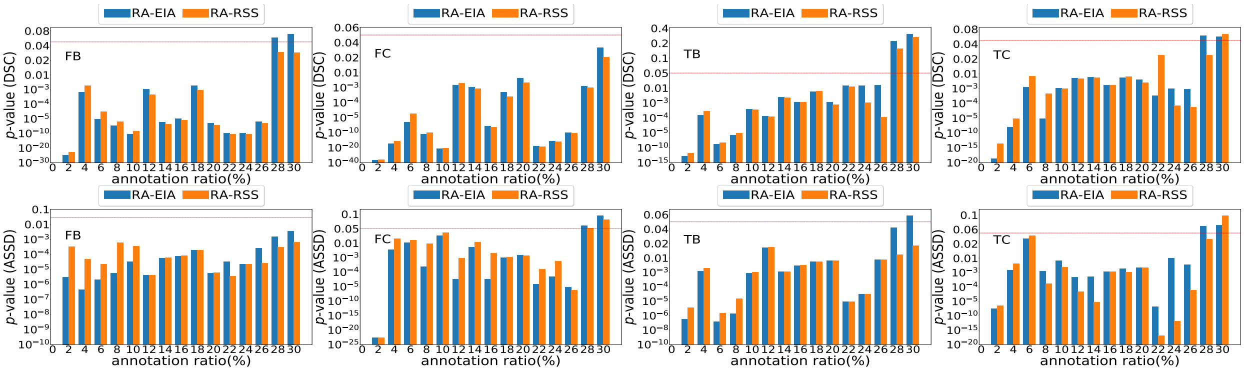


Fig. 11: Significance of performance improvements of employing our RA scheme vs. the EIA and RSS schemes on the Iowa dataset. The performance improvement is statistically significant if the charted p-value is below the red dashed line ($p < 0.05$). Experiments were performed in annotation ratio steps of 2%. To allow the very small (highly significant) p-values (e.g., $p < 0.01$) to be visible, the y-axes are piece-wisely adjusted and labeled to help improve the readability.

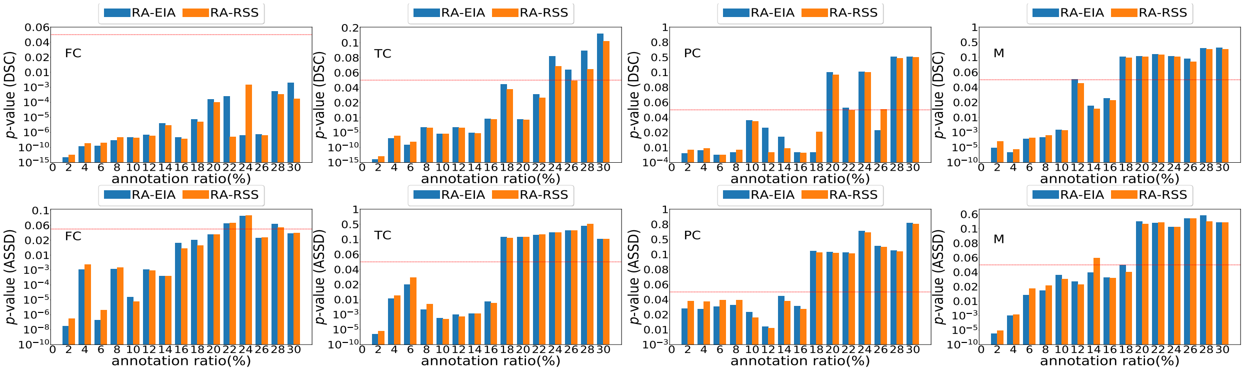


Fig. 12: Significance of performance improvements of employing our RA scheme vs. the EIA and RSS schemes on the iMorphics dataset. The performance improvement is statistically significant if the charted p-value is below the red dashed line ($p < 0.05$). Experiments were performed in annotation ratio steps of 2%. To allow the very small (highly significant) p-values (e.g., $p < 0.01$) to be visible, the y-axes are piece-wisely adjusted and labeled to help improve the readability.

TABLE IV: Ablation study of our method on the iMorphics dataset.

	Femoral Cartilage		Tibial Cartilage		Patellar Bone		Menisci	
	DSC	ASSD	DSC	ASSD	DSC	ASSD	DSC	ASSD
S1 (xy)	0.890±0.022	0.251±0.051	0.876±0.021	0.227±0.074	0.854±0.072	0.378±0.439	0.847±0.025	0.348±0.086
S2 (yz)	0.885±0.020	0.258±0.045	0.873±0.022	0.231±0.048	0.848±0.060	0.382±0.107	0.850±0.027	0.352±0.077
S3 (xz)	0.889±0.020	0.252±0.071	0.873±0.021	0.230±0.045	0.850±0.168	0.382±0.280	0.848±0.023	0.351±0.207
S4 (3D)	0.891±0.020	0.250±0.055	0.877±0.025	0.224±0.067	0.854±0.157	0.376±0.240	0.851±0.024	0.343±0.100
S5 (ensemble)	0.901±0.022	0.238±0.050	0.882±0.048	0.219±0.068	0.871±0.155	0.362±0.252	0.858±0.025	0.314±0.200
S6 (bi-HEMD)	0.902±0.020	0.236±0.082	0.887±0.020	0.213±0.072	0.877±0.051	0.360±0.402	0.864±0.024	0.317±0.201
S7 (self-training)	0.913±0.022	0.215±0.050	0.905±0.024	0.189±0.065	0.896±0.070	0.348±0.033	0.886±0.027	0.297±0.155
S8 (IPM)	0.919±0.020	0.212±0.096	0.909±0.025	0.184±0.068	0.900±0.026	0.348±0.409	0.889±0.024	0.295±0.210

segmentation performance, because the training data we use contribute new information in a more efficient way. Figs. 13 and 14 show that our ensemble and self-training strategies allow detection of small objects and thin boundary areas, despite the annotation sparsity. Our IPM post-processing helps further fine-tune the boundary areas, making the segmentation results more accurate and reliable overall.

V. CONCLUSIONS

We reported a new framework, KCB-Net, for segmenting cartilage and bone surfaces in 3D knee joint MR images. Our method efficiently selects subsets of diverse image slices for expert annotations in a way that the most information-contributing slices are ranked most highly, allowing to train image segmentation models from high-sparsity ratio annotations. In the KCB-Net, three 2D segmentation modules and one 3D module integrating features across multiple scales with edge-aware branches are ensembled to generate pseudo-labels of the un-annotated slices, which are then used to re-train the 3D model. An IPM process is employed to post-process the probability maps generated by the 3D model. Experiments on two large knee datasets show that our new approach outperforms state-of-the-art methods on fully annotated data, and can notably improve segmentation performance when

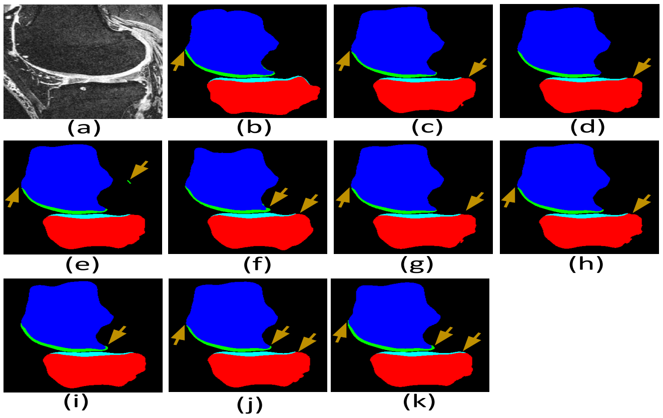


Fig. 13: Visual comparison of component-specific contributions (S1–S8) in our method in the sagittal view on the Iowa dataset. (a) An input 2D slice from a 3D image; (b) ground truth; (c) segmentation obtained by the ensemble method [10]; (d)-(k) segmentations obtained using the S1–S8 components, respectively. Note that our method successfully segments even thin cartilage areas. Arrows point to some spots of interest.

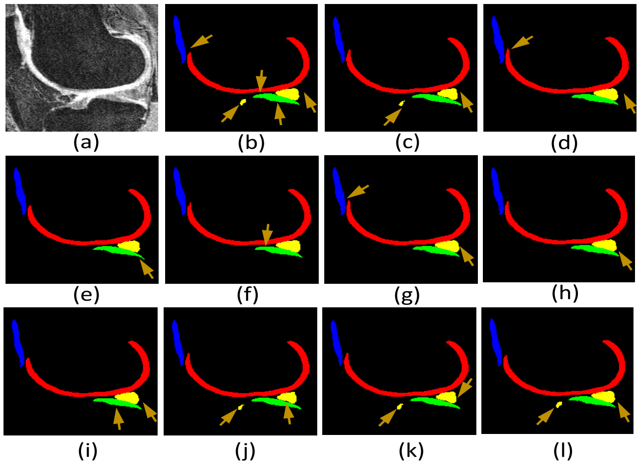


Fig. 14: Visual comparison of component-specific contributions (S1–S8) in our method with the UAD and ensemble [10] methods in the sagittal view on the iMorphics dataset. (a) An input 2D slice from a 3D image; (b) ground truth; (c) segmentation obtained by the ensemble method [10]; (d) segmentation obtained by UAD; (e)-(l) segmentations obtained using our S1–S8 components, respectively. Note that our method can segment the meniscus. Arrows point to some spots of interest.

annotating only small data subsets.

REFERENCES

[1] Paley Orthopedic & Spine Institute, “Anatomy of the knee joint,” https://paleyinstitute.org/centers-of-excellence/cartilage-repair/anatomy-of-the-knee-joint/, 2018.

[2] Y. Yin, X. Zhang, R. Williams, X. Wu, D. D. Anderson, and M. Sonka, “LOGISMOS—layered optimal graph image segmentation of multiple objects and surfaces: Cartilage segmentation in the knee joint,” IEEE Transactions on Medical Imaging, vol. 29, no. 12, pp. 2023–2037, 2010.

[3] S. Kashyap, H. Zhang, K. Rao, and M. Sonka, “Learning-based cost functions for 3-D and 4-D multi-surface multi-object segmentation of knee MRI: Data from the osteoarthritis initiative,” IEEE Transactions on Medical Imaging, vol. 37, no. 5, pp. 1103–1113, 2017.

[4] H. Xie, Z. Pan, L. Zhou, F. A. Zaman, D. Chen, J. B. Jonas, Y. Wang, and X. Wu, “Globally optimal segmentation of mutually interacting surfaces using deep learning,” arXiv preprint arXiv:2007.01259, 2020.

[5] F. Liu, Z. Zhou, H. Jang, A. Samsonov, G. Zhao, and R. Kijowski, “Deep convolutional neural network and 3D deformable approach for tissue segmentation in musculoskeletal magnetic resonance imaging,” Magnetic Resonance in Medicine, vol. 79, no. 4, pp. 2379–2391, 2018.

[6] F. Ambellan, A. Tack, M. Ehlke, and S. Zachow, “Automated segmentation of knee bone and cartilage combining statistical shape knowledge and convolutional neural networks: Data from the osteoarthritis initiative,” Medical Image Analysis, vol. 52, pp. 109–118, 2019.

[7] C. Tan, Z. Yan, S. Zhang, K. Li, and D. N. Metaxas, “Collaborative multi-agent learning for MR knee articular cartilage segmentation,” in International Conference on Medical Image Computing and Computer-Assisted Intervention, 2019, pp. 282–290.

[8] H. Zheng, Y. Zhang, L. Yang, P. Liang, Z. Zhao, C. Wang, and D. Z. Chen, “A new ensemble learning framework for 3D biomedical image segmentation,” in AAAI Conference on Artificial Intelligence, vol. 33, no. 01, 2019, pp. 5909–5916.

[9] L. Yu, J.-Z. Cheng, Q. Dou, X. Yang, H. Chen, J. Qin, and P.-A. Heng, “Automatic 3D cardiovascular MR segmentation with densely-connected volumetric ConvNets,” in International Conference on Medical Image Computing and Computer-Assisted Intervention, 2017, pp. 287–295.

[10] H. Zheng, Y. Zhang, L. Yang, C. Wang, and D. Z. Chen, “An annotation sparsification strategy for 3D medical image segmentation via representative selection and self-training,” in AAAI Conference on Artificial Intelligence, vol. 34, no. 4, 2020, pp. 6925–6932.

[11] Ö. Çiçek, A. Abdulkadir, S. S. Lienkamp, T. Brox, and O. Ronneberger, “3D U-Net: Learning dense volumetric segmentation from sparse annotation,” in International Conference on Medical Image Computing and Computer-assisted Intervention, 2016, pp. 424–432.

[12] H. Zheng, S. M. M. Perrine, M. K. Pitirri, K. Kawasaki, C. Wang, J. T. Richtsmeier, and D. Z. Chen, “Cartilage segmentation in high-resolution 3D micro-CT images via uncertainty-guided self-training with very sparse annotation,” in International Conference on Medical Image Computing and Computer-Assisted Intervention, 2020, pp. 802–812.

[13] J. Chen, L. Yang, Y. Zhang, M. Alber, and D. Z. Chen, “Combining fully convolutional and recurrent neural networks for 3D biomedical image segmentation,” Conference on Neural Information Processing Systems, pp. 3036–3044, 2016.

[14] Z. Guo, H. Zhang, Z. Chen, E. van der Plas, L. Gutmann, D. Thedens, P. Nopoulos, and M. Sonka, “Fully automated 3D segmentation of MR-imaged calf muscle compartments: Neighborhood relationship enhanced fully convolutional network,” Computerized Medical Imaging and Graphics, vol. 87, p. 101835, 2021.

[15] P. Liang, Y. Zhang, Y. Ding, J. Chen, C. S. Madukoma, T. Weninger, J. D. Shrout, and D. Z. Chen, “H-EMD: A hierarchical earth mover’s distance method for instance segmentation,” submitted, 2021.

[16] K. He, X. Zhang, S. Ren, and J. Sun, “Deep residual learning for image recognition,” in IEEE Conference on Computer Vision and Pattern recognition, 2016, pp. 770–778.

[17] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and F.-F. Li, “ImageNet: A large-scale hierarchical image database,” in IEEE Conference on Computer Vision and Pattern Recognition, 2009, pp. 248–255.

[18] L. Yang, Y. Zhang, J. Chen, S. Zhang, and D. Z. Chen, “Suggestive annotation: A deep active learning framework for biomedical image segmentation,” in International Conference on Medical Image Computing and Computer-Assisted Intervention, 2017, pp. 399–407.

[19] L. Zhou, Z. Zhong, A. Shah, B. Qiu, J. Buatti, and X. Wu, “Deep neural networks for surface segmentation meet conditional random fields,” arXiv e-prints, pp. arXiv–1906, 2019.

[20] S. Sun, M. Sonka, and R. R. Beichel, “Graph-based IVUS segmentation with efficient computer-aided refinement,” IEEE Transactions on Medical Imaging, vol. 32, no. 8, pp. 1536–1549, 2013.

[21] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga et al., “PyTorch: An imperative style, high-performance deep learning library,” arXiv preprint arXiv:1912.01703, 2019.

[22] E. Panfilov, A. Tiulpin, S. Klein, M. T. Nieminen, and S. Saarakkala, “Improving robustness of deep learning based knee MRI segmentation: Mixup and adversarial domain adaptation,” in IEEE/CVF International Conference on Computer Vision Workshops, 2019, pp. 450–459.