Medical Image Analysis 67 (2021) 101826

Contents lists available at ScienceDirect

Medical Image Analysis

journal homepage: www.elsevier.com/locate/media

Automatic vertebrae recognition from arbitrary spine MRI images by a category-Consistent self-calibration detection framework

Shen Zhao[a], Xi Wu[b,*], Bo Chen[c], Shuo Li[c,*]

[a] Sun Yat-Sen University, Guangzhou, China
[b] Department of Computer Science, Chengdu University of Information Technology, Chengdu,610225,China
[c] University of Western Ontario, London ON, Canada

ARTICLE INFO

Article history:
Received 5 November 2019
Revised 28 September 2020
Accepted 29 September 2020
Available online 9 October 2020

Keywords:
Vertebrae recognition
Object detection
Message passing
Dictionary learning
Label consistency

ABSTRACT

Accurate vertebrae recognition is crucial in spinal disease localization and successive treatment planning. Although vertebrae detection has been studied for years, reliably recognizing vertebrae from arbitrary spine MRI images remains a challenge. The similar appearance of different vertebrae and the pathological deformations of the same vertebrae makes it difficult for classification in images with different fields of view (FOV). In this paper, we propose a Category-consistent Self-calibration Recognition System (Can-See) to accurately classify the labels and precisely predict the bounding boxes of all vertebrae with improved discriminative capabilities for vertebrae categories and self-awareness of false positive detections. Can-See is designed as a two-step detection framework: (1) A hierarchical proposal network (HPN) to perceive the existence of the vertebrae. HPN leverages the correspondence between hierarchical features and multi-scale anchors to detect objects. This correspondence tackles the image scale/resolution challenge. (2) A Category-consistent Self-calibration Recognition (CSRN) Network to classify each vertebra and refine their bounding boxes. CSRN leverages the dictionary learning principle to preserve the most representative features; it imposes a novel category-consistent constraint to force vertebrae with the same label to have similar features. CSRN then innovatively formulates message passing into the deep learning framework, which leverages the label compatibility principle to self-calibrate the wrong pre-recognitions. Can-See is trained and evaluated on a capacious and challenging dataset of 450 MRI scans. The results show that Can-See achieves high performance (testing accuracy reaches 0.955) and outperforms other state-of-the-art methods.

1. Introduction

Automatic vertebrae recognition (*i.e.*, label classification and bounding box localization) from magnetic resonance imaging (MRI) helps measure the spine's appearance, shape, and geometry, which helps detect local and global abnormalities for the diagnosis of herniation and spinal scoliosis. Automatic vertebrae recognition is thus an essential tool for spine disease diagnosis, medical and surgical treatment planning, as well as postoperative response assessment (Liao et al., 2018). Performing automatic vertebrae recognition accurately and reproducibly for each vertebra is crucial because incorrect recognition may cause mis-diagnosis and wrong-site surgery (surgery on the wrong vertebrae), which is one of the five surgical Never Events in clinical practice (Stahel and Mauffrey, 2014)

However, as shown in Fig. 1, automatic vertebrae recognition in an arbitrary spine image is challenging because (1) The field of view (FOV) varies unpredictably in the input images, e.g., Fig. 1(a) ~ (c) have different FOVs. It is not guaranteed that some specifically-shaped vertebrae (such as the sacrum) exist in the input image, so it is impossible to use these vertebrae to classify the other ones Liao et al. (2018). (2) The input MRI images are often obtained by different MRI settings (such as echo time, repetition time or RF pulses), while the training dataset of a certain setting is often limited. This means that the image characteristics (shape, appearance, texture, resolution, scales, and image intensity distribution) vary widely in the dataset (Yang et al., 2017), e.g., Fig. 1(b) has relatively low resolution. These varieties of the training data impose difficulty on learning representative features to classify the vertebrae. (3) The appearances of different vertebrae are similar due to their repetitive nature (Fig. 1(d)); while the pathological

* Corresponding authors.
E-mail addresses: xi.wu@cuit.edu.cn (X. Wu), slishuo@gmail.com (S. Li).

https://doi.org/10.1016/j.media.2020.101826
1361-8415/

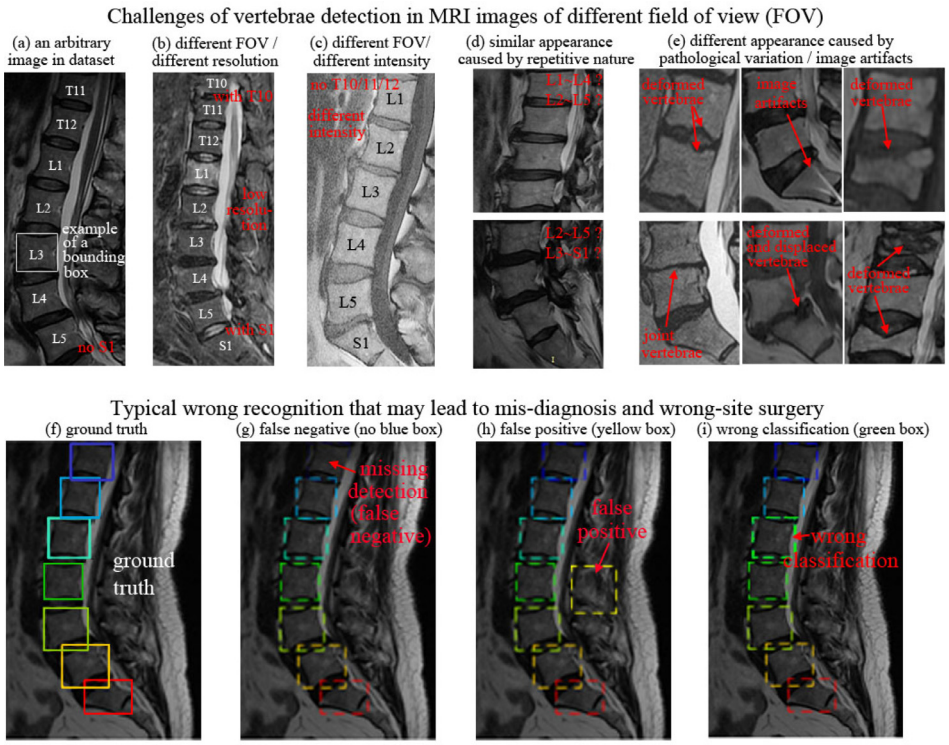

Fig. 1. Challenges of automatic vertebrae recognition in arbitrary spine images (classifying the labels and tight bounding boxes (*i.e.* the white box for L3 in Fig. 1 (a)) of each vertebrae). Fig. 1(a ~ c) show the challenges caused by different fields of view (FOV) and image characteristics (resolution, intensity, acquisition settings, and scales). Fig. 1(d ~ e) show that the repetitive nature, pathological variation, and/or image artifacts make the problem more challenging. Fig. 1(f ~ i) show the ground truth of the vertebrae recognition results and typical wrong recognitions.

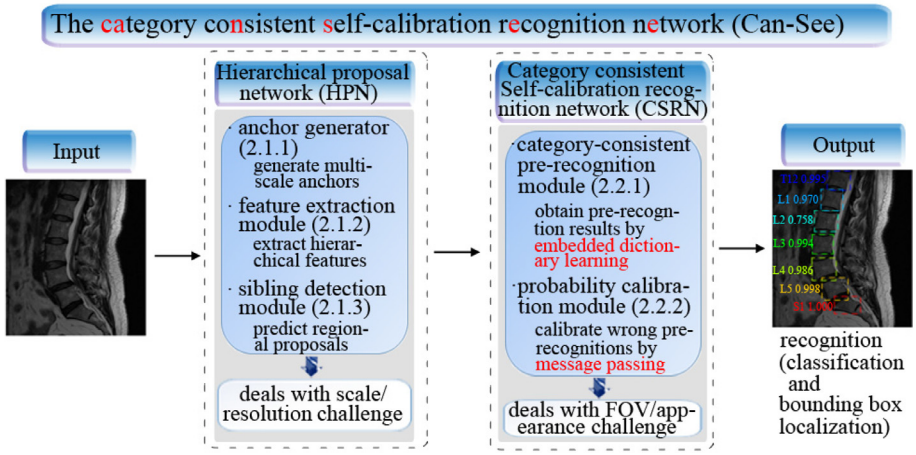

Fig. 2. Overview of Can-see, which is a two-stage network containing: (1) Hierarchical proposal network (HPN) for coarsely localizing regions containing vertebrae (class-agnostic proposals) of different scales and resolutions. (2) Category-consistent self-calibration recognition network (CSRN) for recognizing the class label and bounding box of each vertebra. Category-consistent dictionary learning is integrated into the recognition network for improved discriminative capability; message passing is embedded for automatically calibrating the wrong recognitions caused by different FOV and appearance.

variation and/or image artifacts alters the appearance of the vertebrae in an unknown manner (Fig. 1(e)) (Lootus et al., 2014; Chen et al., 2015). The unpredictable FOV of the input MRI image adds to the problem because it is difficult to classify the vertebrae (e.g., distinguish MRI images containing L5 ~ T11 from those containing L4 ~ T10) due to their similarity in appearance and the pathological variations.

We propose a category-consistent self-calibration recognition network (Can-See) to recognize vertebrae from arbitrary spine MRI images. As shown in Fig. 2, Can-see is composed of two carefully designed networks: a Hierarchical proposal network (HPN) and a Category-consistent Self-calibration Recognition Network (CSRN). These modules are proposed to enhance the robustness to scale/resolution, vertebrae discriminative capabilities, and the self-awareness of false positive detections:

- HPN coarsely locates regions containing vertebrae (called proposals) by matching multi-scale anchors to discriminative hierarchical features. It leverages the principle that an object can be detected by a box that is close enough to and has similar shape with it. HPN is designed to generate anchors to cover all possible scales of the vertebrae in the arbitrary input image, and then estimate the probability of each anchor containing a ver-

tebrae by extracting features for each of them. This design can effectively deal with the multi-scale/resolution challenge.
- CSRN contains a deliberated probability calibration module, which is designed to leverage the principle of message passing to calibrate the predicted class probabilities of the wrong recognitions. CSRN also contains a novel category-consistent module. This module fuses the dictionary learning principle into a deep learning network by a label-consistent k-sparse encoder, which promotes vertebrae of the same category to have similar sparse codes. These strategies can successfully solve the appearance and FOV challenge.

1.1. Related methodology review

1.1.1. Instance detection based on CNNs

Recent instance detection methods are impressive and have achieved great success in many applications, however, they cannot be directly applied to vertebrae detection. This is because they may not be capable of distinguishing vertebrae of similar appearance, especially from an image of unknown FOV. Recent instance detections methods are mainly divided into two categories: two-step detection (such as Faster RCNN (Ren et al., 2015)) and one-step detection (such as YOLO (Redmon et al., 2016) and SSD (Liu et al., 2016)). As reviewed in our previous work (Zhao et al., 2019b), the main difference between two-step detection and one-step detection is whether to use regional proposals to coarsely locate the objects. While methods of both categories have shown good results in several applications (Ren et al., 2015; He et al., 2017; Ben-Ari et al., 2017; Li et al., 2017), two-step detection generally has a higher detection accuracy than one-step detection (Zhao et al., 2018). Recently, two-step detection has been extended to 3D object detection based on point cloud (Zhou and Tuzel, 2018).

In the medical image analysis domain, although the postures of the objects do not change much, a great challenge is the repetitive nature (similar appearances) of different objects and unpredictable deformations of the same objects. To make it more challenging, clinical diagnosis and surgery require a perfect detection accuracy of discriminating similar-appearing vertebrae. Thus, even the more accurate the two-step detection method is chosen, wrong recognitions can still happen in some images (lower part of Fig. 1).

More specifically, there are typically three kinds of wrong cases for vertebrae recognition task: missing detection (false negative, (Fig. 1(g))), false positive (Fig. 1(h)), and wrong classification (Fig. 1(i)). Unfortunately, wrong recognition for even one single vertebra is not tolerable because it may result in mis-diagnosis and wrong-site surgery. Fine-tuning the hyper-parameters of existing Faster RCNN can adjust the number of detected objects in each image, however, this strategy is not robust enough to deal with the image characteristic variance of MRI images. Thus, improvements are needed before this strategy can be used for vertebrae recognition tasks.

1.1.2. Existing work on vertebrae recognition

Existing works on vertebrae recognition use ideas different from the above-mentioned instance detection workflow. Some of these works use classical machine learning methods, while others leverage deep networks. While nontrivial progress has been achieved recently, simultaneously recognizing the labels and bounding boxes of all vertebrae from arbitrary spine images remains a problem.

Classical machine learning methods can accurately recognize vertebrae, however, since they are based on handcrafted feature extraction methods, they may not be generalized to vertebrae of more diverse visual characteristics, i.e., they may not be able to handle more complicated pathological cases. For these algorithms, Lootus (Lootus et al., 2014) presents an accurate and efficient method that combines a classical deformable part model detector with dynamic programming, but it needs the sacrum to be present. Glocker (Glocker et al., 2012) uses regression forests and probabilistic graphic models, but it suffers from narrow FOV (Yang et al., 2017). They then(Glocker et al., 2013) solved this problem by replacing the regression forest with a randomized classification forest to detect the vertebra centroids via pixel-wise dense probabilistic labels training, but it still requires hand-crafted features (Chen et al., 2015), which might not be robust enough for arbitrary input images. Other methods, such as snakes, level-set, dynamic programming and state-space approaches (Zhao et al., 2017; Gao et al., 2017), have also proved to be effective in detection/segmentation tasks of human organs such as carotid intima-media borders and optic disc. These universal methods can be modified to perform vertebrae detection (Kamalakannan et al., 2010) since they are trained to be aware of objects with some certain appearance. However, they still need some parameters to be empirically set, which may again require manual adjustments of these parameters to handle more complicated pathological cases.

With the development of deep learning, convolutional neural networks (CNNs) are more and more frequently used for detection/segmentation tasks (Zhao et al., 2019b; 2019a). Most of these works formulate vertebrae recognition as a centroid point detection task. (Chen et al., 2015) uses CNNs jointly trained with a shape regression model to extract more robust features for vertebrae detection and achieves superior detection performance than traditional methods. Yang (Yang et al., 2017) uses deeply supervised CNN enhanced by message passing to accurately predict pixel-wise probability maps of each vertebrae centroid; Liao (Liao et al., 2018) uses a fully convolutional neural network (FCN) and recurrent neural network (RNN) to localize the centroid of each vertebra. However, directly recognizing the labels and bounding boxes of the vertebrae (rather than the probability map of centroid points) may be more meaningful. Clinically, this reveals relative sizes and positions of vertebrae to perceive pathological deformations; technically, this strategy mitigates the problem of false positives because they provide information about the sizes and overlapping of different recognitions (Yang et al., 2017; Glocker et al., 2013).

1.1.3. Label relationship exploitation

Exploiting the relationships of different recognitions helps improve recognition accuracy; however, this strategy has not been fully studied in instance recognition tasks. Since wrong recognition for even one single vertebra is not clinically tolerable, it is crucial to promote recognition performance using the relationships of different recognitions. Previously, this strategy has been used in image semantic segmentation (Arnab et al., 2016), which uses conditional random fields to leverage the high-order potentials (e.g., the label consistency over super-pixels, which is a kind of label relationship) to correct mistakes in the semantic segmentation work. Then, (Luc et al., 2016) uses the recently popular Generative Adversarial Network (GAN) (Goodfellow et al., 2014) to implicitly leverage more kinds of high-order potentials by assessing the joint configuration of many label variables. We (Zhao et al., 2019b) also attempt to use GAN in an instance detection framework; although GANs have achieved relatively acceptable results in this work, it triggers the thought that is it worthy to use the complicated GAN network (which is relatively unstable to train (Radford et al., 2015)) to trade for the added kinds of implicit high-order potentials?

Compared with semantic segmentation, the advantage of instance detection frameworks is that they can yield recognition results i.e., the class probability vectors (CPVs) for the classification task and the bounding box coordinates for the regression task. **Message passing** (Yedidia et al., 2003) has the potential to leverage the relationship of the recognition results in a more effective way than adversarial networks. This potential is demonstrated by (Yang et al., 2017), who uses the message passing algorithm to

successfully handle the problem of missing response in the task of predicting pixel-wise probability map (the probability of each pixel to be the centroid of a specific vertebra) for vertebrae centroids. However, correcting the wrong recognitions not only involves compensating the missing detections but also correcting the recognitions that have wrong labels and deleting the false positive recognitions. The potential of message passing should be further exploited to tackle these tasks.

The concept of message passing is originally used for exactly inferring the marginal probabilities of each node in a tree-structured probabilistic graphic model. In instance recognition work, the nodes refer to the sorted recognitions, the marginal probabilities refers to the CPVs, the graphic model refers to the collection of all CPVs and their relationships (label compatibility) in the input MRI image, and the messages are the scores that one specific node taking different labels according to its neighbors. During the message passing, the CPV of each recognized vertebrae receives messages from its neighboring recognitions and absorbs them for self-calibration. However, message passing requires the majority of the nodes in the graphic model to have correct CPVs. Fortunately, this demand can be met by arranging the recognition results of the deep CNN-based instance detection framework into a sequential series. Thus, embedding the message passing scheme into the instance detection framework perfectly complements each other.

1.1.4. Label consistency and sparse coding

Sparse feature representations help to promote the performance in deep networks (Liu et al., 2018), however, it has not achieved state-of-the-art performance in the medical imaging domain. Sparse codes can to some extent be achieved by the frequently-used rectified linear units (ReLUs), however, sparsity can not be guaranteed in ReLU. Dictionary learning plays an indispensable role in sparse codes because it guarantees sparsity with improved discriminative capability (Makhzani and Frey, 2013). As expressed by Eq. (1), dictionary learning optimizes both the dictionary $\mathbf{D}$ and the sparse codes $\mathbf{X}$ to minimize the reconstruction error:

$$< \mathbf{D}, \mathbf{X} >= \arg\min_{\mathbf{D},\mathbf{X}} \|\mathbf{Y} - \mathbf{D}\mathbf{X}\|_2^2 \quad \text{s.t.} \forall i, \|\mathbf{x}_i\|_0 \leq T \tag{1}$$

where $\mathbf{Y} \in R^{n \times N}$ are a batch of N input signals (each signal is n-dimensional), $\mathbf{D} = [\mathbf{d}_1 \dots \mathbf{d}_M] \in R^{n \times M}$ is the trainable dictionary matrix (each row $\mathbf{d}_i$ is an atom of the dictionary), $\mathbf{X} = [\mathbf{x}_1 \dots \mathbf{x}_N] \in R^{M \times N}$ is the sparse code of the input signals $\mathbf{Y}$ (each row $\mathbf{x}_j$ is the sparse code of one signal). In each training step, the dictionary $\mathbf{D}$ and the sparse code $\mathbf{x}_i$ for each sample are alternately updated. Classification loss can also be added to Eq. (1) for classification tasks. The learned $\mathbf{x}_i$ enhances the sparsity, which is beneficial to the reconstruction and classification problem (Makhzani and Frey, 2013; Jiang et al., 2013).

However, limited attempts of dictionary learning have been carried out in instance detection frameworks based on CNNs. This is probably because most dictionary methods adopt the above-mentioned iterative training procedure that trains the dictionary off-line, which makes it difficult to be trained together with the CNNs in the instance detection framework. Moreover, the L0 norm loss in the constraint conditions does not have a derivative, which also makes it inappropriate to be trained with the CNNs via back-propagation. Recently, (Liu et al., 2018) proposes a method that integrates dictionary learning with CNNs by a recurrent unit that iteratively updates the sparse codes, however, this method requires the optimal sparse codes as the ground truth, which is often unavailable in instance detection frameworks. (Yang et al., 2017) uses a pre-generated dictionary to reconstruct the vertebrae coordinates of the maximum descending subsequence in the predicted coordinates. This strategy, although achieves high performance for correcting the centroid coordinates, does not make full use of the advantages of the dictionary learning for improved discriminative capability. (Jiang et al., 2013) innovatively introduces the concept of dictionary label, which associates label information with each dictionary atom and uses the KSVD method (Aharon et al., 2006) to solve the minimization problem with L0 norm loss. This method promotes the discriminative capability and achieves good performance in both reconstruction and classification tasks.

Inspired by previous works, we conjecture that integrating the label-consistent dictionary with the instance detection framework promotes feature points with the same class labels to have similar sparse codes, which results in higher recognition accuracy. As discussed above, the KSVD method is used for solving the L0 norm minimization problem, and can not be directly integrated into the instance detection framework. Actually, besides KSVD, many effective methods have been developed to solve the dictionary learning problem these years, such as the ISTA (iterative shrinkage/thresholding algorithm) algorithm (Chambolle et al., 1998), the OMP (orthogonal matching pursuit) algorithm (Pati et al., 1993), the LASSO (least absolute shrinkage and selection operator) algorithm (Tibshirani, 1996), and the DSR (deep sparse code) algorithm (Sharma et al., 2017). However, all these works solve the problem as a two-step work, i.e., they alternately optimize $\mathbf{D}$ and $\mathbf{X}$, which imposes difficulty on integrating them into CNN networks. Thus, it is crucial to develop a variant of the label consistency in (Jiang et al., 2013) that can be embedded into the instance detection framework.

1.2. Contributions

- We propose an accurate clinical tool to simultaneously recognize vertebrae labels and bounding boxes from arbitrary input MRI images of different FOV, resolution, intensity, acquisition settings, and scales. This reveals the shapes and relative positions of different vertebrae for further clinical diagnosis.
- For the first time, we develop a method to integrate the concept of label-consistent dictionary learning to the instance detection framework. This strategy enables end-to-end training of the dictionary and promotes vertebrae of the same labels to have similar sparse codes, which improves the discriminative ability of the recognition framework.
- For the first time, we formulate message passing into a deep learning object recognition network for class probability vectors (CPVs) calibration and provide a simple and effective method to guarantee the inputs to the message passing is valid. This strategy leverages the relationships of the pre-recognized objects for automatically correcting the wrong pre-recognitions with the help of the right ones. It enhances the performance in case where the inputs possess different FOVs and similar appearances. This also benefits other object recognition problems where the coordinates of the target objects have some certain internal spatial relationships.

In this work, we advance our preliminary attempt on vertebrae detection in MICCAI 2019 (Zhao et al., 2019c) in the following aspects: (1) We propose a label-consistent dictionary learning module and integrate it into a deep learning-based instance detection network, which effectively mitigates the intractable wrong recognitions and paves the way for the succeeding message passing. (2) We propose a simple and effective interface to convert the pre-recognition results to a series of nodes in a tree-structured probabilistic graphic model that is needed by the message passing scheme. (3) We carry out in-depth descriptions and discussions on the mechanism of message passing. These discussions provide a more profound understanding of how the label compatibility matrix is trained and how it is used in the testing phase when com-

bined with the CNN-based framework. (4) A more comprehensive review of vertebrae detection, instance detection, message passing, and dictionary learning is conducted to provide a panorama of existing work.

2. Methodology

Our Can-See (Fig. 2) is a tightly integrated deep recognition framework composed of two cascading stages:

1. The hierarchical proposal network (HPN, Section 2.1, Fig. 3) takes the original input MRI slice as input; and yields regional proposals (multi-scale rectangle boxes that coarsely cover the vertebrae) and features corresponding to the proposals as output. It tackles the multi-scale/resolution challenges by three cascading modules:
   a) The anchor generator generates multi-scale anchors at different regular locations to cover vertebrae of all possible sizes and shapes.
   b) The feature extraction module extracts hierarchical discriminative image features corresponding to all anchors of different positions, sizes, and shapes.
   c) The sibling detection module processes these features to predict which anchors contain vertebrae and the coarse locations of the vertebrae (called proposals).

2. The category-consistent self-calibration recognition network (CSRN, Section 2.2, Fig. 4) takes the proposals and their features as input; and yields final recognitions (represented by predicted classes labels and bounding boxes). It tackles the FOV and/or vertebrae appearance challenge by two deliberately designed modules:
   a) The category consistent pre-recognition module calculates sparse codes for each proposal by a embedded k-sparse auto encoder, meanwhile, it imposes a constraint to promote proposals of the same class to have similar sparse codes. These sparse codes are leveraged to yield pre-recognitions (CPVs and bounding box coordinates).
   b) The probability calibration module corrects errors in its input pre-recognitions (caused by FOV variety and/or pathological deformation) by converting them into nodes in a graphical model, and forcing label compatibility among different vertebrae via message passing.

2.1. Hierarchical proposal network (HPN)

2.1.1. The multi-scale anchor generator

The anchor generator (Fig. 3(a)) equidistantly samples grid points from the original input image; and then places boxes of different size and aspect ratio (namely, anchors) centered on the grid points. The key rules of generating the anchors are that the sizes and aspect ratios of the boxes should approximate all possible sizes and aspect ratios of vertebrae in the input images; and the sampling distance of the grid points (i.e., the distance between two neighboring anchors) should be decided to ensure that the anchors are dense enough to cover all vertebrae. Based on the *global a priori* knowledge (typical anatomic morphologies of the vertebrae in our dataset), the sizes of the anchors are chosen to be 16, 32 and 64 pixels; the aspect ratios are chosen to be 1:1, 1:1.5, and 1:2; and the sampling distance is chosen to be 8, 16, and 32 pixels. After these parameters are selected, they are not changed when tackling input MRI slices of different scales, i.e., we no longer need the *individual a priori* knowledge about the anatomic morphologies (size and aspect ratios) of vertebrae in the specific input image. Our anchor generation strategy guarantees that there is one anchor close and similar enough to it, i.e., vertebrae of all scales/aspect ratios at all possible locations can be detected by a similar-shaped anchor in its vicinity.

It should be noted that the anchor generator can be applied to more general cases where the *global a priori* knowledge is more agnostic. In our work, generating anchors of more parameter combinations will not significantly increase recognition performance. However, this may help in cases where we have no *global a priori* knowledge about the objects' sizes and shapes. For example, if the input image is of 512 $\times$ 512, the anchor sizes can be chosen as 1, 2, 4, 8, 16, ... 512 (from the possible smallest object to the largest, sampling by exponential intervals), the sampling distances 2, 4, 8, 16, Q 128 (half the anchor sizes), and the aspect ratios 1:1, 1:2, 2:1, 1:5, 5:1, 1:10, 10:1 (covering objects of different shapes from nearly square to very long-thin strips). This could still acquire high recognition performance, although the computational costs will be higher.

2.1.2. The hierarchical feature extraction module

The feature extraction module (Fig. 3(b)) adopts a Resnet-like network (which is fundamentally the same as the Resnet except for some minor modifications) with a top-down pathway. It takes the original image as input and extracts pyramid feature maps (P1 ~ P3). The detailed network structure is shown in Fig. 3(b), where "Resnet_conv2_x" means the "conv2_x" layer in Table 1 of the original Resnet paper (He et al., 2016). The HPN ground truths, i.e., the first-stage ground truth proposal labels ($p^*_{i_1}$ in Eq. (8) in Section 2.3) and ground truth bounding box corrections ($\boldsymbol{t}^*_{\boldsymbol{i_2}}$ in Eq. (8) in Section 2.3), are used to generate supervision signals. These signals are back-propagated from the sibling networks (Section 2.1.3) to train and update the Resnet-like network.

The reason for using pyramid feature maps is to explicitly combine high- and low-level features for classifying anchors of different sizes. Simply using the output features of Resnet (C3 in Fig. 3(b)) for classification may not be able to detect small objects because C3 has relatively low resolution (Lin et al., 2017). In order to process anchors of different sizes with appropriate features, we feed C3 as well as the intermediate features C1 ~ C2 to a top-down layer as shown in the right half in Fig. 3(b). In the top-down layer, the features P2 ~ P3 are up-sampled and merged with their lower level features using lateral connections. The resulting pyramid feature maps (P1 ~ P3) are of different resolutions, while all of them are semantically strong. Then, anchors are processed with features fitting their sizes (i.e., larger anchors are processed by "smaller" features of higher level) to judge whether they contain vertebrae and regress the first-stage bounding box corrections. This strategy is able to deal with vertebrae of different scales.

In our Resnet-like network, the sizes of intermediate features (C1 ~ C3) are carefully designed based on the input size and the configuration of the anchors. This design is implemented by adjusting the strides of the convolutional layers and the numbers of pooling layers. The principle of this design is, the size of a certain intermediate feature tensor (one of C1 ~ C3) should be corresponding with the number of anchors of a certain scale. For clarity, we first leave the aspect ratios out of consideration, which means that there is only one anchor of a certain scale and at a certain position. Under this circumstance, take anchors of size 16 $\times$ 16 as an example, the sample distance is set to be 8 in our work. Thus, there are 64 $\times$ 64 (512/8 $\times$ 512/8) anchors of this size in an input image of size 512 $\times$ 512. Then, we determine that a certain pyramid feature tensor (one of P1 ~ P3) should be of size "batch size $\times$ 64 $\times$ 64 $\times$ channel numbers". This determination comes from the fact that we feed the pyramid feature tensor into a 3 $\times$ 3 convolutional layer (the gray block in Fig. 3(c), whose output size is set to be "same" with input size) to calculate the weighted sums of the feature vectors in the tensor. These

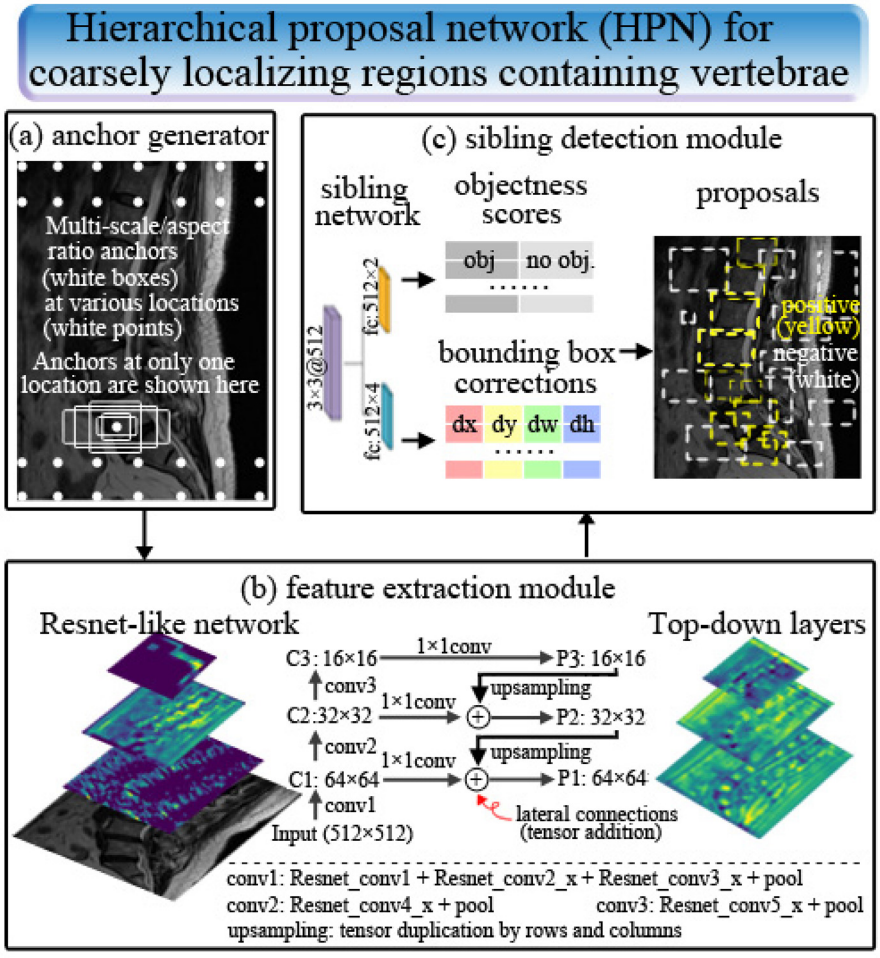

Fig. 3. The hierarchical proposal network (HPN). The anchor generator (Fig. 3(a)) places anchors of different scales/aspect ratios at different locations. The feature extraction module (Fig. 3(b)) extracts discriminative features that are robust to resolution change. The sibling detection module (Fig. 3(c)) simultaneously predicts the objectness scores and bounding box corrections (BBC_1) of each anchor, and then the anchors are refined to multi-scale proposals by BBC_1s. The configuration of anchor scales, feature scales and anchor intervals are elaborately designed to approach the multi-scale challenge.

weighted sums are fed into 1 × 1 convolutional layers for obtaining the objectness scores and BBC_1s of each anchor one by one (the orange and blue blocks in Fig. 3(c)); and the number of the weight sums is the same with the tensor size. This means that the tensor size needs to be equal to the number of anchors, i.e., this tensor should contain 64 × 64 feature vectors to handle the 64 × 64 anchors. Similarly, for anchors of size 32 × 32, we determine another pyramid feature tensor (another one of P1 ~ P3) should be of size "batch size × 32 × 32 × channel numbers"; for anchors of size 64 × 64, we determine the last tensor should be of size "batch size × 16 × 16 × channel numbers". Considering that the sizes of P2 are half of that of P1, and the sizes of P3 are half of that of P2, we decide P1 ~ P3 (and also C1 ~ C3) to be respectively of size "batch size × 64 × 64 × channel numbers", "batch size × 32 × 32 × channel numbers", and "batch size × 16 × 16 × channel numbersg, as shown in Fig. 3(b).

If we consider the issue of aspect ratios, there would be multiple anchors of a certain scale and at a certain position. For example, there are 3 aspect ratios in our work, so there are 3 anchors of a certain scale and at a certain position. However, since the receptive fields of the feature vectors are large enough to cover all 3 anchors, we can use the same feature vector to classify them, which can be implemented by simply changing the output channels of the convolutional layers in the sibling network.

Based on the above discussions, we modify the original Resnet network by keeping only the wanted intermediate feature sizes in our "Resnet-like" network. We design our "conv1" to include one convolutional layer of stride 2 (i.e., the "Resnet_conv1" as in the original Resnet paper (He et al., 2016)) and two pooling layers (the 3 × 3 max pooling layer with stride 2 in Resnet_conv_2 and the down sampling with stride 2 by Resnet_conv_3), which results in an output size of "batch size × 64 × 64 × channel number" (i.e., the desired size of C1/P1). Then, "conv2" and "conv3" are respectively designed to contain one pooling layer, which results in output sizes of "batch size × 32 × 32 × channel number" and "batch size × 16 × 16 × channel number" (the desired size of C2/P2 and C3/P3). This can easily be implemented by disabling the intermediate output of conv2_x in the original Resnet, and denoting the intermediate outputs conv3_x conv5_x as C1 ~ C3 (the left half of Fig. 3(b)). The lateral connections for enhancing hierarchical features in the bottom-up pathway and the top-down pathway are adopted as in (Lin et al., 2017) to obtain P3 ~ P1 (the right half of Fig. 3(b)). This design preserves the inherent advantages of Resnet, while retaining fewer anchor sizes and intermediate features according to our work may be beneficial to save computation. The shortcut connections can prevent degradation problem and highlight the slight changes of the network parameters (Yang et al., 2018), and thereby distinguish the subtle appearance discrepancy

of different vertebrae by changes of the parameters. Moreover, the Resnet features are proved to be robust to the resolution change of the input images.

We clarify that the feature pyramid network has been proposed in (Lin et al., 2017), and the modifications to Resnet does not change its nature, so they are NOT considered as our contribution. However, the above discussion is beneficial to future readers to have a better understanding of the concepts, principles, and usage of the object recognition workflow. The original faster RCNN papers (He et al., 2016; Lin et al., 2017) do not discuss how to choose the sizes of anchors and features, nor do they discuss the internal relationships between the two sizes. In our paper, we clarify this workflow so that even readers who have no knowledge of faster RCNN can have better understanding without the need for extracting fragmented knowledge from the original literature.

2.1.3. The sibling detection module

The sibling detection module (Fig. 3(c)) sends the ablated features P1 ~ P3 into a 3 × 3 convolutional layer and two sibling 1 × 1 convolutional layers to predict objectness scores (probabilities of containing vertebra) and first-stage bounding box corrections (BBC_1) of different anchors. Then, as inspired by Faster RCNN (Ren et al., 2015), the BBC_1s are applied to the anchors to obtain coarse detection boxes (called proposals as in (Ren et al., 2015)). Proposals with high objectness scores are locations where vertebrae probably exist.

As in our previous work (Zhao et al., 2019a), non-maximum suppression (NMS) and hard negative mining (HNM) are used to preserve negative proposals with high objectness scores (which are difficult to recognize from positive ones). For each image, a total of N proposals are preserved and fed into the succeeding network; among the N proposals, the negative proposals are 4 times (this ratio is empirically determined) more than the positive ones. This removes the "obvious" negative proposals and retains the hard ones that are difficult to recognize from positive ones. Using these hard negatives to train the succeeding CSRN helps reduce redundancy and improve its discriminability.

Summarization for Section 2.1. The motivation of our HPN is to coarsely locate regions containing vertebrae (class-agnostic proposals) of different scales and resolutions. We adopt three tightly integrated modules in HPN. These modules generate multi-scale anchors and extract hierarchical features to predict the existence of vertebrae in each anchor. The correspondence of the anchor scale and the feature scale are leveraged to tackle the scale/resolution challenge and locate vertebrae from an arbitrary input image. All weight and bias variables in HPN are trainable.

2.2. Category-consistent self-calibration recognition network (CSRN)

2.2.1. The category-consistent pre-recognition module

The overall workflow. As shown in Fig. 4(a), the category-consistent pre-recognition module simultaneously yields preliminary classification results (CPVs) and bounding box regression results by a multi-task dictionary-embedded deep network.

This module takes the proposals and the shared hierarchical features P3 ~ P1 as its input. Similar to our previous work (Zhao et al., 2019b), the category-consistent pre-recognition module also adopts ROI-pooling (Ren et al., 2015) to choose one feature from the shared P3 ~ P1 according to the feature level calculated by the size of each proposal:

$$k = \lfloor k_0 + \log_2(\sqrt{wh}/h_0) \rfloor \tag{2}$$

where w and h are the width and height of the proposal. The parameters in this equation are the same as in (Zhao et al., 2019b), except that we change k_0 to be 3, which result in the 64 × 64 vertebrae corresponding to P_3, 32 × 32 vertebrae corresponding to P_2, and 16 × 16 vertebrae corresponding to P_1. The levels for the very few proposals larger than 64 × 64 (or smaller than 16 × 16) are cut off to 3 (or 1). After feature map selection, the chosen features are cropped by the corresponding proposals, then each cropped feature is resized to 7 × 7 using bilinear interpolation (Fig. 4(a2)) for better bounding box coordinate accuracy. Next, the cropped and resized features are fed into 2 cascading convolutional layers (the first one has a kernel size of 7 × 7, and the second 1 × 1); the output channels of the second convolutional layer is 1024. Then, the dimensionalities with size 1 are squeezed so that the feature corresponding to each proposal is a vector of size 1024 (Fig. 4(a3)).

The feature vectors are then fed into two siblings branches to pre-recognize class-aware labels and bounding boxes: (1) A label-consistent dictionary learning layer that classifies each proposal by calculating its class probability vector (CPV, Fig. 4(a4)). A CPV is a vector whose elements are the probabilities of the pre-recognized vertebrae having different labels; the predicted label is the index of CPVs maximum, and the recognition confidence score is the maximum value. A category-consistent constraint is used to promote proposals with the same label to have similar sparse codes. (2) A simple fully connected layer to regress the second-stage bounding box corrections (BBC_2) for all classes for each proposal. After the classification, the BBC_2 corresponding to the predicted label is chosen and applied to the proposals to obtain the pre-recognition bounding boxes. Since the simple fully connected layer can achieve high performance for the regression task (as demonstrated in our preliminary work (Zhao et al., 2019c)), we do not use dictionary learning layer for this branch.

The dictionary learning layer integrated with CNNs. The dictionary learning layer takes the above-mentioned feature vectors (Fig. 4(a3), denoted as $\mathbf{Y}$ for the entire batch of images) as input. It outputs sparse codes $\mathbf{X}$ for succeeding classification/bounding box regression tasks. $\mathbf{Y}$ contains N_b vectors ($N_b = b \times N$, where b means the batch size, N means the number of preserved proposals after HPN in one image). For each input feature vector $\mathbf{y}_i$, its sparse code $\mathbf{x}_i$ encodes the most representative features for further tasks (such as classification). As mentioned in the introduction section, typical methods for obtaining sparse codes $\mathbf{X}$ is to alternatively solve Eq. (1), where $\mathbf{D}$ is the trainable dictionary matrix. However, the alternative training strategy of $\mathbf{D}$ and $\mathbf{X}$ makes it difficult to embed dictionary learning into deep recognition networks.

We design a dictionary learning layer that uses the k sparse auto-encoder strategy as the basic skeleton, which can be embedded into the CNN-based instance recognition framework for end-to-end training. Hereinafter, we will first detail how it is implemented, then demonstrate its rationality by analyzing its similarities and differences between classical solutions such as KSVD, and lastly, summarize its advantages.

The implementation of k sparse auto-encoder is simple: for each sample $\mathbf{y}_i$ (a row in $\mathbf{Y}$, a 1 × 1024 vector), its sparse codes $\mathbf{x}_i$ is constructed by multiplying a tied weight matrix $\mathbf{W}$ (which is the transpose matrix of the dictionary $\mathbf{D}$ as in (Makhzani and Frey, 2013)) with the input and then preserving the k largest units. The indices of these preserved units are collected and denoted as the support set $\mathcal{S}$. All other units are set to 0 to get $\mathbf{x}_i$. Then, the dictionary $\mathbf{D}$ is multiplied to $\mathbf{x}_i$ to reconstruct $\mathbf{y}_i$. Mean absolute error ($\sum_i \|\mathbf{y}_i - \mathbf{D}\mathbf{x}_i\|_2^2$) are used as the reconstruction loss. The sparse codes $\mathbf{x}_i$ is used for the succeeding classification task, namely, the most representative features are selected for classification. The k sparse auto-encoder method updates the dictionary $\mathbf{D}$ by Stochastic Gradient Descent (SGD) during back-propagation; it then updates the sparse codes $\mathbf{X}$ (as well as the support set $\mathcal{S}$) in the feed-forward phase.

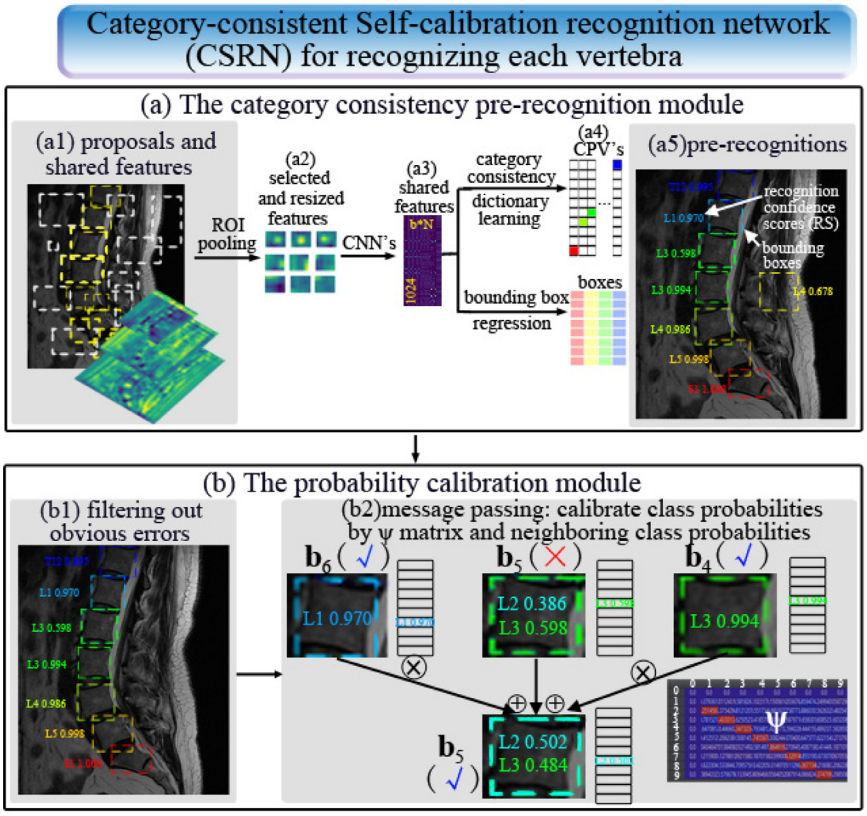

Fig. 4. The Category-consistent Self-calibration recognition network (CSRN). The category-consistent pre-recognition module (Fig. 4(a)) classifies the proposals into class-aware labels and refines proposals to bounding boxes. This module takes advantage of category-consistent dictionary learning for more discriminative features and yields pre-recognitions. The probability calibration module (Fig. 4(b)) filters out "easy" wrong recognitions and uses message passing to correct "hard" wrong detections. The mechanism of message passing is shown in Fig. 4(b2), where the CPV of one pre-recognition ($\mathbf{b_5}$ in this figure) receives messages from its neighbors ($\mathbf{b_4}$ and $\mathbf{b_6}$) via the $\mathbf{\Psi}$ matrix. The messages contain CPVs of all other pre-recognitions and helps adjust the $\mathbf{b_5}$ to reach label compatibility. This self-calibration process corrects the recognition errors caused by appearance deformation in arbitrary FOVs.

This design, although simple, can serve as an effective substitute to classical alternatively-trained dictionary learning for finding out the most representative features for each $\mathbf{y}_i$, while being able to be trained end-to-end with the deep recognition network. The rationality of integrating the k sparse auto-encoder into the CNN network is demonstrated by a detailed comparison with KSVD.

We first briefly describe the KSVD training procedure for clarity. In KSVD, $\mathbf{D}$ and $\mathbf{X}$ are updated alternately in two stages (sparse coding stage and dictionary updating stage). (1) In the sparse coding stage, $\mathbf{D}$ is assumed to be fixed, and each sparse code $\mathbf{x}_i$ are solved by minimizing the reconstruction loss $\|\mathbf{y}_i - \mathbf{D}\mathbf{x}_i\|_2^2$ with the sparsity constraint $\|\mathbf{x}_i\|_0 \leq k$ using some pursuit methods (Sulam et al., 2018) such as OMP (Pati et al., 1993). (2) In the dictionary updating stage, the dictionary is updated row by row by solving the singular value decomposition (SVD) of the error $\mathbf{E}_{j,\mathcal{S}} = \sum_j \mathbf{y}_{j,\mathcal{S}} - \mathbf{D}_{\setminus j}\mathbf{x}_{j,\mathcal{S}}$, where $\mathbf{x}_{i,\mathcal{S}}$ means every sample relevant to the jth row of $\mathbf{D}$, i.e., its ith element is not 0; $\mathbf{D}_{\setminus i}$ is acquired by setting the ith row of $\mathbf{D}$ to zero. After decomposing $\mathbf{E}_{i,\mathcal{S}}$ into $\mathbf{E}_{i,\mathcal{S}} = \mathbf{U}\mathbf{\Delta}\mathbf{V}$, the first row of $\mathbf{U}$ is used to update $\mathbf{d}_i$ (the i^{th} row of $\mathbf{D}$), the first column of $\mathbf{V}$ multiplied by $\mathbf{\Delta}(0, 0)$ is used to update $\mathbf{X}_{\mathcal{S}}$. SVD is performed for k times to update all k atoms (rows) of $\mathbf{D}$ (Aharon et al., 2006). After all columns of $\mathbf{D}$ are updated, the procedure comes back to the sparse coding stage of the next iteration.

Although the update of $\mathbf{D}$ and $\mathbf{X}$ are not the same in the two methods, they are still similar in essence. Firstly, in the sparse coding stage, as mentioned in (Papyan et al., 2017), the forward pass of CNN is equivalent to the layered thresholding algorithm, which is used to approximate the sparse codes. In other words, although we do not use the pursuit methods to obtain the exact sparse codes, the estimated ones are close to them. Actually, we have also tried to replace the forward feed process with the OMP algorithm to obtain the sparse codes, the results are no better than directly using matrix multiplication, giving strong evidence that the forward pass can yield proper sparse codes (Makhzani and Frey, 2015). Secondly, in the dictionary training stage of the k sparse auto-encoder, the update of $\mathbf{D}$ using SGD is an effective substitution of the SVD method:

$$\begin{aligned}\frac{\partial \|\mathbf{y}_i - \mathbf{D}\mathbf{x}_i\|_2^2}{\partial \mathbf{D}^T} &= 2(\mathbf{y}_i - \mathbf{D}\mathbf{x}_i)\frac{\partial \|\mathbf{y}_i - \mathbf{D}\mathbf{x}_i\|}{\partial \mathbf{D}^T} \\ &= \begin{cases} 2(\mathbf{y}_i - \mathbf{D}\mathbf{x}_i)\frac{\partial \|-\mathbf{D}\mathbf{x}_i\|}{\partial \mathbf{D}^T} & rows\ of\ \mathbf{D} \in \mathcal{S}_i \\ 0 & otherwise \end{cases} \\ &= \begin{cases} -2(\mathbf{y}_i - \mathbf{D}\mathbf{x}_i)\mathbf{x}_i^T & rows\ of\ \mathbf{D} \in \mathcal{S}_i \\ 0 & otherwise \end{cases} \end{aligned} \tag{3}$$

The derivative in Eq. (3) actually updates the dictionary twice in one back-propagation procedure. The first update on the decoder weights (the back-propagation from the output reconstruction layer to the hidden sparse coding layer) serves as an substitution of the SVD. Since the derivative can reach the preceding layers by propagating through the support set $\mathcal{S}$, it can be integrated with CNNs during training. The second update (the back-propagation from the hidden sparse codes to the input) optimizes the encoder for a better sparse code $\mathbf{X}$ in the next feed forward phase. Although one step of SGD is not able to provide the optimal dictionary, the training method of $\mathbf{D}$ is less critical in creating

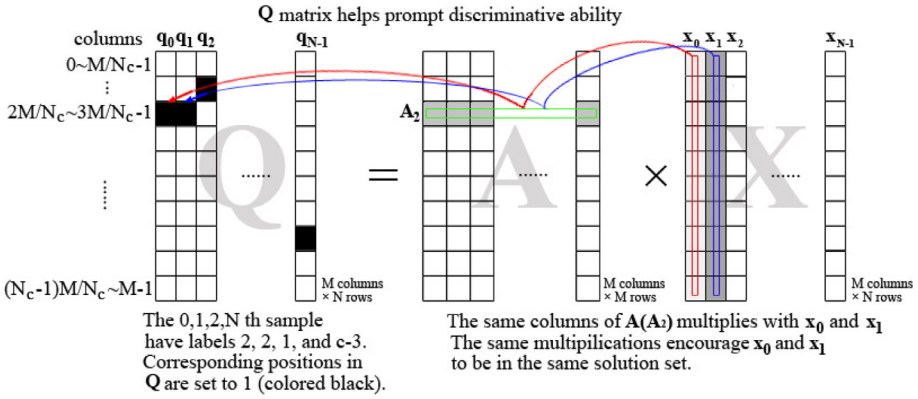

Fig. 5. The mechanism of the label-consistent strategy helping to prompt feature discriminative ability. The **Q** matrix promotes the features of samples having the same labels to be similar by forcing them into the solution set of a system of linear equations. Note that each grid means $\frac{M}{N_c}$ columns of the **A** and **Q** matrix.

a suitable dictionary (Coates and Ng, 2011). After several epochs of training, the SGD method is also able to achieve high performance.

The advantage of our design to embed k sparse auto-encoder into deep recognition network is that: (1) We use a forward pass in the deep network for obtaining **X** instead of performing L0 normalization optimization (which do not have a derivative). This helps the dictionary learning layer to be integrated into the CNN-based instance recognition network (the dictionary **D** is trained with the CNNs in an end-to-end manner). (2) The alternative solution of **X** and **D** is avoided. Actually, **X** is not considered as a trainable weight but a hidden unit in the integrated convolutional networks. (3) Label-consistent constraints (detailed in the next part) can be easily imposed on the sparse codes obtained by k sparse auto-encoder for improved discriminative ability. (4) The sparse codes **X** can be easily leveraged for the classification task as in (Jiang et al., 2013) by a linear classifier (i.e., a fully connected layer that yields the pre-recognized CPVs). The reconstruction task and the classification task can be jointly trained in a unified objective function (detailed in Section 2.3).

Label-consistent strategy for improved discriminative capability. Inspired by (Jiang et al., 2013), we impose a label-consistent constraint to the sparse codes **X** Eq. (3) on top of the above-mentioned k sparse auto-encoder. This constraint forces samples with the same labels to have similar sparse codes, which helps to improve the discriminative capability for recognizing different vertebrae:

$$L_{cc} = \sum_{i=1}^{N}\sum_{j=1}^{K}(\mathbf{Q}-\mathbf{AX})_{i,j}Q_{i,j} \tag{4}$$

where **A** is the transition matrix, **Q** is the ground truth *discriminative label* matrix, which is the key component for label consistency (the subscript *i, j* means the element of the *i*th column and *j*th row of matrix $\mathbf{Q}-\mathbf{AX}$ or **Q**).

The **Q** matrix improves label consistency by comprehensively concerning the labels of samples and dictionary atoms. **Q** is a matrix of $M \times N$; each row $\mathbf{q}_i$ ($M \times 1$ vector, $i \in [0, N-1]$) is defined as the discriminative label of each input sample $\mathbf{y}_i$; each column $\mathbf{q}_j^T$ ($1 \times N$ vector, $j \in [0, M-1]$) corresponds to an atom of the dictionary. In order to obtain **Q**, we first define the *labels of dictionary atoms* by uniformly assigning labels to all of them. If the dictionary has M rows and there are N_c labels, we assign the 0th $\sim (\frac{M}{N_c}-1)^{th}$ dictionary atoms to have label 0, $(\frac{M}{N_c})^{th} \sim (\frac{2M}{N_c}-1)^{th}$ dictionary atoms to have label 1, ... *etc*. With this in mind, we set the values of the label-consistent matrix **Q** row by row. For each row $\mathbf{q}_i$, if the sample it represents has the same label with the dictionary labels, these positions in $\mathbf{q}_i$ are set to be 1, and the other positions in $\mathbf{q}_i$ are set to be 0. For example, as shown in Fig. 5, the 0th sample corresponds to the 0th row of **Q**, and the $(\frac{2M}{N_c})^{th} \sim (\frac{3M}{N_c}-1)^{th}$ columns of $\mathbf{q}_0$ are assigned dictionary label 2, so if the 0th sample have label 2, the $(\frac{2M}{N_c})^{th} \sim (\frac{3M}{N_c}-1)^{th}$ columns in $\mathbf{q}_0$ are set to 1 (the black block in the 0th row of Fig. 5) and the others are set to 0 (the other white blocks in the row).

The mechanism that **Q** helps to prompt the discriminative ability of the dictionary is that, different elements in $\mathbf{q}_j^T$ (a column in **Q**) are approximated by multiplying the same column in **A** with different rows in **X**, i.e., the sparse codes of samples with the same labels are constrained onto the solution set of $\mathbf{A}_k\mathbf{x} = \mathbf{1}$ ($\mathbf{A}_k$ is a submatrix of **A** containing some of its columns, indicating positions where the dictionary label is the same with the sample label). For example, as shown in Fig. 5, suppose the 0th and 1st sample both have label 2, then the elements of the $(\frac{2M}{N_c})^{th} \sim (\frac{3M}{N_c}-1)^{th}$ columns in $\mathbf{q}_0$ and $\mathbf{q}_1$ are all set to 1 (black blocks in the 0th and 1st row of **Q** in Fig. 5). Since these elements are obtained by the multiplication of the $(\frac{2M}{N_c})^{th} \sim (\frac{3M}{N_c}-1)^{th}$ columns of **A** (denoted as $\mathbf{A}_2$) and the 0th and 1st rows of **X** (denoted as $\mathbf{x}_0$ and $\mathbf{x}_1$ respectively), $\mathbf{x}_0$ and $\mathbf{x}_1$ would be encouraged onto the same solution set $\mathbf{A}_2\mathbf{x} = \mathbf{1}$; and their similarity is thus increased. The resulting sparse codes can be easily transformed to the same (or very close) CPVs by the succeeding fully connected layer. In this way, the sparse codes are trained to be most discriminative in the sparse feature space.

The label-consistent strategy is easy to be trained with the instance detection CNN framework. Thanks to the *k* sparse auto-encoder, an analytical link between the sparse code and the dictionary is established, so we do not need to solve for $\frac{\partial \mathbf{x}_S}{\partial \mathbf{D}_S}$ using the complicated derivative with matrix inverse as in (Jiang et al., 2013). The discriminative sparse code error Eq. (4) can be minimized together with the classification loss and the reconstruction loss using the SGD algorithm. In this way, the label-consistent strategy is integrated into the pre-recognition network, which prompts samples of the same class to have similar sparse codes and boosts the classification performance.

Summarization. The motivation of this subsection is to pre-recognize vertebrae (classify each vertebra and regress its bounding box) from the proposals and features, and to guarantee that most pre-recognitions are correct by improved discriminate ability of similarly-appearing vertebrae in different FOV's. We develop a deep network with an embedded dictionary learning layer based on k sparse auto-encoders for this task, which enables end-to-end training with the CNN networks. We then further implement a label-consistent constraint to improve the discriminate ability. The CNN weights and biases, the dictionary matrix **D**, and the transition matrix **A** are the trainable variables in this subsection.

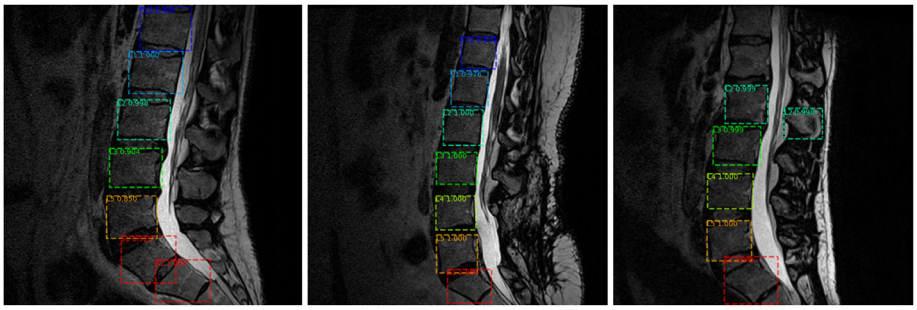

Fig. 6. Supplementary examples showing the wrongly pre-recognized cases. The appearances of the wrongly pre-recognized vertebrae are not significantly different with the correctly pre-recognized ones, however, their CPVs may be wrongly calculated because of the repetitive nature of vertebrae. This is the essence of wrong recognitions.

2.2.2. The probability calibration module

The overall workflow. As shown in Fig. 4(b), our probability calibration module effectively yields more accurate classified vertebrae and more precise bounding boxes than the pre-recognized ones through message passing.

The probability calibration module takes the pre-recognitions (the outputs of the category-consistent pre-recognition module) as input. These pre-recognitions are correct for the vast majority of the vertebrae, however, wrong pre-recognitions still happens in some images. We first analyze the reasons that lead to wrong pre-recognitions. As shown in Fig. 1(f ~ i), these wrong recognitions are mainly divided into three categories: missing detection (false negative), false positive, and wrong classification. Although these wrong recognitions seem different from each other, they are the same in essence: the class probability vectors (detailed in Section 2.2.1) are wrongly predicted. For example, the false negative in Fig. 1(f) is caused by wrongly predicting the probability of label 0 (background) to be larger than label 7 (T12), i.e., the 7th element in the CPV is wrongly predicted to be smaller than the 0th element. Also, the false positive in Fig. 1(g) is caused by wrongly predicting the probability of label 2 (L5) to be larger than label 0 (background); the wrong classification in Fig. 1(h) is caused by wrongly predicting the probability of label 4 (L2) to be larger than label 5 (L1). The appearances of the wrongly pre-recognized vertebrae are not significantly different from the correctly pre-recognized ones (as shown in Fig. 1(f) ~ (i) and also in the supplementary cases Fig. 6). The repetitive nature of vertebrae confuses different vertebrae, while the pre-recognition module independently predicts the labels of different vertebrae without leveraging the label relationship between neighboring vertebrae (e.g., it can not be aware that predicting two neighboring vertebrae to be the same label is obviously wrong). The repetitive nature and the independent recognition strategy may cause some vertebrae may be wrongly pre-recognized (i.e., their CPVs are wrongly calculated).

With this in mind, we design our message passing self-calibration module to efficiently correct the wrong recognitions. Firstly, an adaptive threshold interface is designed to remove the relatively easier false positives at wrong positions (as shown in Fig. 1(g)) by identifying their x coordinates. This procedure preserves only the pre-recognitions at approximately the correct positions and constructs an appropriate sequence of "coordinate-label (indicated by CPVs)" pairs for the succeeding message passing algorithm. The sequence is sorted using the y coordinates of the pre-recognitions and used as the input to the message passing module. Then, the message passing module is designed to carefully calibrate the CPVs of the missing recognitions and wrong recognitions at approximately the correct locations (as shown in Fig. 1(f) and (h)). This procedure leverages the label compatibility between neighboring vertebrae; it uses the CPVs of the correct pre-recognitions to calibrate the wrong ones. Since BBC_2s are predicted for each class in the pre-recognition module, the bounding boxes can be automatically refined by choosing the BBC_2 of the correct class. In other words, the BBC_2s are automatically refined by choosing the correct class label. In this way, our probability calibration module yields both calibrated CPVs and bounding boxes with much higher accuracy.

The adaptive threshold interface. The adaptive threshold interface aims to remove the "easy" false positives and sorts the remaining vertebrae to construct a sequence. The x and y coordinates of the correct vertebrae have some internal rules, i.e., if the y coordinates are sorted form small to large, the x coordinates should form a smooth curve; this curve can be either monotonic or not, but it should not contain distinct outliers even if the patient has spinal diseases that can lead to vertebrae displacement in the x direction. Thus, we can simply use the thought of removing outliers to remove these false positives with wrong x coordinates.

An adaptive threshold is intended for finding obvious outliers from the input pre-recognitions, while being insensitive to the pathological displacement of vertebrae. To find this threshold, we adopt the Savitzky-Golay filter (Savitzky and Golay, 1964) that fits each point in a curve using the weighted average of M neighboring points. This procedure does not contain any trainable variables; it processes the pre-recognitions to effectively extract the smooth x coordinates curve. Then, during training, the deviation between the x coordinates before and after SG smoothing is calculated, and the largest deviation is set as the x coordinate deviation threshold. Since the x coordinates of the bounding boxes are trained to be very close to the ground truth, the threshold represents the maximum degree of outlying that could be caused by the pathological displacements. During testing, the same smoothing procedure is implemented to the pre-recognitions; if the deviation of a pre-recognition before and after the smoothing operation is larger than the threshold, this pre-recognition is regarded as an outlier (false positive) and deleted.

It should be noted that more information (such as the y coordinates of neighboring pre-recognitions and the recognition scores) can also be used to remove false positives. This may lead to more complicated machine learning methods. In our work, simply using the x coordinates deviation threshold is enough for identifying these "easy" false positives.

Message passing module for CPV calibration. The message passing module takes the above-mentioned sorted sequences of pre-recognized CPVs as input. The sequence of an image is regarded as a chain structured graphic model; which is evolved by message passing for self-calibration of the "hard" wrong pre-recognitions. During this evolvement, messages are passed between neighboring nodes (recognitions) to calibrate the CPVs and optimize the overall label probability distribution by a label compatibility matrix $\mathbf{\Psi}$. The $\mathbf{\Psi}$ matrix is a trainable parameter of shape $N_c \times N_c$, N_c is the number of classes, an arbitrary element $\Psi(a,$

b) means the compatibility score of two consecutive nodes having labels a and b respectively. Eq. (5) theoretically reveals how messages are calculated, and how it affects the CPV of a specific vertebra via the $\mathbf{\Psi}$ matrix. Next, we first detail the physical meaning of Eq. (5) in our vertebrae recognition task; and then show how it is integrated into the CNN based network; and lastly demonstrate its role during training and testing:

$$\hat{\mathbf{b}}_\mathbf{i} = \frac{1}{Z}\mathbf{b}_\mathbf{i} \otimes \prod_{j\in\nu(i)} \mathbf{m}_\mathbf{ji}, \quad \text{where} \quad \mathbf{m}_\mathbf{ji} = \sum_{label_j\in\mathcal{L}} \mathbf{b}_\mathbf{j} \otimes \mathbf{\Psi} \otimes \prod_{k\in\nu(j)\setminus i} \mathbf{m}_\mathbf{kj} \quad (5)$$

where:

- $\hat{\mathbf{b}}_\mathbf{i}$ is the belief of the ith pre-recognition, which encodes its CPV and is updated by the element wise product of its initial value ($\mathbf{b}_\mathbf{i}$) and the message it receives from all its neighbors $\{j\}$ ($\mathbf{m}_\mathbf{ji}$). The initial state $\mathbf{b}_\mathbf{i}$ is calculated by multiplying a feature matrix $\mathbf{\Phi}$ to the pre-recognized CPVs. The notation $\prod_{j\in\nu(i)}$ means multiplying all the messages sent to node i by its neighboring nodes. Z is a normalization constant to force the CPVs' sum to 1.
- $\mathbf{m}_\mathbf{ji}$ is the message from the jth to the ith pre-recognition, which is a combination of the beliefs and the label compatibility matrix $\mathbf{\Psi}$ by element wise product ($\otimes$) of: (i) the belief of the jth pre-recognition ($\mathbf{b}_\mathbf{j}$), (ii) each row of $\mathbf{\Psi}$, and (iii) the message that flow into the jth pre-recognition from its neighbors except the ith ($\mathbf{m}_\mathbf{kj}$).
- The notation $\sum_{label_j\in\mathcal{L}}$ means summing over all label possibilities of node j, which means that all possible pathways from different labels of node j to node i are considered when calculating the message it sends to node i; thus, the message is comprehensive and is able to reflect the possibility of node i having all labels.

In order to train message passing together with the CNN-based networks, we perform the following deductions and modifications to formulate Eq. (5) into matrix addition and multiplication. These make it possible to train message passing together with the CNN-based networks:

(1) Node vicinity based on pre-recognitions

Since the pre-recognition network and the adaptive threshold interface provides pre-recognitions at approximate correct positions, the structure of the graphic model can be simplified. In the sorted pre-recognition sequences, the ith one only neighbors the $i-1$th and $i+1$th (*i.e.*, the belief $\hat{\mathbf{b}}_\mathbf{i}$ in Eq. (5) is simplified to $\frac{1}{Z}\mathbf{b}_\mathbf{i} \otimes \mathbf{m}_{\mathbf{i-1,i}} \otimes \mathbf{m}_{\mathbf{i+1,i}}$), and the messages it receives ($\mathbf{m}_{\mathbf{i-1,i}}/\mathbf{m}_{\mathbf{i+1,i}}$) are only dependent on the $i-2/i+2$th pre-recognition (*e.g.*, $\mathbf{m}_{\mathbf{i-1,i}} = \sum_{label_{i-1}\in\mathcal{L}} \mathbf{b}_{\mathbf{i-1}} \otimes \mathbf{\Psi} \otimes \mathbf{m}_{\mathbf{i-2,i-1}}$). Thus, by substituting the expression of message $\mathbf{m}$ in turn, the beliefs of all pre-recognized vertebrae are combined by the $\mathbf{\Psi}$ matrix and absorbed into $\hat{\mathbf{b}}_\mathbf{i}$.

(2) Construction of messages for formulating Eq. (5) into CNNs

We implement Eq. (5) in the logarithm domain to change the multiplication operation to the addition operation, which helps prevent arithmetic overflow. The product operation in Eq. (5) thus becomes addition, and the summing operation becomes logsumexp. Considering the above-mentioned node vicinity, Eq. (5) evolves into Eq. (6):

$$\begin{aligned}
\hat{\mathbf{b}}_\mathbf{i} &= exp(-Z + \mathbf{b}_\mathbf{i} + \mathbf{m}_{\mathbf{i-1,i}} + \mathbf{m}_{\mathbf{i+1,i}}) \\
Z &= \log \sum_{label_i\in\mathcal{L}} exp(\mathbf{b}_\mathbf{i} + \mathbf{m}_{\mathbf{i-1,i}} + \mathbf{m}_{\mathbf{i+1,i}}) \\
\mathbf{m}_{\mathbf{i-1,i}} &= \log \sum_{label_{i-1}\in\mathcal{L}} exp\left(\mathbf{b}_{\mathbf{i-1}} \oplus \mathbf{\Psi} \oplus \mathbf{m}^\mathbf{T}_{\mathbf{i-2,i-1}}\right) \\
\mathbf{m}_{\mathbf{i+1,i}} &= \log \sum_{label_{i+1}\in\mathcal{L}} exp\left(\mathbf{b}_{\mathbf{i+1}} \oplus \mathbf{\Psi} \oplus \mathbf{m}^\mathbf{T}_{\mathbf{i+2,i+1}}\right)
\end{aligned} \quad (6)$$

where $\hat{\mathbf{b}}_\mathbf{i}$ and the messages $\mathbf{m}$ are all modified into the logarithm domain. The normalization factor Z means summing up all label possibilities of node i, which forms the softmax operation in calculating $\hat{\mathbf{b}}_\mathbf{i}$. The summation of $\mathbf{b}_{\mathbf{i-1}} \oplus \mathbf{\Psi} \oplus \mathbf{m}^\mathbf{T}_{\mathbf{i-2,i-1}}$ is performed by duplicating the vectors into matrices. $\mathbf{b}_{\mathbf{i-1}}$ ($N_c \times 1$ vector) is duplicated by row and $\mathbf{m}^\mathbf{T}_{\mathbf{i-2,i-1}}$ ($1 \times N_c$ vector) is duplicated by column into $N_c \times N_c$ matrices to sum up with $\mathbf{\Psi}$. In this way, an arbitrary element in this summation (e.g., the one at the ath column and bth row) means the score of "the $i-1$th node having label a" plus the label compatibility score of "the i-1th node having label a while the ith node having label b", and the message that the $i-1$th mode receives. This is a comprehensive score concerning the class probabilities of nodes i-1 and i as well as their label compatibility. Thus, the logexpsum operation over all elements in a column means considering all possible labels of the $i-1$th node and constructs the forward message from the $i-1$th to the ith node. The backward messages are similarly constructed by the fourth equation in (5). This makes it possible to construct bi-directional messages using matrix operations of the $\mathbf{b}_\mathbf{i}$ and $\mathbf{\Psi}$.

(3) Enhancing $\mathbf{\Psi}$ matrix by joint compatibility scores

Although directly using Eq. (6) enables the training of the $\mathbf{\Phi}$ and $\mathbf{\Psi}$ matrix, we slightly modify the calculation of final CPV $\hat{\mathbf{C}}$ into Eq. (7) to enhance the function of the $\mathbf{\Psi}$ matrix.

$$\hat{\mathbf{c}}_\mathbf{i} = \sum_{label_{i+1}\in\mathcal{L}} exp\left[-Z + (\mathbf{b}_\mathbf{i} + \mathbf{b}_{\mathbf{i+1}}) \oplus \mathbf{\Psi} \oplus (\mathbf{m}_{\mathbf{i-1,i}} + \mathbf{m}_{\mathbf{i+2,i+1}})^\mathbf{T}\right] \quad (7)$$

where $(\mathbf{b}_\mathbf{i} + \mathbf{b}_{\mathbf{i+1}}) \oplus \mathbf{\Psi} \oplus (\mathbf{m}_{\mathbf{i-1,i}} + \mathbf{m}_{\mathbf{i+2,i+1}})^\mathbf{T}$ means first performing $\mathbf{b}_\mathbf{i} + \mathbf{b}_{\mathbf{i+1}}$ and $\mathbf{m}_{\mathbf{i-1,i}} + \mathbf{m}_{\mathbf{i+2,i+1}}$ separately as vectors, and then duplicating the two summations by row/columns to $N_c \times N_c$ matrices, and lastly adding up the three matrices. The summation of $\mathbf{m}_{\mathbf{i-1,i}}$ and $\mathbf{m}_{\mathbf{i+2,i+1}}$ considers all possible values of the earlier nodes (the 0th to $i-1$th of the forward message, and the $N-1$th to $i+2$th for the backward). Then, we take into consideration the compatibility of the ith and $i+1$th node by adding the $\mathbf{\Psi}$ matrix, $\mathbf{b}_\mathbf{i}$ and $\mathbf{b}_{\mathbf{i+1}}$. This results in a joint score of the ith and the $i+1$th node. Lastly, the "marginal" probability (calibrated CPV) of the ith node can be calculated by summing up the labels over the $i+1$th node.

Till now, we have converted the message passing procedure into matrix addition and logsumexp operations; and the whole procedure is summarized in Algorithm 1. After the final CPVs are calculated, the message passing loss is calculated by $\text{loss}_{\text{mp}} = -\sum_{i=1}^{N}\sum_{j=1}^{N_c} \mathbf{y}_\mathbf{i} \ln \hat{\mathbf{c}}_\mathbf{i}(j)$, where $\mathbf{y}_\mathbf{i}$ is the one-hot ground truth label of the ith vertebrae. The message passing loss is minimized together with the losses in the pre-recognition network to train $\mathbf{\Phi}$ and $\mathbf{\Psi}$ (the two trainable variables in the message passing module).

During training, the $\mathbf{\Psi}$ matrix is trained by forcing the calibrated CPVs to approach the ground truth class. During testing, the $\mathbf{\Psi}$ matrix is leveraged to calculate and calibrate the CPVs of each vertebra using its neighboring vertebrae. The $\mathbf{\Psi}$ matrix can be trained correctly only when the pre-recognitions are correct; similarly, the CPVs can be calibrated appropriately only when most of the pre-recognitions are correct and the $\mathbf{\Psi}$ matrix is properly trained. Fortunately, these can be both guaranteed by the pre-recognition network: (1) When the training of the pre-recognition network (and also the HPN) reaches stability, the training label accuracy is very high (near 100%), which provides reliable CPVs to train the $\mathbf{\Psi}$ matrix. There are no false positives or wrong predictions to perturb the training of $\mathbf{\Psi}$ matrix. (2) When using the trained $\mathbf{\Psi}$ matrix for testing, thanks to our HPN and label-consistent pre-recognize network, the majority of our pre-recognitions have acceptable CPV's with correct labels and high recognition scores. Also, filtering out the "easy" false positives (the yellow box in Fig. 4(a2)) at wrong positions helps construct a cor-

Algorithm 1: message passing.

Require: $Pre-recognition\,CPVs$: $\mathbf{C}$, $feature\,matrix$: $\mathbf{\Phi}$, $label\,compatibility\,matrix$: $\mathbf{\Psi}$
Ensure: $Calibrated\,CPVs$: $\hat{\mathbf{C}}$
1. Calculate the initial belief score matrix: $\mathbf{B} \leftarrow matrix\_multiply(\mathbf{C}, \mathbf{\Phi})$
2.Calculate bi-directional message ($\mathbf{m}_{i-1,i}$ and $\mathbf{m}_{i+1,i}$ in 6):
/* $\mathbf{b}_i$ means the ith column of $\mathbf{B}$, subscript dup means duplication of vectors into matrices*/
cumulative $\leftarrow 0$
for $i = N-2 \sim 0$
  Calculate each message:
  $\mathbf{m}_{i+1,i} \leftarrow \log \sum_{rows} exp(\mathbf{b}_{i,dup} \oplus \mathbf{\Psi} \oplus \mathbf{cumulative}^T_{dup})$
  Update cumulative message:
  $\mathbf{cumulative} += \mathbf{m}_{i+1,i}$
endfor
cumulative $\leftarrow 0$
for $i = 1 \sim N-1$
  Calculate each message:
  $\mathbf{m}_{i-1,i} \leftarrow \log \sum_{columns} exp(\mathbf{b}_{i,dup} \oplus \mathbf{\Psi} \oplus \mathbf{cumulative}^T_{dup})$
  Update cumulative message:
  $\mathbf{cumulative} += \mathbf{m}_{i-1,i}$
endfor
3.Calibrate the CPV of each node ($\hat{\mathbf{c}}_i$ in 7) using the bi-directional messages:
for $i = 0 \sim N-1$
  Calculate joint score of two nodes:
  $\mathbf{T}_i \leftarrow (\mathbf{b_i} + \mathbf{b_{i+1}})_{dup} \oplus \mathbf{\Psi} \oplus (\mathbf{m_{i-1,i}} + \mathbf{m_{i+2,i+1}})^T_{dup}$
  Calculate normalization factor:
  $Z_i = \log \sum_{all} exp(\mathbf{T}_i)$
  Calculate calibrated CPV:
  $\mathbf{C}(i,:) = \sum_{rows} exp(\mathbf{T}_i - Z_i)$
return C

rect input sequence to the message passing. Thus, the adjustments of the CPVs are in the correct direction; the undesired CPVs are calibrated using Eq. (5) (Fig. 4(b2)) and their maxima are promoted to appear at the correct indices.

Summarization. The motivation of this subsection is the self-awareness and self-calibration of the wrongly pre-recognized vertebrae. We first purpose an interface for finding "easy" wrong recognitions and converting pre-recognitions to CPV sequence, and then develop a message passing module to correct more difficult wrong pre-recognitions using the compatibility of neighboring labels. The matrices $\mathbf{\Phi}$ and $\mathbf{\Psi}$ are the trainable variables in this subsection.

Summarization for Section 2.2 (CSRN). This section contains the main contributions of our paper, which leverages the regional proposals (output of Section 2.1) for accurately recognizing different vertebrae. It proposes two cascading modules: category-consistent pre-recognition module (subsection 2.2.1) and a probability calibration module (subsection 2.2.2). The pre-recognition module encodes a dictionary learning layer, which is based on a k sparse auto-encoder with label-consistent constraints, into an end-to-end deep recognition network. These designs achieve pre-recognitions that are correct for the vast majority of the vertebrae. The calibration module includes a cascading adaptive threshold interface and a message passing module. The easy wrong pre-recognitions are first tackled by the adaptive threshold interface, and then the hard ones are corrected by passing messages among the CPV sequences of an image. The pre-recognition module and calibration module mutually benefit each other and achieves high vertebrae recognition accuracy.

2.3. Objective function and training strategy

The objective function of the detection framework includes three parts: the HPN loss L_{HPN}, the category-consistent pre-recognition loss L_{Pre}, and the message passing loss L_{mp}. These losses respectively correspond to the three terms in Eq. (8).

$$
\begin{aligned}
L &= L_{HPN} + L_{Pre} + L_{mp} \\
L_{HPN} &= \frac{\lambda_1}{N_1}\sum_{i_1=1}^{N_1} L_{cls}(\boldsymbol{p_{i_1}}, p^*_{i_1}) + \frac{\lambda_2}{N_2}\sum_{i_2=1}^{N_2} L_{loc}(\boldsymbol{t_{i_2}}, \boldsymbol{t^*_{i_2}}) \\
L_{Pre} &= \frac{\lambda_3}{N_3}\sum_{i_3=1}^{N_3} L_{cls}(\boldsymbol{q_{i_3}}, q^*_{i_3}) + \frac{\lambda_4}{N_4}\sum_{i_4=1}^{N_4} L_{loc}(\boldsymbol{u}^{q^*_{i_4}}_{\boldsymbol{i_4}}, \boldsymbol{u^*_{i_4}}) \\
&\quad + \frac{\lambda_5}{N_3}\sum_{i_5=1}^{N_3} L_{rec}(\boldsymbol{Y_{i_3,rec}}, \boldsymbol{Y^*_{i_3}}) + \frac{\lambda_6}{N_3}\sum_{i_6=1}^{N_3} L_{cc}(\boldsymbol{Q_{i_3}}, \boldsymbol{Q^*_{i_3}}) \\
L_{mp} &= \frac{\lambda_3}{N_3}\sum_{i_3=1}^{N_3} L_{cls}(\boldsymbol{q_{i_3,mp}}, q^*_{i_3})
\end{aligned}
\tag{8}
$$

L_{HPN} is the same as the RPN loss in our previous work in (Zhao et al., 2019b). $p^*_{i_1}$ is the corresponding ground truth label for each anchor on whether it contains a vertebra; $\boldsymbol{t^*_{i_2}}$ is the ground truth bounding box correction value for transforming each anchor to its corresponding ground truth bounding box. As mentioned in our previous work (Zhao et al., 2019b), $p^*_{i_1}$ and $\boldsymbol{t^*_{i_2}}$ can be derived from the manually labelled ground truth for the recognition task (i.e., $q^*_{i_3}$ and $\boldsymbol{u^*_{i_4}}$). These details will be shown in the codes which will be released in the near future.

L_{Pre} consists of four terms: the classification loss L_{cls}, the reconstruction loss L_{rec}, the category-consistent loss L_{cc}, and the bounding box loss L_{loc}. L_{cls} is the cross-entropy loss between the pre-recognized CPVs and the ground truth label. This is a multi-class classification task that aims at classifying the positive proposals into N_c vertebrae labels (different vertebrae have different labels), while classifying all negative proposals into the background label. L_{loc} is the smooth L1 loss ((Lin et al., 2017)) of the predicted box and the ground truth box for the positive recognitions. The reconstruction loss L_{rec} and the category-consistent loss L_{cc} also uses smooth L1 loss, where $\mathbf{Y^*}$ and $\mathbf{Q^*}$ are respectively the input features and the discriminative labels, $\mathbf{Y_{i_3,rec}}$ and $\mathbf{Q_{i_3}}$ are the reconstructions and the calculated discriminative labels. N3 is the total number of selected positive and negative proposals, which are the same with the classification task in L_{Pre} and detailed in (Zhao et al., 2019b).

The message passing loss L_{mp} is similar to L_{cls}, except that the predictions are acquired after the message passing procedure.

The weights ($\lambda_1 \sim \lambda_7$) in Eq. (8) are selected based on the previous experience of the detection task of our previous work (Zhao et al., 2019b) as well as the original label-consistent KSVD paper (Jiang et al., 2013). Except for the reconstruction loss λ_5, which is set to be 0.1, all other weights are set to be 1. λ_5 and λ_6 are set to be linearly decaying as a function of epoch. The reconstruction loss has a smaller weight because this task is relatively unimportant than the other tasks. Also, as reported in (Makhzani and Frey, 2013; Coates and Ng, 2011), the optimal sparse codes for classification do not exactly match those for reconstruction. Moreover, as the training proceeds, the classification loss and regression loss becomes gradually smaller than the reconstruction loss and category-consistent loss after several epochs. In order to maintain the dominance of the main losses, we set λ_5 and λ_6 to be linearly decaying following (Lee et al., 2015).

All losses can be minimized through training our integrated network Can-See. However, since the message passing algorithm requires the majority of its inputs are correct, we first train the

Table 1
Summarization for the modules in the methodology part.

Module names	Inputs	Outputs	Trainable variables	motivations	innovations
Anchor generator (2.1.1)	Input MRI scan	Anchors of different sizes and aspect ratios	None	Cover vertebrae of all possible sizes and shapes throughout the input MRI scan	None
Feature extraction module (2.1.2)	Input MRI scan	Hierarchical features corresponding to all anchors	CNN weights and biases	Extract hierarchical features for processing anchors of different sizes and shapes (tackle the scale/resolution challenge)	None
Sibling detection module (2.1.3)	Anchors and features (obtained in 2.1.1 and 2.1.2)	Proposals	CNN weights and biases	Find out anchors containing vertebrae and predict the coarse locations of vertebrae	None
Category consistent pre-recognition module (2.2.1)	Proposals and hierarchical features (obtained in 2.1.2 and 2.1.3)	Pre-recognitions (classified CPV's and bounding boxes)	CNN weights and biases; dictionary matrix **D**; transition matrix **A**	Pre-recognize vertebrae with improved discriminate ability of similarly-appearing vertebrae in different FOV's	Embedded dictionary learning layer with label-consistent constraints
Probability calibration module (2.2.2)	Pre-recognitions (obtained in 2.2.1)	Final calibrated recognitions (classified CPV's and bounding boxes)	feature matrix $\mathbf{\Phi}$; label compatibility matrix $\mathbf{\Psi}$	Calibrate the wrong pre-recognitions with self-exploitation of label relationships for further tackling the FOV/appearance challenge	Embedded message passing probability calibration

HPN and pre-recognition network for 20,000 steps (~ 111 epochs) to stabilize the training and make the training outputs reliable pre-recognition results. Then, the grading network is trained together with the detection task and adversarial module for another 10,000 steps (~ 55 epochs).

2.4. Summarization for the methodology part

We summarize the names, inputs, outputs, trainable variables, motivations, and innovations of all modules used in our work by the Table 1. In all, we recognize vertebrae from the input MRI images by firstly extracting regional proposals and hierarchical features by HPN, and then pre-recognize each proposal using the label consistent k sparse auto-encoder dictionary learning layer in the label-consistent pre-recognition module of CSRN; lastly, the CPVs of the wrong pre-recognitions are calibrated in the calibration module in CSRN. The total loss function is minimized in an end-to-end manner for vertebrae recognition.

3. Data and experiments

3.1. Data and implementation

Can-See has been intensively evaluated using a challenging dataset including 450 MRI images acquired from different medical centers. The images are of different image characteristics (such as vertebrae appearance, image resolution, intensity distribution), acquisition settings (which result in T1, T2, PD, CUBE, TSE, and STIR images) and FOV (containing S ~ T12, S ~ T11, S ~ T10, L5 ~ T11, L5 ~ T10, L4 ~ T10, each FOV has ~ 75 images). Besides, the vertebrae in the images also have different appearances because of pathological deformation. 2D slices (not necessarily the mid-sagittal slices) of each 3D MRI scan are automatically extracted (the 3D MRI scan is a 3D volume tensor, extracting a 2D matrix from the 3D tensor using operations such as "sliced = volume[:, :, t]" gives the t^{th} sagittal 2D slice of the 3D scan) and resized to 512 × 512 (which is used as the input in Fig. 2) without manual cropping. The detection ground truth is labeled on each MRI image using our lab tool according to the clinical criterion.

Can-See is implemented in Python 3.6 on Tensorflow 1.9.0 and trained using a momentum optimizer with exponential learning rate decay. The batch size is 2, the initial learning rate is 1e-3, the decay factor is 0.96 per 10 epochs, and the learning momentum is 0.9. The training is implemented on an NVIDIA GTX1080 GPU. We use the standard five-fold cross-validation for evaluation. The number of images of each FOV is kept approximately the same in the training/testing dataset in each fold. The five results from the folds are averaged to produce a single result of the different criteria.

3.2. Evaluation criteria

Extensive experiments are conducted to validate the effectiveness of our Can-See qualitatively and quantitatively.

3.2.1. Qualitative performance evaluation

In order to visually demonstrate the accuracy and robustness to image characteristics of Can-See, we choose images of different FOV, MRI acquisition settings, vertebrae appearance, vertebrae numbers, image resolution, and intensity distribution. The chosen images are from different folds in our five-fold cross-validation to verify the reproducibility of our FAR network when the training and testing data vary. The detection box and the ground truth box of the critical vertebrae are both demonstrated to prove the excellent vertebrae detection performance.

3.2.2. Quantitative performance evaluation

Four metrics are used to evaluate the detection performance:

(1) Image recognition accuracy. This is defined by the percentage of images with all its vertebrae correctly detected, i.e., the ratio $\frac{correctly\ recognized\ images}{all\ images}$. This is a rather strict metric because an image is considered as correctly recognized only if all vertebrae in the image are correctly recognized, any single false positive, false negative or wrong classification would cause the image to be regarded as wrongly recognized.

(2) Identification rate (IDR). This measures the accuracy of the individual vertebra classification, i.e., the percentage of vertebrae that have been correctly detected. This criterion has been widely used for evaluating methods that detect the centroid points (Glocker et al., 2012; Chen et al., 2015; Yang et al., 2017). A vertebra is regarded as correctly recognized only if the predicted label is the same with that of the closest ground truth centroid's label, and that the localization error is less than 20mm (Glocker et al., 2012). Although our work aims at finding out the bounding box of the vertebrae, we also calculate the centroid point coordinates using predicted boxes to compare our method with the state-of-the-art.

(3) mAP_{75}, which is a comprehensive metric that considers the precision, recall as well as the IoU (Intersection-over-union) with

the ground truth boxes of an object recognition network. This metric is used in many state-of-the-art object detection networks such as Faster RCNN (Ren et al., 2015), Mask RCNN (He et al., 2017), and YOLO (Redmon et al., 2016) for the evaluation of their performance. Detailed descriptions of this metric have been provided in section 3.3.3 of our previous work (Zhao et al., 2019b).

(4) mIoU, which is the average IoU of the detection and ground truth bounding boxes of all vertebrae in all images.

3.3. Intra- and linter-comparison experiments

3.3.1. Ablation experiments for intra-comparison

Ablation experiments following the same five-fold cross-validation protocol are carried out to respectively prove the necessity of the dictionary learning layer and the self-calibration module. First, the dictionary learning layer is removed (annotated as "without dictionary") as a comparison experiment to demonstrate the importance of the label-consistent discriminative features. Second, the self-calibration module is removed (annotated as "without self-calibration") to prove the strengths of the message passing algorithm to leverage the label compatibility. Third, both the dictionary learning layer and the self-calibration module are removed (annotated as "baseline") for proving the abilities of the detection framework, and also the necessity of integrating the proposed modules.

3.3.2. Inter-comparison experiments

Inter-comparison experiments concerning four other state-of-the-art methods are carried out to demonstrate the strengths of Can-See. The four methods are originally proposed in (Liao et al., 2018; Yang et al., 2017; Ren et al., 2015; Zhao et al., 2019b) with necessary modifications. For example, the method in (Zhao et al., 2019b) is originally designed for vertebrae recognition and spondylolisthesis grading using a recognition network with GAN. We modify it by deleting the grading branch and using the same feature extraction network with the present work (which helps to compare the performance of GAN and message passing in correcting wrong recognitions). Also, since some of these methods only detect the centroid points of each vertebra, they can not be evaluated by the mAP_{75} and mIoU metrics. We also calculate the centroid point location error of our method to compare it with these methods. Since only (Ren et al., 2015) has published their code, we make our best effort to re-implement (Yang et al., 2017) and a 2D version of (Liao et al., 2018) and adjust their hyper-parameters for better performance. All these four methods are trained with the same batch size and training steps, and they are performed with the same five-fold cross-validation protocol.

4. Results and discussion

4.1. Comprehensive analysis

4.1.1. Qualitative evaluation results of can-See

Fig. 7 demonstrates that Can-See achieves high vertebrae classification accuracy and bounding box localization precision. Typical qualitative results are shown in Fig. 7, which are of different FOV, acquisition settings, vertebrae appearance, vertebrae numbers, and image resolution from different folds in our five-fold cross-validation. For example, Fig. 7(a) is a T1 sequence containing S ~ T12, Fig. 7(b) is a low-resolution STIR (Short-TI Inversion Recovery) sequence containing S ~ T10, Fig. 7(c) is a T1-FSE (Fast Spin Echo) sequence containing L5 ~ T11 with L1 pathologically deformed, Fig. 7(d) is a T1-TSE(Turbo spin echo) sequence containing L4 ~ T10, which is difficult to distinguish with images containing L5 ~ T11. Despite all these challenges, Can-See achieves high performance; the detected boxes (dashed) have correct labels and high overlaps with their ground truth boxes (solid). This means that Can-See is robust to changes in image characteristics and able to achieve high and reproducible performance when the training and testing data varies.

4.1.2. Quantitative evaluation results of can-See

Fig. 8 demonstrates the high performance of Can-See evaluated by all four metrics.

(1) The black bars mean image recognition accuracy. The first black bar is the average image recognition accuracy for all tested images, and the following black bars are those for individual FOVs. It is seen that the first bar reaches 0.955 ± 0.024, which means that all the vertebrae are correctly recognized in 95.5% of the input images without any false positive/negatives. For individual FOVs, image recognition accuracies are respectively 0.981 ± 0.017, 0.967 ± 0.028, 0.960 ± 0.039, 0.970 ± 0.030, 0.967 ± 0.022, and 0.889 ± 0.086 for FOVs S ~ T10, S ~ T11, S ~ T12, L5 ~ T10, L5 ~ T11, and L4 ~ T10. Even for the most difficult FOV (L4 ~ T10, which tends to be confused with FOV L5 ~ T11 without message passing), the image recognition accuracy is still as high as 0.889. This high accuracy can be attributed to the fact that discriminative features in the pre-recognition module help distinguish different FOVs, and the message passing module effectively calibrates the wrong pre-recognitions.

(2) The red bars mean the identification rate (IDR) for individual vertebra. Similar to the black bars, the first red bar means the IDR of all images, while the rest mean those for individual FOVs. The mean IDR reaches 0.974 ± 0.022 and shows a high classification performance for individual vertebra. For different FOVs, the IDRs are generally larger than 0.95. For the most difficult FOV L4 ~ T10, the IDR is still high, which demonstrates the robustness of Can-See to FOV changes. The standard deviations over different folds are small, which again shows the robustness to changes in the training and testing data. The IDRs are slightly higher than the corresponding image recognition accuracies, which means that IDR is a relatively easier metric for recognition accuracy.

(3) The blue bars mean the mAP_{75} averaged among different tested images. The first blue bar shows an average mAP_{75} of 0.972 ± 0.019, which comprehensively shows excellent recognition accuracy and precise vertebrae bounding boxes locations. Vertebrae of different classes are detected and correctly classified in almost all images. For different FOVs, the mAP_{75}s are generally larger than 0.94, which shows that vertebrae are correctly classified regardless of FOVs, even if evaluated under a relatively high IoU threshold (IoU $\geq$ 0.75), i.e., the recognized vertebrae's bounding boxes overlap well with the ground truth boxes with the correct label.

(4) The pink bars mean the average mIoU reaches 0.928 ± 0.006, i.e., the recognized bounding boxes have high overlaps with the ground truths. Similar to the above metrics, the mIoUs are high for different FOVs, which again shows that the performance of Can-See is robust to FOV changes.

The ability of our Can-See tackling this challenge may be attributed to the mutual beneficial effect between the pre-recognition module and the probability calibration module. On one hand, the pre-recognition module shows strong ability to capture representative features (such as vertebrae appearance and orientations) for classifying vertebrae with distinguishable characteristics. Furthermore, the label consistent dictionary learning strategy can improve this classification performance by forcing vertebrae of the

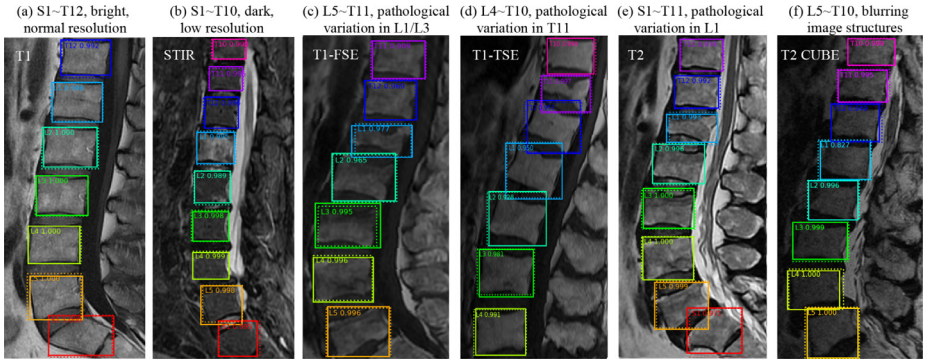

Fig. 7. Can-See achieves high vertebrae detection performance on a challenging dataset of different FOV, image characteristics and acquisition settings. The dotted boxes are the detection boxes with confidence scores, and the solid boxes are ground truth boxes.

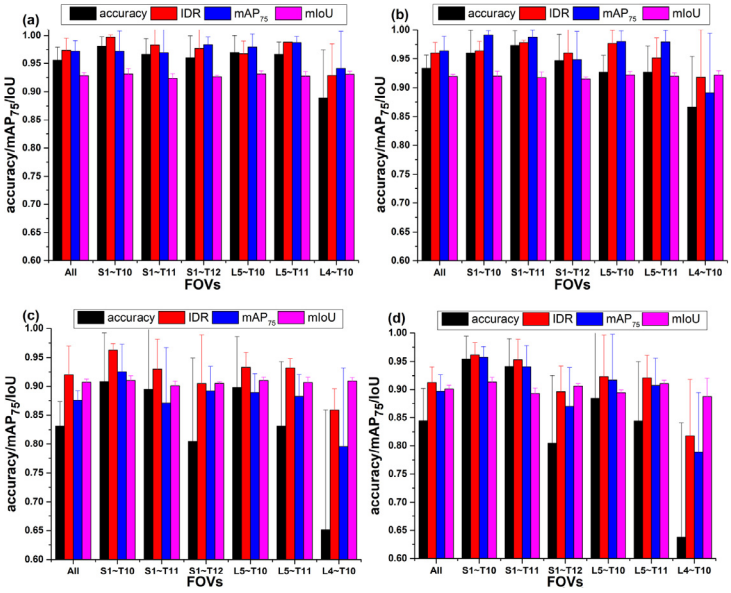

Fig. 8. Four evaluation metrics indicate that Can-See can accurately classify and localize vertebrae in images of different FOVs, image characteristics, and vertebrae appearances. Fig. 8(a) ~ (d) means the evaluation results of the four metrics for Can-See, Hi-scene(Zhao et al., 2019c)(ablation experiment without dictionary learning layer), ablation experiment without message passing, and the baseline method (ablation experiment without either module). The black, red, blue and pink bars respectively mean the image recognition accuracy, IDR, mAP_{75}, and mIoU for different FOVs. By comparing fig. 8(a) with (c) (and (b) with (d)), it can be seen that the message passing significantly enhances the performance of all metrics. By comparing fig. 8(c) with (d), it is seen that the dictionary learning layer does not improve the image recognition accuracy without message passing, but it improves the IDR; it also helps reduce wrong recognitions with high recognition scores which are intractable for the message passing to self-calibrate. In this way, the dictionary helps enhance the overall performance when the message passing is present (as shown by a comparison between Fig. 8(a) with (b)). (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

same label in different images to have closer features, while vertebrae of different labels in the same images to have farther features. On the other hand, our probability calibration module effectively exploits the label relationship between the different vertebrae for improved recognition performance. It leverages the classification results of the distinguishable vertebrae to help classify those who do not have distinguishable characteristics by exploiting the label relationships using message passing.

For example, the top vertebrae may be T11 in one image, and T10 in another; the appearances of T11 and T10 may be similar, i.e., these two vertebrae do not have distinguishable characteristics. However, some other vertebrae (such as L1, L5, and S) in the image may have better distinguishable characteristics, e.g., S has a particular appearance, L1 is at the transition regions of different vertebral sections, and the centerline orientation of L5 begins to show greater deviation with its upper vertebrae (as shown in Fig. 6a, the centerline orientation of L5 leans left compared with those of other vertebrae). These distinguishable vertebrae can be recognized with high confidence (i.e., their recognition scores are high as mentioned in Section 2.2). These high recognition scores are leveraged to recognize T10 and T11 by the message passing module, e.g., if the third (or fourth) vertebrae shows strong evidence of being L1, then the top vertebrae could probably be T11 (or T10).

The high recognition performance of our Can-See is beneficial for clinical diagnosis of different spinal diseases such as diskitis, disk narrowing, and compression fracture. For example, Can-See automatically presents that L1 vertebra in Fig. 7(c) and T11 vertebra in Fig. 7(d) has abnormal aspect ratios, which means that these vertebrae may be suffering from “vertebra plana”, which means flattened vertebra and may be caused by compression fractures. In terms of treatment planning, our Can-See indicates that back bracing can be applied to this patient to provide external support for limiting the motion of fractured vertebrae. A short period of bed rest could be applied for pain management, however, prolonged inactivity should be avoided because it may raise the risk for future

Table 2
Ablation experiments demonstrating the effect of category-consistent strategy and the probability calibration module. The message passing algorithm in the probability calibration module helps enhance the performance by calibrating the wrong CPVs, while the category-consistent strategy in the pre-recognition module helps mitigate the intractable wrong recognitions and pave way for the message passing.

Method	Loc-Err (pixel)	image accuracy	IDR	mAP_{75}	mIoU
Can-See	4.872 ± 2.846	0.955 ± 0.094	0.974 ± 0.022	0.972 ± 0.019	0.928 ± 0.052
Without probability calibration module	5.972 ± 4.076	0.831 ± 0.112	0.920 ± 0.050	0.875 ± 0.076	0.906 ± 0.085
Without dictionary learning layer (Hi-scene)	5.199 ± 3.063	0.933 ± 0.124	0.963 ± 0.036	0.964 ± 0.025	0.919 ± 0.036
Baseline method	6.136 ± 4.108	0.844 ± 0.168	0.912 ± 0.068	0.866 ± 0.085	0.874 ± 0.155

compression fractures. If the patient feels that the chronic pain persists, surgical procedures such as vertebroplasty, kyphoplasty and spinal fusion surgery may be needed. In all, although the variability of image characteristics in arbitrary MRI images leads to unusual difficulties, Can-see can acquire accurate recognition results.

4.2. Comparative experiments

4.2.1. Intra-comparison experiments

As shown in Table 2, the category-consistent strategy in the pre-recognition module and the probability calibration module both contribute to superior performance of recognition accuracy. As a reference, our Can-See on average achieves 0.955 ± 0.024 image recognition accuracy and 0.974 ± 0.022 testing IDR (first row in Table 2).

(1) After only preserving the convolutional neural networks in HPN and CSRN (both the category-consistent strategy and the probability calibration module are removed) as the baseline method, the testing image recognition accuracy decreases to 0.844 ± 0.168 and testing IDR decreases to 0.912 ± 0.068 (fourth row in Table 2). This not only demonstrates the effectiveness of our designed category-consistent pre-recognition module and probability calibration module but also proves that the baseline recognition network is capable of extracting correct features and distinguishing subtle feature discrepancies of different vertebrae with similar appearances. This discrimination ability helps the majority of the recognized vertebrae to be correct, which lays the foundation for using the category consistency and probability calibration strategies for further improvements.

(2) If the probability calibration module is removed, only the category-consistent dictionary learning layer is functioning compared with the baseline method. In this case, the testing image recognition accuracy is 0.831 ± 0.112 and testing IDR is 0.920 ± 0.050 (second row in Table 2). It is interesting that although the dictionary learning layer helps improve the overall performance in Can-See, this improvement is not significant without the probability calibration module. Neither image recognition accuracy nor IDR shows significant improvements compared with the baseline method (in some cases there are even declines) when the category-consistent dictionary learning layer is used alone. Interestingly, the mechanism that dictionary learning layer benefits Can-See lies in the fact that it reduces the number of intractable wrong cases (for example, an image where six out of seven vertebrae are wrongly predicted, which is hard to calibrate). The number of wrong recognitions with high recognition scores is decreased. Although the dictionary learning layer results in somewhat more missing detections, they are mostly because their recognition scores do not reach some threshold (i.e., 0.8 in our experiments). This can be easily calibrated by the message passing algorithm, namely, the dictionary learning layer helps improve the overall performance by paving the way for the successive probability calibration module.

(3) If the category-consistent dictionary learning layer is removed, only the message passing algorithm is functioning compared with the baseline method (which is the case in our preliminarily work Hi-scene (Zhao et al., 2019c)). In this case, the testing image recognition accuracy is 0.933 ± 0.124 and testing IDR is 0.963 ± 0.036 (third row in Table 2). On one hand, it is shown that the message passing algorithm significantly increases the image recognition accuracy. There are adequate convolutional layers in the baseline method, which are capable of capturing most of the vertebrae in the image; however, there might be a few wrong recognitions in each image because of the similarity of different vertebrae, which results in a relatively high IDR but low image recognition accuracy. The message passing algorithm can effectively calibrate the CPVs of the wrong pre-recognitions using those of the right ones, which results in a significant rise of the image recognition accuracy. On the other hand, comparing with Can-See, it is also demonstrated that the category-consistent strategy in the pre-recognition module plays a role in the recognition performance enhancement. It helps the features of vertebrae with the same labels to be closer to each other by constraining them into the solution set of the same liner inhomogeneous equation systems. Without message passing, the benefits of the dictionary learning layer seem not significant; however, the cooperation with message passing makes it a feasible way for increased recognition accuracy.

Conclusively, Can-See achieves higher image recognition accuracy and IDR than its ablated versions. The combination of the category-consistent strategy and probability calibration module contributes to accurate vertebrae recognition from arbitrary MRI images.

4.2.2. Inter-comparison experiments

As mentioned in Section 3.3.2, four powerful methods are used to perform vertebrae recognition. The results in Table 3 show that: (1) Our Can-See outperforms the state-of-the-art methods on the same datasets in terms of all four metrics used. Comparing with the state-of-the-art classification networks, Can-See shows significant advantages by an average of ~ 9% image recognition accuracy. (2) The message passing method is more beneficial than GAN in vertebrae recognition. (3) HPN and CSRN in Can-See reinforce the mutual benefit between each other for better recognition performance.

The second column in Table 3 reveals the results of a method using fully connected network (FCN) and LSTM. As mentioned in the (Liao et al., 2018), long-range contextual information, which concerns the sequential order of the vertebrae similar to our message passing method, is considered using the LSTM. LSTM and message passing both leverages the label relationship of different vertebrae. The advantage of our work is that the pre-recognition module and the adaptive threshold method effectively eliminates

Table 3
Comparison with the state-of-the-art. The comparison shows Can-See reinforces the mutual benefit between HPN and CSRN; it achieves high vertebrae recognition performance and outperforms the other methods (FCN-LSTMLiao et al. (2018), DI2INYang et al. (2017), Faster-RCNNRen et al. (2015), and FAR Zhao et al. (2019b)).

Method	Loc-Err (pixel)	image accuracy	IDR	mAP_{75}	mIoU
Can-See	4.872 ± 2.846	0.955 ± 0.024	0.974 ± 0.022	0.972 ± 0.019	0.928 ± 0.052
FCN-LSTM	7.923 ± 4.788	0.727 ± 0.231	0.816 ± 0.188	–	–
DI2IN	6.891 ± 5.056	0.835 ± 0.183	0.904 ± 0.179	–	–
Faster-RCNN	7.124 ± 3.259	0.831 ± 0.106	0.884 ± 0.098	0.829 ± 0.130	0.844 ± 0.162
FAR	6.387 ± 3.643	0.878 ± 0.135	0.931 ± 0.108	0.938 ± 0.156	0.910 ± 0.177

the false positives, which forms reliable input for the succeeding label calibration procedures (LSTM or message passing). Otherwise, the unreliable inputs would harm the performance of label calibration when they are fed into it.

The third column in Table 3 reveals the results of a method using a U-net-like network and message passing. According to (Yang et al., 2017), the U-net like network is used to finding the centroid points of each vertebra; and then message passing is leveraged for response enhancement (i.e., correcting false negatives). Although this work and our work both use message passing, our work achieves a better performance because our HPN and pre-recognition network in SRN successfully handles images of arbitrary scales/characteristics/FOVs. The pre-recognized vertebrae are more precise and contains fewer false positives compared with the predicted pixel-wise probability maps for vertebrae centroid, which gives play to the message passing algorithm to pass the correct probabilities for better CPV calibration performance. In other words, the HPN and the CSRN in Can-See reinforce the mutual benefit between each other.

The fourth column in Table 3 is the recognition results of Faster-RCNN, which is similar to the baseline method in the ablation experiments. This method, after fine-tuning the hyper-parameters, may have similar results with the baseline method. The IDR is relatively high; however, since this method does not leverage the relationship of different recognitions, there might still be a few wrong recognitions in some images, which affects the overall image recognition accuracy especially for the difficult FOVs such as L4 ~ T10 and L5 ~ T10.

The comparison between the first and last column in Table 3 shows the advantage of message passing to GAN (Goodfellow et al., 2014). GAN is a popular novel algorithm for enhancing label compatibility by implicitly leveraging the internal higher-order potentials of the vertebrae coordinates. The discriminator of GAN uses the label relationships of neighboring recognitions to distinguish ground truth recognitions from predicted ones obtained from the generator, i.e., the recognition network. This strategy should prompt the generator to yield more reliable recognitions, however, to our surprise, we found that although we have tried different training configurations (e.g., when to start training the discriminator, the learning rate configuration, the number of layers in the discriminator), the recognition results by GAN is far less satisfactory compared with those using message passing. This may be due to the mechanism of GANs. GANs uses convolutional layers to extract image features and judge whether its input is ground truth or those generated by the generator. This strategy helps to generate images with the same distribution of input data (input images), however, it mimics the overall distribution of the input image to generate diversified images. Whereas, in the current task, the outputs of the generator are the coordinates (instead of the pixel intensities), which do not need diversified results with the same overall distribution (instead, it requires individual coordinates to be accurate). Moreover, GAN implicitly learns the distributions of the coordinates without an explicit definition, which may raise a problem that GAN may not be able to automatically learn the most needed distribution knowledge for correctly revealing the relationships to monitor the generator. On the contrary, it may learn some redundant information, which does not benefit the recognition performance. Also, it is demonstrated by many pieces of literature that GAN is difficult to train and may suffer from convergence problems (Hjelm et al., 2017; Gulrajani et al., 2017; Zhang et al., 2018). Thus, GAN might not be suitable for the current recognition task; although it may help improve the recognition performance to some extent, it is sub-optimal for this task. As a comparison, our message passing can leverage the label relationships of neighboring vertebrae, which directly calibrates the CPVs and results in a better performance with a lower computational cost.

5. Conclusion

In this paper, we develop a Category-Consistent Self-calibration Detection Framework (Can-See) to recognize vertebrae in arbitrary spine MRI images. It consists of two novel networks: (1) A hierarchical proposal network for perceiving the existence of vertebrae of arbitrary scale/aspect ratio; and (2) A category-consistent self-calibration recognition network for pre-recognizing the label and bounding box of each vertebra and automatically correcting wrong pre-recognitions caused by FOV variety and pathological deformations. Its performance and effectiveness are demonstrated by extensive experiments. Codes will be released in the near future after cleaning up some details, and readers are welcome to ask for the codes in advance.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

CRediT authorship contribution statement

Shen Zhao: Conceptualization, Methodology, Software, Writing - original draft. **Xi Wu:** Data curation, Formal analysis. **Bo Chen:** Data curation, Formal analysis. **Shuo Li:** Supervision, Methodology, Project administration.

References

Aharon, M., Elad, M., Bruckstein, A., 2006. K-Svd: an algorithm for designing overcomplete dictionaries for sparse representation. IEEE Trans. Signal Process. 54 (11), 4311–4322.
Arnab, A., Jayasumana, S., Zheng, S., Torr, P.H., 2016. Higher Order Conditional Random Fields in Deep Neural Networks. In: European Conference on Computer Vision. Springer, pp. 524–540.
Ben-Ari, R., Akselrod-Ballin, A., Karlinsky, L., Hashoul, S., 2017. Domain Specific Convolutional Neural Nets for Detection of Architectural Distortion in Mammograms. In: Biomedical Imaging (ISBI 2017), 2017 IEEE 14th International Symposium on. IEEE, pp. 552–556.
Chambolle, A., De Vore, R.A., Lee, N.-Y., Lucier, B.J., 1998. Nonlinear wavelet image processing: variational problems, compression, and noise removal through wavelet shrinkage. IEEE Trans. Image Process. 7 (3), 319–335.
Chen, H., Shen, C., Qin, J., Ni, D., Shi, L., Cheng, J.C., Heng, P.A., 2015. Automatic Localization and Identification of Vertebrae in Spine Ct via a Joint Learning

Model with Deep Neural Networks. In: International Conference on Medical Image Computing and Computer-Assisted Intervention. Springer, pp. 515–522.
Coates, A., Ng, A.Y., 2011. The Importance of Encoding deldDelversus delilns delilt versus Training with Sparse Coding and Vector Quantization. In: Proceedings of the 28th international conference on machine learning (ICML-11), pp. 921–928.
Gao, Z., Xiong, H., Liu, X., Zhang, H., Ghista, D., Wu, W., Li, S., 2017. Robust estimation of carotid artery wall motion using the elasticity-based state-space approach. Med Image Anal 37, 1–21.
Glocker, B., Feulner, J., Criminisi, A., Haynor, D.R., Konukoglu, E., 2012. Automatic Localization and Identification of Vertebrae in Arbitrary Field-of-view Ct Scans. In: International Conference on Medical Image Computing and Computer-Assisted Intervention. Springer, pp. 590–598.
Glocker, B., Zikic, D., Konukoglu, E., Haynor, D.R., Criminisi, A., 2013. Vertebrae Localization in Pathological Spine Ct via Dense Classification from Sparse Annotations. In: International Conference on Medical Image Computing and Computer-Assisted Intervention. Springer, pp. 262–270.
Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., Bengio, Y., 2014. Generative Adversarial Nets. In: Advances in neural information processing systems, pp. 2672–2680.
Gulrajani, I., Ahmed, F., Arjovsky, M., Dumoulin, V., Courville, A.C., 2017. Improved Training of Wasserstein Gans. In: Advances in neural information processing systems, pp. 5767–5777.
He, K., Gkioxari, G., Dollár, P., Girshick, R., 2017. Mask R-cnn. in: Computer Vision (ICCV). In: 2017 IEEE International Conference on. IEEE, pp. 2980–2988.
He, K., Zhang, X., Ren, S., Sun, J., 2016. Deep Residual Learning for Image Recognition. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 770–778.
Hjelm, R.D., Jacob, A.P., Che, T., Trischler, A., Cho, K., Bengio, Y., 2017. Boundary-seeking generative adversarial networks. arXiv preprint arXiv:1702.08431.
Jiang, Z., Lin, Z., Davis, L.S., 2013. Label consistent k-svd: learning a discriminative dictionary for recognition. IEEE Trans Pattern Anal Mach Intell 35 (11), 2651–2664.
Kamalakannan, S., Gururajan, A., Sari-Sarraf, H., Long, R., Antani, S., 2010. Double-edge detection of radiographic lumbar vertebrae images using pressurized open dgvf snakes. IEEE Trans. Biomed. Eng. 57 (6), 1325–1334.
Lee, C.-Y., Xie, S., Gallagher, P., Zhang, Z., Tu, Z., 2015. Deeply-supervised Nets. In: Artificial Intelligence and Statistics, pp. 562–570.
Li, J., Liang, X., Wei, Y., Xu, T., Feng, J., Yan, S., 2017. Perceptual Generative Adversarial Networks for Small Object Detection. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 1222–1230.
Liao, H., Mesfin, A., Luo, J., 2018. Joint vertebrae identification and localization in spinal ct images by combining short-and long-range contextual information. IEEE Trans Med Imaging 37 (5), 1266–1275.
Lin, T.-Y., Dollár, P., Girshick, R., He, K., Hariharan, B., Belongie, S., 2017. Feature Pyramid Networks for Object Detection. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 2117–2125.
Liu, W., Anguelov, D., Erhan, D., Szegedy, C., Reed, S., Fu, C.-Y., Berg, A.C., 2016. Ssd: Single Shot Multibox Detector. In: European conference on computer vision. Springer, pp. 21–37.
Liu, Y., Chen, Q., Chen, W., Wassell, I., 2018. Dictionary Learning Inspired Deep Network for Scene Recognition. In: Thirty-Second AAAI Conference on Artificial Intelligence..
Lootus, M., Kadir, T., Zisserman, A., 2014. Vertebrae Detection and Labelling in Lumbar Mr Images. In: Computational methods and clinical applications for spine imaging. Springer, pp. 219–230.
Luc, P., Couprie, C., Chintala, S., Verbeek, J., 2016. Semantic segmentation using adversarial networks. arXiv preprint arXiv:1611.08408.
Makhzani, A., Frey, B., 2013. K-Sparse autoencoders. arXiv preprint arXiv:1312.5663.
Makhzani, A., Frey, B.J., 2015. Winner-take-all Autoencoders. In: Advances in neural information processing systems, pp. 2791–2799.
Papyan, V., Romano, Y., Elad, M., 2017. Convolutional neural networks analyzed via convolutional sparse coding. The Journal of Machine Learning Research 18 (1), 2887–2938.
Pati, Y.C., Rezaiifar, R., Krishnaprasad, P.S., 1993. Orthogonal Matching Pursuit: Recursive Function Approximation with Applications to Wavelet Decomposition. In: Proceedings of 27th Asilomar conference on signals, systems and computers. IEEE, pp. 40–44.
Radford, A., Metz, L., Chintala, S., 2015. Unsupervised representation learning with deep convolutional generative adversarial networks. arXiv preprint arXiv:1511.06434.
Redmon, J., Divvala, S., Girshick, R., Farhadi, A., 2016. You Only Look Once: Unified, Real-time Object Detection. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 779–788.
Ren, S., He, K., Girshick, R., Sun, J., 2015. Faster R-cnn: Towards Real-time Object Detection with Region Proposal Networks. In: Advances in neural information processing systems, pp. 91–99.
Savitzky, A., Golay, M.J., 1964. Smoothing and differentiation of data by simplified least squares procedures. Anal. Chem. 36 (8), 1627–1639.
Sharma, P., Abrol, V., Sao, A.K., Sharma, P., Abrol, V., Sao, A.K., 2017. Deep-sparse-representation-based features for speech recognition. IEEE/ACM Transactions on Audio, Speech and Language Processing (TASLP) 25 (11), 2162–2175.
Stahel, P.F., Mauffrey, C., 2014. Patient safety in surgery. Springer.
Sulam, J., Papyan, V., Romano, Y., Elad, M., 2018. Multilayer convolutional sparse modeling: pursuit and dictionary learning. IEEE Trans. Signal Process. 66 (15), 4090–4104.
Tibshirani, R., 1996. Regression shrinkage and selection via the lasso. Journal of the Royal Statistical Society: Series B (Methodological) 58 (1), 267–288.
Yang, D., Xiong, T., Xu, D., Huang, Q., Liu, D., Zhou, S.K., Xu, Z., Park, J., Chen, M., Tran, T.D., et al., 2017. Automatic Vertebra Labeling in Large-scale 3D Ct Using Deep Image-to-image Network with Message Passing and Sparsity Regularization. In: International Conference on Information Processing in Medical Imaging. Springer, pp. 633–644.
Yang, Z., Luo, T., Wang, D., Hu, Z., Gao, J., Wang, L., 2018. Learning to Navigate for Fine-grained Classification. In: Proceedings of the European Conference on Computer Vision (ECCV), pp. 420–435.
Yedidia, J.S., Freeman, W.T., Weiss, Y., 2003. Understanding belief propagation and its generalizations. Exploring artificial intelligence in the new millennium 8, 236–239.
Zhang, J., Shu, Y., Xu, S., Cao, G., Zhong, F., Liu, M., Qin, X., 2018. Sparsely Grouped Multi-task Generative Adversarial Networks for Facial Attribute Manipulation. In: 2018 ACM Multimedia Conference on Multimedia Conference. ACM, pp. 392–401.
Zhao, R., Liao, W., Zou, B., Chen, Z., Li, S., 2019a. Weakly-supervised simultaneous evidence identification and segmentation for automated glaucoma diagnosis.
Zhao, S., Gao, Z., Zhang, H., Xie, Y., Luo, J., Ghista, D., Wei, Z., Bi, X., Xiong, H., Xu, C., et al., 2017. Robust segmentation of intima–media borders with different morphologies and dynamics during the cardiac cycle. IEEE J Biomed Health Inform 22 (5), 1571–1582.
Zhao, S., Wu, X., Chen, B., Li, S., 2019b. Automatic spondylolisthesis grading from mris across modalities using faster adversarial recognition network. Medical Image Analysis. 101533
Zhao, S., Wu, X., Chen, B., Li, S., 2019c. Automatic Vertebrae Recognition from Arbitrary Spine Mri Images by a Hierarchical Self-calibration Detection Framework. In: International Conference on Medical Image Computing and Computer-Assisted Intervention. Springer, pp. 316–325.
Zhao, Z.-Q., Zheng, P., Xu, S.-t., Wu, X., 2018. Object detection with deep learning: a review. arXiv preprint arXiv:1807.05511.
Zhou, Y., Tuzel, O., 2018. Voxelnet: End-to-end Learning for Point Cloud Based 3D Object Detection. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 4490–4499.