Multi-Instance Deep Learning: Discover Discriminative Local Anatomies for Bodypart Recognition

Zhennan Yan, Yiqiang Zhan*, Zhigang Peng, Shu Liao, Yoshihisa Shinagawa, Shaoting Zhang, *Senior Member, IEEE*, Dimitris N. Metaxas, *Fellow, IEEE*, and Xiang Sean Zhou

***Abstract*—In general image recognition problems, discriminative information often lies in local image patches. For example, most human identity information exists in the image patches containing human faces. The same situation stays in medical images as well. "Bodypart identity" of a transversal slice—which bodypart the slice comes from—is often indicated by local image information, e.g., a cardiac slice and an aorta arch slice are only differentiated by the mediastinum region. In this work, we design a multi-stage deep learning framework for image classification and apply it on bodypart recognition. Specifically, the proposed framework aims at: 1) discover the local regions that are discriminative and non-informative to the image classification problem, and 2) learn a image-level classifier based on these local regions. We achieve these two tasks by the two stages of learning scheme, respectively. In the pre-train stage, a convolutional neural network (CNN) is learned in a multi-instance learning fashion to extract the most discriminative and and non-informative local patches from the training slices. In the boosting stage, the pre-learned CNN is further boosted by these local patches for image classification. The CNN learned by exploiting the discriminative local appearances becomes more accurate than those learned from global image context. The key hallmark of our method is that it automatically discovers the discriminative and non-informative local patches through multi-instance deep learning. Thus, no manual annotation is required. Our method is validated on a synthetic dataset and a large scale CT dataset. It achieves better performances than state-of-the-art approaches, including the standard deep CNN.**

***Index Terms*—CNN, discriminative local information discovery, multi-instance, multi-stage.**

I. Introduction

OVER the course of the recent decades, more and more automatic image analysis algorithms have been developed to assist clinicians in the interpretation and assessment of medical images. Some of the algorithms are designed for fundamental image analysis tasks, such as anatomical landmark detection and organ segmentation, while the others are implemented for comprehensive computer-aided-diagnosis (CAD) systems. Since different organ systems have highly diverse characteristics, medical image analysis methods/models are often designed/trained for specific anatomies to incorporate prior knowledge, e.g., organ shape [1]–[4]. To benefit real-world clinical workflows, these algorithms are desired to be invoked automatically for applicable datasets. Therefore, it is important to automatically identify the human bodypart contained in the medical image in the first place. However, compared to the extensively investigated organ segmentation and landmark detection topics, automatic bodypart recognition (identify the human bodypart contained in the medical image) is still less explored.

In fact, auto-bodypart recognition algorithm may benefit radiological workflow in different aspects. For example, the current imaging workflow requires the planning of the scanning range in topogram or scout scans. With a very reliable and fast bodypart recognition algorithm, this planning step may be conducted on-the-fly to significantly save scanning time. Another example is the bodypart-based query in PACS system. Since the bodypart information in DICOM header is not very reliable [5], an automatic bodypart recognition will enable content-based image retrieval and improve the retrieval precision. Besides the aforementioned "direct" benefits, bodypart recognition algorithm also paves the way to other higher level medical image interpretation tasks. Bodypart recognition can serve as an initialization module for anatomy detection or segmentation algorithms. Given the bodypart information, the search range of the following detection/segmentation algorithms can be reduced, hence, the algorithm speed and robustness are improved. Moreover, with the availability of more and more intelligent medical image analysis algorithms, radiologists hope that medical images could have been "pre-processed" by all applicable auto-algorithms before being loaded for manual reading. In this way, the automatic results can be displayed instantaneously in the reading room to speed up the reading process. In this scenario, a robust bodypart recognition algorithm again becomes important to gate the intelligent algorithms properly for meaningful results.

It is worth noting that although DICOM header includes bodypart information, text-based retrieval methods still face

Manuscript received December 27, 2015; accepted February 01, 2016. Date of publication February 03, 2016; date of current version April 29, 2016. This work of Zhennan Yan was mainly accomplished during an internship in Siemens Healthcare. *Asterisk indicates corresponding author.*

Z. Yan and D. N. Metaxas are with the Department of Computer Science, Rutgers University, Piscataway, NJ 08854 USA (e-mail: zhennany@cs.rutgers.edu; dnm@cs.rutgers.edu).

*Y. Zhan is with the Siemens Healthcare, Malvern, PA 19355 USA (e-mail: yiqiang.zhan@siemens.com

Z. Peng, S. Liao, Y. Shinagawa, and X. S. Zhou are with the Siemens Healthcare, Malvern, PA 19355 USA.

S. Zhang is with the Department of Computer Science, University of North Carolina, Charlotte, NC 28223 USA (e-mail: szhang16@uncc.edu).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TMI.2016.2524985

0278-0062 © 2016 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission.
See http://www.ieee.org/publications_standards/publications/rights/index.html for more information.

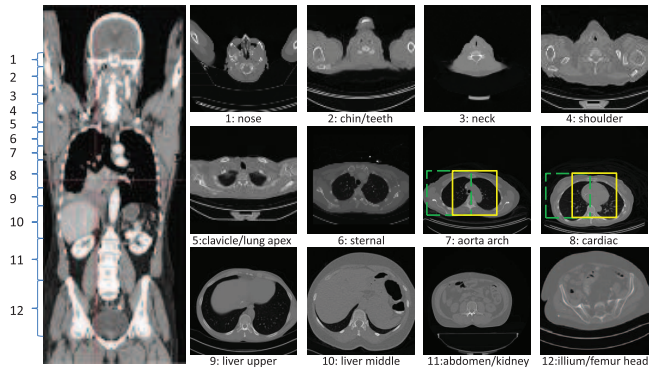

Fig. 1. Definition of body sections. Human body is divided into 12 continuous parts. Each parts may cover different ranges due to the variability of anatomies.

three major challenges. First, it may contain around 15% errors [5] and thereby limit the accuracy of text-based bodypart recognition. Second, text information in DICOM is highly abstract and may not precisely describe the anatomies contained in the scan. For example, it is difficult to tell if a scan with DICOM tag (0018,0015)="TSPINE" includes the mid-part of the liver. In addition, the multi-language nature of DICOM bodypart information becomes another barrier for text-based retrieval. On the contrary, a reliable image-based bodypart recognition algorithm can tackle all these three challenges by leveraging the intrinsic anatomical appearance information.

CT and MR are two common forms of medical imaging scans. A CT/MR sequence is usually a 3D volume image consisted of a series of 2D slices. This paper focuses on the bodypart identification in a 2D transversal slice, namely "slice-based bodypart recognition". Specifically, we divide human body into continuous sections according to anatomical context as shown in Fig. 1. Given a 2D transversal slice, the task of a slice-based bodypart recognition algorithm is to identify which section the slice belongs to. Although 3D volume always contain more comprehensive anatomy information, based on which the bodypart recognition can be more accurate, this study only aims at slice-based bodypart recognition for two reasons. First, in some real-world systems, 3D volume is not always accessible. For example, in a client-server application, the server end might only receive the 3D volume data slice-by-slice due to the limited network speed but need to output bodypart information instantaneously. Second, 2D slice-based bodypart recognition provides the foundation of 3D bodypart identification. Given the bodypart identities of all slices of a 3D volume, the 3D bodypart can be straightforwardly derived.

Slice-based bodypart recognition is essentially a multi-class image classification problem, which has been extensively studied for decades. In general, image classification algorithms consist of feature extraction and classification modules. Based on the different design principles of these two modules, various image classification algorithms can be categorized into three groups. The first group uses carefully hand-crafted features followed by classifiers without feature selection capability, e.g., SVM and logistic regression [6], [7]. The second group extends the feature set to a feature pool derived from some feature basis, e.g., Haar mother functions. Since the feature pool often includes thousands of features, the following classification modules need to have the feature selection capability, e.g., Adaboost [8], random forest [9]. The third group comes from the latest achievements of the deep learning research. Instead of designing any features, those algorithms [10], [11] aim to learn both the features and classifiers jointly from the data. As the features are learned for specific image classification tasks, they often have more discriminative power, hence, achieve better classification performance than those ad-hoc designed ones.

In slice-based bodypart recognition, it is difficult to "design" common features that work well for different body parts, due to diverse appearances in different body sections and large variability between subjects. Thus, deep learning technology, which learns features and classifiers simultaneously, becomes a promising solution. However, slice-based bodypart recognition has its unique challenge which might not be solved by standard deep learning. As shown in Fig. 1, although image 7 and 8 come from aorta arch and cardiac sections, respectively, their global appearance characteristics are quite similar. For these two slices, the discriminative information only resides in the local mediastinum region (indicated by the yellow boxes). The rest areas are just "non-informative" for classification. Although the standard deep learning framework is able to learn some low-level and abstract features from global context, it cannot learn local patches that are most discriminative for bodypart recognition. The "non-informative" regions here may mislead the classifier to recognize these two sections as identical. Hence, the classification power of the learned deep network may be limited. In fact, this problem also exists in general image classification/recognition applications. For example, in face recognition, Taigman *et al.* [12] shows that deep learning can show its power only after the face (the local region of interest) is properly localized. However, while face is a well defined object and can be detected by mature algorithms, the discriminative local regions for bodypart recognition are not easy to define, not to mention that the effort to build these local detectors might be quite large.

In summary, two key questions need to be answered to tackle the challenge of the slice-based bodypart recognition. First, which local regions are discriminative or non-informative for bodypart recognition? Second, how to learn the local bodypart identifiers without time-consuming manual annotations? We answer these questions using a multi-stage deep learning scheme. In the pre-train stage, a convolutional neural network (CNN) is learned in a multi-instance learning fashion to "discover" the most discriminative local patches. Specifically, each slice is divided into several local patches. The deep network thus receives a set of labeled slices (bags), each containing multiple local patches (instances). The loss function of the CNN is adapted in a way that as long as one local patch (instance) is correctly labeled, the class of corresponding slice (bag) is considered to be correct. In this way, the pre-trained CNN will be more sensitive to the discriminative local patches than others. Based on the responses of the pre-trained CNNs, discriminative and non-informative local patches are selected to further boost the pre-trained CNN. This is the second stage (namely boosting stage) of our learning scheme. At run-time, a sliding window approach is employed to apply the boosted CNN to the

subject image. As the CNN only has peaky responses on the discriminative local patches, it essentially identifies bodypart by focusing on the most distinctive local information and discarding non-informative local regions. Thanks to its ability to "discover" local discriminative local patches, this method is expected to identify bodypart more accurately and robust than global image context-based approaches.

The major contributions of this work include: 1. A multi-stage deep learning strategy is proposed to identify anatomical body parts by using discriminative local information; 2. The propsoed method does not require annotations of the discriminative local patches. Instead, it automatically *discovers* these local patches through multi-instance deep learning. Thus, our solution becomes highly scalable. 3. We validate our method on a large number of synthetic images and 7000+ CT slices. In both experiments, it shows superior performance than state-of-the-art methods.

II. Related Work

In this section, we will review relevant studies in two categories. First, we will review different bodypart recognition methods proposed in medical image analysis domain. Second, we will also review representative image recognition methods in general computer vision community, since slice-based bodypart recognition is technically similar to these problems.

Several bodypart recognition systems in medical imaging domain has been introduced in this decade. Park *et al.* [13] proposed an algorithm to determine the body parts using energy information from Wavelet Transform. Look-up tables are designed to classify the imaging modality and body parts. Hong *et al.* [14] proposed a framework to identify different body parts from a whole-body scan. The method starts from establishing global reference frame and the head location. After determining the bounding box of the head, other body parts including neck, thorax cage, abdomen and pelvis, are localized one by one using different algorithms. In general, these approaches employ ad-hoc designed features and algorithms to identify major body parts which have globally variant appearances. Recently, more learning-based approaches are proposed for bodypart recognition. All of these methods essentially resort to the detection of specific organs or landmarks. In [15], Zhan *et al.* trained multiple, organ-specific classifiers and optimize the schedule of organ detections based on information theory. In [16], Criminisi *et al.* utilized regression forests for anatomy detection and localization and obtained better accuracy than their classification approach in [17]. In [18], Donner *et al.* also trained regressor for anatomical landmark detection. However, since these organ/landmark-based recognition methods rely on a number of organ/landmark detectors, large manual annotation efforts are required in the training stage.

Technically, slice-based bodypart recognition is an image classification problem, which has been extensively studied in computer vision and machine learning communities. In general, existing image classification methods can be categorized into two groups, global information-based and local information-based. Global information-based approaches extract features from the whole image. Conventional approaches often rely on carefully designed/selected features, e.g., gist [19], SIFT [20], Histogram of Oriented Gradients (HOG) [21] and their variants. These features are extracted on either dense grids or a few interested points, organized as bag of words to provide statistical summary of the spatial scene layouts without any object segmentation [22]. The framework of global representations followed by classical classifiers has been widely used in scene recognition [22] and image classification [23], [24]. With the latest advances of machine learning technology, deep learning based algorithms [10], [25] have shown their superior in these tasks due to the ability of learning expressive nonlinear features and classifier simultaneously.

Roth *et al.* [26] presented a method for anatomy-specific classification of medical images using deep convolutional networks (ConvNets). They applied a trained deep CNN on 2D axial CT images to classify 5 bodyparts (neck, lungs, liver, pelvis and legs) and obtained the state-of-the-art accuracy (5.9% error). Their results demonstrated the power of deep learning in bodypart recognition. However, real-world applications may require a finer grained differentiation beyond these 5 bodyparts, e.g., aortic arch vs cardiac sections. Due to the globally-similar and locally-different appearance characteristics of these body sections, the CNNs trained on the whole axial images may not be able to differentiate them effectively.

Global information-based approaches achieved good performances in some image classification problems. However, it is not sufficient or appropriate to recognize images whose characteristics are exhibited by local objects, e.g., jumbled image recognition [27], multi-label image classification [11], [28], and scene classification [29], [30]. On the contrary, local information-based approaches can achieve better performance here. In [31], Szegedy *et al.* utilized CNN for local object detection and recognition and achieves state-of-the-art performance on Pascal VOC database. However, the training stage requires manually annotated object bounding boxes, which is often time consuming. To avoid explicit local region or object annotation, different approaches have been emerged. Felzenszwalb *et al.* [32] designed a part-based deformable model using local information of object parts for object recognition and detection. They assumed a star-structured model for object, then treated the object's part locations as latent variables during training. In [33], Singh *et al.* used unsupervised clustering method and one-vs-all linear SVM to train classifier for each cluster to discover the discriminative patches which can be used as visual words in spatial pyramid based classification. In another pioneer work, Wei *et al.* [11] applied an existing general objectness detection (BING [34]) to produce some candidate local windows from a given image, which are used to do multi-label. Recently, several studies have emerged to apply multi-instance learning (MIL) [31], [35]–[37] combined with CNN to better utilize local information in weakly supervised classification or segmentation tasks. For example, Wu *et al.* proposed a deep multi-instance learning framwork in a weakly supervised setting for image classification and auto-annotation based on object and keyword proposals [38]. The object proposals are generated by existing methods, e.g., BING [34], and the keyword proposals are crawled from web texts using Baidu image search engine. Pinheiro *et al.* combined CNN and MIL to do pixel labeling in [39]. They built their segmentation

network over the already trained Overfeat [40] and achieved better performance on Pascal VOC dataset than other weakly supervised methods. Papandreou *et al.* proposed a semantic image segmentation method in weakly or semi-supervised setting (with bouding boxes or image-level labels) [41]. They combined CNN with a conditional random field (CRF) [42] in a EM algorithm to inference the pixel-level labels. Hou *et al.* extended the EM based label inference method and combine the patch-level classifiers to predict image-level label in gigapixel resolution image classification [43].

Inspired by the latest advances in deep learning research, we propose a multi-instance multi-stage deep learning framework to recognize bodypart in CT slices. Compared to other learning-based approaches, our method only requires "weak" supervision at a global level, i.e., bodypart labels at image level. Our framework is able to discover the discriminative "local" information automatically. In this way, the annotation efforts in the training stage are dramatically reduced. This is in particular meaningful for medical image applications, since the annotations in medical images often require clinical expertise and high cost. This work is extended from our IPMI paper [44] by fine-tuning the framework and re-organizing data sets for more extensive evaluations to explore the benefits and limitations of our proposed method.

III. Methodology

We design a multi-stage deep learning framework to discover local discriminative patches and build local classifiers. We start this section by the problem statement and notation definitions. The first learning stage is introduced in Section III-B, which aims to learn representative local image features in a supervised multi-instance fashion. Then we describe the second learning stage in Section III-C, in which some discriminative and non-informative local patches are extracted from images and used to boost learning to obtain a patch-based classifier for image recognition. In Section III-D, we discuss the run-time image classification strategy using the learned CNN model. At last, we show details of the implementation in Section III-E.

A. Problem Statement

Definitions: Slice-based bodypart recognition is a typical multi-class image classification problem for a learning algorithm. Denote $\mathbf{X}$ as the input slice/image, K as the number of body sections (classes), and $l \in \{1, \ldots, K\}$ as the corresponding class label of $\mathbf{X}$. The learning algorithm aims to find a function $\mathcal{O} : \mathbf{X} \rightarrow l$. In traditional image classification frameworks, $\mathcal{O}$ is often defined as $\mathbb{C}(\mathbb{F}(\mathbf{X}))$, where $\mathbb{F}(\mathbf{X})$ and $\mathbb{C}(.)$ denote the feature extractors and classifiers, respectively.

In the context of convolutional neural network (CNN), $\mathcal{O}$ becomes a multi-layer neural network. An example of standard CNN is shown in Fig. 2, it has two convolutional layers (C1, C3), each followed by a max-pooling layer (S2, S4), one fully connected hidden layer (H5) receiving outputs of the last pooling layer, and one logistic regression (LR) layer (O6) as the output layer. In CNN, $\mathbb{F}(\mathbf{X})$ becomes multiple nonlinear layers, which aim to extract image features in a local-to-global fashion. $\mathbb{C}(.)$ is implemented by the LR layer, whose output is a K-dimension vector $R(k), k \in \{1, \ldots, K\}$ representing

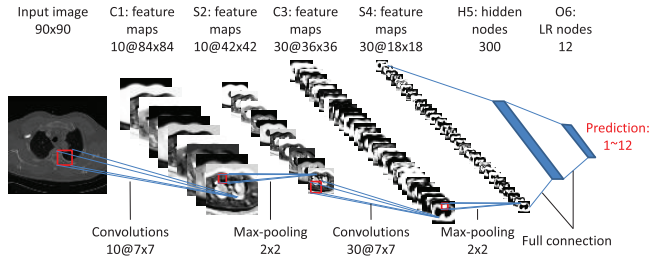

Fig. 2. Illustration of one standard CNN architecture and the outputs of each layer (similar to LeNet [45]).

the probability of $\mathbf{X}$ belonging to each class k. Mathematically, $R(k)$ can be described as a conditional probability as $R(k) = \mathbf{P}(k|\mathbf{X}; \mathbf{W})$. Here, $\mathbf{W}$ denote the CNN coefficients, which include the weights of convolutional filters, hidden nodes, LR nodes, as well as the bias vectors. The final predicted label l is determined by the argument of the maximum element (class with the highest probability) in R.

Given a set of training images $\mathcal{T} = \{\mathbf{X}_m, m = 1, \ldots, M\}$, with corresponding discrete labels $l_m \in \{1, \ldots, K\}$, the training algorithm of CNN aims to minimize the loss function:

$$L_1(\mathbf{W}) = \sum_{\mathbf{X}_m \in \mathcal{T}} -\log(\mathbf{P}(l_m|\mathbf{X}_m; \mathbf{W})) \qquad (1)$$

where $\mathbf{P}(l_m|\mathbf{X}_m; \mathbf{W})$ indicates the probability of image $\mathbf{X}_m$ being correctly classified as class l_m using network coefficients $\mathbf{W}$.

CNN has shown impressive performance in image classification tasks [10], [11]. The success shall be attributed to its capability of modeling complex nonlinear functions and leveraging the context information of neighboring pixels. In these successful applications, standard CNN is conducted as a *global* learning scheme, which takes the entire image as input. However, in slice-based bodypart recognition, distinctive information often comes from *local* patches (as shown in Fig. 1) and these local patches are distributed "inconsistently" at different positions of the slices. The intrinsic conflicts between "global learning scheme" and "local distinctive information" may limit the performance of standard CNN in bodypart recognition. (One may argue that CNN can still learn local features through its convolutional layers. However, this situation only holds while local features always appear at the similar location across different images, which is not the case of bodypart recognition.) We design a toy example to illustrate this problem. As shown in Fig. 3(a), we randomly position and combine 4 types of geometry elements, square, circle, triangle and diamond to synthesize two classes of binary images. While circle and diamond are allowed to appear in any classes, triangle and square are exclusively owned by Class1 and Class2, respectively (ref Section IV-A for more details). Using standard CNN that takes the whole image as input, the classification accuracy is $\sim 83\%$ (row "SCNN" of Table I(a)). It implies that standard CNN does not discover and learn the discriminative local patches: "triangle" and "square". (Otherwise, the accuracy shall be much higher due to the significant differences between "triangle" and "square".) This problem will become trivial if we have the prior knowledge of the discriminative local patches

TABLE I
CLASSIFICATION ACCURACIES ON SYNTHETIC DATA IN TERMS OF RECALL, PRECISION AND F1 SCORE (%).

	(a) Triangle and square									(b) Diamond and circle								
	Recall			Precision			F1			Recall			Precision			F1		
Class	1	2	Total	1	2	Total	1	2	Total	1	2	Total	1	2	Total	1	2	Total
LR	78.7	83.4	81.1	82.6	79.7	81.1	80.6	81.5	81.1	70.3	62.5	66.4	65.2	67.8	66.5	67.7	65.0	66.5
SVM	84.5	81.2	82.9	81.8	84.0	82.9	83.1	82.6	82.9	69.0	63.1	66.1	65.2	67.1	66.1	67.0	65.0	66.1
SCNN	84.2	82.4	83.3	82.7	83.9	83.3	83.5	83.2	83.3	91.8	94.2	93	94.1	92.0	93.0	92.9	93.1	93.0
PCNN	99.6	99.7	99.7	99.7	99.6	99.7	99.7	99.7	99.7	99.6	**100**	99.8	**100**	99.6	99.8	99.8	99.8	99.8
BCNN1	98.4	99.7	99.1	99.7	98.4	99.1	99.0	99.1	99.1	95.6	**100**	97.8	**100**	95.8	97.9	97.8	97.9	97.9
BCNN2	**100**	**100**	**100**	**100**	**99.9**	**100**	**100**	**100**	**100**	**99.9**	99.9	**99.9**	99.9	**99.9**	**99.9**	**99.9**	**99.9**	**99.9**

and build local classifiers on them. However, in bodypart recognition, it is not easy to figure out the most discriminative local patches for different body sections. In addition, even with *adhoc* knowledge, annotating local patches and training local classifiers often takes large effort. The solution thus becomes non-scalable when body-sections are re-defined or the imaging modalities are changed.

To leverage the local information, more important, to automatically "discover" the discriminative local patches for different body sections, we design a two-stage CNN learning framework. It consists of pre-train and boosting stages, which will be detailed next.

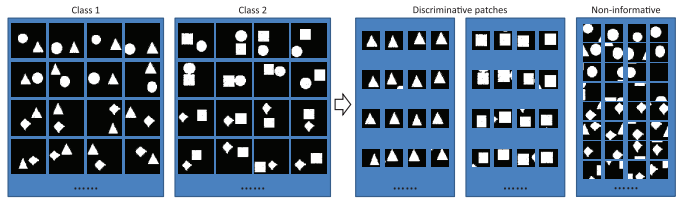

Fig. 3. A synthetic toy example. (a) Synthetic images of two classes. (b) The discriminative and non-informative local patches selected by the pre-trained CNN model. Note that we never "tell" the algorithm that these two classes are differentiable by triangle and square.

B. Learning Stage I: Multi-Instance CNN Pre-Train

In order to exploit the local information, CNN should take *discriminative* local patches instead of the entire slice as its input. Here, the key problem is how to automatically *discover* these local patches through learning. This is the major task of the first stage of our CNN learning framework. A multi-instance learning strategy is designed to achieve this goal.

Given a training set $\mathcal{T} = \{\mathbf{X}_m, m = 1, \ldots, M\}$ with corresponding labels l_m. Each training image, $\mathbf{X}_m$, is divided into a set of local patches defined as $\mathcal{L}(X_m) = \{\mathbf{x}_{mn}, n = 1, \ldots, N\}$. These local patches become the basic training samples of the CNN and their labels are inherited from the original images, i.e., all $\mathbf{x}_{mn} \in \mathcal{L}(X_m)$ share the same label l_m. While the structure of CNN is still the same as the standard one, the loss function is adapted as:

$$L_2(\mathbf{W}) = \sum_{X_m \in \mathcal{T}} -\log(\max_{\mathbf{x}_{mn} \in \mathcal{L}(\mathbf{X}_m)} \mathbf{P}(l_m|\mathbf{x}_{mn}; \mathbf{W})), \quad (2)$$

where $\mathbf{P}(l_m|\mathbf{x}_{mn}; \mathbf{W})$ is the probability that the local patch $\mathbf{x}_{mn}$ is correctly classified as l_m using CNN coefficients $\mathbf{W}$.

The new loss function is different from (1) by adopting a multi-instance learning criterion. Here, each original training slice $\mathbf{X}_m$ is treated as a bag consisting of multiple instances (local patches), $\{\mathbf{x}_{mn}\}$. Within each bag (slice), only the instance with the highest probability to be correctly classified is counted in the loss function. Such instance is considered as the most discriminative local patch of the image slice. Let R_{mn} be the output vector of the CNN on local patch $\mathbf{x}_{mn}$. The l_mth component of R_{mn} represents the probability of $\mathbf{x}_{mn}$ being correctly classified. As illustrated in Fig. 4, for each training image $\mathbf{X}_m$, only the local patch that has the highest response at the l_mth component of R_{mn} (indicated by the yellow and purple boxes for two training images, respectively), contributes to the loss function and drives the update of network coefficients $\mathbf{W}$ during the backward propagation. Accordingly, the learned CNN is expected to have high responses on discriminative local patches. In other words, the most discriminative local patches for each image class are automatically *discovered* after the CNN training. Fig. 3(b) shows the discovered discriminative patches (containing triangle or square) for the image classification task in toy example. This is exactly in accordance to the fact that these two classes are only distinguishable by "triangle" and "square". It proves that our method is able to *discover* the key local patches without manual annotation.

To ensure that the learned CNN will have stable high responses on discriminative local patches, a spatial continuity factor is further incorporated into the loss function as:

$$L_3(\mathbf{W}) = \sum_{X_m \in \mathcal{T}} -\log(\max_{\mathbf{x}_{mn} \in \mathcal{L}(\mathbf{X}_m)} \sum_{\mathbf{x} \in \mathfrak{N}(\mathbf{x}_{mn})} \mathbf{P}(l_m|\mathbf{x}; \mathbf{W})). \quad (3)$$

Here, $\mathfrak{N}(\mathbf{x}_{mn})$ denotes the local patches in the neighborhood of $\mathbf{x}_{mn}$. Based on (3), for each training slice, the local patch to be counted in the loss function is not the most *individually* discriminative one (i.e., with the highest probability of being correctly classified), but the one whose neighboring patches and itself are *overall* most discriminative. In this way, the selected discriminative local patches will be robust to image translation and artifacts.

C. Learning Stage II: CNN Boosting

In the second stage of our learning framework, the main task is to boost the pre-trained CNN using selected local patches, which is illustrated in Fig. 5.

The first type of selected local patches are the discriminative ones, i.e., these local patches on which the pre-trained CNN

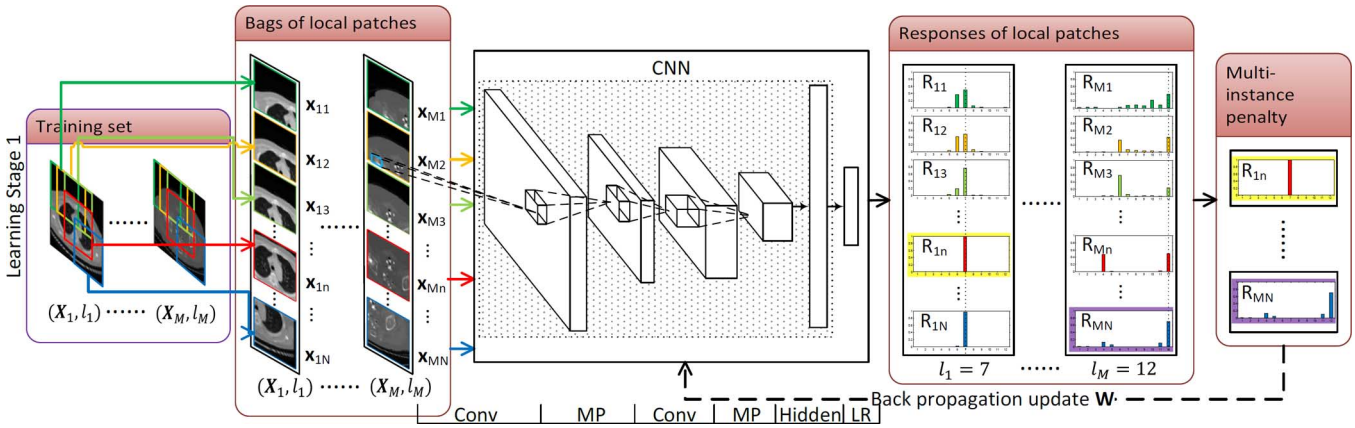

Fig. 4. Illustration of pre-train learning stage.

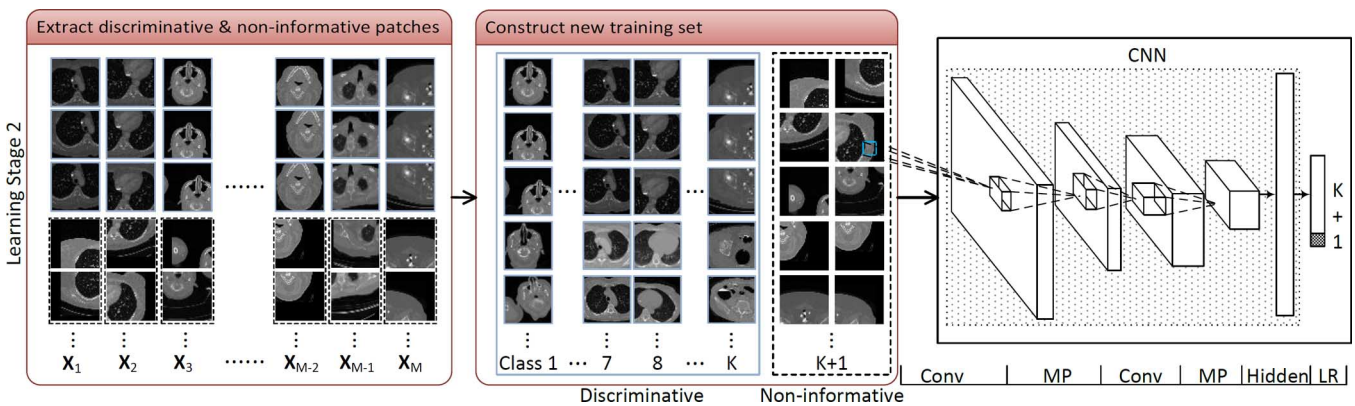

Fig. 5. Illustration of boosting learning stage.

have high responses at the corresponding classes. For each image $\mathbf{X}_m$, we select D discriminative local patches as:

$$\mathbf{A}_m = \underset{\mathbf{x}_{mn} \in \mathcal{L}(\mathbf{X}_m)}{\arg\max_{\mathrm{D}}} \mathbf{P}(l_m|\mathbf{x}_{mn}; \hat{\mathbf{W}}). \tag{4}$$

Here, $\hat{\mathbf{W}}$ is the coefficients of the pre-trained CNN. $\mathbf{P}(l_m|\mathbf{x}_{mn}; \hat{\mathbf{W}})$ denotes the response of the pre-trained CNN on the local patch $\mathbf{x}_{mn}$ corresponding to the correct class l_m. $\arg\max_{\mathrm{D}}(.)$ is the operator that returns the arguments of the largest D elements.

We noticed that apart from the discriminative local patches, the remaining regions cannot be completely ignored in the boosting stage for two reasons. First, only selecting discriminative patches to boost classifier may lead to overfitting problems. Second, some "confusing" local patches may mislead the bodypart recognition. For example, the patches containing lung regions (green dashed boxes in Fig. 1) appear in both aortic arch and cardiac sections. For these "confusing" patches, CNN may generate similarly high responses for both aortic arch and cardiac classes. (Note that since the pre-trained CNN is only ensured to correctly classify one local patch per slice, the responses of the remaining patches are not guaranteed.) At run-time, when CNN is applied to the confusing patches, the high responses on multiple classes may induce wrong bodypart identification. Therefore, the algorithm should select these "confusing" regions as the second type of local patches in boosting stage to suppress their responses for *all* classes (body sections).

To this end, we introduce a new "non-informative" class (patches in dashed box in Fig. 5) besides the existing training classes. This class includes two kinds of local patches: 1) local patches where the pre-trained CNN has higher responses on wrong classes, and 2) local patches where the pre-trained CNN has "flat" responses across all classes. Denote $\mathbf{P}(k|\mathbf{x}_{mn}; \hat{\mathbf{W}})$ as the kth output of the pre-trained CNN on $\mathbf{x}_{mn}$, i.e., the probability of $\mathbf{x}_{mn}$ belonging to class k, the non-informative local patches of a training slice $\mathbf{X}_m$ are defined as:

$$\begin{aligned}\mathbf{B}_m = &\{\mathbf{x}_{mn} | \underset{k \in \{1,\ldots,K\}}{\arg\max} \mathbf{P}(k|\mathbf{x}_{mn}; \hat{\mathbf{W}}) \neq l_m\} \\ &\cup \{\mathbf{x}_{mn} | \underset{k \in \{1,\ldots,K\}}{\text{entropy}} \mathbf{P}(k|\mathbf{x}_{mn}; \hat{\mathbf{W}}) > \theta\}.\end{aligned} \tag{5}$$

Recall the toy example, Fig. 3(b) shows the selected discriminative and non-informative local patches. When the discriminative patches from Class1 and Class2 only contain triangle or square, respectively, the non-informative patches may include circle, diamond or background. This is exactly in accordance to the fact that these two classes are only distinguishable by "triangle" and "square".

After introducing the additional non-informative class, the CNN structure keeps the same as the pre-trained CNN, except the LR layer has an additional output (see shadowed box in the rightmost diagram of Fig. 5) and the corresponding connections to the hidden layer. Since the pre-trained CNN already captured some discriminative local appearance characteristics, all

network layers except the last one are initialized by inheriting their coefficients from the pre-trained CNN. These coefficients are further adapted by minimizing (6):

$$L_4(\mathbf{W}) = \sum_{\mathbf{x}\in\mathbf{A}\bigcup\mathbf{B}} -\log(\mathbf{P}(l|\mathbf{x};\mathbf{W})). \quad (6)$$

Here, $\mathbf{A} = \bigcup_{\{m=1,\ldots,M\}} \mathbf{A}_m$ and $\mathbf{B} = \bigcup_{\{m=1,\ldots,M\}} \mathbf{B}_m$ denote the discriminative and non-informative local patches selected from all training images, respectively. Note that since the non-informative local patches are not belonging to any body section class now, their responses on any body section class can be effectively suppressed during the CNN boosting stage.

D. Run-Time Classification

The two-stage CNN learning algorithm is summarized as follows.

Input:

Scalars M, N, K, dataset $(\mathbf{X}_m, l_m), \forall m \in \{1,\ldots,M\}$, CNN architecture

Output:

Boosted CNN coefficients $\mathbf{W}^{opt}$

1: Partition $\mathbf{X}_m$ into N overlapping local regions $\mathbf{x}_{mn}$

2: Pre-train CNN on $(\mathbf{x}_{mn}, l_m)$ using multi-instance loss function **(3)**, and obtain optimized $\hat{\mathbf{W}}$

3: Extract $\mathbf{A}_m$ and $\mathbf{B}_m$ according to **(4)** and **(5)**

4: Assign label l_m to each instance of $\mathbf{A}_m$, and label K+1 to each of $\mathbf{B}_m$

5: Modify pre-trained CNN by adding one unit to LR, layers except LR inherit coefficients

6: Boost CNN on set $\mathbf{A} \cup \mathbf{B}$ using loss function **(6)**, and obtain optimized $\mathbf{W}^{opt}$

At runtime, the boosted CNN is applied for bodypart recognition in a sliding window fashion. The sliding window partitions a testing image $\mathbf{X}$ into N overlapping local patches $\mathcal{L}(\mathbf{X}) = \{\mathbf{x}_n, n = 1,\ldots,N\}$. For each local patch $\mathbf{x}_n$, the boosted CNN outputs a response vector with $K+1$ components $\{\mathbf{P}(k|\mathbf{x}_n;\mathbf{W}^{opt}) \mid k = 1,\ldots,K+1\}$, where $\mathbf{W}^{opt}$ denotes the optimal coefficients of (6). The class of the local patch $\mathbf{x}_n$ is then determined as:

$$c(\mathbf{x}_n) = \underset{k\in\{1,\ldots,K+1\}}{\arg\max} \mathbf{P}(k|\mathbf{x}_n;\mathbf{W}^{opt}). \quad (7)$$

Since the class $K+1$ is an artificially constructed non-informative one, local patches belong to this class should be ignored in body section determination. Simply, the class (body section) of the testing slice $\mathbf{X}$ can be determined by its most discriminative patch $\mathbf{x}_{n^*}$ as:

$$C(\mathbf{X}) = c(\mathbf{x}_{n^*}), \quad (8)$$

$$\mathbf{x}_{n^*} = \underset{\mathbf{x}_n\in\mathcal{L}(\mathbf{X});c(\mathbf{x}_n)\neq K+1}{\arg\max} \mathbf{P}(c(\mathbf{x}_n)|\mathbf{x}_n;\mathbf{W}^{opt}). \quad (9)$$

However, it is possible that the detected $\mathbf{x}_{n^*}$ is an outlier with different prediction than its neighbors. It would be more robust to fuse the prediction around its neighborhood to label the image. Therefore, we combine the class probabilities of the neighboring patches around the most discriminative patch to derive the image class as:

$$C(\mathbf{X}) = \underset{k\in\{1,\ldots,K\}}{\arg\max} \sum_{\mathbf{x}_n\in\mathfrak{N}(\mathbf{x}_{n^*})} \mathbf{P}(k|\mathbf{x}_n;\mathbf{W}^{opt}). \quad (10)$$

E. Implementation Details

In this study, we assume one middle level discriminative patch from an image is enough for the image classification. To discover the patches which are discriminative and representative for their image categories, the patch size should not be too small to include semantic information for the discriminative objects. To simplify analysis, we slide a fixed-size window to extract overlapping patches from images for training and testing. The patch size and step size is specified in experimental settings. We further analyze the sensitivity of patch size later. In learning stage II, since the patches per image are overlapping, the non-informative patches are selected with a spatial constraint that they should not appear neighboring to the discriminative patches.

In each of the two training stages, we train a CNN model similar to Fig. 2. The following strategies are employed to improve the performance of the learned CNN. First, Rectified Linear Units (ReLUs) are used to map the neurons' output in convolutional layers. As shown in [10], [46], ReLUs demonstrates faster convergence and better performance than sigmoid functions. Second, to incorporate larger variability in our training samples, hence, increase the robustness of the CNN, we augment data using label-preserving transformations [25], [47]. Specifically, we simply apply up to 10% (relate to image size) random translation to increase training data samples. Third, the "dropout" strategy [48] is employed to reduce the risk of "over-fitting". It forces half of the neurons randomly "dropped out" at each training iteration. In this way, the complex correlation of neurons is reduced and more robust features can be learned. Finally, as the training set may be too large to load into memory at one time, we trained our model using a mini batch of samples at each iteration. The optimization is implemented by stochastic gradient descent with a momentum term β [49] and a weight decay term γ. For a weight $\omega \in \mathbf{W}$, its update at iteration t is defined as

$$\omega^{(t)} = \omega^{(t-1)} + \Delta\omega^{(t)}, \quad (11)$$

where

$$\Delta\omega^{(t)} = \beta \cdot \Delta\omega^{(t-1)} - \epsilon \cdot (\delta\omega^{(t-1)} + \gamma * \omega^{(t-1)}) \quad (12)$$

$\delta\omega^{(t-1)}$ is the gradient of weight based on current batch of samples.

The learning process is conducted on a training subset and a validation subset. It won't stop until the error rate on validation subset is smaller than a threshold ξ or a predefined maximum number of epochs is reached. Besides, the learning may stop earlier if it cannot reach smaller error since the current smallest

one after a number of patient iterations. Our algorithm is implemented in Python using Theano library [50]. To leverage the highly parallelable property of CNN training, we trained our models on a 64-bit desktop with i7–2600 (3.4 GHz) CPU, 16GB RAM and NVIDIA GTX-660 3GB GPU.

IV. Experiments

A. Image Classification on Synthetic Data

We first validate our method on a synthetic data set, which has been briefly introduced as a toy example in Section III-A. It is constructed by 4 types of geometry elements: triangle, square, circle and diamond. The size of all synthetic images are 60×60 with black background (intensity value 0). The basic geometry elements are created within a bounding-box 20×20 and random intensity values in $[1, 255]$. These elements are then resized with random scales up to 10% in height and width, and randomly positioned on the image background. In constructing the two image classes, we ensure that the triangle and square are the "distinctive" element and only appear in Class1 and Class2, respectively. Besides the distinctive element, each image has another element, which is randomly chosen from circle or diamond by coin flipping. (Some examples of the synthetic images are shown in Fig. 3(a).) Overall, we create 2000 training, 2000 validation and 2000 for testing samples (1000 for each class). Let TP (true positive) denote the number of samples belonging to class k and correctly classified; FN (false negative) denote the number of samples belonging to class k but misclassified; FP (false positive) denote the number of samples not belonging to class k but misclassified as class k. Classification accuracies are reported in terms of recall, precision and F1 score as

$$Recall = \frac{TP}{TP+FN}, precision = \frac{TP}{TP+FP}, \quad (13)$$

$$F1score = 2\frac{precision \cdot recall}{precision + recall}. \quad (14)$$

The classification accuracy of standard CNN algorithm is 83.3%, as shown in the "SCNN" row in Table I(a). This inferior performance results from the fact that the global CNN learning scheme may not learn the most discriminative local patches. On the contrary, the most discriminative and non-informative local patches are effectively discovered by our two-stage learning framework as shown in Fig. 3(b). The discovered pattern exactly matches the rule of generating these two classes. By leveraging these local patches, our classification accuracy can reach 100% ("BCNN2" row in Table I(a)).

A comparison study is conducted using: (1) logistic regression (LR); (2) SVM; (3) standard CNN, similar to LeNet [45], trained on whole image (SCNN); (4) local patch-based CNN without boost, i.e., the CNN trained by pre-train stage only (PCNN); (5) local patch-based CNN boosted without additional non-informative class (BCNN1); (6) local patch-based CNN boosted with both discriminative and non-informative patches (BCNN2). Methods (1)–(3) represent conventional learning (using image intensities directly as features) and deep learning approaches. Methods (4), (5) are two variants of our proposed one (6), which are presented to verify the effects of each component of our method. The parameters of LR and SVM are optimized using grid search with cross-validation. All CNN-related methods use the same intermediate architecture: one convolutional layer with 10 5×5 filters, one max-pooling layer with 2×2 kernel, one hidden layer of 300 nodes, and finally followed by a LR layer to output response. The patch size for all patch-based CNNs is 30×30. There are 36 patches extracted from each 60×60 image through a sliding window with 6-pixel step size.

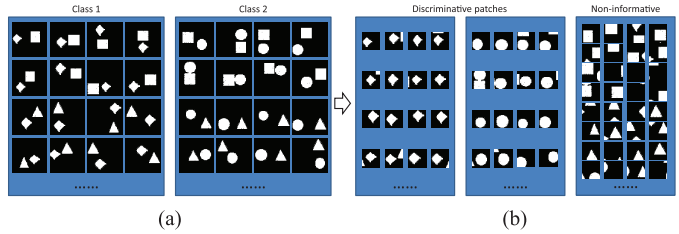

Fig. 6. The second toy example. (a) Synthetic images of two classes distinguished by diamond and circle. It is important to note that we used the same image samples as in Fig. 3, but re-labeled the images into two classes based on different rules. (b) The discriminative and non-informative local patches discovered by the pre-trained CNN model.

As shown in Table I(a), standard deep learning method (SCNN) is better than LR, which indicates deep learning can learn good features from raw data. By leveraging the local discriminative information, PCNN gets $\approx$ 16% improvement from SCNN. Among our local patch-based CNNs (PCNN, BCNN1 and BCNN2), BCNN1, which is trained on extracted discriminative (without non-informative) patches, is worse than PCNN due to overfitting. BCNN2, which includes all designed components, achieves the best performance.

To further prove the adaptivity of our algorithm, we re-labeled the synthetic data using Diamond and circle as distinctive elements in class 1 and class 2, respectively (see Fig. 6(a)). In other words, although the synthetic data are exactly the same, the local patches to distinguish the two classes become different. This is in analogy to real-world problems where the datasets are identical but the classification goal is changed. After conducting the pre-train algorithm, the extracted local patches from the learned model are shown in Fig. 6(b). Again, the extracted local patches contain the most discriminative information, diamond and circle. The classification accuracies are shown in Table I(b). Again, our two-stage learning framework BCNN2 achieves the best performance among all comparison methods. This result demonstrates that our multi-instance CNN learning can *adaptively* learn discriminative local regions for specific classification tasks without any local level annotations.

B. Bodypart Recognition of CT Slices

In the second experiment, we applied our method in bodypart recognition of transversal CT slices. As shown in Fig. 1, transversal slices of CT scans are categorized into 12 body sections (classes). Our dataset includes 7489 transversal CT slices. They were collected from scans of 675 patients with very different ages (1–90 years old). The imaging protocols were different: 31 different reconstruction kernels, $0.281\ \text{mm} - 1.953\ \text{mm}$ in-slice pixel resolution. We organize

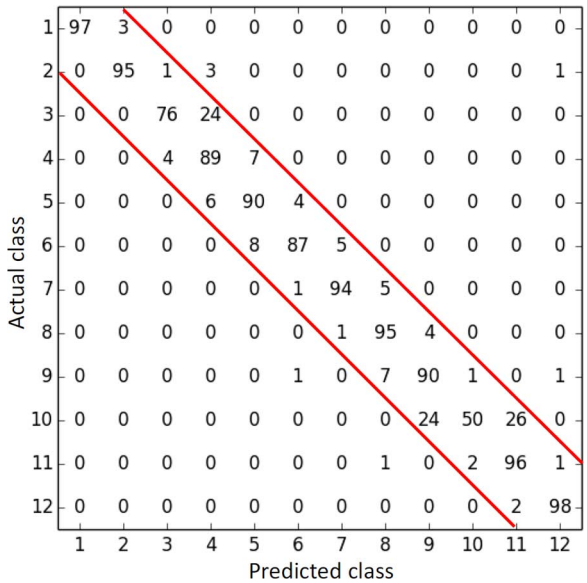

Fig. 7. Confusion matrix of BCNN2 on CT data. Values are normalized to $0 \sim 100$ in each row.

such a dataset with large variance to test the robustness of the proposed method. The whole dataset is divided into 2413 (225 patients) training, 656 (56 patients) validation and 4043 (394 patients) testing subsets. In this experiment, we augment data by simply applying up to 10% random translations in training and validation subsets to make them three times larger.

Our preprocessing includes two different steps: image sub-sampling and image cropping. First, all images are re-sampled to have 4 mm $\times$ 4 mm pixel resolution and 90$\times$90 in size. Then, cropping operation extracts 50$\times$50 local patches from each image with 10-pixel step size. Thus, 25 local patches are extracted per image. Our CNN has similar structure as in Fig. 2. C1 layer has 20 9$\times$9 filters. C3 layer has 40 9$\times$9 filters. Two sub-sampling layer, S2 and S4, use 2$\times$2 max-pooling. H5 layer has 600 hidden nodes. LR layer, O6, has 12 output nodes in pre-train stage, or 13 output nodes in boosting stage.

As shown in the "BCNN2" row of Table II, our method can achieve the classification accuracy (F1 score) at 92.23%. Fig. 7 shows more detailed classification performance by the confusion matrix. Most errors appear close to the diagonal line, which means most mis-classifications happen in the neighboring body sections. Quantitatively, 99.2% of the testing cases have "less-than-one neighboring class error" (within the red line corridor of Fig. 7). In practice, this kind of errors are already acceptable for some use cases and they may be further fixed by post-processing algorithms. For example, the 0.8% gross errors can be further suppressed by a simple label smoothing after classifications of a series of continuous slices for 3D bodypart identification. The learning process takes 440 epochs ($\approx$ 25 hours) in stage I and 70 epochs ($\approx$ 1 hour) in stage II.

For comparison, tested image classification methods include: (1) LR1, (2) LR2, (3) SVM1, (4) SVM2, (5) CaffeNet, (6) SCNN, (7) SCNN_a, (8) PCNN, (9) BCNN1, and (10) our proposed BCNN2. In LR1 and SVM1 methods, we use bag-of-word model with dense SIFT features to train logistic regressor and SVM classifier, respectively. While LR2 and SVM2 methods simply replace SIFT by HOG features. Same as the previous experiment, the LR and SVM parameters were optimized using grid search with cross-validation on the same training and validation sets as other comparison methods. Then, the optimized models were applied on the same testing set to produce results for fair comparisons. SCNN method is the standard CNN that takes the whole slice as input. SCNN_a method is the same as SCNN except trained by six times more augmented data samples with random transformations, rotations and scalings. Method (8), (9) are the variants of (10) as described in Section IV-A. Similar network structure is used in all CNN-based methods, (6)-(10), except different input and output sizes in patch-based CNNs (8)-(10). CaffeNet [51] has the similar structure as AlexNet [10] with a minor variation, which is trained on whole images without cropping. We noticed that training of CaffeNet with 50$\times$50 cropping doesn't converge. This observation shows that our proposed method is not merely a kind of data augmentation via image cropping. The discriminative and non-informative patches discovered by multi-instance learning are the keys to success. BCNN1 is trained on extracted discriminative (without non-informative) patches from learning stage I. Although the trained classifier focuses more on discriminative patches, ambiguous local patches across different classes (e.g., upholding arms may look similar to neck) are completely ignored and thereby mislead the classifier at runtime. Thus, the performance of BCNN1 is worse than PCNN and close to the SCNN. Compared to its variants, the proposed BCNN2 achieves the best performance (even better than much deeper CNN, CaffeNet), which proves the necessity of using all strategies designed in our method. In addition, we noted that the SCNN_a trained with more augmented data is even inferior to the SCNN due to overfitting (training error: SCNN_a 4.4% vs. SCNN 5%; testing error: SCNN_a 14.7% vs. SCNN 12.3%). It shows that the global CNN cannot learn the anatomy characteristics from more augmented data but overfit them. As shown in Table II, the overfitting problem is more severe in neck (column 3) and liver upper (column 9) sections. These two sections happen to have subtle global appearance differences compared to their neighboring sections and are thus prone to overfitting. The online classification time of each method is about (1) 4 ms, (2) 3 ms, (3) 5 ms, (4) 4 ms, (5) 3 ms, (6) 4 ms, (7) 4 ms, (8) 10 ms, (9) 11 ms, (10) 11 ms per image, respectively.

In this application of bodypart recognition, the most discriminative patch samples for each class are shown in Fig. 8 as well as the some samples of non-informative (kind of misleading) patches. From this figure, we observe that the proposed method "magically" extracts meaningful local patches for each class without any prior information, and these representative and discriminative local patches can significantly improve the classification task comparing with the global image information.

To investigate whether the standard CNNs can discover the required discriminative features at some intermediate layers or they completely miss them, we did extra experiments to train linear SVM classifier on the learned hidden activation on each layer of the baseline CNN (SCNN). Totally 5 classifiers were

TABLE II
CLASSIFICATION ACCURACIES ON CT DATA IN TERMS OF RECALL, PRECISION AND F1 SCORE (%).

	Recall												
Class	1	2	3	4	5	6	7	8	9	10	11	12	Total
LR1	48.54	63.64	33.33	69.95	39.09	43.37	47.66	81.43	25.82	53.68	41.62	88.79	63.37
LR2	67.96	64.50	42.59	74.24	38.64	42.17	56.25	78.51	37.09	65.79	60.41	91.48	68.71
SVM1	41.75	64.07	46.30	76.26	36.36	45.18	47.27	78.66	31.46	52.90	50.25	88.57	64.63
SVM2	76.70	81.39	39.82	79.55	54.09	63.86	69.92	84.06	44.60	64.47	75.64	96.53	76.75
CaffeNet	71.85	64.94	**87.96**	85.10	74.09	80.72	57.81	94.59	80.75	78.95	85.28	97.53	84.74
SCNN	84.47	93.51	72.22	88.89	80.46	80.12	86.72	95.47	77.93	77.63	78.43	96.30	87.73
SCNN_a	81.55	96.54	43.52	92.68	79.09	**90.96**	93.75	88.45	51.17	64.47	**90.36**	92.60	84.76
PCNN	87.38	92.21	**87.96**	93.18	**90.00**	74.70	**94.53**	95.47	81.69	81.05	**90.36**	92.49	90.21
BCNN1	54.37	83.12	69.44	**94.95**	75.91	82.53	93.36	95.03	**84.98**	72.63	85.53	96.75	87.78
BCNN2	**88.35**	**96.97**	80.56	91.67	86.82	87.35	93.75	**95.61**	79.81	**87.11**	87.82	**99.33**	**92.21**
	Precision												
Class	1	2	3	4	5	6	7	8	9	10	11	12	Total
LR1	72.46	63.09	63.16	64.42	55.48	54.14	54.46	72.15	55.00	57.14	47.40	67.87	62.21
LR2	65.42	64.78	61.33	68.53	53.80	46.36	56.92	76.06	59.40	64.43	62.80	78.92	67.74
SVM1	75.44	61.93	65.79	64.81	53.69	51.37	53.78	74.00	56.30	58.26	49.62	72.15	63.72
SVM2	89.77	72.03	72.88	72.25	56.94	57.92	72.76	82.62	62.50	70.81	71.64	90.54	76.39
CaffeNet	98.67	**98.04**	41.49	86.41	**93.68**	83.23	87.57	84.25	66.15	77.32	84.42	99.09	86.84
SCNN	87.88	87.45	82.11	87.35	87.62	83.13	84.73	92.36	76.50	75.06	83.51	96.73	87.72
SCNN_a	96.55	78.80	77.05	81.19	89.69	79.06	80.54	94.38	83.85	74.70	66.17	98.33	85.75
PCNN	96.77	93.83	81.20	87.86	85.35	**96.88**	90.30	95.33	78.73	**83.92**	77.73	**99.76**	90.69
BCNN1	**100.00**	94.58	54.75	76.27	91.26	93.20	91.92	95.59	77.68	82.64	78.74	97.08	88.62
BCNN2	96.81	91.80	**88.78**	**90.30**	90.52	89.51	**91.95**	**95.75**	**80.95**	82.13	**91.78**	98.66	**92.25**
	F1 score												
Class	1	2	3	4	5	6	7	8	9	10	11	12	Total
LR1	58.14	63.36	43.64	67.07	45.87	48.16	50.83	76.51	35.14	55.36	44.32	76.93	62.78
LR2	66.67	64.64	50.27	71.27	44.97	44.16	56.58	77.27	45.67	65.10	61.58	84.74	68.22
SVM1	53.75	62.98	54.35	70.07	43.36	48.08	50.31	76.26	40.36	55.45	49.94	79.52	64.17
SVM2	82.72	76.42	51.50	75.72	55.48	60.75	71.32	83.33	52.06	67.49	73.58	93.44	76.57
CaffeNet	83.15	78.13	56.38	85.75	82.74	81.96	69.65	89.12	72.73	78.13	84.85	98.31	85.78
SCNN	86.14	90.38	76.85	88.11	83.89	81.60	85.71	93.89	77.21	76.33	80.89	96.52	87.73
SCNN_a	88.42	86.77	55.62	86.56	84.06	84.59	86.64	91.32	63.56	69.21	76.39	95.38	85.25
PCNN	91.84	93.01	84.44	90.44	87.61	84.35	92.37	95.40	80.18	82.46	83.57	95.99	90.45
BCNN1	70.44	88.48	61.22	84.59	82.88	87.54	92.64	95.31	**81.17**	77.31	82.00	96.91	88.20
BCNN2	**92.39**	**94.32**	**84.47**	**90.98**	**88.63**	**88.42**	**92.84**	**95.68**	80.38	**84.55**	**89.75**	**98.99**	**92.23**

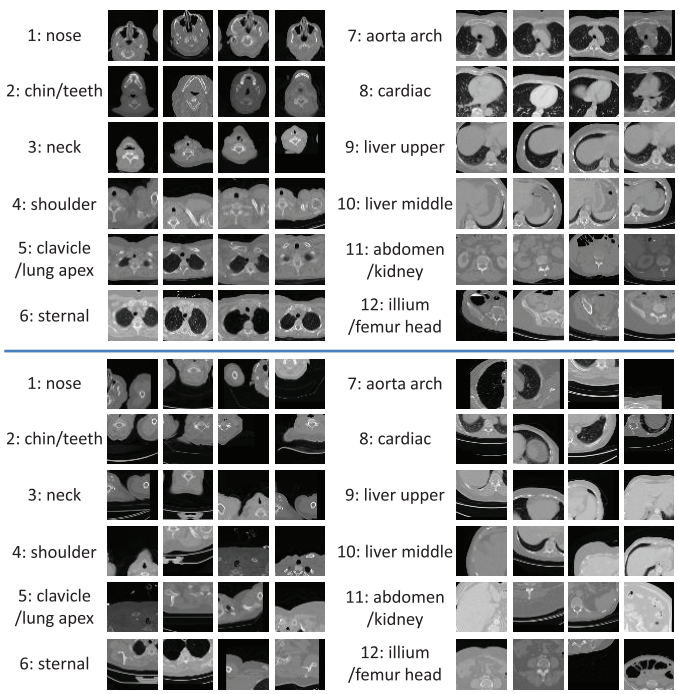

Fig. 8. Automatically discovered discriminative and non-informative patches from each class through multi-instance learning.

learned from features on layers (C1, S2, C3, S4, H5). The feature sizes are 134480, 33620, 43560, 10240 and 600, respectively. The F1-scores on testing set are 0.75, 0.77, 0.86, 0.86 and 0.88, respectively. Compared with reported F1-score of SCNN ($\approx$ 0.88), we conclude that (1) features on higher layers are better for the classification task; (2) although the learned features in SCNN are discriminative to some extent, the more representative and discriminative local features can only be discovered in our proposed patch based learning algorithm.

C. Sensitivity Experiments

To evaluate the robustness of the trained models, we apply different scales of random linear translation on the testing data and compute the classification error rates. The Fig. 9(a) shows the results. From the plots, we can see that our proposed method BCNN2 has the best robustness regarding to the random translation of testing samples. Although the training and validation subsets have been augmented using up to 11% random translation, the other approaches do not perform as well as the proposed method when the testing samples have larger translations. In this situation, retraining the models on augmented dataset with larger translation could be a solution. However, the re-training costs cannot be overlooked and fixing it after the fact is not efficient in practice.

As one of the important parameters in BCNN2 method, step size of sliding window testing is investigated regarding to the accuracies (shown in Fig. 9(b)). The running times for step sizes 1, 5, 10, 15, 20, 25, and 30 pixels are 541.1, 30.6, 11.7, 5.3, 5.2,

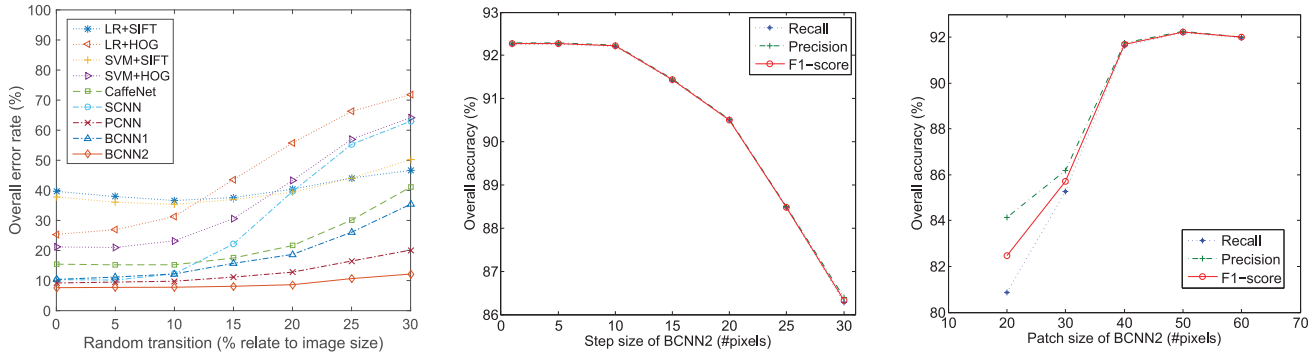

Fig. 9. Performance analyses on the sensitivity of parameters. (a) Classification errors vs. scales of random translations on testing data. (b) Classification accuracies vs. step size of sliding window in BCNN2. (c) Classification accuracies vs. patch size in BCNN2.

3.6 and 3.4 ms per image respectively. Considering the balance of running time and accuracy, step size 10 or 15 should be a reasonable choice in this experiment. The effect of patch size to the classification accuracy is also investigated as shown in Fig. 9(c). We can see from the plot that (1) the patch size should not be too small to capture the discriminative information (size 20 or 30); (2) the performance is not very sensitive to the local patch size once it is big enough to include discriminative information (sizes from 40 to 60 in this task).

We also conducted two extra experiments to test other variants of our proposed method. First, we use (2) with a bit larger patch size (70×70) rather than the (3) to accommodate for the neighbors information. The final classification accuracy in terms of F1-score becomes 91.67%, a little worse than that of the proposed BCNN2 (92.23%). Second, instead of using the run-time classification strategy by (10), we can simply use the (8) as in [44], or majority voting of predictions from all partitioned patches in the slice to predict image classification. The F1 score drops $\approx 2\%$ and $\approx 4\%$, respectively.

V. Conclusions

In this paper, a novel multi-stage deep learning framework is presented to tackle the bodypart recognition problem. Its key novelty is to automatically exploit the local information through CNN, and discover the discriminative and non-informative local patches via multi-instance learning. It is worth noting that since no manual annotations are required to label these local patches, our method becomes very scalable. The proposed method is evaluated on a synthetic dataset and a large scale CT dataset. The experimental results show clear improvements compared with state-of-the-art methods. It is proved that the success of the proposed method does not result from more augmented training samples but its capability of *discovering* local characteristics of different bodyparts. This supervised discriminative patch discovery and classification method can be easily applied to other image classification tasks where local information is critical to distinguish different classes. Our proposed framework can also be extended to 3D cases using 3D convolutional filters. In future, we plan to investigate extracting multi-scale patches from images and exploring some sophisticated algorithms to further improve the performance in the boosting stage.

References

[1] T. F. Cootes, C. J. Taylor, D. H. Cooper, and J. Graham, "Active shape models-their training and application," *Comput. Vis. Image Understand.*, vol. 61, no. 1, pp. 38–59, 1995.
[2] S. Zhang *et al.*, "Deformable segmentation via sparse shape representation," in *Proc. MICCAI*, 2011, pp. 451–458.
[3] S. Zhang, Y. Zhan, and D. N. Metaxas, "Deformable segmentation via sparse representation and dictionary learning," *Med. Image Anal.*, vol. 16, no. 7, pp. 1385–1396, 2012.
[4] S. Zhang *et al.*, "Towards robust and effective shape modeling: Sparse shape composition," *Med. Image Anal.*, vol. 16, no. 1, pp. 265–277, 2012.
[5] M. O. Gueld *et al.*, "Quality of dicom header information for image categorization," in *Proc. SPIE Med. Imag.*, 2002, pp. 280–287.
[6] Y. Rejani and S. T. Selvi, "Early detection of breast cancer using SVM classifier technique," *ArXiv Preprint ArXiv:0912.2314*, 2009.
[7] V. F. Van Ravesteijn *et al.*, "Computer-aided detection of polyps in CT colonography using logistic regression," *IEEE Trans. Med. Imag.*, vol. 29, no. 1, pp. 120–131, Jan. 2010.
[8] Y. Freund and R. E. Schapire, "A desicion-theoretic generalization of on-line learning and an application to boosting," in *Computational Learning Theory*. New York: Springer, 1995, pp. 23–37.
[9] L. Breiman, "Random forests," *Mach. Learn.*, vol. 45, no. 1, pp. 5–32, 2001.
[10] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Adv. Neural Inf. Process. Syst.*, 2012, pp. 1097–1105.
[11] Y. Wei *et al.*, "CNN: Single-label to multi-label," *ArXiv Preprint ArXiv:1406.5726*, 2014.
[12] Y. Taigman, M. Yang, M. Ranzato, and L. Wolf, "Deepface: Closing the gap to human-level performance in face verification," in *Proc. IEEE Int. Conf. Comput. Vis. Pattern Recognit.*, 2014, pp. 1701–1708.
[13] J. Park, G. Kang, S. Pan, and P. Kim, "A novel algorithm for identification of body parts in medical images," in *Fuzzy Syst. Knowl. Discovery*, 2006, pp. 1148–1158.
[14] L. Hong and S. Hong, "Methods and apparatus for automatic body part identification and localization," U.S. Patent App. 11/933 518, May 15, 2008.
[15] Y. Zhan, X. S. Zhou, Z. Peng, and A. Krishnan, "Active scheduling of organ detection and segmentation in whole-body medical images," in *Proc. MICCAI*, 2008, pp. 313–321.
[16] A. Criminisi, J. Shotton, D. Robertson, and E. Konukoglu, "Regression forests for efficient anatomy detection and localization in CT studies," in *Proc. MICCAI Med. Comput. Vis. Recognit. Tech. Appl. Med. Imag.*, 2011, pp. 106–117.
[17] A. Criminisi, J. Shotton, and S. Bucciarelli, "Decision forests with long-range spatial context for organ localization in CT volumes," in *Proc. MICCAI*, 2009, pp. 69–80.
[18] R. Donner, B. H. Menze, H. Bischof, and G. Langs, "Global localization of 3D anatomical structures by pre-filtered hough forests and discrete optimization," *Med. Image Anal.*, vol. 17, no. 8, pp. 1304–1314, 2013.
[19] A. Oliva and A. Torralba, "Building the gist of a scene: The role of global image features in recognition," *Progr. Brain Res.*, vol. 155, pp. 23–36, 2006.

[20] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *Int. J. Comput. Vis.*, vol. 60, no. 2, pp. 91–110, 2004.

[21] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," in *Proc. IEEE Int. Conf. Comput. Vis. Pattern Recognit.*, 2005, vol. 1, pp. 886–893.

[22] S. Lazebnik, C. Schmid, and J. Ponce, "Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories," in *Proc. IEEE Int. Conf. Comput. Vis. Pattern Recognit.*, 2006, vol. 2, pp. 2169–2178.

[23] J. Yang, K. Yu, Y. Gong, and T. Huang, "Linear spatial pyramid matching using sparse coding for image classification," in *Proc. IEEE Int. Conf. Comput. Vis. Pattern Recognit.*, 2009, pp. 1794–1801.

[24] J. Sánchez and F. Perronnin, "High-dimensional signature compression for large-scale image classification," in *Proc. IEEE Int. Conf. Comput. Vis. Pattern Recognit.*, 2011, pp. 1665–1672.

[25] D. Ciresan, U. Meier, and J. Schmidhuber, "Multi-column deep neural networks for image classification," in *Proc. IEEE Int. Conf. Comput. Vis. Pattern Recognit.*, 2012, pp. 3642–3649.

[26] H. R. Roth *et al.*, "Anatomy-specific classification of medical images using deep convolutional nets," in *Proc. IEEE Int. Symp. Biomed. Imag.*, 2015, pp. 101–104.

[27] D. Parikh, "Recognizing jumbled images: the role of local and global information in image classification," in *Proc. Int. Conf. Comput. Vis.*, 2011, pp. 519–526.

[28] Z.-J. Zha *et al.*, "Joint multi-label multi-instance learning for image classification," in *Proc. IEEE Int. Conf. Comput. Vis. Pattern Recognit.*, 2008, pp. 1–8.

[29] O. Maron and A. L. Ratan, "Multiple-instance learning for natural scene classification," in *Proc. Int. Conf. Mach. Learn.*, 1998, vol. 98, pp. 341–349.

[30] M. Juneja, A. Vedaldi, C. Jawahar, and A. Zisserman, "Blocks that shout: Distinctive parts for scene classification," in *Proc. IEEE Int. Conf. Comput. Vis. Pattern Recognit.*, 2013, pp. 923–930.

[31] C. Szegedy, A. Toshev, and D. Erhan, "Deep neural networks for object detection," in *Adv. Neural Inf. Process. Syst.*, 2013, pp. 2553–2561.

[32] P. F. Felzenszwalb, R. B. Girshick, D. McAllester, and D. Ramanan, "Object detection with discriminatively trained part-based models," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 32, no. 9, pp. 1627–1645, Sep. 2010.

[33] S. Singh, A. Gupta, and A. A. Efros, "Unsupervised discovery of mid-level discriminative patches," in *Proc. Eur. Conf. Comput. Vis.*, 2012, pp. 73–86.

[34] M.-M. Cheng, Z. Zhang, W.-Y. Lin, and P. Torr, "BING: Binarized normed gradients for objectness estimation at 300 fps," in *Proc. IEEE Int. Conf. Comput. Vis. Pattern Recognit.*, 2014, pp. 3286–3293.

[35] O. Maron and T. Lozano-Pérez, "A framework for multiple-instance learning," *Adv. Neural Inf. Process. Syst.*, pp. 570–576, 1998.

[36] S. Andrews, I. Tsochantaridis, and T. Hofmann, "Support vector machines for multiple-instance learning," in *Adv. Neural Inf. Process. Syst.*, 2002, pp. 561–568.

[37] Q. Zhang and S. A. Goldman, "EM-DD: An improved multiple-instance learning technique," in *Adv. Neural Inf. Process. Syst.*, 2001, pp. 1073–1080.

[38] J. Wu, Y. Yu, C. Huang, and K. Yu, "Deep multiple instance learning for image classification and auto-annotation," in *Proc. IEEE Int. Conf. Comput. Vis. Pattern Recognit.*, 2015, pp. 3460–3469.

[39] P. O. Pinheiro and R. Collobert, "From image-level to pixel-level labeling with convolutional networks," in *Proc. IEEE Int. Conf. Comput. Vis. Pattern Recognit.*, 2015, pp. 1713–1721.

[40] P. Sermanet *et al.*, "Overfeat: Integrated recognition, localization and detection using convolutional networks," *ArXiv Preprint ArXiv:1312.6229*, 2013.

[41] G. Papandreou, L. Chen, K. Murphy, and A. L. Yuille, "Weakly- and semi-supervised learning of a DCNN for semantic image segmentation," *ArXiv Preprint ArXiv:1502.02734v2* 2015 [Online]. Available: http://arxiv.org/abs/1502.02734

[42] P. Krähenbühl and V. Koltun, "Efficient inference in fully connected CRFs with Gaussian edge potentials," *ArXiv Preprint ArXiv:1210.5644*, 2012.

[43] L. Hou *et al.*, "Efficient multiple instance convolutional neural networks for gigapixel resolution image classification," *ArXiv Preprint ArXiv:1504.07947*, 2015.

[44] Z. Yan *et al.*, "Bodypart recognition using multi-stage deep learning," in *Proc. Int. Conf. Inf. Process. Med. Imag.*, 2015, pp. 449–461.

[45] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proc. IEEE*, vol. 86, no. 11, pp. 2278–2324, Nov. 1998.

[46] V. Nair and G. E. Hinton, "Rectified linear units improve restricted boltzmann machines," in *Proc. Int. Conf. Mach. Learn.*, 2010, pp. 807–814.

[47] P. Y. Simard, D. Steinkraus, and J. C. Platt, "Best practices for convolutional neural networks applied to visual document analysis," in *Proc. IEEE Comput. Soc. 12th Int. Conf. Document Anal. Recognit.*, 2003, vol. 2, p. 958.

[48] G. E. Hinton, N. Srivastava, A. Krizhevsky, I. Sutskever, and R. R. Salakhutdinov, "Improving neural networks by preventing co-adaptation of feature detectors," *ArXiv Preprint ArXiv:1207.0580*, 2012.

[49] N. Qian, "On the momentum term in gradient descent learning algorithms," *Neural Netw.*, vol. 12, no. 1, pp. 145–151, 1999.

[50] J. Bergstra *et al.*, "Theano: A CPU and GPU math expression compiler," in *Proc. Python Sci. Comput. Conf.*, Jun. 2010.

[51] Y. Jia *et al.*, "Caffe: Convolutional architecture for fast feature embedding," *ArXiv Preprint ArXiv:1408.5093*, 2014.