CHAPTER 4

Multi-Instance Multi-Stage Deep Learning for Medical Image Recognition

Zhennan Yan*, Yiqiang Zhan†, Shaoting Zhang‡, Dimitris Metaxas*, Xiang Sean Zhou†

Rutgers University, Piscataway, NJ, United States Siemens Healthcare, Malvern, PA, United States† University of North Carolina at Charlotte, Charlotte, NC, United States‡*

CHAPTER OUTLINE

4.1 Introduction 83
4.2 Related Work 85
4.3 Methodology 87
4.3.1 Problem Statement and Framework Overview 87
4.3.2 Learning Stage I: Multi-Instance CNN Pre-Train 88
4.3.3 Learning Stage II: CNN Boosting 90
4.3.4 Run-Time Classification 92
4.4 Results 93
4.4.1 Image Classification on Synthetic Data 93
4.4.2 Body-Part Recognition on CT Slices 95
4.5 Discussion and Future Work 99
References 100

4.1 INTRODUCTION

In medical image analysis, clinical information representation and extraction is the primary goal, and the basis of many complicated frameworks. Many automatic or semi-automatic algorithms have been developed in this community [1–4]. They were designed to assist clinicians or researchers to interpret and assess medical images in different applications, from fundamental tasks, e.g. anatomical landmark detection [5,6] and organ segmentation [7–9], to complicated computer aided diagnosis (CAD) systems [10–12]. Since different organ systems have highly diverse characteristics, medical image analysis models are usually trained or designed for specific anatomies to incorporate appropriate prior knowledge. Such ad-hoc models using handcraft features are not easily extensible to different use cases.

Deep Learning for Medical Image Analysis. DOI: 10.1016/B978-0-12-810408-8.00006-7

In the last decade, deep learning [13] methods have shown successful outcomes in different applications, including signal processing [14], object recognition [15], natural language processing [16], etc. Their attraction comes from the automatic feature learning ability in a deep network architecture. The deep network architecture uses multiple layers of simple but nonlinear activation functions to transform the input data to multiple levels of feature representation, from low (detail) level to high (abstract) level. The network is able to learn such hierarchical feature representations in unsupervised or supervised way from a large amount of training data by itself. Such learned hierarchical features have been proved to be superior to ad-hoc designed ones in a wide range of practical applications [17]. Image recognition [15] is one of the most promising applications, where deep learning learns good representation of visual features. Thus, deep learning should be able to benefit the medical image recognition as well. The goal of this chapter is to introduce recent progress of deep learning based methods in medical image recognition, in particular a medical image recognition algorithm using multi-stage multi-instance deep learning.

A common form of medical data is imaging scans, e.g. CT and MR. A typical imaging scan is a 3D volume image consisted of a series of 2D slices. Although image scans are usually acquired for specific body parts in most clinical activities, whole body scans are also used to give a holistic view of diseases. For large scale medical image analysis, the data are very likely collected from different institutes with various hardware and protocols. These complicated data set always need complex preprocessing, such as data selection, alignment and anatomy localization, before applying some automatic frameworks for a research study. In such cases, it is important to have a reliable automatic module that can recognize image contents in the first place. For example, we would like to have a program, which can act like a medical professional who can tell quickly anatomies and their rough positions in an image at a glance. With the correct first impression, automated workflow can conduct easy preprocessing and other higher level jobs (for example, detection and segmentation) using appropriate method or model.

In this chapter, we mainly focus on the medical image recognition task to identify anatomies contained in the input image, namely “body-part recognition”. Although organ segmentation and landmark detection topics have been extensively investigated, the automatic body-part recognition (identifying the human body parts contained in the medical image) is still less explored. This task can be easily defined as an image classification problem in 2D setting. For example, assuming the human body is divided into continuous sections according to anatomical context as shown in Fig. 4.1, the task is, in fact, a multi-class image classification problem. Given a 2D transversal slice, the goal is to identify what body section the image belongs to. Although 3D volume always contain more comprehensive anatomy information, a 2D slice-based anatomy recognition algorithm provides the foundation of 3D recognition, and is efficient and practical when 3D data is not available. This chapter does not intend to introduce a new method but to present our existing work on deep learning [18,19].

It is worth noting that although DICOM header includes anatomy information, text-based retrieval is not an ideal choice due to three major challenges. First, it may

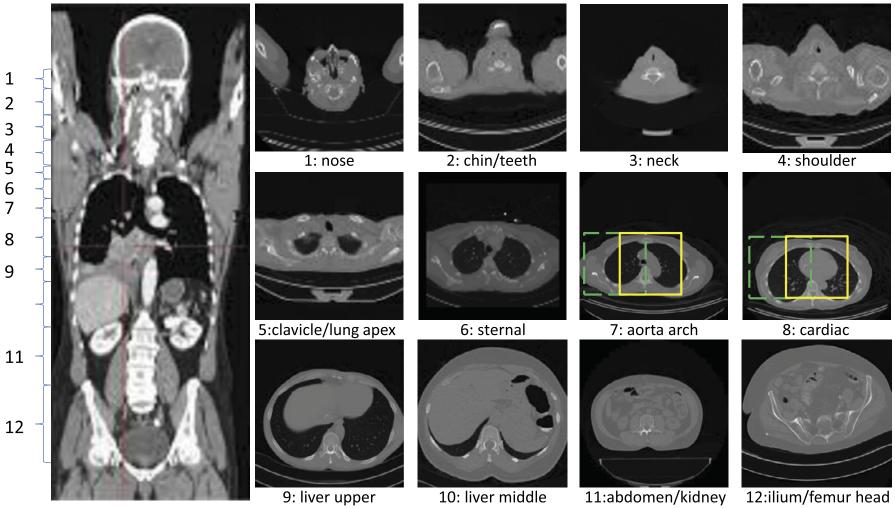


FIGURE 4.1

Definition of body sections. Human body is divided into 12 continuous parts. Each part may cover different ranges due to the variability of anatomies. Yellow boxes indicate the discriminative local regions in aorta arch class and cardiac class, while green boxes indicate ambiguous local regions for this classification task.

contain 15% of errors in DICOM headers [20]. Second, text information in DICOM is sometimes too abstract to precisely describe the anatomies contained in the scan. Third, the multi-language supporting of DICOM header becomes another barrier for text-based retrieval. On the contrary, a reliable image-based anatomy recognition algorithm can tackle all these three challenges by learning the intrinsic anatomical appearance information.

4.2 RELATED WORK

Deep learning utilizes massive amounts of computational power and achieves state-of-the-art results on various challenging tasks [17,13]. Among different deep learning methods, convolutional neural network (CNN) [21] based algorithms are more suitable in image related tasks, since images have highly correlated intensities in local regions and some local signals or statistics are invariant to location. In computer vision community, different CNN based methods [15,22,23] have shown their superiority in image classification tasks compared to conventional approaches with carefully designed/selected features, e.g. SVM and logistic regression [24,25] with SIFT [26] or HOG [27] features. Despite this success, their application in medical image analysis remains to be fully explored.

Recently, researchers have began to apply deep learning in CAD tasks, including detection [28–30], segmentation [31–34], disease classification [35], etc. These methods are proposed for specific anatomies and tasks. As discussed before, it would be useful to have an image-based anatomy recognition method in the CAD system. Roth et al. [36] presented a method for anatomy-specific classification of medical images using CNN. They applied a standard deep CNN on 2D axial CT images to classify 5 anatomies (neck, lungs, liver, pelvis, and legs) and obtained the state-of-the-art accuracy (5.9% error). In this slice-based anatomy recognition, the standard CNN is conducted as a *global* learning scheme, which takes the entire image as input. The CNN successfully learned feature representations to capture the diverse appearances in the five body sections. However, this standard CNN is not easily scalable to handle more detailed anatomy recognition (as shown in Fig. 4.1) effectively, since distinctive information often comes from *local* patches and these local patches are distributed "inconsistently" at different positions of the slices. As shown in Fig. 4.1, aorta arch section and cardiac section have globally similar appearance characteristics, while the discriminative information only resides in the local mediastinum region (indicated by the yellow boxes). The other areas are just "non-informative" or misleading for classification purposes. 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 body-part recognition.

In fact, this problem also exists in general image classification/recognition applications in computer vision. Researchers are trying to leveraging local region information to train CNN for recognition or classification. For example, in face recognition [37], the authors first detect and align face regions properly before training CNN. In another pioneer work, Wei et al. [23] applied an existing objectness detector [38] to produce some local region proposals from a given image, and used them to train multi-label CNN classifier. Similarly, Girshick [39] trained a region-based CNN (fast R-CNN) based on existing object proposals and used a multi-task loss function to learn the classifier and bounding-box regressor for efficient object detection. Then, Ren et al. [40] proposed a faster R-CNN detection by using a region proposal network which shares convolutional features with the detection network. Despite promising results these methods generate, they all require manual annotations of local regions of objects in images for training. However, the discriminative local regions for body section recognition are not easy to define, not to mention that the effort to build these local detectors might be quite large. Note that organ/landmark detector based anatomy recognition approaches are limited due to the tedious manual annotation efforts in the training stage and complicated inference efforts in testing stage [41–43].

To avoid explicit local region or object annotation, several studies [44–46] have emerged to incorporate multi-instance learning (MIL) [47] with CNN to better utilize local information in weakly supervised learning fashion. Yan et al. [18,19] proposed a fine-grained body-part recognition method based on CNN and MIL. It only requires image level label to "discover" the discriminative local regions automatically and use multi-stage learning to utilize the discovered local information to generate state-of-the-art results for image classification. In this way, the annotation efforts for local regions in the training stage are totally eliminated. This is in particular meaningful for medical image applications, since the annotations in medical images always require clinical expertise and high cost.

In this chapter, we introduce the technical details of the multi-stage multi-instance deep learning for medical image classification, specifically with the use case of body-part recognition in image slices.

4.3 METHODOLOGY

4.3.1 PROBLEM STATEMENT AND FRAMEWORK OVERVIEW

Definitions. Slice-based body-part recognition is a typical multi-class image classification problem for a learning algorithm. Denote by $\mathbf{X}$ the input slice/image, by K the number of body sections (classes), and by $l \in \{1, \ldots, K\}$ the corresponding class label of $\mathbf{X}$. The learning algorithm aims to find a function $\mathcal{O} : \mathbf{X} \to l$. In traditional image classification frameworks, $\mathcal{O}$ is often defined as $\mathbb{C}(\mathbb{F}(\mathbf{X}))$, where $\mathbb{F}(\mathbf{X})$ and $\mathbb{C}(\cdot)$ 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. 4.2 (similar to LeNet [21]), 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. $\mathbb{C}(\cdot)$ is implemented by the LR layer, whose output is a K-dimension vector $R(k), k \in \{1, \ldots, K\}$, representing the probability of $\mathbf{X}$ belonging to each class k. Mathematically, $R(k)$ can be described as a conditional probability $R(k) = \mathbf{P}(k|\mathbf{X}; \mathbf{W})$. Here, $\mathbf{W}$ denotes 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

$$L_1(\mathbf{W}) = \sum_{\mathbf{X}_m \in \mathcal{T}} -\log(\mathbf{P}(l_m|\mathbf{X}_m; \mathbf{W})), \tag{4.1}$$

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 $\mathbf{W}$.

Here, a multi-instance multi-stage deep learning framework is designed to “discover” discriminative and non-informative local regions for precise image classification without time-consuming local manual annotations, and learn classifier in the meantime. In the first stage, a CNN is learned in a multi-instance learning fashion

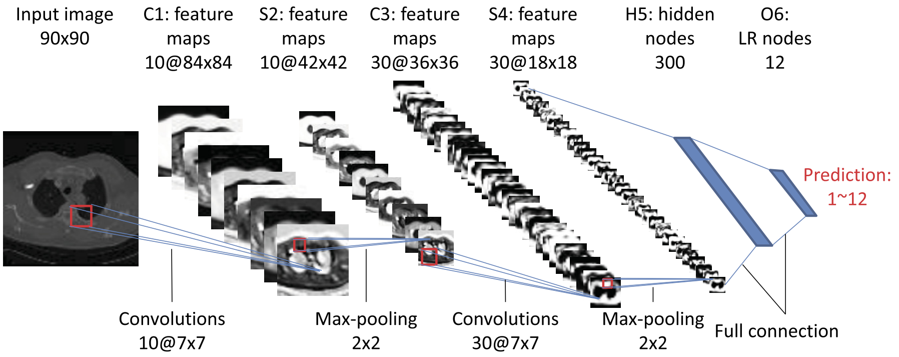


FIGURE 4.2

Illustration of one standard CNN architecture and the outputs of each layer.

to discover the most discriminative local patches. Specifically, each image is divided into several local patches. The deep network thus receives a set of labeled images (bags), each containing multiple local patches (instances). The loss function of the CNN is chosen in a way that as long as one local patch (instance) is correctly classified, the labeling of corresponding image slice (bag) is considered to be correct. In this way, the pre-trained CNN will be more sensitive (responds with significantly higher probability at the correct label) 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 in 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 target image. As the CNN is sensitive to the discriminative local patches, it essentially identifies a body part by focusing on the most distinctive local information and discarding non-informative local regions.

4.3.2 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 a 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 $\mathcal{L}(\mathbf{X}_m) = \{\mathbf{x}_{mn}, n = 1, \ldots, N\}$. These local patches become the basic training samples of the CNN and their labels are inherited from the original images, i.e., all $\mathbf{x}_{mn} \in \mathcal{L}(\mathbf{X}_m)$ share the same label l_m. While the structure of CNN is still the same as the standard

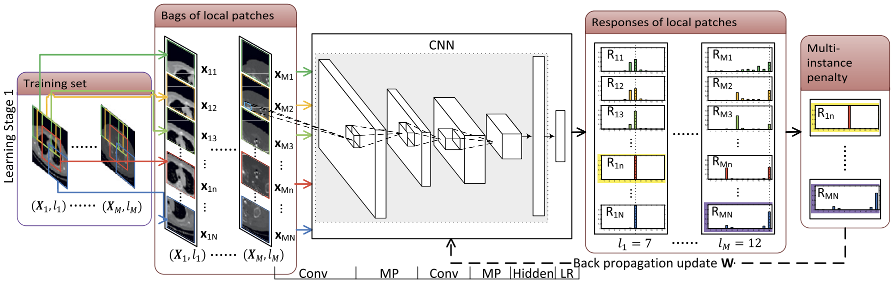


FIGURE 4.3

Illustration of the pre-train stage. In this stage, the CNN is trained in a multi-instance fashion. For instance, yellow highlighted response of instance $\mathbf{x}_{1n}$ from image $\mathbf{X}_1$ and purple highlighted response of instance $\mathbf{x}_{MN}$ from image $\mathbf{X}_M$ are picked to compute the loss to update the CNN parameters in a training iteration. They are picked because they have higher response on the correct label than other instances from the same image. Those local patches which can be easily and correctly classified are considered as the discriminative information for image classification. After training, the CNN will become sensitive to those discriminative local regions.

one, the loss function is

$$L_2(\mathbf{W}) = \sum_{\mathbf{X}_m \in \mathcal{T}} -\log\Big(\max_{\mathbf{x}_{mn} \in \mathcal{L}(\mathbf{X}_m)} \mathbf{P}(l_m | \mathbf{x}_{mn}; \mathbf{W})\Big), \tag{4.2}$$

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 Eq. (4.1) 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. This 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. 4.3, 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.

We design a toy example to illustrate this discovery ability. As shown in Fig. 4.4A, four types of geometry elements, namely, square, circle, triangle, and diamond, are randomly positioned and combined to generate two classes of binary images. While circle and diamond are allowed to appear in any classes, triangle and square are ex-

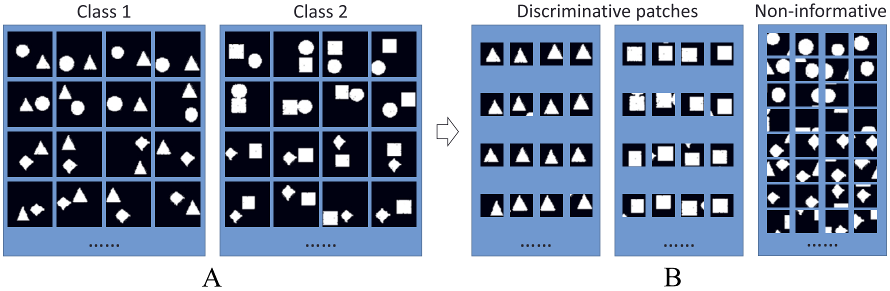


FIGURE 4.4

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.

clusively owned by Class 1 and Class 2, respectively. Fig. 4.4B shows the discovered discriminative patches (containing triangle or square) for the image classification task in toy example. This is exactly in accordance with 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. Of course, this problem would become trivial if we have prior knowledge of the discriminative local patches and build specific classifiers on them. However, in real-world recognition tasks, it is not easy to figure out the most discriminative local patches for different classes. In addition, even with *ad hoc* knowledge, annotating local patches and training local classifiers often takes large effort. The solution thus becomes non-scalable.

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

$$L_3(\mathbf{W}) = \sum_{\mathbf{X}_m \in \mathcal{T}} -\log\Big(\max_{\mathbf{x}_{mn} \in \mathcal{L}(\mathbf{X}_m)} \sum_{\mathbf{x} \in \mathfrak{N}(\mathbf{x}_{mn})} \mathbf{P}(l_m|\mathbf{x};\mathbf{W})\Big). \tag{4.3}$$

Here, $\mathfrak{N}(\mathbf{x}_{mn})$ denotes the local patches in the neighborhood of $\mathbf{x}_{mn}$. Based on Eq. (4.3), 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.

4.3.3 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. 4.5.

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

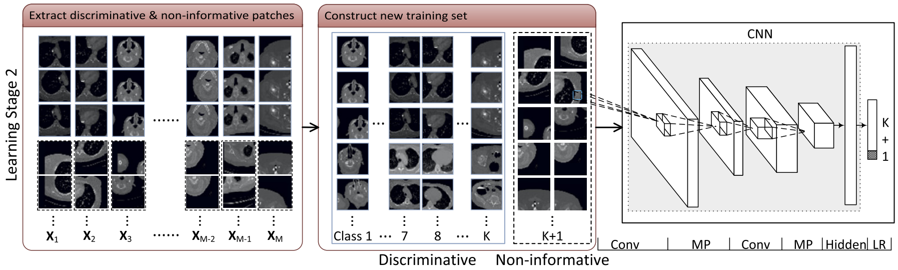


FIGURE 4.5

Illustration of boosting stage. In this stage, CNN architecture is modified by adding a non-informative class in output layer. The parameters are inherited from the pre-trained CNN and fine-tuned using the discriminative and non-informative local regions extracted from each class by the pre-trained CNN.

classes. For each image $\mathbf{X}_m$, we select D discriminative local patches as

$$\mathbf{A}_m = \underset{\mathbf{x}_{mn} \in \mathcal{L}(\mathbf{X}_m)}{\mathrm{argmax}_{\mathrm{D}}} \; \mathbf{P}(l_m | \mathbf{x}_{mn}; \hat{\mathbf{W}}). \tag{4.4}$$

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. $\mathrm{argmax}_{\mathrm{D}}(\cdot)$ 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 body-part recognition. For example, the patches containing lung regions (green dashed boxes in Fig. 4.1) 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 body-part 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. 4.5) besides the existing training classes. This class includes two kinds of local patch: (i) local patch where the pre-trained CNN has higher responses on wrong classes, and (ii) local patch where the pre-trained CNN has "flat" responses across all classes. Denote by $\mathbf{P}(k|\mathbf{x}_{mn}; \hat{\mathbf{W}})$ 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:

$$\begin{aligned}\mathbf{B}_m = \{\mathbf{x}_{mn} | \underset{k\in\{1,\ldots,K\}}{\operatorname{argmax}} \ \mathbf{P}(k|\mathbf{x}_{mn}; \hat{\mathbf{W}}) \neq l_m\} \\ \cup \{\mathbf{x}_{mn} | \underset{k\in\{1,\ldots,K\}}{\operatorname{entropy}} \ \mathbf{P}(k|\mathbf{x}_{mn}; \hat{\mathbf{W}}) > \theta\}.\end{aligned} \tag{4.5}$$

Recalling the toy example, Fig. 4.4B shows the selected discriminative and non-informative local patches. When the discriminative patches from Class 1 and Class 2 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", and other components are misleading. It demonstrates the discovery ability of the method.

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. 4.5) 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 Eq. (4.6):

$$L_4(\mathbf{W}) = \sum_{\mathbf{x}\in\mathbf{A}\bigcup\mathbf{B}} -\log(\mathbf{P}(l|\mathbf{x}; \mathbf{W})). \tag{4.6}$$

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 do not belong to any body section class now, their responses on any body section class can be effectively suppressed during the CNN boosting stage.

4.3.4 RUN-TIME CLASSIFICATION

At runtime, the boosted CNN is applied for body-part 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 $\{\mathbf{P}(k|\mathbf{x}_n; \mathbf{W}^{opt})|k = 1, \ldots, K+1\}$, where $\mathbf{W}^{opt}$ denotes the optimal coefficients of Eq. (4.6). The class of the local patch $\mathbf{x}_n$ is then determined as

$$c(\mathbf{x}_n) = \underset{k\in\{1,\ldots,K+1\}}{\operatorname{argmax}} \ \mathbf{P}(k|\mathbf{x}_n; \mathbf{W}^{opt}). \tag{4.7}$$

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. The most discriminative patch $\mathbf{x}_{n^*}$ in the image is selected as the most peaky correctly labeled one excluding the non-informative patches:

$$\mathbf{x}_{n^*} = \underset{\mathbf{x}_n \in \mathcal{L}(\mathbf{X}); c(\mathbf{x}_n) \neq K+1}{\operatorname{argmax}} \mathbf{P}(c(\mathbf{x}_n)|\mathbf{x}_n; \mathbf{W}^{opt}). \tag{4.8}$$

To generate reliable classification of the image **X**, the effect of possible outlier $\mathbf{x}_{n^*}$ with different prediction of its neighbors can be suppressed by a label fusion of patches to label the image. A simple choice of label fusion is combining the class probabilities in the neighborhood around the most discriminative patch:

$$C(\mathbf{X}) = \underset{k \in \{1,\ldots,K\}}{\operatorname{argmax}} \sum_{\mathbf{x}_n \in \mathfrak{N}(\mathbf{x}_{n^*})} \mathbf{P}(k|\mathbf{x}_n; \mathbf{W}^{opt}). \tag{4.9}$$

4.4 RESULTS

In this study, we mainly compare the multi-instance multi-stage CNN with standard CNN on image classification tasks. In implementation of CNNs, Rectified Linear Units (ReLUs) [48] and Dropout strategy [49] are employed. Dropout rate is 0.5. Data is augmented by up to 10% (relate to image size) random translations to increase training samples. The image patches for multi-instance learning are extracted by fixed-size sliding window with overlapping. As the training set may be too large to load into memory at one time, the models are learned using a mini batch of samples at each iteration. The optimization is implemented by stochastic gradient descent with a momentum term β [50] and a weight decay. The learning process is conducted on a training subset and a validation subset. It will stop if either the error rate on validation subset drops below a threshold or a predefined maximum number of epochs is reached. The framework is implemented in Python using Theano on a 64-bit desktop with i7-2600 (3.4 GHz) CPU, 16 GB RAM and NVIDIA GTX-660 3 GB GPU. Classification accuracies are reported in terms of recall, precision and F_1 score as

$$recall = \frac{TP}{TP + FN}, \quad precision = \frac{TP}{TP + FP}, \tag{4.10}$$

$$F_1 = 2\frac{precision \cdot recall}{precision + recall}, \tag{4.11}$$

where 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.

4.4.1 IMAGE CLASSIFICATION ON SYNTHETIC DATA

A synthetic data set, which has been briefly introduced as a toy example in Section 4.3.2, is constructed by 4 types of geometry elements: triangle, square, circle,

Table 4.1 Classification accuracies (%) on synthetic data set as shown in Fig. 4.4. Class 1 contains triangle; Class 2 contains square

	Triangle and square								
	Recall			**Precision**			F_1		
Class	**1**	**2**	**Total**	**1**	**2**	**Total**	**1**	**2**	**Total**
SCNN	84.2	82.4	83.3	82.7	83.9	83.3	83.5	83.2	83.3
PCNN	99.6	99.7	99.7	99.7	99.6	99.7	99.7	99.7	99.7
BCNN1	98.4	99.7	99.1	99.7	98.4	99.1	99.0	99.1	99.1
BCNN2	**100**	**100**	**100**	**100**	**99.9**	**100**	**100**	**100**	**100**

and diamond. Each synthetic image (60 × 60) contains two of the geometry elements at random positions on black background (intensity value 0). The basic geometry elements are roughly 20 × 20 with variance in height and width. They have random intensity values in $[1, 255]$. In constructing the two image classes, we ensure that the triangle and square are the "distinctive" element and only appear in Class 1 and Class 2, respectively. Circle or diamond is evenly picked as the second element in each image. Some examples of the synthetic images are shown in Fig. 4.4A. Overall, we create 2000 training, 2000 validation, and 2000 for testing samples (balanced distribution for each class).

Four different variants of CNN are compared: (i) standard CNN, as shown in Fig. 4.2, trained on whole image (SCNN); (ii) local patch-based CNN without boost, i.e., the CNN trained by pre-train stage only (PCNN); (iii) local patch-based CNN boosted without additional non-informative class (BCNN1); (iv) local patch-based CNN boosted with both discriminative and non-informative patches (BCNN2). Method (i) represents standard CNN learning (using features extracted from whole image). Methods (ii) and (iii) are two variants of our proposed method (iv), which are presented to verify the effects of each component of our method. All CNNs use the same intermediate architecture: one convolutional layer with 10 5 × 5 filters, one max-pooling layer with a 2 × 2 kernel, one hidden layer of 300 nodes, and finally it is followed by an 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.

As shown in Table 4.1, by leveraging the local discriminative information, PCNN gets $\approx 16\%$ improvement from SCNN. It implies that standard CNN does not fully discover and learn the discriminative local patches, "triangle" and "square". On the contrary, the most discriminative and non-informative local patches are effectively discovered by the CNN with multi-instance learning as shown in Fig. 4.4B. Among our local patch-based CNNs (PCNN, BCNN1, and BCNN2), BCNN1 is worse than PCNN due to overfitting on discriminative patches (because the parameters of BCNN1 are initialized by those of PCNN, and refined by training with the extracted discriminative patches only). BCNN2 is similar to BCNN1 but refined by training with discriminative as well as non-informative patches and achieves the best performance.

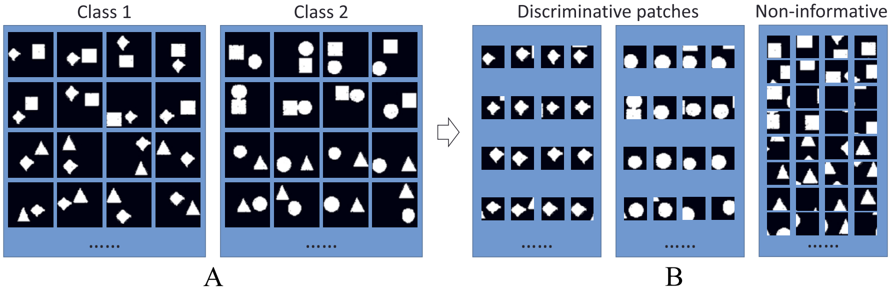


FIGURE 4.6

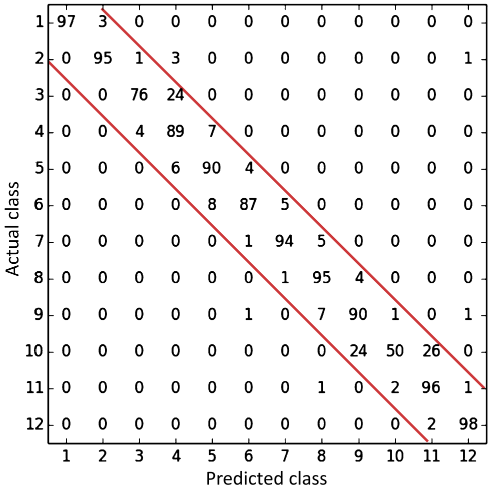
The second toy example. (A) Synthetic images of two classes distinguished by diamond and circle. It is important to note that we used the same image samples as in Fig. 4.3, but re-labeled the images into two classes based on different rules. (B) The discriminative and non-informative local patches discovered by the pre-trained CNN model.

To further prove the key discovery ability 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. 4.6A). 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 first stage learning, the extracted local patches from the learned CNN are shown in Fig. 4.6B. Again, the extracted local patches contain the most discriminative information, diamond and circle. This result demonstrates that our multi-instance CNN learning can *adaptively* learn discriminative local regions for specific classification task without any local level annotations.

4.4.2 BODY-PART RECOGNITION ON CT SLICES

A dataset of 7489 transversal CT slices was collected from whole body scans of 675 patients with very different ages (1–90 year old). The imaging protocols were different: 31 different reconstruction kernels, 0.281–1.953 mm in-slice pixel resolution. This dataset with large variance is good to validate the robustness of the proposed method in practice. As shown in Fig. 4.1, transversal slices of CT scans are categorized into 12 body sections (classes). The body part recognition problem is defined as image classification of the transversal CT slices. The whole dataset is divided into 2413 (225 patients) training, 656 (56 patients) validation, and 4043 (394 patients) testing subsets. We augment data by 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 × 4 mm pixel resolution and 90 × 90 in size. Then, cropping operation (for multi-instance learning) extracts 50 × 50 local patches from each image with 10-pixel step size. Thus, 25 local patches



	1	2	3	4	5	6	7	8	9	10	11	12
1	97	3	0	0	0	0	0	0	0	0	0	0
2	0	95	1	3	0	0	0	0	0	0	0	1
3	0	0	76	24	0	0	0	0	0	0	0	0
4	0	0	4	89	7	0	0	0	0	0	0	0
5	0	0	0	6	90	4	0	0	0	0	0	0
6	0	0	0	0	8	87	5	0	0	0	0	0
7	0	0	0	0	0	1	94	5	0	0	0	0
8	0	0	0	0	0	0	1	95	4	0	0	0
9	0	0	0	0	0	1	0	7	90	1	0	1
10	0	0	0	0	0	0	0	0	24	50	26	0
11	0	0	0	0	0	0	0	1	0	2	96	1
12	0	0	0	0	0	0	0	0	0	0	2	98

FIGURE 4.7

Confusion matrix of BCNN2 on CT data. Values are normalized to 0–100 in each row. Classes are defined in Fig. 4.1.

are extracted per image. Our CNN has similar structure as in Fig. 4.2. C1 layer has 20 9 × 9 filters. C3 layer has 40 9 × 9 filters. Two sub-sampling layer, S2 and S4, use 2 × 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. The learning process takes 440 epochs (≈ 25 hours) in first stage and 70 epochs (≈ 1 hour) in the second stage.

Fig. 4.7 shows detailed classification performance by the confusion matrix. Most errors appear close to the diagonal line, which means most misclassifications happen between the neighboring body sections. Quantitatively, the classification error is 7.79%, 90% of which being "less-than-one neighboring class error" (within the red line corridor of Fig. 4.7). In practice, this type of error is acceptable for some use cases. The remaining gross error (0.8%) can be further suppressed by a simple label smoothing after classifications of a series of continuous slices for 3D body-part identification.

For quantitative comparison (in Table 4.2), tested CNN models include: (i) CaffeNet, (ii) SCNN, (iii) SCNN_a, (iv) PCNN, (v) BCNN1, and (vi) our proposed BCNN2. 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. Methods (iv) and (v) are the variants of (vi) as described in Section 4.4.1. Similar network structure is used in methods (ii)–(vi), except for different input and output sizes. CaffeNet [51] has a similar structure as AlexNet [15] with a minor variation, which is trained on whole images without cropping. We noticed that training of CaffeNet with 50 × 50

Table 4.2 Classification accuracies on CT data (%). Classes are defined in Fig. 4.1

Class	F_1 score					
	CaffeNet	SCNN	SCNN_a	PCNN	BCNN1	BCNN2
1	83.15	86.14	88.42	91.84	70.44	**92.39**
2	78.13	90.38	86.77	93.01	88.48	**94.32**
3	56.38	76.85	55.62	84.44	61.22	**84.47**
4	85.75	88.11	86.56	90.44	84.59	**90.98**
5	82.74	83.89	84.06	87.61	82.88	**88.63**
6	81.96	81.6	84.59	84.35	87.54	**88.42**
7	69.65	85.71	86.64	92.37	92.64	**92.84**
8	89.12	93.89	91.32	95.4	95.31	**95.68**
9	72.73	77.21	63.56	80.18	**81.17**	80.38
10	78.13	76.33	69.21	82.46	77.31	**84.55**
11	84.85	80.89	76.39	83.57	82	**89.75**
12	98.31	96.52	95.38	95.99	96.91	**98.99**
Total	85.78	87.73	85.25	90.45	88.2	**92.23**

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 key 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 in the last column of Table 4.2 (significantly 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 and tends to overfit them. As shown in Table 4.2, 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 is about 3, 4, 4, 10, 11, and 11 ms per image for methods (i) through (vi), respectively. A more detailed comparison can be found in [19].

The discovered discriminative patch samples and non-informative (kind of misleading) patches for each class in the CT dataset are shown in Fig. 4.8. From this figure, we observe that the proposed method "magically" extracts meaningful local patches for each class without any prior information, and these discovered local in-

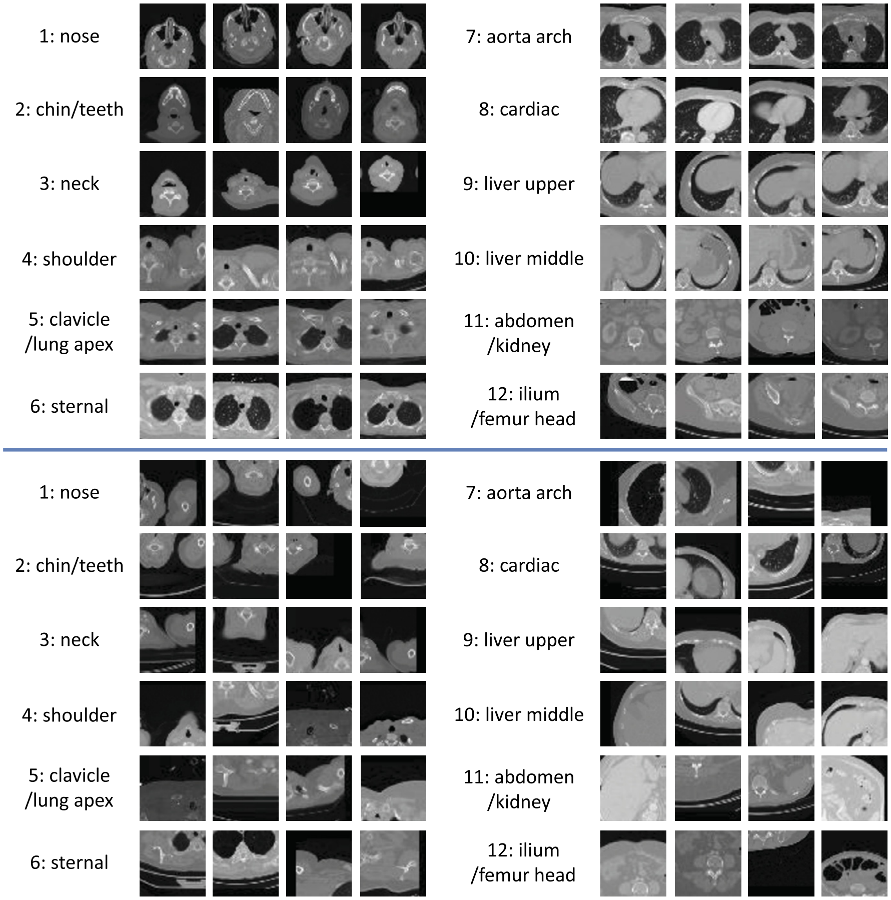


FIGURE 4.8

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

formation can significantly improve the classification task comparing with the global image information. It is also noted that the discriminative patches of liver middle class contain only a small part of the liver and a narrow band of the left lung bottom. The corresponding non-informative patches may contain a large piece of the liver, since it can appear in different classes (e.g. liver upper and abdomen/kidney). This observation indicates that the proposed method just finds the discriminative information from the data, and does not guarantee that some particular "desired" element will be considered as discriminative.

As one of the important parameters in BCNN2 method, step size of sliding window testing is investigated regarding to the accuracies (shown in Fig. 4.9, left). The

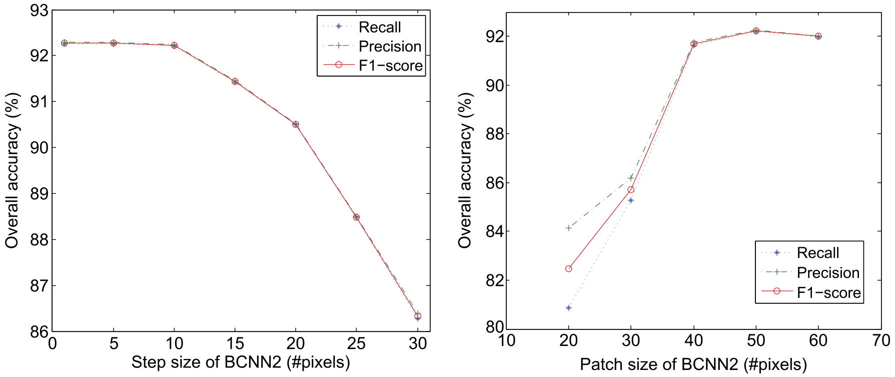


FIGURE 4.9

Performance analyses on the sensitivity of parameters in BCNN2. (Left) Classification accuracies vs. step size of sliding window. (Right) Classification accuracies vs. patch size.

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, 3.6, and 3.4 ms per image, respectively. Considering the balance of running time and accuracy, step size 10 or 15 is a reasonable choice in this task. The effect of patch size to the classification accuracy is also investigated as shown in Fig. 4.9, right. We can see from the plot that (i) the patch size should not be too small to capture the discriminative information (size 20 or 30); (ii) 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).

4.5 DISCUSSION AND FUTURE WORK

In this chapter, we introduced a multi-instance multi-stage deep learning framework for medical image classification. The framework discovers the discriminative and non-informative local patches via multi-instance learning, and employs multi-stage learning strategy to learn CNN for the image classification task. From the validations on synthetic dataset and a large scale CT dataset, we observed clear improvements compared with other state-of-the-art methods. It is proved that the success of the proposed method against the standard CNN does not result from more augmented training samples (see the results of SCNN_a vs SCNN), but rather results from its capability of *discovering* local characteristics of different body parts.

In fact, the method may benefit radiological workflow in different aspects. For example, a very reliable and fast anatomy recognition algorithm can make the planning of the scanning range in topogram or scout scans be conducted on-the-fly to significantly save scanning time. In another example, an automatic image-based recognition will enable content-based image retrieval and improve the query precision in PACS system. Besides, it can serve as an initialization module for other higher level medical image interpretation tasks, e.g. anatomy detection or segmentation, to make the workflow more efficient. With the precise search ranges, detection/segmentation speed and robustness can be benefited. Moreover, it can help in medical image preprocessing by gating the applicable auto-algorithms before being loaded for manual reading. In this way, the meaningful and automatic results can be displayed instantaneously in the reading room to speed up radiologists' reading process.

It is worth noting that since no manual annotations are required to label these local patches, our method becomes very scalable. This weakly 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. It can be used to discover and extract discriminative information in different classes to help in obtaining better insight in some problems. One limitation of this method is the identical patch size for different classes [19]. It requires prior knowledge to ensure that the chosen patch size is able to include the discriminative information in a fixed size region. It could be better to incorporate strategies like multi-scale convolution [52] or multi-scale image patch [53] to discover and recognize different-size discriminative local regions in different classes. Another limitation is the sliding-window and multiple stage pipeline. Like the improvement from R-CNN [54] to fast R-CNN [39] and faster R-CNN [40], it may be possible to simplify the multi-stage pipeline to single-stage training, and use part of the convolutional features of image for local regions to avoid the sliding-window strategy in training and testing.

This 2D slice based body part recognition can be trivially applied in 3D image data by labeling each slice one by one. Considering that no more than 7% error locating between continuous sections is acceptable in practice, the only gross error (less than 1%) can be easily eliminated by a smoothing filter on the predicted label distribution. This framework can also be extended to handle 3D cases. One way is to treat multiple slice as a multi-channel input (like the RGB image). Another way is using 3D convolutional filters in the CNN.

Another possible direction of improvement would be the multi-modal deep learning [55,56]. It is not limited in just multiple image modalities [57]. Since shape prior models [58–60] have shown the benefits in many applications, the 3D geometric information should be able to act as a complementary modality besides visual appearance to be incorporated in deep learning frameworks [61,62].

REFERENCES

1. T. McInerney, D. Terzopoulos, Deformable models in medical image analysis: a survey, Med. Image Anal. 1 (2) (1996) 91–108.
2. J.A. Maintz, M.A. Viergever, A survey of medical image registration, Med. Image Anal. 2 (1) (1998) 1–36.
3. D.L. Pham, C. Xu, J.L. Prince, Current methods in medical image segmentation, Annu. Rev. Biomed. Eng. 2 (1) (2000) 315–337.
4. D.L. Hill, P.G. Batchelor, M. Holden, D.J. Hawkes, Medical image registration, Phys. Med. Biol. 46 (3) (2001) R1.

5. M. Betke, H. Hong, D. Thomas, C. Prince, J.P. Ko, Landmark detection in the chest and registration of lung surfaces with an application to nodule registration, Med. Image Anal. 7 (3) (2003) 265–281.
6. Y. Zheng, M. John, R. Liao, J. Boese, U. Kirschstein, B. Georgescu, S.K. Zhou, J. Kempfert, T. Walther, G. Brockmann, et al., Automatic aorta segmentation and valve landmark detection in C-arm CT: application to aortic valve implantation, in: Medical Image Computing and Computer-Assisted Intervention, Springer, 2010, pp. 476–483.
7. D. Shen, S. Moffat, S.M. Resnick, C. Davatzikos, Measuring size and shape of the hippocampus in MR images using a deformable shape model, NeuroImage 15 (2) (2002) 422–434.
8. H. Ling, S.K. Zhou, Y. Zheng, B. Georgescu, M. Suehling, D. Comaniciu, Hierarchical, learning-based automatic liver segmentation, in: IEEE International Conference on Computer Vision and Pattern Recognition, IEEE, 2008, pp. 1–8.
9. C. Li, R. Huang, Z. Ding, J.C. Gatenby, D.N. Metaxas, J.C. Gore, A level set method for image segmentation in the presence of intensity inhomogeneities with application to MRI, IEEE Trans. Image Process. 20 (7) (2011) 2007–2016.
10. R. Bellotti, F. De Carlo, G. Gargano, S. Tangaro, D. Cascio, E. Catanzariti, P. Cerello, S.C. Cheran, P. Delogu, I. De Mitri, et al., A CAD system for nodule detection in low-dose lung CTs based on region growing and a new active contour model, Med. Phys. 34 (12) (2007) 4901–4910.
11. L.A. Meinel, A.H. Stolpen, K.S. Berbaum, L.L. Fajardo, J.M. Reinhardt, Breast MRI lesion classification: improved performance of human readers with a backpropagation neural network computer-aided diagnosis (CAD) system, J. Magn. Reson. Imaging 25 (1) (2007) 89–95.
12. K. Doi, Current status and future potential of computer-aided diagnosis in medical imaging, Br. J. Radiol. 78 (suppl_1) (2005) s1–s19.
13. Y. LeCun, Y. Bengio, G. Hinton, Deep learning, Nature 521 (7553) (2015) 436–444.
14. G. Hinton, L. Deng, D. Yu, G.E. Dahl, A.r. Mohamed, N. Jaitly, A. Senior, V. Vanhoucke, P. Nguyen, T.N. Sainath, et al., Deep neural networks for acoustic modeling in speech recognition: the shared views of four research groups, IEEE Signal Process. Mag. 29 (6) (2012) 82–97.
15. A. Krizhevsky, I. Sutskever, G.E. Hinton, ImageNet classification with deep convolutional neural networks, in: Advances in Neural Information Processing Systems, 2012, pp. 1097–1105.
16. I. Sutskever, O. Vinyals, Q.V. Le, Sequence to sequence learning with neural networks, in: Advances in Neural Information Processing Systems, 2014, pp. 3104–3112.
17. Y. Bengio, A. Courville, P. Vincent, Representation learning: a review and new perspectives, IEEE Trans. Pattern Anal. Mach. Intell. 35 (8) (2013) 1798–1828.
18. Z. Yan, Y. Zhan, Z. Peng, S. Liao, Y. Shinagawa, D.N. Metaxas, X.S. Zhou, Bodypart recognition using multi-stage deep learning, in: International Conference on Information Processing in Medical Imaging, Springer, 2015, pp. 449–461.
19. Z. Yan, Y. Zhan, Z. Peng, S. Liao, Y. Shinagawa, S. Zhang, D. Metaxas, X. Zhou, Multi-instance deep learning: discover discriminative local anatomies for bodypart recognition, IEEE Trans. Med. Imaging (2016) 1332–1343.
20. M.O. Gueld, M. Kohnen, D. Keysers, H. Schubert, B.B. Wein, J. Bredno, T.M. Lehmann, Quality of DICOM header information for image categorization, in: Medical Imaging, International Society for Optics and Photonics, 2002, pp. 280–287.

21. Y. LeCun, L. Bottou, Y. Bengio, P. Haffner, Gradient-based learning applied to document recognition, Proc. IEEE 86 (11) (1998) 2278–2324.
22. C. Szegedy, A. Toshev, D. Erhan, Deep neural networks for object detection, in: Advances in Neural Information Processing Systems, 2013, pp. 2553–2561.
23. Y. Wei, W. Xia, J. Huang, B. Ni, J. Dong, Y. Zhao, S. Yan, CNN: single-label to multi-label, arXiv:1406.5726, 2014.
24. C. Cortes, V. Vapnik, Support-vector networks, Mach. Learn. 20 (3) (1995) 273–297.
25. D.W. Hosmer Jr., S. Lemeshow, Applied Logistic Regression, John Wiley & Sons, 2004.
26. D.G. Lowe, Distinctive image features from scale-invariant keypoints, Int. J. Comput. Vis. 60 (2) (2004) 91–110.
27. N. Dalal, B. Triggs, Histograms of oriented gradients for human detection, in: IEEE International Conference on Computer Vision and Pattern Recognition, vol. 1, IEEE, 2005, pp. 886–893.
28. D. Yang, S. Zhang, Z. Yan, C. Tan, K. Li, D. Metaxas, Automated anatomical landmark detection on distal femur surface using convolutional neural network, in: IEEE International Symposium on Biomedical Imaging, IEEE, 2015, pp. 17–21.
29. H.C. Shin, H.R. Roth, M. Gao, L. Lu, Z. Xu, I. Nogues, J. Yao, D. Mollura, R.M. Summers, Deep convolutional neural networks for computer-aided detection: CNN architectures, dataset characteristics and transfer learning, IEEE Trans. Med. Imaging 35 (5) (2016) 1285–1298.
30. S. Albarqouni, C. Baur, F. Achilles, V. Belagiannis, S. Demirci, N. Navab, AggNet: deep learning from crowds for mitosis detection in breast cancer histology images, IEEE Trans. Med. Imaging 35 (5) (2016) 1313–1321.
31. Y. Guo, G. Wu, L.A. Commander, S. Szary, V. Jewells, W. Lin, D. Shen, Segmenting hippocampus from infant brains by sparse patch matching with deep-learned features, in: Medical Image Computing and Computer-Assisted Intervention, Springer, 2014, pp. 308–315.
32. O. Ronneberger, P. Fischer, T. Brox, U-Net: convolutional networks for biomedical image segmentation, in: Medical Image Computing and Computer-Assisted Intervention, Springer, 2015, pp. 234–241.
33. T. Brosch, L. Tang, Y. Yoo, D. Li, A. Traboulsee, R. Tam, Deep 3D convolutional encoder networks with shortcuts for multiscale feature integration applied to multiple sclerosis lesion segmentation, IEEE Trans. Med. Imaging 35 (5) (2016) 1229–1239.
34. P. Moeskops, M.A. Viergever, A.M. Mendrik, L.S. de Vries, M.J. Benders, I. Isgum, Automatic segmentation of MR brain images with a convolutional neural network, IEEE Trans. Med. Imaging 35 (5) (2016) 1252–1261.
35. M. Anthimopoulos, S. Christodoulidis, L. Ebner, A. Christe, S. Mougiakakou, Lung pattern classification for interstitial lung diseases using a deep convolutional neural network, IEEE Trans. Med. Imaging 35 (5) (2016) 1207–1216.
36. H.R. Roth, C.T. Lee, H.C. Shin, A. Seff, L. Kim, J. Yao, L. Lu, R.M. Summers, Anatomy-specific classification of medical images using deep convolutional nets, in: IEEE International Symposium on Biomedical Imaging, 2015, pp. 101–104.
37. Y. Taigman, M. Yang, M. Ranzato, L. Wolf, DeepFace: closing the gap to human-level performance in face verification, in: IEEE International Conference on Computer Vision and Pattern Recognition, IEEE, 2014, pp. 1701–1708.
38. M.M. Cheng, Z. Zhang, W.Y. Lin, P. Torr, BING: binarized normed gradients for objectness estimation at 300 fps, in: IEEE International Conference on Computer Vision and Pattern Recognition, IEEE, 2014, pp. 3286–3293.

39. R. Girshick, Fast R-CNN, in: IEEE International Conference on Computer Vision, 2015, pp. 1440–1448.
40. S. Ren, K. He, R. Girshick, J. Sun, Faster R-CNN: towards real-time object detection with region proposal networks, in: Advances in Neural Information Processing Systems, 2015, pp. 91–99.
41. Y. Zhan, X.S. Zhou, Z. Peng, A. Krishnan, Active scheduling of organ detection and segmentation in whole-body medical images, in: Medical Image Computing and Computer-Assisted Intervention, Springer, 2008, pp. 313–321.
42. A. Criminisi, J. Shotton, D. Robertson, E. Konukoglu, Regression forests for efficient anatomy detection and localization in CT studies, in: Medical Computer Vision. Recognition Techniques and Applications in Medical Imaging, Springer, 2011, pp. 106–117.
43. R. Donner, B.H. Menze, H. Bischof, G. Langs, Global localization of 3D anatomical structures by pre-filtered Hough Forests and discrete optimization, Med. Image Anal. 17 (8) (2013) 1304–1314.
44. P.O. Pinheiro, R. Collobert, From image-level to pixel-level labeling with convolutional networks, in: IEEE International Conference on Computer Vision and Pattern Recognition, 2015, pp. 1713–1721.
45. G. Papandreou, L.C. Chen, K. Murphy, A.L. Yuille, Weakly- and semi-supervised learning of a DCNN for semantic image segmentation, arXiv:1502.02734, 2015.
46. J. Wu, Y. Yu, C. Huang, K. Yu, Deep multiple instance learning for image classification and auto-annotation, in: IEEE International Conference on Computer Vision and Pattern Recognition, IEEE, 2015, pp. 3460–3469.
47. O. Maron, T. Lozano-Pérez, A framework for multiple-instance learning, in: Advances in Neural Information Processing Systems, 1998, pp. 570–576.
48. V. Nair, G.E. Hinton, Rectified linear units improve restricted Boltzmann machines, in: International Conference on Machine Learning, 2010, pp. 807–814.
49. N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, R. Salakhutdinov, Dropout: a simple way to prevent neural networks from overfitting, J. Mach. Learn. Res. 15 (1) (2014) 1929–1958.
50. N. Qian, On the momentum term in gradient descent learning algorithms, Neural Netw. 12 (1) (1999) 145–151.
51. Y. Jia, E. Shelhamer, J. Donahue, S. Karayev, J. Long, R. Girshick, S. Guadarrama, T. Darrell, Caffe: convolutional architecture for fast feature embedding, in: Proceedings of the ACM International Conference on Multimedia, ACM, 2014, pp. 675–678.
52. P. Sermanet, Y. LeCun, Traffic sign recognition with multi-scale convolutional networks, in: The 2011 International Joint Conference on Neural Networks (IJCNN), IEEE, 2011, pp. 2809–2813.
53. P. Felzenszwalb, D. McAllester, D. Ramanan, A discriminatively trained, multiscale, deformable part model, in: IEEE International Conference on Computer Vision and Pattern Recognition, IEEE, 2008, pp. 1–8.
54. R. Girshick, J. Donahue, T. Darrell, J. Malik, Rich feature hierarchies for accurate object detection and semantic segmentation, in: IEEE International Conference on Computer Vision and Pattern Recognition, 2014, pp. 580–587.
55. J. Ngiam, A. Khosla, M. Kim, J. Nam, H. Lee, A.Y. Ng, Multimodal deep learning, in: International Conference on Machine Learning, 2011, pp. 689–696.
56. A. Karpathy, F.F. Li, Deep visual-semantic alignments for generating image descriptions, in: IEEE International Conference on Computer Vision and Pattern Recognition, 2015, pp. 3128–3137.

57. W. Zhang, R. Li, H. Deng, L. Wang, W. Lin, S. Ji, D. Shen, Deep convolutional neural networks for multi-modality isointense infant brain image segmentation, NeuroImage 108 (2015) 214–224.
58. T.F. Cootes, C.J. Taylor, D.H. Cooper, J. Graham, Active shape models – their training and application, Comput. Vis. Image Underst. 61 (1) (1995) 38–59.
59. S. Zhang, Y. Zhan, M. Dewan, J. Huang, D.N. Metaxas, X.S. Zhou, Towards robust and effective shape modeling: sparse shape composition, Med. Image Anal. 16 (1) (2012) 265–277.
60. S. Zhang, Y. Zhan, D.N. Metaxas, Deformable segmentation via sparse representation and dictionary learning, Med. Image Anal. 16 (7) (2012) 1385–1396.
61. Z. Wu, S. Song, A. Khosla, F. Yu, L. Zhang, X. Tang, J. Xiao, 3D shapenets: a deep representation for volumetric shapes, in: IEEE International Conference on Computer Vision and Pattern Recognition, 2015, pp. 1912–1920.
62. Y. Fang, J. Xie, G. Dai, M. Wang, F. Zhu, T. Xu, E. Wong, 3D deep shape descriptor, in: IEEE International Conference on Computer Vision and Pattern Recognition, 2015, pp. 2319–2328.