Chapter 4
Robust Landmark Detection in Volumetric Data with Efficient 3D Deep Learning

Yefeng Zheng, David Liu, Bogdan Georgescu, Hien Nguyen and Dorin Comaniciu

Abstract Recently, deep learning has demonstrated great success in computer vision with the capability to learn powerful image features from a large training set. However, most of the published work has been confined to solving 2D problems, with a few limited exceptions that treated the 3D space as a composition of 2D orthogonal planes. The challenge of 3D deep learning is due to a much larger input vector, compared to 2D, which dramatically increases the computation time and the chance of over-fitting, especially when combined with limited training samples (hundreds to thousands), typical for medical imaging applications. To address this challenge, we propose an efficient and robust deep learning algorithm capable of full 3D detection in volumetric data. A two-step approach is exploited for efficient detection. A shallow network (with one hidden layer) is used for the initial testing of all voxels to obtain a small number of promising candidates, followed by more accurate classification with a deep network. In addition, we propose two approaches, i.e., separable filter decomposition and network sparsification, to speed up the evaluation of a network. To mitigate the over-fitting issue, thereby increasing detection robustness, we extract small 3D patches from a multi-resolution image pyramid. The deeply learned image features are further combined with Haar wavelet-like features to increase the detection accuracy. The proposed method has been quantitatively evaluated for carotid artery bifurcation detection on a head-neck CT dataset from 455 patients. Compared to the state of the art, the mean error is reduced by more than half, from 5.97 mm to 2.64 mm, with a detection speed of less than 1 s/volume.

4.1 Introduction

An anatomical landmark is a biologically meaningful point on an organism, which can be easily distinguished from surrounding tissues. Normally, it is consistently present across different instances of the same organism so that it can be used to

Y. Zheng (✉) · D. Liu · B. Georgescu · H. Nguyen · D. Comaniciu
Medical Imaging Technologies, Siemens Healthcare, Princeton, NJ, USA
e-mail: yefeng.zheng@siemens.com

L. Lu et al. (eds.), *Deep Learning and Convolutional Neural Networks for Medical Image Computing*, Advances in Computer Vision and Pattern Recognition, DOI 10.1007/978-3-319-42999-1_4

establish anatomical correspondence within the population. There are many applications of automatic anatomical landmark detection in medical image analysis. For example, landmarks can be used to align an input volume to a canonical plane on which physicians routinely perform diagnosis and quantification [1, 2]. A detected vascular landmark provides a seed point for automatic vessel centerline extraction and lumen segmentation [3, 4]. For a nonrigid object with large variation, a holistic detection may not be robust. Aggregation of the detection results of multiple landmarks on the object may provide a more robust solution [5]. In some applications, the landmarks themselves provide important measurements for disease quantification and surgical planning (e.g., the distance from coronary ostia to the aortic hinge plane is a critical indicator whether the patient is a good candidate for transcatheter aortic valve replacement [6]).

Various landmark detection methods have been proposed in the literature. Most of the state-of-the-art algorithms [1–6] apply machine learning (e.g., support vector machines, random forests, or boosting algorithms) on a set of handcrafted image features (e.g., SIFT features or Haar wavelet-like features). However, in practice, we found some landmark detection problems (e.g., carotid artery bifurcation landmarks in this work) are still too challenging to be solved with the current technology.

Deep learning [7] has demonstrated great success in computer vision with the capability to learn powerful image features (either supervised or unsupervised) from a large training set. Recently, deep learning has been applied in many medical image analysis problems, including body region recognition [8], cell detection [9], lymph node detection [10], organ detection/segmentation [11, 12], cross-modality registration [13], and 2D/3D registration [14]. On all these applications, deep learning outperforms the state of the art.

However, several challenges are still present in applying deep learning to 3D landmark detection. Normally, the input to a neural network classifier is an image patch, which increases dramatically in size from 2D to 3D. For example, a patch of 32×32 pixels generates an input of 1024 dimensions to the classifier. However, a $32 \times 32 \times 32$ 3D patch contains 32,768 voxels. Such a big input feature vector creates several challenges. First, the computation time of a deep neural network is often too slow for a real clinical application. The most widely used and robust approach for object detection is the *sliding window* based approach, in which the trained classifier is tested on each voxel in the volume. Evaluating a deep network on a large volume may take several minutes. Second, as a rule of thumb, a network with a bigger input vector requires more training data. With enough training samples (e.g., over 10 million in ImageNet), deep learning has demonstrated impressive performance gain over other methods. However, the medical imaging community is often struggling with limited training samples (often in hundreds or thousands) due to the difficulty to generate and share images. Several approaches can tackle or at least mitigate the issue of limited training samples. One approach is to reduce the patch size. For example, if we reduce the patch size from $32 \times 32 \times 32$ voxels to $16 \times 16 \times 16$, we can reduce the input dimension by a factor of eight. However, a small patch may not contain enough information for classification. Alternatively, instead of sampling a 3D patch, we can sample on three orthogonal planes [15] or even a 2D patch with a random

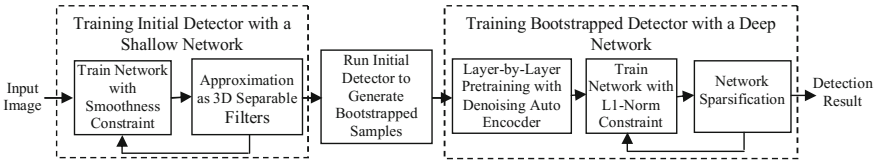

Fig. 4.1 Training procedure of the proposed deep network based 3D landmark detection method

orientation [10]. Although they can effectively reduce the input dimension, there is a concern on how much 3D information is contained in 2D planes.

In this work we tackle the above challenges in the application of deep learning for 3D anatomical structure detection (focusing on landmarks). Our approach significantly accelerates the detection speed, resulting in an efficient method that can detect a landmark in less than one second. We apply a two-stage classification strategy (as shown in Fig. 4.1). In the first stage, we train a shallow network with only one small hidden layer (e.g., with 64 hidden nodes). This network is applied to test all voxels in the volume in a sliding window process to generate 2000 candidates for the second-stage classification. The second network is much bigger with three hidden layers (each has 2000 nodes) to obtain more discriminative power. Such a cascaded classification approach has been widely used in object detection to improve detection efficiency and robustness.

In this work we propose two techniques to further accelerate the detection speed: separable filter approximation for the first-stage classifier and network sparsification for the second-stage classifier. The weights of a node in the first hidden layer are often treated as a filter (3D in this case). The response of the first hidden layer over the volume can be calculated as a convolution with the filter. Here, a neighboring patch is shifted by only one voxel; however, the response needs to be recalculated from scratch. In this work we approximate the weights as separable filters using tensor decomposition. Therefore, a direct 3D convolution is decomposed as three one-dimensional convolutions along the x, y, and z axis, respectively. Previously, such approximation has been exploited for 2D classification problems [16, 17]. However, in 3D, the trained filters are more difficult to be approximated as separable filters. We propose a new training cost function to enforce smoothness of the filters so that they can be approximated with high accuracy. The second big network only applies on a small number of candidates that have little correlation. Separable filter approximation does not help to accelerate classification. However, many weights in a big network are close to zero. We propose to add L1-norm regularization to the cost function to drive majority of the weights (e.g., 90%) to zero, resulting in a sparse network with increased classification efficiency without deteriorating accuracy.

The power of deep learning is on the automatic learning of a hierarchical image representation (i.e., image features). Instead of using the trained network as a classifier, we can use the responses at each layer (including the input layer, all hidden layers, and the output layer) as features and feed them into other state-of-the-art classifiers (e.g., boosting). After years of feature engineering, some handcrafted features

have considerable discriminative power for some applications and they may be complimentary to deeply learned features. In this work we demonstrate that combining deeply learned features and Haar wavelet-like features, we can reduce the detection failures.

The remainder of this chapter is organized as follows. In Sect. 4.2 we present a new method to train a shallow network with separable filters, which are efficient in a sliding window based detection scheme to prune the landmark candidates. Section 4.3 describes a sparse network that can effectively accelerate the evaluation of a deep network, which is used to further test the preserved landmark candidates. We present a feature fusion approach in Sect. 4.4 to combine Haar wavelet-like features and deeply learned features to improve the landmark detection accuracy. Experiments on a large dataset in Sect. 4.5 demonstrate the robustness and efficiency of the proposed method. This chapter concludes with Sect. 4.6. Please note, an early version of this work was published in [18].

4.2 Training Shallow Network with Separable Filters

A fully connected multilayer perceptron (MLP) neural network is a layered architecture. Suppose the input is a n_0-dimensional vector $[X_1^0, X_2^0, \ldots, X_{n_0}^0]$. The response of a node X_j^1 of the first hidden layer is

$$X_j^1 = g\left(\sum_{i=1}^{n_0} W_{i,j}^0 X_i^0 + b_j^0\right), \tag{4.1}$$

for $j = 1, 2, \ldots, n_1$ (n_1 is the number of nodes in the first hidden layer). Here, $W_{i,j}^0$ is a weight; b_j^0 is a bias term; And, $g(.)$ is a nonlinear function, which can be sigmoid, hypo-tangent, restricted linear unit (ReLU), or other forms. In this work we use the sigmoid function

$$g(x) = \frac{1}{1 + e^{-x}}, \tag{4.2}$$

which is the most popular nonlinear function. If we denote $\mathbf{X}^0 = [X_1^0, \ldots, X_{n_0}^0]^T$ and $\mathbf{W}_j^0 = [W_{1,j}^0, \ldots, W_{n_0,j}^0]^T$, Eq. (4.1) can be rewritten as $X_j^1 = g\left((\mathbf{W}_j^0)^T \mathbf{X}^0 + b_j^0\right)$. Multiple layers can be stacked together using Eq. (4.1) as a building block. For a binary classification problem as this work, the output of the network can be a single node $\hat{X}$. Suppose there are L hidden layers, the output of the neural network is $\hat{X} = g\left((\mathbf{W}^L)^T \mathbf{X}^L + b^L\right)$. During network training, we require the output to match the class label Y (with 1 for the positive class and 0 for negative) by minimizing the squared error $E = ||Y - \hat{X}||^2$.

In object detection using a sliding window based approach, for each position hypothesis, we crop an image patch (with a predefined size) centered at the position

hypothesis. We then serialize the patch intensities into a vector as the input to calculate response $\hat{X}$. After testing a patch, we shift the patch by one voxel (e.g., to the right) and repeat the above process again. Such a naive implementation is time consuming. Coming back to Eq. (4.1), we can treat the weights of a node in the first hidden layer as a filter. The first term of the response is a dot-product of the filter and the image patch intensities. Shifting the patch over the whole volume is equivalent to convolution using the filter. Therefore, alternatively, we can perform convolution using each filter $\mathbf{W}_j^0$ for $j = 1, 2, \ldots, n_1$ and cache the response maps. During object detection, we can use the cached maps to retrieve the response of the first hidden layer.

Although such an alternative approach does not save computation time, it gives us a hint for speedup. With a bit abuse of symbols, suppose $\mathbf{W}_{x,y,z}$ is a 3D filter with size $n_x \times n_y \times n_z$. Let us further assume that $\mathbf{W}_{x,y,z}$ is separable, which means we can find three one-dimensional vectors, $\mathbf{W}_x$, $\mathbf{W}_y$, $\mathbf{W}_z$, such that

$$\mathbf{W}_{x,y,z}(i, j, k) = \mathbf{W}_x(i).\mathbf{W}_y(j).\mathbf{W}_z(k) \tag{4.3}$$

for any $i \in [1, n_x]$, $j \in [1, n_y]$, and $k \in [1, n_z]$. The convolution of the volume with $\mathbf{W}_{x,y,z}$ is equivalent to three sequential convolutions with $\mathbf{W}_x$, $\mathbf{W}_y$, and $\mathbf{W}_z$ along its corresponding axis. Sequential convolution with one-dimensional filters is much more efficient than direct convolution with a 3D filter, especially for a large filter. However, in reality, Eq. (4.3) is just an approximation of filters learned by a neural network and such a rank-1 approximation is poor in general. In this work we search for S sets of separable filters to approximate the original filter as

$$\mathbf{W}_{x,y,z} \approx \sum_{s=1}^{S} \mathbf{W}_x^s.\mathbf{W}_y^s.\mathbf{W}_z^s. \tag{4.4}$$

Please note, with a sufficient number of separable filters (e.g., $S \geq \min\{n_x, n_y, n_z\}$), we can reconstruct the original filter perfectly.

To achieve detection efficiency, we need to cache $n_1 \times S$ filtered response maps. If the input volume is big (the size of a typical CT scan in our dataset is about 300 MB) and n_1 is relatively large (e.g., 64 or more), the cached response maps consume a lot of memory. Fortunately, the learned filters $\mathbf{W}_1^0, \ldots, \mathbf{W}_{n_1}^0$ often have strong correlation (i.e., a filter can be reconstructed by a linear combination of other filters). We do not need to maintain a different filter bank for each $\mathbf{W}_i^0$. The separable filters in reconstruction can be drawn from the same bank,

$$\mathbf{W}_i^0 \approx \sum_{s=1}^{S} c_{i,s}.\mathbf{W}_x^s.\mathbf{W}_y^s.\mathbf{W}_z^s. \tag{4.5}$$

Here, $c_{i,s}$ is the combination coefficient, which is specific for each filter $\mathbf{W}_i^0$. However, $\mathbf{W}_x^s$, $\mathbf{W}_y^s$, and $\mathbf{W}_z^s$ are shared by all filters. Equation (4.5) is a rank-S decomposition of a 4D tensor $[\mathbf{W}_1^0, \mathbf{W}_2^0, \ldots, \mathbf{W}_{n_1}^0]$, which can be solved using [19].

Using 4D tensor decomposition, we only need to convolve the volume S times (instead of $n_1.S$ times using 3D tensor decomposition) and cache S response maps. Suppose the input volume has $N_x \times N_y \times N_z$ voxels. For each voxel, we need to do $n_x n_y n_z$ multiplications using the original sliding window based approach. To calculate the response of a hidden layer with n_1 nodes, the total number of multiplications is $n_1 n_x n_y n_z N_x N_y N_z$. Using the proposed approach, to perform convolution with S set of separable filters, we need do $S(n_x + n_y + n_z)N_x N_y N_z$ multiplications. To calculate the response of n_1 hidden layer nodes, we need to combine the S responses using Eq. (4.5), resulting in $n_1 S N_x N_y N_z$ multiplications. The total number of multiplications is $S(n_x + n_y + n_z + n_1)N_x N_y N_z$. Suppose $S = 32$, $n_1 = 64$, the speedup is 62 times for a $15 \times 15 \times 15$ patch.

To achieve significant speedup and save memory footprint, we need to reduce S as much as possible. However, we found, with a small S (e.g., 32), it was more difficult to approximate 3D filters than 2D filters [16, 17]. Nonlinear functions $g(.)$ are exploited in neural networks to bound the response to a certain range (e.g., [0, 1] using the sigmoid function). Many nodes are saturated (with an output close to 0 or 1) and once a node is saturated, its response is not sensitive to the change of the weights. Therefore, a weight can take an extremely large value, resulting in a non-smooth filter. Here, we propose to modify the objective function to encourage the network to generate smooth filters

$$E = ||Y - \hat{X}||^2 + \alpha \sum_{i=1}^{n_1} ||\mathbf{W}_i^0 - \overline{\mathbf{W}_i^0}||^2. \tag{4.6}$$

Here, $\overline{\mathbf{W}_i^0}$ is the mean value of the weights of filter $\mathbf{W}_i^0$. So, the second term measures the variance of the filter weights. Parameter α (often takes a small value, e.g., 0.001) keeps a balance between two terms in the objective function. The proposed smooth regularization term is different to the widely used L2-norm regularization, which is as follows

$$E = ||Y - \hat{X}||^2 + \alpha \sum_{j=1}^{L} \sum_{i=1}^{n_j} ||\mathbf{W}_i^0||^2. \tag{4.7}$$

The L2-norm regularization applies to all weights, while our regularization applies only to the first hidden layer. Furthermore, L2-norm regularization encourages small weights, therefore shrinks the capacity of the network; while our regularization encourages small variance of the weights.

The training of the initial shallow network detector is as follows (as shown in the left dashed box of Fig. 4.1). (1) Train a network using Eq. (4.6). (2) Approximate the learned filters using a filter bank with S ($S = 32$ in our experiments) sets of separable

filters to minimize the error of Eq. (4.5). The above process may be iterated a few times (e.g., three times). In the first iteration, the network weights and filter bank are initialized with random values. However, in the following iterations, they are both initialized with the optimal values from the previous iteration.

Previously, separable filter approximation has been exploited for 2D classification problems [16, 17]. We found 3D filters were more difficult to be approximated well with a small filter bank; therefore, we propose a new objective function to encourage the network to generate smooth filters for higher separability. Furthermore, unlike [17], we also iteratively retrain the network to compensate the loss of accuracy due to approximation.

4.3 Training Sparse Deep Network

Using a shallow network, we can efficiently test all voxels in the volume and assign a detection score to each voxel. After that, we preserve 2000 candidates with the largest detection scores. The number of preserved candidates is tuned to have a high probability to include the correct detection (e.g., hypotheses within one-voxel distance to the ground truth). However, most of the preserved candidates are still false positives. In the next step, we train a deep network to further reduce the false positives. The classification problem is now much tougher and a shallow network does not work well. In this work we use a big network with three hidden layers, each with 2000 nodes.

Even though we only need to classify a small number of candidates, the computation may still take some time since the network is now much bigger. Since the preserved candidates are often scattered over the whole volume, separable filter decomposition as used in the initial detection stage does not help to accelerate the classification. After checking the values of the learned weights of this deep network, we found most of weights were very small, close to zero. That means many connections in the network can be removed without sacrificing classification accuracy. Here, we apply L1-norm regularization to enforce sparse connection

$$E = ||Y - \hat{X}||^2 + \beta \sum_{j=1}^{L} \sum_{i=1}^{n_j} ||\mathbf{W}_i^j||. \tag{4.8}$$

Parameter β can be used to tune the number of zero weights. The higher β is, the more weights converge to zero. With a sufficient number of training epochs, part of weights converges exactly to zero. In practice, to speed up the training, we periodically check the magnitude of weights. The weights with a magnitude smaller than a threshold are set to zero and the network is refined again. In our experiments, we find that 90% of the weights can be set to zero after training, without deteriorating the classification accuracy. Thus, we can speed up the classification by roughly ten times.

The proposed acceleration technologies can be applied to different neural network architectures, e.g., a multilayer perceptron (MLP) and a convolutional neural network (CNN). In this work we use the MLP. While the shallow network is trained with back-propagation to directly minimize the objective function in Eq. (4.6), the deep network is pretrained using the denoising auto-encoder criterion [7] and then fine-tuned to minimize Eq. (4.8). The right dashed box of Fig. 4.1 shows the training procedure of the sparse deep network.

4.4 Robust Detection by Combining Multiple Features

To train a robust neural network based landmark detector on limited training samples, we have to control the patch size. The optimal patch size was searched and we found a size of $15 \times 15 \times 15$ achieved a good trade-off between detection speed and accuracy. However, a small patch has a limited field-of-view, thereby may not capture enough information for classification. In this work we extract patches on an image pyramid with multiple resolutions. A small patch in a low-resolution volume has a much larger field-of-view at the original resolution. To be specific, we build an image pyramid with three resolutions (1 mm, 2 mm, and 4-mm resolution, respectively). The intensities of patches from multiple resolutions are concatenated into a long vector to feed the network. As demonstrated in Sect. 4.5, a multi-resolution patch can improve the landmark detection accuracy.

Deep learning automatically learns a hierarchical representation of the input data. Representation at different hierarchical levels may provide complementary information for classification. Furthermore, through years' of feature engineering, some handcrafted image features can achieve quite reasonable performance on a certain task. Combining effective handcrafted image features with deeply learned hierarchical features may achieve even better performance than using them separately.

In this work we propose to use probabilistic boosting-tree (PBT) [20] to combine all features. A PBT is a combination of a decision tree and AdaBoost, by replacing a weak classification node in the decision tree with a strong AdaBoost classifier [21]. Our feature pool is composed of two types of features: Haar wavelet-like features $(h_1, h_2, \ldots, h_m)$ and neural network features r_i^j (where r_i^j is the response of node i at layer j). If $j = 0$, r_i^0 is an input node, representing the image intensity of a voxel in the patch. The last neural network feature is actually the response of the output node, which is the classification score by the network. This feature is the strongest feature and it is always the first selected feature by the AdaBoost algorithm.

Given 2000 landmark candidates generated by the first detection stage (Sect. 4.2), we evaluate them using the bootstrapped classifier presented in this section. We preserve 250 candidates with the highest classification score and then aggregate them into a single detection as follows. For each candidate we define a neighborhood, which is a $8 \times 8 \times 8\,\text{mm}^3$ box centered on the candidate. We calculate the total vote of each candidate as the summation of the classification score of all neighboring candidates. (The score of the current candidate is also counted since it is neighboring

to itself.) The candidate with the largest vote is picked and the final landmark position is the weighted average (according to the classification score) of all candidates in its neighborhood.

4.5 Experiments

In this section we validate the proposed method on carotid artery bifurcation detection. The carotid artery is the main vessel supplying oxygenated blood to the head and neck. The common carotid artery originates from the aortic arch and runs up toward the head before bifurcating to the external carotid artery (supplying blood to face) and internal carotid artery (supplying blood to brain). Examination of the carotid artery helps to assess the stroke risk of a patient. Automatic detection of this bifurcation landmark provides a seed point for centerline tracing and lumen segmentation, thereby making automatic examination possible. However, as shown in Fig. 4.2a, the internal/external carotid arteries further bifurcate to many branches and there are other vessels (e.g., vertebral arteries and jugular veins) present nearby, which may cause confusion to an automatic detection algorithm.

We collected a head-neck CT dataset from 455 patients. Each image slice has 512×512 pixels and a volume contains a variable number of slices (from 46 to 1181 slices). The volume resolution varies too, with a typical voxel size of $0.46 \times 0.46 \times 0.50\,\text{mm}^3$. To achieve a consistent resolution, we resample all input volumes to 1.0 mm. A fourfold cross validation is performed to evaluate the detection accuracy and determine the hyper parameters, e.g., the network size, smoothness constraint α in Eq. (4.6), sparsity constraint β in Eq. (4.8). There are two carotid arteries (left versus right) as shown in Fig. 4.2. Here, we report the bifurcation detection accuracy of the right carotid artery (as shown in Table 4.1) with different approaches. The detection accuracy of the left carotid artery bifurcation is similar.

The rough location of the carotid artery bifurcation can be predicted by other landmarks using a landmark network [22]. However, due to the challenge of the task, the prediction is not always accurate. We have to crop a box as large as $50 \times 50 \times 100\,\text{mm}^3$ around the predicted position to make sure the correct position of the carotid artery bifurcation is covered. To have a fair comparison with [4], in the following experiments, the landmark detection is constrained to this box for all compared methods.

For each approach reported in Table 4.1, we follow a two-step process by applying the first detector to reduce the number of candidates to 2000, followed by a bootstrapped detection to further reduce the number of candidates to 250. The final detection is picked from the candidate with the largest vote from other candidates.

The value of a CT voxel represents the attenuation coefficient of the underlying tissue to X-ray, which is often represented as a Hounsfield unit. The Hounsfield unit has a wide range from -1000 for air to 3000 for bones/metals and it is normally represented with a 12-bit precision. A carotid artery filled with contrasted agent

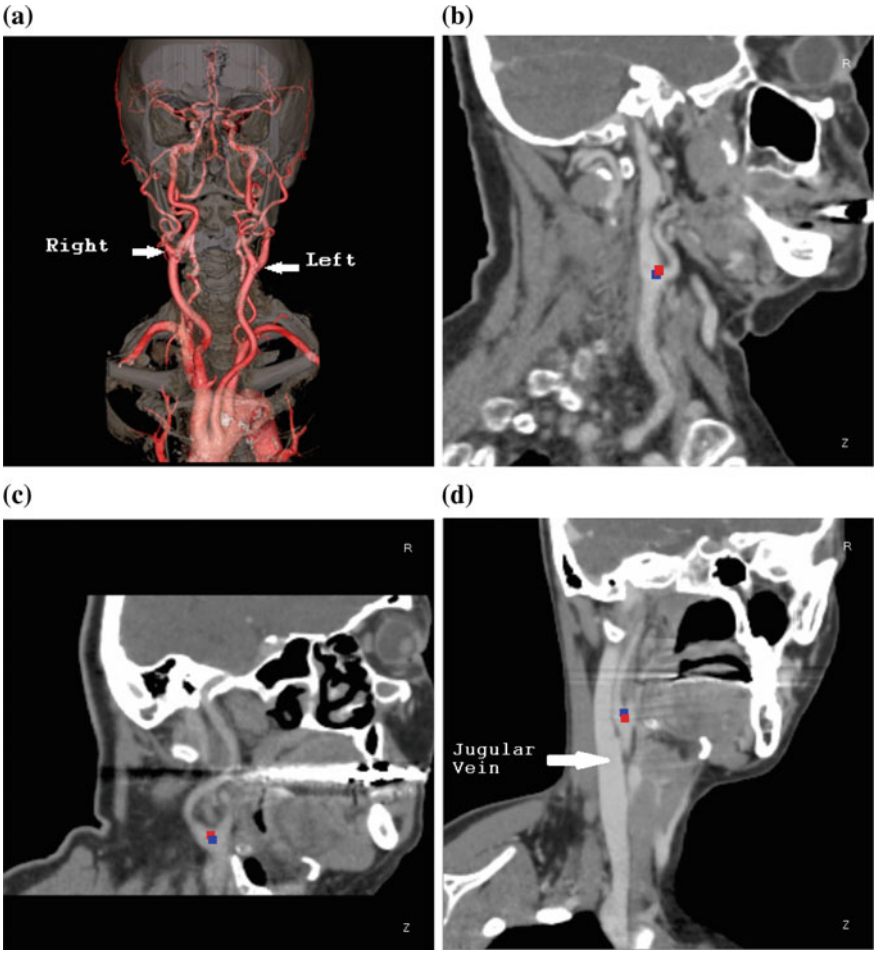

Fig. 4.2 Carotid artery bifurcation landmark detection in head-neck CT scans. **a** 3D visualization of carotid arteries with *white arrows* pointing to the *left* and *right* bifurcations (image courtesy of http://blog.remakehealth.com/). **b–d** A few examples of the *right* carotid artery bifurcation detection results with the ground truth labeled as *blue dots* and detected landmarks in *red*

Table 4.1 Quantitative evaluation of carotid artery bifurcation detection accuracy on 455 CT scans based on a fourfold cross validation. The errors are reported in millimeters

	Mean	Std	Median	80th Percentile
Haar + PBT	5.97	6.99	3.64	7.84
Neural network (Single resolution)	4.13	9.39	1.24	2.35
Neural network (Multi-resolution)	3.69	6.71	1.62	3.25
Network features + PBT	3.54	8.40	1.25	**2.31**
Haar + network + PBT	**2.64**	**4.98**	**1.21**	2.39

occupies only a small portion of the full Hounsfield unit range. Standard normalization methods of neural network training (e.g., linear normalization to [0, 1] using the minimum and maximum value of the input, or normalizing to zero-mean and unit-variance) do not work well for this application. In this work we use a window based normalization. Intensities inside the window of $[-24, 576]$ Hounsfield unit is linearly transformed to [0, 1]; Intensities less than -24 are truncated to 0; And, intensities higher than 576 are truncated to 1.

Previously, Liu et al. [4] used Haar wavelet-like features + boosting to detect vascular landmarks and achieved promising results. Applying this approach on our dataset, we achieve a mean error of 5.97 mm and the large mean error is caused by too many detection outliers. The neural network based approach can significantly improve the detection accuracy with a mean error of 4.13 mm using a $15 \times 15 \times 15$ patch extracted from a single resolution (1 mm). Using patches extracted from an image pyramid with three resolutions, we can further reduce the mean detection error to 3.69 mm. If we combine features from all layers of the network using the PBT, we achieve slightly better mean accuracy of 3.54 mm. Combining the deeply learned features and Haar wavelet-like features, we achieve the best detection accuracy with a mean error of 2.64 mm. We suspect that the improvement comes from the complementary information of the Haar wavelet-like features and neural network features. Figure 4.2 shows the detection results on a few typical datasets.

The proposed method is computationally efficient. Using the speedup technologies presented in Sects. 4.2 and 4.3, it takes 0.92 s to detect a landmark on a computer with a six-core 2.6 GHz CPU (without using GPU). For comparison, the computation time increases to 18.0 s if we turn off the proposed acceleration technologies (namely, separable filter approximation and network sparsification). The whole training procedure takes about 6 h and the sparse deep network consumes majority of the training time.

4.6 Conclusions

In this work we proposed 3D deep learning for efficient and robust landmark detection in volumetric data. We proposed two technologies to speed up the detection using neural networks, namely, separable filter decomposition and network sparsification. To improve the detection robustness, we exploit deeply learned image features trained on a multi-resolution image pyramid. Furthermore, we use the boosting technology to incorporate deeply learned hierarchical features and Haar wavelet-like features to further improve the detection accuracy. The proposed method is generic and can be retrained to detect other 3D landmarks or the center of organs.

References

1. Zhan Y, Dewan M, Harder M, Krishnan A, Zhou XS (2011) Robust automatic knee MR slice positioning through redundant and hierarchical anatomy detection. IEEE Trans Med Imag 30(12):2087–2100
2. Schwing AG, Zheng Y (2014) Reliable extraction of the mid-sagittal plane in 3D brain MRI via hierarchical landmark detection. In: Proceedings of the international symposium on biomedical imaging, pp 213–216
3. Zheng Y, Tek H, Funka-Lea G, Zhou SK, Vega-Higuera F, Comaniciu D (2011) Efficient detection of native and bypass coronary ostia in cardiac CT volumes: anatomical versus pathological structures. In: Proceedings of the international conference on medical image computing and computer assisted intervention, pp 403–410
4. Liu D, Zhou S, Bernhardt D, Comaniciu D (2011) Vascular landmark detection in 3D CT data. In: Proceedings of the SPIE medical imaging, pp 1–7
5. Zheng Y, Lu X, Georgescu B, Littmann A, Mueller E, Comaniciu D (2009) Robust object detection using marginal space learning and ranking-based multi-detector aggregation: application to automatic left ventricle detection in 2D MRI images. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 1343–1350
6. Zheng Y, John M, Liao R, Nottling A, Boese J, Kempfert J, Walther T, Brockmann G, Comaniciu D (2012) Automatic aorta segmentation and valve landmark detection in C-arm CT for transcatheter aortic valve implantation. IEEE Trans Med Imaging 31(12):2307–2321
7. Vincent P, Larochelle H, Lajoie I, Bengio Y, Manzagol PA (2010) Stacked denoising autoencoders: learning useful representations in a deep network with a local denoising criterion. J Mach Learn Res 11:3371–3408
8. Yan Z, Zhan Y, Peng Z, Liao S, Shinagawa Y, Metaxas DN, Zhou, XS (2015) Bodypart recognition using multi-stage deep learning. In: Proceedings of the information processing in medical imaging, pp 449–461
9. Liu F, Yang L (2015) A novel cell detection method using deep convolutional neural network and maximum-weight independent set. In: Proceedings of the international conference on medical image computing and computer assisted intervention, pp 349–357
10. Roth HR, Lu L., Seff A, Cherry KM, Hoffman J, Wang S, Liu J, Turkbey E, Summers RM (2014) A new 2.5D representation for lymph node detection using random sets of deep convolutional neural network observations. In: Proceedings of the international conference on medical image computing and computer assisted intervention, pp 520–527
11. Carneiro G, Nascimento JC, Freitas A (2012) The segmentation of the left ventricle of the heart from ultrasound data using deep learning architectures and derivative-based search methods. IEEE Trans Image Process 21(3):968–982
12. Ghesu FC, Krubasik E, Georgescu B, Singh V, Zheng Y, Hornegger J, Comaniciu D (2016) Marginal space deep learning: efficient architecture for volumetric image parsing. IEEE Trans Med Imag 35(5):1217–1228
13. Cheng X, Zhang L, Zheng Y (2016) Deep similarity learning for multimodal medical images. Comput Methods Biomech Biomed Eng Imaging Vis 4:1–5
14. Miao S, Wang ZJ, Zheng Y, Liao R (2016) Real-time 2D/3D registration via CNN regression. In: Proceedings of the IEEE international symposium on biomedical imaging, pp 1–4
15. Prasoon A, Petersen K, Igel C, Lauze F, Dam E, Nielsen M (2013) Deep feature learning for knee cartilage segmentation using a triplanar convolutional neural network. In: Proceedings of the international conference on medical image computing and computer assisted intervention, vol 8150, pp 246–253
16. Rigamonti R, Sironi A, Lepetit V, Fua P (2013) Learning separable filters. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 2754–2761
17. Denton E, Zaremba W, Bruna J, LeCun Y, Fergus R (2014) Exploiting linear structure within convolutional networks for efficient evaluation. In: Advances in neural information processing systems, pp 1–11

18. Zheng Y, Liu D, Georgescu B, Nguyen H, Comaniciu D (2015) 3D deep learning for efficient and robust landmark detection in volumetric data. In: Proceedings of the international conference medical image computing and computer assisted intervention, pp 565–572
19. Acar E, Dunlavy DM, Kolda TG (2011) A scalable optimization approach for fitting canonical tensor decompositions. J Chemom 25(2):67–86
20. Tu Z (2005) Probabilistic boosting-tree: learning discriminative methods for classification, recognition, and clustering. In: Proceedings of the international conference on computer vision, pp 1589–1596
21. Freund Y, Schapire RE (1997) A decision-theoretic generalization of on-line learning and an application to boosting. J Comput Syst Sci 55(1):119–139
22. Liu D, Zhou S, Bernhardt D, Comaniciu D (2010) Search strategies for multiple landmark detection by submodular maximization. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 2831–2838