Marginal Space Deep Learning: Efficient Architecture for Detection in Volumetric Image Data

Florin C. Ghesu[1,2], Bogdan Georgescu[1], Yefeng Zheng[1], Joachim Hornegger[2], and Dorin Comaniciu[1]

[1] Imaging and Computer Vision, Siemens Corporate Technology, Princeton NJ, USA
[2] Pattern Recognition Lab, Friedrich-Alexander-Universität, Erlangen-Nürnberg

Abstract. Current state-of-the-art techniques for fast and robust parsing of volumetric medical image data exploit large annotated image databases and are typically based on machine learning methods. Two main challenges to be solved are the low efficiency in scanning large volumetric input images and the need for manual engineering of image features. This work proposes Marginal Space Deep Learning (MSDL) as an effective solution, that combines the strengths of efficient object parametrization in hierarchical marginal spaces with the automated feature design of Deep Learning (DL) network architectures. Representation learning through DL automatically identifies, disentangles and learns explanatory factors directly from low-level image data. However, the direct application of DL to volumetric data results in a very high complexity, due to the increased number of transformation parameters. For example, the number of parameters defining a similarity transformation increases to 9 in 3D (3 for location, 3 for orientation and 3 for scale). The mechanism of marginal space learning provides excellent run-time performance by learning classifiers in high probability regions in spaces of gradually increasing dimensionality, for example starting from location only (3D) to location and orientation (6D) and full parameter space (9D). In addition, for parametrized feature computation, we propose to simplify the network by replacing the standard, pre-determined feature sampling pattern with a sparse, adaptive, self-learned pattern. The MSDL framework is evaluated on detecting the aortic heart valve in 3D ultrasound data. The dataset contains 3795 volumes from 150 patients. Our method outperforms the state-of-the-art with an improvement of 36%, running in less than one second. To our knowledge this is the first successful demonstration of the DL potential to detection in full 3D data with parametrized representations.

1 Introduction

Effective data representation is essential for the performance of machine learning algorithms [1]. This motivates a large effort invested into handcrafting features, which encompass the underlying observation in a learning space easy to tackle. For this purpose, complex data preprocessing and transformation pipelines are used to design representations that can ensure an effective learning process.

This type of approach is however subject to severe limitations, since it targets exclusively human ingenuity to disentangle and understand prior information hidden in the data and then use such knowledge for feature engineering [2, 3].

Representation learning through Deep Learning (DL) addresses these limitations and is aimed to expand the scope and general applicability of machine learning algorithms [1]. This is achieved by applying a mechanism that supports the joint learning of the underlying phenomena and the required features. The capability of automatically identifying and disentangling data-describing attributes directly from low-level image data eliminates the need for complex, manual prerequisites. Hierarchical representations encoded by deep neural networks (NN) [1, 4] are used to effectively model this learning approach. Such deep architectures outperform state-of-the-art classifiers on a variety of publicly available benchmark tests [5–8]. Nonetheless, the current applications of these architectures concentrate on 2D data, with no generic extension to any 3D image modality. Capturing the complex appearance of a 3D object and supporting the efficient scanning of high-dimensional spaces are not straightforward, given the increased number of parameters (9 to describe a rigid transformation in 3D).

In this work we propose novel *sparse deep neural networks* for learning parametrized representations from 3D medical image modalities and supporting the effective parsing of volumetric medical image data. We use the concept of network simplification through sparsity injection to replace the standard, predetermined sampling pattern used for handcrafted features, with an adaptive, sparse, self-learned pattern. This brings a considerable increase in computational performance and also serves as regularization against overfitting. Our method for imposing sparsity is based on an iterative learning process using a greedy approach. In order to address the problem of efficiently scanning large parameter spaces for detecting objects in 3D images, we propose MSDL, a novel integration of our sparse deep neural network into the Marginal Space Learning (MSL) pipeline [3]. The Probabilistic Boosting Tree (PBT) classifier [9] used in the MSL framework is replaced with our generic, sparse feature-learning engine, which we apply in marginal spaces of increasing dimensionality to estimate the rigid transformation parameters of the target object. The proposed framework combines the computational efficiency of MSL with the potential of DL architectures. We evaluate the framework for the problem of detecting the pose of a bounding box enclosing the aortic valve in 3D ultrasound images of the heart (see Figure 1).

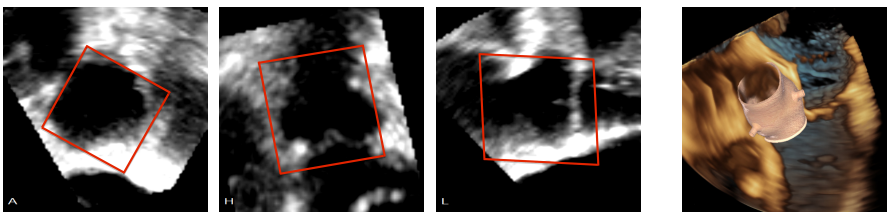

Fig. 1. Planar cuts displaying the bounding box of the aortic valve in a transesophageal ultrasound volume, as well as the 3D geometry of the valve depicted in the last image.

2 Related work

Representation learning, also known under the header of *deep learning* or *feature learning*, is a rapidly developing field in the machine learning community. Correlated with the increase in computational power, recent publications show remarkable result improvements for tasks ranging from speech recognition, object recognition, natural language processing to transfer learning.

For the generic task of object recognition/tracking the impact of this technology started with the break of supremacy of the support vector machines on the MNIST image classification problem [5, 6]. This motivated a further improvement through the introduction of multi-column deep neural networks [8] or state-of-the-art network regularization techniques based on a random dropping of units [7]. For more specific tasks within the medical imaging field, stacked sparse autoencoders are applied for multiple organ detection and classification [10]. Using the same pixel-based classification approach, deep neural architectures are also used for the segmentation of brain structures [11]. More recent publications present solutions to emulate 3D learning tasks using 2D feature fusion from predetermined planar cuts [12] or representation sets from random observations.

All investigated methods are devised for 2D or hybrid image modalities, with no extension or direct solution for parsing 3D data. The application of deep learning for object detection with high-dimensional representations is, to the best of our knowledge, not attempted yet.

3 Method

In the following we present our Marginal Space Deep Learning architecture for efficiently estimating the anisotropic similarity transformation parameters of an object in a 3D image. We model the pose of the sought object by using a bounding box, defined by 9 parameters: $\boldsymbol{T} = (t_x, t_y, t_z)$ for the translation, $\boldsymbol{R} = (\phi_x, \phi_y, \phi_z)$ for the orientation and $\boldsymbol{S} = (s_x, s_y, s_z)$ for the anisotropic scale of the object (see Figure 1 for the aortic valve examples). We tackle the object detection problem with machine learning, by training a classifier which can decide if a given parametrized volume patch contains the target object or not.

3.1 Sparse Deep Learning Architectures

A deep Neural Network (NN) is a powerful feature-learning engine, built on hierarchies of data representations [4]. Structurally, the network architecture can be divided into multiple layers, organized and connected hierarchically. In such networks, data representations are obtained by applying learned filters or kernels over representations defined in the previous layer. The same holds for fully connected layers, where the kernel size is restricted to the size of the underlying representation map. As such, a deep NN can be defined by the parameters $(\boldsymbol{w}, \boldsymbol{b})$, where $\boldsymbol{w} = (\boldsymbol{w}^{(1)}, \boldsymbol{w}^{(2)}, \cdots, \boldsymbol{w}^{(n)})^\top$ represents the list of concatenated kernel parameters over n layers and $\boldsymbol{b}$ encodes the biases of all neurons contained in the

Algorithm 1 Learning algorithm with iterative threshold-enforced sparsity

```
Pre-training stage using all weights (small # epochs)
for # iterations do
    for all filter i with sparsity do
        p - proportion of absolute smallest non-zero w_j^(i) ← 0
        Re-normalize to preserve ||w^(i)||_1
    end for
    Train network on active coefficients (small # epochs)
end for
```

network. The underlying learning problem is supervised, meaning that for a given set of input patches $\boldsymbol{X}$ (i.e. observations), we are given a corresponding set of class assignments $\boldsymbol{y}$, specifying if the patches contain the target object or not. Considering the independence of the input observations, using the Maximum Likelihood Estimation (MLE) method, we learn the network parameters in order to maximize the likelihood function:

$$\left(\hat{\boldsymbol{w}}, \hat{\boldsymbol{b}}\right) = \arg\max_{\boldsymbol{w},\boldsymbol{b}} \mathcal{L}(\boldsymbol{w}, \boldsymbol{b}) = \arg\max_{\boldsymbol{w},\boldsymbol{b}} \prod_{i=1}^{m} p(y^{(i)} | \boldsymbol{x}^{(i)}; \boldsymbol{w}, \boldsymbol{b}), \tag{1}$$

where m represents the number of training samples. For a linear regression model this is equivalent to minimizing the least square distance between the estimated output $\hat{\boldsymbol{y}}$ and the ground truth reference $\boldsymbol{y}$ [4]. We solve this with the Stochastic Gradient Descent (SGD) method, based on the back-propagation algorithm to update the network coefficients according to the computed gradient [13].

As presented in [4], it is conjectured that most networks are oversized for the underlying task. Starting from this observation and the need for optimal runtime performance, we propose a novel network simplification technique based on the injection of sparsity. Defining the network response function as $\mathcal{R}(\,\cdot\,; \boldsymbol{w}, \boldsymbol{b})$, we aim to find a sparsity map $\boldsymbol{s}$ over the network parameters, such that the response residual ϵ given by:

$$\epsilon = \|\mathcal{R}(X; \boldsymbol{w}, \boldsymbol{b}) - \mathcal{R}(X; \boldsymbol{w} \odot \boldsymbol{s}, \boldsymbol{b})\|, \text{ where } s_i \in \mathbb{R}^+, \forall i, \tag{2}$$

is minimal, where $\odot$ denotes the element-wise multiplication of vectors. For this, we apply an iterative learning process, enforcing sparsity in a gradual manner in the layers of the neural network by removing weights with smallest absolute value, in other words with minimal impact on the network response. Algorithm 1 presents the training method.

By using this kind of approach we learn adaptive, sparse features, more specifically in the first layer we learn an adaptive sampling pattern over the input. This is used to replace the standard uniform sampling pattern defined in handcrafted features, eliminating the need for feature engineering. The sparsity enforcement is essential for efficient feature computation under different transformations, bringing a speed-wise improvement of two orders of magnitude. Also, by simplifying the model, the network is more robust against overfitting.

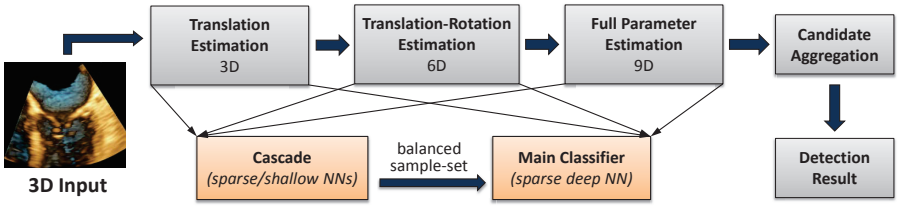

Fig. 2. Scheme depicting the Marginal Space Deep Learning pipeline.

3.2 Marginal Space Deep Learning

In order to perform the detection with the introduced classifier in a given volumetric image I, we aim to find the parameters $(\boldsymbol{T}, \boldsymbol{R}, \boldsymbol{S})$ such that we maximize the posterior probability $p(\boldsymbol{T}, \boldsymbol{R}, \boldsymbol{S}|I)$ over the space of all possible transformations, namely:

$$\left(\hat{\boldsymbol{T}}, \hat{\boldsymbol{R}}, \hat{\boldsymbol{S}}\right) = \arg\max_{\boldsymbol{T},\boldsymbol{R},\boldsymbol{S}} p(\boldsymbol{T}, \boldsymbol{R}, \boldsymbol{S}|I). \tag{3}$$

Due to the exponential increase of the number of pose hypotheses with respect to the dimensionality of the pose parameter space, an exhaustive search is impractical. To address this we propose *Marginal Space Deep Learning*, a framework combining the concept of Marginal Space Learning [3] with the presented sparse DL architecture. We split the parameter space in marginal sub-spaces of increasing dimensionality, learning the underlying classifier only in high probability regions, estimating consequently the translation, orientation and scale of the target object. This approach is expressed by the factorization of the posterior probability as:

$$p(\boldsymbol{T}, \boldsymbol{R}, \boldsymbol{S}|I) = p(\boldsymbol{T}|I)p(\boldsymbol{R}|\boldsymbol{T}, I)p(\boldsymbol{S}|\boldsymbol{T}, \boldsymbol{R}, I). \tag{4}$$

We use our sparse DL-based classifier to estimate in turn the posterior probabilities $p(\boldsymbol{T}|I)$, $p(\boldsymbol{T}, \boldsymbol{R}|I)$ and $p(\boldsymbol{T}, \boldsymbol{R}, \boldsymbol{S}|I)$ which are then used to obtain the factors contained in Eq. 4, using the relations: $p(\boldsymbol{R}|\boldsymbol{T}, I) = \frac{p(\boldsymbol{T},\boldsymbol{R}|I)}{p(\boldsymbol{T}|I)}$ and $p(\boldsymbol{S}|\boldsymbol{T}, \boldsymbol{R}, I) = \frac{p(\boldsymbol{T},\boldsymbol{R},\boldsymbol{S}|I)}{p(\boldsymbol{T},\boldsymbol{R}|I)}$. Using this kind of approach, as shown in [3], we achieve a speed-up of 6 orders of magnitude compared to the exhaustive search.

A challenge arising with the use of deep neural networks as discriminating engine in each stage of the marginal space pipeline, is the high class imbalance. This imbalance can reach ratios of $1:1000$ positive to negative samples. A deep architecture cannot be trained with an SGD approach on such an unbalanced set and simply re-weighting the penalties for the network cost function further worsens the vanishing gradient effect. Instead, we propose to use a *negative filtering cascade* of classifiers to hierarchically eliminate as many negatives as possible, while preserving the positive samples across cascade stages. More specifically, in each stage of the cascade we employ a simple, shallow, sparse neural network and manually tune its decision boundary to eliminate the maximum number of true negatives. The remaining samples are propagated through to the next cascade stage where the same filtering procedure is repeated, unless we achieve a

balanced sample set. In order to train a network within the cascade, we iterate at epoch level over the complete positive sample set, while at each batch level, we randomly sample the negative space to obtain a balanced training batch. Figure 2 shows a schematic visualization of the complete framework.

4 Experimental Results

For a comprehensive evaluation, we refer to the problem of detecting the aortic valve in 3D Ultrasound volumes and compare the results to the reference state-of-the-art MSL approach [3]. We model the pose of the aortic valve using a bounding box defined in a 9-dimensional parameter space (see Section 3). The dataset used for evaluation stems from 150 patients. Over multiple acquisitions and time frames we extracted a set of 3795 volumes. The size of the frames present a high variation between $100\times100\times50$ and $250\times250\times150$ voxels, at an original isotropic resolution of 0.75 mm, adjusted to 3 mm for our experiments. The intensity of each volume is normalized to unit range and annotated with the ground truth box, enclosing the aortic valve at an average scale of $32 \times 32 \times 28$ mm [14]. To set up the training environment for both the proposed MSDL pipeline and the reference MSL approach, the dataset is split randomly at patient level in a $90\% - 10\%$ proportion to determine the training set and the validation examples for testing.

The meta-parameters defining the sub-space sampling and candidate propagation in the MSL pipeline, as well as the network dependent parameters for both the main classifier and the cascade used in each stage are estimated using a grid search. For the proposed MSDL approach we distinguish in our experiments two variants: **MSDL-tes** (using the gradual sparsity enforcement technique) and **MSDL-full** (using all weights in the network). We achieve with 0% false negative rate the sample balancing, using in each cascade less than 3 shallow networks. All used networks are composed of fully connected layers of nodes with sigmoid activation. For all 3 marginal spaces we use the same architecture for the cascade and main classifier; cascade: 2 layers = 5832 (sparse) $\times$ 60 $\times$ 1 and main classifier: 4 layers = 5832 (sparse) $\times$ 150 $\times$ 80 $\times$ 50 $\times$ 1 hidden units.

Table 1. Comparison of the performance of the state-of-the-art MSL [3] and the proposed MSDL framework. The measures used to quantify the quality of the results w.r.t to the groundtruth data are the error of the position of the box and mean corner distance (both measured in millimetres). The superior results are displayed in bold.

	Position Error [mm]				**Corner Error [mm]**			
	Training Data		**Test Data**		**Training Data**		**Test Data**	
	MSL	MSDL	MSL	MSDL	MSL	MSDL	MSL	MSDL
Mean	3.24	**1.66**	3.52	**2.26**	5.73	**3.29**	6.49	**4.57**
Median	2.91	**1.51**	3.31	**2.04**	5.21	**3.02**	6.22	**3.98**
STD	1.83	**0.99**	1.60	**1.13**	2.58	**1.44**	**2.06**	2.07

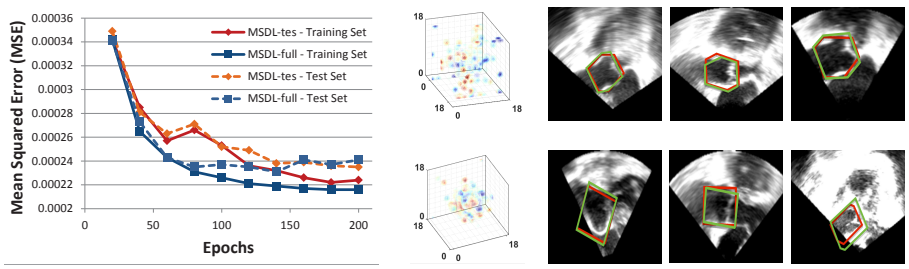

Fig. 3. Left: Plot depicting the error progression on the training set and hold-out test set in the translation stage, with the highlight on the impact of the sparsification on the accuracy. Centre: Example sparse patterns for translation (top) and full space (bottom), note more compact representation for the latter due to better data alignment. Right: Example images showing detection results for different patients from the test set. In order to capture also the underlying image information (i.e. the anatomy) we use 2D planar cuts through the volume. Please note that depending on the cutting plane, the visualized boxes can be viewed as complex polygons (ground-truth shown in red, detection shown in green).

To quantify the results, we consider the position error of the center of the box and the mean corner distance error (measuring the estimation accuracy of the full transformation). The latter measure represents the average distance between the 8 corners of the detected box and the ground truth box. Table 1 shows the obtained results. The MSDL approach outperforms the state-of-the-art MSL method by improving the mean position error by 36%. Figure 3(left) shows the error measured during training for MSDL-tes and MSDL-full. The error variation on the training data is explained by the injection of sparsity. As can be seen, the enforced sparsity acts as regularization on the hold-out test set, preventing the network from overfitting the data. As such applying sparsity minimally impacts the accuracy on unseen data. In Figure 3(center) we illustrate an example of the learned sparse weights showing a more distributed pattern on the translation stage and more compact (and around the aortic root) on the full parameters estimation stage, due to better data alignment. Qualitative results are depicted in Figure 3(right). In terms of time performance, running the full MSDL pipeline requires under 0.5 seconds compared to 1.9 seconds for MSL (using only CPU). By imposing sparsity we achieve a speed-up of $\times 300$ compared to MSDL-full, hence the large computational benefit of the network simplification.

5 Conclusion

This work introduces the MSDL framework for efficient and robust scanning of 3D volumetric medical image data. We proposed to tackle the parameter estimation in hierarchical sub-spaces of increasing dimension by using a deep neural architecture, simplified through sparsity injection. The training of such a classifier is based on an iterative learning process. Within the pipeline, the

described learning engine is preceded by a negative sample filtering cascade of shallow sparse neural networks, which addresses the high class imbalance associated with each learning space. By using this kind of approach, the need for complex handcrafted features is eliminated. In terms of performance our method outperforms the state-of-the-art MSL for the problem of detecting the aortic valve in 3D ultrasound images. For future work, we plan on evaluating the framework on more complex problems, with the target of completing the detection pipeline with the full segmentation of the shape.

References

1. Bengio, Y., Courville, A.C., Vincent, P.: Unsupervised Feature Learning and Deep Learning: A Review and New Perspectives. CoRR **abs/1206.5538** (2012)
2. Lowe, D.G.: Object recognition from local scale-invariant features. In: ICCV. Volume 2. (1999) 1150–1157
3. Zheng, Y., Barbu, A., Georgescu, B., Scheuering, M., Comaniciu, D.: Four-Chamber Heart Modeling and Automatic Segmentation for 3-D Cardiac CT Volumes Using Marginal Space Learning and Steerable Features. IEEE TMI **27**(11) (2008) 1668–1681
4. Lecun, Y., Bottou, L., Bengio, Y., Haffner, P.: Gradient-based learning applied to document recognition. Proceedings of the IEEE **86**(11) (1998) 2278–2324
5. Hinton, G.E., Osindero, S., Teh, Y.W.: A Fast Learning Algorithm for Deep Belief Nets. NIPS **18**(7) (2006) 1527–1554
6. Bengio, Y., Lamblin, P., Popovici, D., Larochelle, H., Montral, U.D., Qubec, M.: Greedy layer-wise training of deep networks. In: NIPS, MIT Press (2007)
7. Krizhevsky, A., Sutskever, I., Hinton, G.E.: ImageNet Classification with Deep Convolutional Neural Networks. In: NIPS. Curran Associates, Inc. (2012) 1097–1105
8. Ciresan, D.C., Meier, U., Schmidhuber, J.: Multi-column Deep Neural Networks for Image Classification. CoRR **abs/1202.2745** (2012)
9. Tu, Z.: Probabilistic Boosting-Tree: Learning Discriminative Models for Classification, Recognition, and Clustering. In: IEEE 10th ICCV. ICCV (2005) 1589–1596
10. Shin, H.C., Orton, M., Collins, D.J., Doran, S.J., Leach, M.O.: Stacked Autoencoders for Unsupervised Feature Learning and Multiple Organ Detection in a Pilot Study Using 4D Patient Data. IEEE PAMI **35**(8) (2013) 1930–1943
11. Ciresan, D., Giusti, A., Gambardella, L.M., Schmidhuber, J.: Deep Neural Networks Segment Neuronal Membranes in Electron Microscopy Images. In Pereira, F., Burges, C., Bottou, L., Weinberger, K., eds.: NIPS. Curran Associates, Inc. (2012) 2843–2851
12. Roth, H.R., Lu, L., Seff, A., Cherry, K.M., Hoffman, J., Wang, S., Liu, J., Turkbey, E., Summers, R.M.: A New 2.5D Representation for Lymph Node Detection Using Random Sets of Deep Convolutional Neural Network Observations. In: MICCAI. (2014) 520–527
13. Rumelhart, D.E., Hinton, G.E., Williams, R.J.: Parallel Distributed Processing: Explorations in the Microstructure of Cognition. MIT Press (1986) 318–362
14. Ionasec, R.I., Voigt, I., Georgescu, B., Wang, Y., Houle, H., Vega-Higuera, F., Navab, N., Comaniciu, D.: Patient-specific modeling and quantification of the aortic and mitral valves from 4-D cardiac CT and TEE. IEEE Trans Med Imaging **29**(9) (Sep 2010) 1636–1651