Lifted Auto-Context Forests for Brain Tumour Segmentation

Loic Le Folgoc, Aditya V. Nori, Siddharth Ancha, and Antonio Criminisi

Microsoft Research Cambridge, UK.

Abstract. We revisit Auto-Context Forests for brain tumour segmentation in multi-channel magnetic resonance images. Semantic context is progressively built and refined via successive layers of Decision Forests (DFs). This sequential approach allows to decompose complex segmentation tasks into a series of simpler subtasks, *e.g.* one that encodes the hierarchical structure between labels. Our contribution is threefold. 1) *Improved generalization:* we introduce an efficient node-splitting criterion based on hold-out estimates of generalization to enhance the discriminative power of nodes in decision ensembles. 2) *Increased compactness:* at a tree-level we derive a principled, assumption-free trade-off between data-fit and model complexity, thereby yielding shallow discriminative ensembles trained orders of magnitude faster. 3) *Guided semantic bagging:* we expose latent data-space semantics captured by forest pathways and specialize subsequent classifiers over the resulting semantic regions to boost accuracy. The proposed framework is practical: the per-layer training is fast, modular and robust. It was a top performer in the MICCAI 2016 BRATS (Brain Tumour Segmentation) challenge, and this paper aims to discuss and provide details about the challenge entry.

1 Introduction

The past few years have witnessed a vast body of machine learning (ML) techniques for the automatic segmentation of medical images. Decision forests [22,19], and more recently, deep neural networks [12] have yielded state-of-the-art results at MICCAI BRATS (Brain Tumour Segmentation) challenges. Our approach builds upon the framework of DFs, investigating efficient and generic ways to increase their accuracy and generalisation.

Decision trees partition the feature space via a sequence of *learnt* binary decisions (giving rise to a binary tree structure) into a collection of disjoint subsets (at tree leaves). Learnt decisions favour 'pure' tree leaves, *e.g.* the majority of test points routed to a given leaf should share the same label. In the DF framework, the predictions of a collection of *randomised* decision trees are aggregated. Randomisation occurs at training time: tree node decisions are weakly discriminative learners chosen among random subsets of candidate decisions [1,9] evaluated on random subsets of training data [4]. In principle, the mechanism increases the discriminative power and generalisation of the algorithm. Indeed, fast generic intensity-based weak learners chained within *deep* tree architectures (millions of nodes or more) have convincingly demonstrated their performance, see *e.g.* [7].

Still in practice, more expressive features yield more discriminative node decisions, and therefore improved segmentation accuracy. Tustison et al. [19,18] incorporate features derived from the local intensity context and texture, from elastic registration and other task-specific processing steps. Supervoxels have also been used successfully for the brain tumour segmentation task [8].

In this paper, we depart from hand-crafting powerful features for a complementary approach that is entirely generic and free of additional computations. Section 4 introduces an efficient node-splitting criterion based on cross-validation estimates that improves the feature *selection* during the training stage. As a result, learnt features are more discriminative and generalize better to unseen data: we refer to this process as *lifting* the node decisions. Furthermore the proposed cost function induces a natural stopping condition to grow or prune decision trees, resulting in an *Ockham razor*-like, principled trade-off between tree complexity and training accuracy. We show that lifted DFs outperform standard DFs for a fraction of their computational resources, using compact, shallow tree architectures (several dozens or hundreds of nodes).

We exploit these computational gains to revisit *auto-context* [15,16] segmentation forests (section 5). In the DF framework, reasoning about semantics is indirect: the algorithm reasons about the *intensity* of neighbouring voxels to decide which label assignment is likely. We want the algorithm to reason directly about semantics, *e.g.* classify a voxel of interest based on neighbouring voxels' *label assignment* probabilities. In [14], Shotton et al. train two successive forest layers, augmenting the latter with the output of the former. We extend the approach, experimenting with a meta-architecture of cascaded DFs that naturally intertwine high-level *semantic* reasoning with *intensity*-based low-level reasoning. The framework aims to be practical: the per-layer training is simple, modular and robust. Furthermore this sequential design enables the decomposition of complex segmentation tasks into series of simpler subtasks that, for instance, exploit the hierarchical structure between labels (*e.g.* whole tumour, tumour core, enhancing tumour parts). Beyond auto-context, we contribute with a clustering mechanism that exposes the latent data-space semantics encoded within DF pathways. The revealed semantics are exploited to automatically guide classification in subsequent layers (section 5.2).

We discuss results and details of the BRATS challenge entry in section 7.

2 Background: Random Forests for Segmentation

We begin with a brief summary of the randomized decision forest framework for image segmentation. Image segmentation is cast as a voxelwise classification task. Let $I = \{I_j\}_{j=1..J}$ the set of input channels in a multichannel image, x a pixel and $c \in \mathcal{C} = \{1 \dots K\}$ the label to predict. We define $\mathbf{x} = \{x, I\} \in \mathcal{X}$ as the feature vector of x. DF classifiers predict the probability $p(c|\mathbf{x})$ for voxel x to have label c given its feature representation $\mathbf{x}$ by aggregating predictions of an ensemble $\mathcal{T}$ of $t = 1 \cdots |\mathcal{T}|$ decision trees. Simple averaging is widely used, so that

$$p(c|\mathbf{x}) = 1/|\mathcal{T}| \cdot \sum_{t=1}^{|\mathcal{T}|} p_t(c|\mathbf{x}) \, ,$$

where $p_t(c|\mathbf{x})$ stands for the t-th tree prediction. The class of maximum probability is then returned (maximum a posteriori assignment).

Tree predictions are obtained as follows. Points $\mathbf{x}$ are routed down the tree from the root node by evaluating at every internal (*split*) node $n \in \mathcal{S}$ along the path a routing function $h_n(\mathbf{x}) \triangleq [f(\mathbf{x}, \boldsymbol{\theta}_n) \leq \tau_n] \in \{0,1\}$, moving down to the left child n_L whenever $h_n(\mathbf{x}) = 1$ and to the right child n_R otherwise, until a leaf node $n(\mathbf{x})$ is reached. Here $f: \mathcal{X} \times \Theta \mapsto \mathbb{R}$ is a weak learner parametrized by some node-specific feature $\boldsymbol{\theta}_n \in \Theta$ and $\tau_n \in \mathbb{R}$ a node-specific threshold. Examples of weak learners are given in section 3. Each terminal (*leaf*) node $n \in \mathcal{L}$ is paired with a class predictor $p_n(c|\mathbf{x}) \triangleq p_n(c)$ such that $0 \leq p_n(c) \leq 1$, $\sum_{c=1}^{K} p_n(c) = 1$. The tree then predicts class c with probability $p_{n(\mathbf{x})}(c)$.

Decision trees are trained in a supervised manner from training data $D = \{\mathbf{x}_i, c_i\}_{i=1}^{N}$ with known labels c_i, greedily and recursively, starting from a single root node. Training a node n consists of finding the optimal feature $\boldsymbol{\theta}_n^*$ and threshold τ_n^* so that the node training data $D_n = \{\mathbf{x}_i, c_i\}_{i \in \mathcal{I}_n}$ is split between left and right children n_L and n_R in a way that maximizes class purity. Specifically, $\psi^* = (\boldsymbol{\theta}_n^*, \tau_n^*)$ and the resulting split $D_n = D_{n_L}(\psi) \coprod D_{n_R}(\psi)$ maximize the Information Gain IG$(\psi; D_n)$ of Eq. (1):

$$\sum_{\epsilon=\{L,R\}} \frac{|D_{n_\epsilon}(\psi)|}{|D_n|} \sum_{c \in \mathcal{C}} p_{n_\epsilon}(c;\psi) \log p_{n_\epsilon}(c;\psi) - \sum_{c \in \mathcal{C}} p_n^*(c) \log p_n^*(c) \,, \tag{1}$$

where $p_{n_\epsilon}(c;\psi) \triangleq |\{i \in \mathcal{I}_{n_\epsilon}(\psi) | c_i = c\}| \,/\, |D_{n_\epsilon}(\psi)|$ is the empirical class distribution in the training data D_{n_ϵ} for the child node n_ϵ. The optimum $\psi^* \triangleq \arg\max_\psi \mathrm{IG}(\psi; D_n)$ is found by exhaustive search after proper quantization of thresholds τ_n. Trees are grown up to some maximum depth or until too few training examples reach a given node.

Last but not least, random forests introduce randomization in the training of each tree via feature and data *bagging*. For the t-th tree and at node $n \in \mathcal{V}_t$, only a random subset $\Theta' \subsetneq \Theta$ of candidate features is considered for training [1,9]. Similarly, only a random subset $D_n' \subsetneq D_n$ of training examples sampled with(out) replacement is used [3]. Data bagging is implemented both at an image level (random image subsets) and at a voxel level (random voxel subsets). At test time (for a previously unseen image) of course, each voxel on the image grid is sent through the forest for its label to be predicted.

Note that we do *not* make use of class rebalancing schemes. Training samples are often weighted according to the relative frequency of their class whenever summing over training examples. This aims to correct classifier bias in favor of the more frequent class, in case of large class imbalance. Of course class rebalancing induces the opposite bias against more prevalent classes, which can be inextricable in a multilabel setting. Section 5.2 discusses an alternative strategy to naturally correct for distribution imbalance.

3 Fast scale-space context-sensitive features

$3D$ medical image segmentation demands scalable, robust, high precision solutions. We revisit scale-space representations to craft fast, expressive, compactly parametrized features, as a simple alternative to the popular integral or Haar-like features [20].

Background: integral features. Integral features are based on intensity averages within anisotropic cuboids offset from the point of interest [6]. Cuboid averages are computed in constant time by probing the value of a precomputed integral map at the cuboid vertices [20]. For instance, $f(\mathbf{x},\boldsymbol{\theta}) \triangleq \sum_{x'\in\mathcal{C}_2} I_{j_2}(x') - \sum_{x'\in\mathcal{C}_1} I_{j_1}(x')$ computes the difference of responses in cuboids $\mathcal{C}_1$ and $\mathcal{C}_2$ of size $\boldsymbol{s_1} = (s_1^x, s_1^y, s_1^z)$ and $\boldsymbol{s_2} = (s_2^x, s_2^y, s_2^z)$, centered at offset locations $x + \boldsymbol{o}_1$ and $x + \boldsymbol{o}_2$, in distinct channels I_{j_1} and I_{j_2}. Here $\boldsymbol{\theta} = (j_1, j_2, \boldsymbol{o}_1, \boldsymbol{o}_2, \boldsymbol{s}_1, \boldsymbol{s}_2)$ is a 14-dimensional feature.

Proposed scale-space representation. During node training, it is crucial for sufficiently strong weak-learners to be reachable within the budget allocated to feature sampling and optimization. Hence reducing the feature parametrization while maintaining expressivity is key. In integral features, the sophistication of probing anisotropic cuboids with a continuous range of edge lengths comes at a cost w.r.t. parametric complexity. We restrict ourselves to a small *finite* range of *isotropic* averages. We augment the original set of input channels with their smoothed counterparts under separable Gaussian filtering at scales σ_1, σ_2... Given s scales, $f(\mathbf{x},\boldsymbol{\theta}) \triangleq I_{j_2}(x + \boldsymbol{o}_2) - I_{j_1}(x + \boldsymbol{o}_1)$ computes the difference of responses in channels j_ϵ mod s at scales j_ϵ / s with offset $\boldsymbol{o}_\epsilon$ from voxel x ($\epsilon = 1, 2$). Here $\boldsymbol{\theta} = (j_1, j_2, \boldsymbol{o}_1, \boldsymbol{o}_2)$ is an 8-dimensional feature.

A single point is probed for every 8 cuboid vertices probed under integral features, as well as circumventing many boundary checks. For all practical purposes ($s = 2, 3$) byte[] storage of scale-space maps limits the memory overhead relative to integral maps (short[] storage).

Fast rotation invariant features. We illustrate how to go beyond *directional* context and account for natural local invariances with an example of fast, multiscale, approximately rotation invariant feature. Let $\phi_1 \cdots \phi_{12}$ stand for the coordinates of an axis-aligned, centered icosahedron of radius r. Denoting by $\boldsymbol{\theta} = (j_1, j_2, r)$ the 3-dimensional feature, $f(\mathbf{x},\boldsymbol{\theta}) \triangleq \text{Median}_{v=1}^{12} |I_{j_2}(x + \phi_v) - I_{j_1}(x)|$ gives a robust summary of intensity variations around point x.

4 Lifting Decision Forests by Minimizing Cross-Validation Error Estimates

4.1 A cautionary look at Information Gain maximization

We follow the notations of section 2 but drop the node index n for convenience. Information gain maximization w.r.t. (feature, threshold) parameters $\psi = (\boldsymbol{\theta}, \tau)$ can be shown to be equivalent to a joint maximum likelihood estimation (MLE) of $\phi \triangleq (\psi, p_L, p_R)$, the node parameters and children' class predictors. Details are omitted for brevity. In other words, decision trees are usually grown by greedily, recursively splitting leaf nodes by likelihood maximization:

$$\phi^* = \arg\max_{\phi} \; \mathcal{CF}(D;\phi) \triangleq \frac{p(D;\phi)}{p^*(D)} \,. \tag{2}$$

In Eq. (2), the denominator is the data likelihood using the current leaf node predictor, whereas the numerator is the data likelihood when splitting this node with parameters ϕ into left and right children. The denominator is constant w.r.t. ϕ, as optimization of the current node has precedence in the recursive schedule.

Unfortunately MLE is subject to overtting. With DFs the risk is twofold. At a node-level, weak learners with poor generalization may be selected. The deeper the trees, the more likely it is to happen, since the training data is split between an exponentially increasing number of nodes. At a tree-level, the lack of principled control of model capacity negatively impacts generalization. Indeed the information gain of Eq. (2) is strictly positive as long as 1) training samples remain at the node of interest 2) the data distribution is not pure. As a result trees generally grow to the maximum allowed depth, with little control over generalization. Medical image segmentation tasks often call for large trees of weak learners to be grown (tree depth 20–30, millions of nodes). Due to computational constraints, few such trees can be grown (a few dozens at most), so that model averaging across randomized trees is insufficient to balance tree overfitting. As an efficient alternative to MLE that can be directly used to control tree growth and generalization, we propose to maximize the predictive score as obtained from *cross-validation* estimates.

4.2 Maximizing Cross Validation Estimates of Generalization

We derive Cross-Validation Estimates (CVE) of the predictive score as follows. At any given node, the (potentially bagged) training data $D = D^{\text{CV}} \coprod D^{\text{T}}$ is randomly divided in two disjoint subsets, a tuning subset D^{T} and a validation subset D^{CV}. The optimization problem is then defined as:

$$\phi^* = \arg\max_{\phi} \frac{p(D^{\text{CV}}; \boldsymbol{\theta})}{p^*(D^{\text{CV}})}, \quad \text{s.t.} \\ \big(\tau_{\boldsymbol{\theta}}, p_\epsilon(\cdot; \boldsymbol{\theta})\big) = \arg\max_{\tau, p_\epsilon} p_\epsilon(D^{\text{T}}_\epsilon; \phi) \tag{3}$$

where $p(D^{\text{CV}}; \boldsymbol{\theta}) \triangleq p(D^{\text{CV}}; \boldsymbol{\theta}, \tau_{\boldsymbol{\theta}}, p_\epsilon(\cdot; \boldsymbol{\theta}))$. The key change is that parameters are now constrained to be tuned on D^{T} whereas the final feature score is computed on D^{CV}. While a *k-fold* estimate could be used instead in Eq. (3), the *hold-out* procedure has the benefit of efficiency and added randomness.

A simple two stage procedure is followed, akin to that of MLE.For each candidate feature $\boldsymbol{\theta}$, the best threshold $\tau_{\boldsymbol{\theta}}$ is found by IG maximization on the tuning subset D^{T}. Left and right children' class predictors are set to the empirical tuning data histograms. Instead of directly returning the IG as a feature score, the cross-entropy between tuning and validation empirical distributions is computed and returned. As we merely replace the ML feature score by a CV score computed from subsets of readily available data, the computational complexity remains unchanged.

Key to the proposed approach is that Eq. (3) takes negative values whenever no candidate split yields superior generalization to the current leaf node model. Based on this remark, we implement a greedy scheme to control the tree complexity, where branches that do not increase the score are pruned in a single bottom-up pass as post-processing, similarly to [13]. We further use a simple heuristic to drastically reduce training time, growing trees *in stages* up to any desired maximum depth, successively pruning score-decreasing branches and regrowing remaining, non-pruned ones.

5 Auto-context forests for brain tumour segmentation

We investigate cascaded DF architectures: the classifier consists of layers of DFs partially or fully connected via their output posterior maps. Each layer solves a separate subtask, improves upon the current prediction or both. This architecture naturally allows to interleave high-level *semantic* reasoning with *intensity*-based low-level reasoning. We demonstrate this via two ideas, a) *auto-context*: allowing downstream layers to reason about semantics captured in upstream layers; and b) *decision pathway clustering*: latent data-space semantics are revealed by clustering *decision pathways* and cluster-specific DFs are trained.

5.1 Building and training Auto-Context Forests

The process of cascading DFs is illustrated in Fig. 5.1. Since DFs rely on generic context-sensitive features that disregard the exact nature of input channels (cf. section 3), we simply proceed by augmenting the set of input channels for subsequent layers with output posterior maps from previous layers.

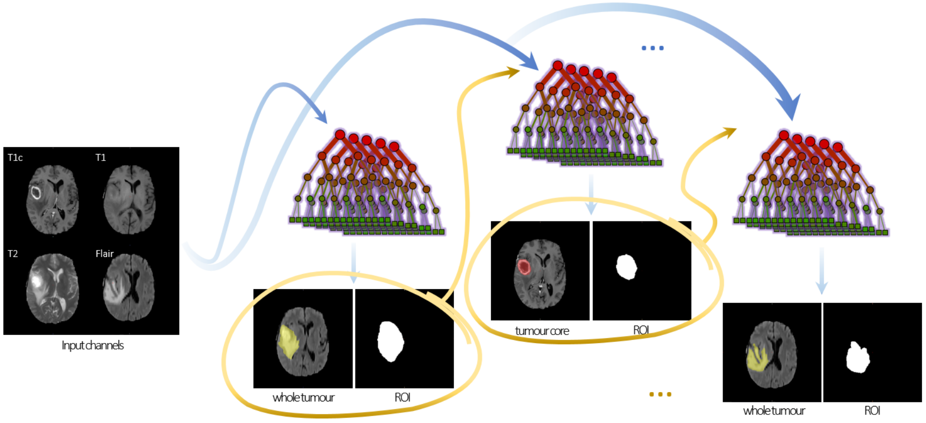

Fig. 1. Auto-Context Segmentation Forests. In this schematic example, layer 2 solves a segmentation task distinct from that of layers 1, 3 but the interleaving allows to exploit joint dependencies.

Layers are trained sequentially, one at a time in a greedy manner (following section 2, 4.2). The main appeal of the proposed approach is that it is practical, modular, fast and reliable. Specifically, we renark that the BRATS challenge labels define a *nested* structure: the whole tumour (WT) is formed of the edema (ED) and tumour core (TC). The tumour core itself is subdivided into enhancing tumour parts (ET) and other parts of the core that are only indirectly relevant to the task: the necrotic core (NC) and non-enhancing remaining parts (NE). Classically these labels would be restructured into mutually exclusive classes (ED, ET, NC, NE and the background BG) so as to recast the task as one of multiway classification. Instead the proposed framework allows to directly use these hierarchical dependencies. While many variants can reasonably be built, the final architecture that we used for the BRATS challenge consists of layers of binary DFs, alternating between prediction of WT, TC and ET.

5.2 Exposing Latent Semantics in Decision Forests for guided bagging

Given Auto-Context Forests, a natural idea is to progressively refine the region of interest (ROI) after each layer, starting from an initial over-segmentation (*e.g.* the full image). Downstream DFs are trained on tighter ROIs that exclude irrelevant background clutter, thereby increasing their accuracy. This closely relates to class rebalancing and boosting. ROIs are usually obtained via simple mathematical morphology. With such an approach, it is crucial to maintain high recall throughout the procedure, as false negatives may be definitively excluded. We investigate an alternative approach that circumvents these limitations, making ROI refinement a computational convenience.

The proposed approach exposes and exploits the latent semantics already captured within a given DF as follows. Each data point is identified with the collection of tree paths that it traverses at test time. A metric d_{DP} is defined over such collections of tree paths (*decision pathways*), assigning smaller distance between points following similar paths across many trees, and data clusters are identified by k-means w.r.t. d_{DP}. Then, cluster-specific DFs are trained over the corresponding training data. At test time, data points are assigned to the cluster with closest centroid and the corresponding DF is used for prediction.

The underlying assumption is that data points that are clustered together will share common underlying semantics, as they jointly satisfy many predicates. A wide range of metrics can be designed and for the sake of simplicity, we define (given a collection $\mathcal{T}$ of trees):

$$d_{\text{DP}}^{\mathcal{T}}(\mathbf{x}_i, \mathbf{x}_j) \triangleq \sum_{t=1}^{|\mathcal{T}|} \left(\frac{1}{2}\right)^{\text{depth}_t^{\mathcal{T}}(\mathbf{x}_i, \mathbf{x}_j)} , \tag{4}$$

where $\mathbf{x}_i, \mathbf{x}_j$ are two points, and $\text{depth}_t^{\mathcal{T}}(\mathbf{x}_i, \mathbf{x}_j)$ is the depth of the deepest common node in both paths for the t-th tree ($+\infty$ if the paths are identical).

6 BRATS challenge details: model, pre-processing & settings

6.1 BRATS 2015 dataset

6.2 Training

Preprocessing. Image masks are defined from the FLAIR modality, masking out voxels of intensity 0. The image contrast is standardized: the distribution of voxel intensities within the mask is brought to a preset, common median and mean absolute error by affine remapping. As a result images are all normalized within the same intensity range, so that the following step is mostly implementation specific. We further window intensity values to make threshold quantization easier when training DFs: intensities are thresholded to lie between some minimum and maximum values and brought within a byte range.

Initialization: SMM/MRF. An SMM-MRF layer is used to locate the region of interest (ROI) for the whole tumor. The likelihood for each of the five mutually exclusive ground truth classes is modelled using a Student Mixture (SMM) with spatially varying (BG) or fixed (other classes) proportions [2], as a suitably modified variant of [5]. An MRF prior is assumed over BG, ED and TC. The model is similar in spirit to [21,11]. The model is purely unsupervised: we do not use white/grey matter and cerebro-spinal fluid labels in the current implementation. However the learnt components for the background SMM are highly correlated to those labels. We assume another MRF prior over voxel assignments to the background SMM components, encoding the rough intuition that WM/GM/CSF should smoothly vary spatially. Variational Bayesian inference is used at training and test time. Both MRFs define fully connected cliques over the image, with Gaussian decay of pairwise potentials w.r.t. the distance of voxel centers. For this choice of potentials, the dependencies induced by MRF priors in variational updates can be efficiently computed via Gaussian filtering. Inference over $3D$ volumes is very fast both at training (seconds or minutes) and test time (seconds).

Auto-context architecture. 9 layers of binary DFs are cascaded, cycling between WT, TC and ET probabilities. All layers use the original, raw image channels. The first layers have their input augmented with probability maps from the upstream SMM/MRF, the subsequent layers use probability maps output by previous DFs. In addition, the first 3 layers are also passed the prior probability ”atlas” maps returned by the spatially-varying background SMM/MRF model.

ROI refinement. For computational convenience, subsequent layers run on ROIs rather than the full image. For instance, the second WT binary DF only tests points within the mask provided by the first WT binary DF. Similarly at training time, the second layer is trained on subsets of image voxels within the respective image ROIs output by the first layer. The first, second and third layers use masks obtained as dilated versions of the segmentation masks output by the previous layer (dilated resp. by 15mm, 10mm, 5mm).

Other settings. DFs come with a number of parameters, most of which where not found to drastically affect the pipeline accuracy. Between 100 and 200 candidate features per node. 100 thresholds.

6.3 BRATS 2016 test dataset

The pipeline described above is fully automated. To the authors' knowledge, the BRATS 2015 training dataset pre-processing includes: rigid registration (as well as resampling to a common image geometry), bias field correction and skull removal [10]. The BRATS

2016 test dataset contains a number of images for which the preprocessing is not done or only partial (cf. Fig. **??**). To cope with that, our pipeline was modified to include rigid registration and resampling, bias field correction [17] and skull removal as part of a semi-automatic pre-processing step.

6.4 Experimental setting & running time

The proposed approach is implemented in C# and F#. All experiments were performed on a 3.6GHz Intel Xeon processor system with 16GB RAM running Microsoft Windows 10. Training on the BRATS 2015 dataset took 6 to 7 hours (including ”testing” on the whole dataset). Testing took about 20s per image.

7 Experiments & Results

7.1 BRATS benchmark: Multi-modal MR brain tumor segmentation

Report running times! Compare to the literature! Try to do the BRATS 2013 leaderboard comparison (array of numbers). BRATS 2015 accuracy vs. number of training images, with varying train/test subsets. Compare to AutoGlioS baseline + to 1-layer CVE forests. Report predicted accuracy (training).

7.2 Multi-organ segmentation from CT scans

8 Discussion

This is where we mention the BRATS 2016 challenge. The interesting part is to discuss numbers found in the literature on the 2015 training set (scores in the 90’s) and the outcome at the BRATS challenge. We even have the scores for the 2016 test set, very interesting to discuss it.

Also, recall that many application-specific improvements to the DF framework have been proposed in the past years (or multi-stage procedures which DFs are only a part of) and they would also benefit us here (*e.g.* segmenting WM/GM/CSF in brain applications). The standpoint of the paper is complementary, proposing generic, efficient, widely applicable improvements to DF training that will improve accuracy in most settings.

Might be worth pointing out once more the fast experimentation / practical aspect of the approach. Compare training times to the literature here (keep the results section as clean, concise and factual as possible).

9 Conclusion

We described a principled way to train DFs using hold-out estimates of the predictive error, *lifting* the accuracy and generalization of individual nodes and of the DF altogether. The proposed node-splitting cost function induces a natural trade-off between prediction accuracy and model complexity: following an *Occam razor*-like principle branches only grow as long as a clear gain in generalization can be evidenced. We find that shallow lifted trees formed of a few dozens or hundreds of nodes outperform conventional deep trees formed of millions of nodes. This is of practical interest: it makes training, tuning and experimenting with randomized decision forests much more straightforward.

We exploit this benefit to experiment within the framework of auto-context forests, on challenging multi-class and multi-organ medical image segmentation tasks. Auto-context forests directly encode contextual cues about semantics, rather than merely raw intensities. We investigate several mechanisms to boost the accuracy of the sequence of DFs: ROI refinement, natural for image segmentation tasks; as well as a novel form of guided bagging. Data points are clustered via an inexpensive k-means scheme, based on the collection of decision paths they follow, and subsequent layers train multiple cluster-specific DFs.

Acknowledgment

The authors would like to thank... Microsoft Inria Joint Centre

References

1. Amit, Y., Geman, D.: Shape quantization and recognition with randomized trees. Neural computation 9(7), 1545–1588 (1997) 1, 3
2. Archambeau, C., Verleysen, M.: Robust bayesian clustering. Neural Networks 20(1), 129–138 (2007) 8
3. Breiman, L.: Bagging predictors. Machine learning 24(2), 123–140 (1996) 3
4. Breiman, L.: Random forests. Machine learning 45(1), 5–32 (2001) 1
5. Cordier, N., Delingette, H., Ayache, N.: A patch-based approach for the segmentation of pathologies: Application to glioma labelling. IEEE Transactions on Medical Imaging 35(4) (2015) 8
6. Criminisi, A., Robertson, D., Konukoglu, E., Shotton, J., Pathak, S., White, S., Siddiqui, K.: Regression forests for efficient anatomy detection and localization in computed tomography scans. Medical image analysis 17(8), 1293–1303 (2013) 4
7. Criminisi, A., Shotton, J.: Decision forests for computer vision and medical image analysis. Springer Science & Business Media (2013) 1
8. Geremia, E., Menze, B.H., Ayache, N.: Spatially adaptive random forests. In: 2013 IEEE 10th International Symposium on Biomedical Imaging. pp. 1344–1347. IEEE (2013) 2
9. Ho, T.K.: The random subspace method for constructing decision forests. IEEE transactions on pattern analysis and machine intelligence 20(8), 832–844 (1998) 1, 3
10. Menze, B., Jakab, A., Bauer, S., Kalpathy-Cramer, J., Farahani, K., Kirby, J., Burren, Y., Porz, N., Slotboom, J., Wiest, R., et al.: The multimodal brain tumor image segmentation benchmark (brats). IEEE Transactions on Medical Imaging 34(10), 1993–2024 (2015) 8
11. Menze, B.H., Van Leemput, K., Lashkari, D., Weber, M.A., Ayache, N., Golland, P.: A generative model for brain tumor segmentation in multi-modal images. In: International Conference on Medical Image Computing and Computer-Assisted Intervention. pp. 151–159. Springer (2010) 8

12. Pereira, S., Pinto, A., Alves, V., Silva, C.A.: Deep convolutional neural networks for the segmentation of gliomas in multi-sequence mri. In: International Workshop on Brainlesion: Glioma, Multiple Sclerosis, Stroke and Traumatic Brain Injuries. pp. 131–143. Springer (2015) 1
13. Quinlan, J.R.: Simplifying decision trees. International journal of man-machine studies 27(3), 221–234 (1987) 6
14. Shotton, J., Johnson, M., Cipolla, R.: Semantic texton forests for image categorization and segmentation. In: Computer vision and pattern recognition, 2008. CVPR 2008. IEEE Conference on. pp. 1–8. IEEE (2008) 2
15. Tu, Z.: Auto-context and its application to high-level vision tasks. In: Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on. pp. 1–8. IEEE (2008) 2
16. Tu, Z., Bai, X.: Auto-context and its application to high-level vision tasks and 3D brain image segmentation. IEEE Transactions on Pattern Analysis and Machine Intelligence 32(10), 1744–1757 (2010) 2
17. Tustison, N., Gee, J.: N4itk: Nicks n3 itk implementation for mri bias field correction. Insight Journal (2009) 9
18. Tustison, N.J., Shrinidhi, K., Wintermark, M., Durst, C.R., Kandel, B.M., Gee, J.C., Grossman, M.C., Avants, B.B.: Optimal symmetric multimodal templates and concatenated random forests for supervised brain tumor segmentation (simplified) with antsr. Neuroinformatics 13(2), 209–225 (2015) 2
19. Tustison, N., Wintermark, M., Durst, C., Avants, B.: Ants andarboles. Multimodal Brain Tumor Segmentation p. 47 (2013) 1, 2
20. Viola, P., Jones, M.: Rapid object detection using a boosted cascade of simple features. In: Computer Vision and Pattern Recognition, 2001. CVPR 2001. Proceedings of the 2001 IEEE Computer Society Conference on. vol. 1, pp. I–511. IEEE (2001) 4
21. Zhang, Y., Brady, M., Smith, S.: Segmentation of brain mr images through a hidden markov random field model and the expectation-maximization algorithm. IEEE transactions on medical imaging 20(1), 45–57 (2001) 8
22. Zikic, D., Glocker, B., Konukoglu, E., Criminisi, A., Demiralp, C., Shotton, J., Thomas, O., Das, T., Jena, R., Price, S.: Decision forests for tissue-specific segmentation of high-grade gliomas in multi-channel MR. In: International Conference on Medical Image Computing and Computer-Assisted Intervention. pp. 369–376. Springer (2012) 1