Improving Decision Ensemble Generalization with Hold-Out Error Estimates: Application to Auto-Context Image Segmentation

Anonymous CVPR submission

Paper ID 2053

Abstract

We introduce an efficient node-splitting criterion that enhances the generalization and discriminative power of nodes in decision ensembles. At an ensemble-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. We demonstrate improved accuracy and generalization on real-world datasets: our case study focuses on high precision, computationally intensive medical image segmentation tasks. To that end, we revisit auto-context forests: semantic context is progressively built and refined via successive layers of decision forests. This sequential design enables the decomposition of complex segmentation tasks into series of simpler subtasks that, for instance, exploit the hierarchical structure between labels. The approach is practical: the per-layer training is simple, modular and robust. In addition, we show that forest pathways capture latent semantics in data-space. Consequently, specializing classifiers over the resulting semantic regions is found to boost accuracy. The proposed approach was joint winner of the MICCAI 2016 BRATS (Brain Tumour Segmentation) challenge.

1. Introduction

This paper investigates efficient and generic ways to increase the accuracy and generalisation of randomized decision forests (DFs), with applications to structured classification tasks such as medical image segmentation. DFs are a practical choice for a variety of machine learning tasks such as classification and regression, widely adopted in the fields of computer vision and medical imaging [11, 4] because of their simplicity, flexibility and efficiency when handling large datasets. They have been used with success for the localisation and segmentation of multiple organs [22, 30, 12] and of anomalies [17, 45, 40] in medical volumes.

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, 19] evaluated on random subsets of training data [6]. 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 *e.g.*, for brain tumour segmentation [45], for organ localisation [10].

Still in practice, more expressive features yield more discriminative node decisions, and therefore improved segmentation accuracy. Tustison et al. [40, 41] incorporate features derived from the local intensity context and texture, from elastic registration and other task-specific processing steps. Texture-based features, volumetric features and more recently supervoxels have proven popular and successful in various applications [18, 27, 26, 9].

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 *Occam's 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* [38, 39] segmentation forests (section 5). In the

aforementioned body of work, 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 [35], 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.

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 apply these techniques to the challenging task of multi-modal MR brain tumour segmentation and achieve results on par with the state-of-the-art (section 7). We single out the impact of each individual contribution over the course of the article. We demonstrate promising results on the task of multi-organ segmentation from abdominal CT scans. A variant of the proposed methodology jointly won the MICCAI 2016 BRATS (Brain Tumour Segmentation) challenge, where it compared favorably with state-of-the-art CNN architectures.

2. Datasets

Might as well accustom the reader to the idea that we are going to apply our ideas to medical imaging from the beginning :)

3. Background on Random Forests

The following summary introduces the necessary notations and background on random forests for classification.

Let $\mathbf{x} \in \mathcal{X}$ be a point to classify (identified with its feature vector), and $c \in \mathcal{C} = \{1 \dots K\}$ the label to predict. A decision tree $(T, \mathbf{f}, \mathbf{p})$ is a triplet consisting of a proper binary tree structure $T = (\mathcal{V}, \mathcal{E}_L, \mathcal{E}_R)$ with its associated set $\mathbf{h} = \{h_n\}_{n \in \mathcal{S}}$ of binary-valued node routing functions and its associated set $\mathbf{p} = \{p_n\}_{n \in \mathcal{L}}$ of leaf class predictors. Each node in $\mathcal{V} = \mathcal{I} \coprod \mathcal{L}$ is either a leaf node $n \in \mathcal{L}$ with no child ($\mathcal{E}_L(n) = \mathcal{E}_R(n) = \varnothing$) or a split node $n \in \mathcal{S}$ with exactly one left and right children ($n_L \in \mathcal{V}$ s.t. $\mathcal{E}_L(n) = \{n_L\}$, $n_R \in \mathcal{V}$ s.t. $\mathcal{E}_R(n) = \{n_R\}$). Each leaf node $n \in \mathcal{L}$ is associated a class predictor p_n, assigning probability $p_n(c|\mathbf{x})$ for $\mathbf{x}$ to be of class c. A simple categorical distribution is usually assumed at each leaf, discarding remaining feature vector information: $p_n(c|\mathbf{x}) = p_n(c)$ such that $0 \leq p_n(c) \leq 1$, $\sum_{c=1}^{K} p_n(c) = 1$. Split nodes $n \in \mathcal{S}$ are paired with a routing function $h_n(\mathbf{x}) \triangleq [f(\mathbf{x}, \boldsymbol{\theta}_n) \leq \tau_n] \in \{0, 1\}$, with $f : \mathcal{X} \times \Theta \mapsto \mathbb{R}$ a weak learner parametrized by some node-specific feature $\boldsymbol{\theta}_n \in \Theta$ and $\tau_n \in \mathbb{R}$ a node-specific threshold.

Tree testing. Points $\mathbf{x}$ are routed down the tree starting from the root node by evaluating routing functions $h_n(\mathbf{x})$ along the path, 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. The tree then predicts class c with probability $p_{n(\mathbf{x})}(c)$.

Decision forest. A decision forest is a collection $\mathcal{T}$ of $t = 1 \cdots |\mathcal{T}|$ trees along with a rule to aggregate tree predictions. Simple averaging is widely used and yields $p(c|\mathbf{x}) = 1/|\mathcal{T}| \cdot \sum_{t=1}^{|\mathcal{T}|} p_t(c|\mathbf{x})$, 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 training. The optimal decision tree $(T, \mathbf{f}, \mathbf{p})$ is learned during a supervised training phase from training data $D = \{\mathbf{x}_i, c_i\}_{i=1}^{N}$ with known labels c_i. Training is done greedily and recursively, starting from a single root node. Let $n \in \mathcal{L}$, and $D_n = \{\mathbf{x}_i, c_i\}_{i \in \mathcal{I}_n}$ be the subset of training points reaching this node. If the stopping condition isn't reached (*e.g.* maximum tree depth or minimum number of training samples), n is changed into a split node with node feature $\boldsymbol{\theta}_n^*$ and threshold τ_n^*, pointing to left and right leaf nodes n_L and n_R paired with respective predictors $p_{n_L}^*$ and $p_{n_R}^*$. The learned node parameters $\psi^* = (\boldsymbol{\theta}_n^*, \tau_n^*)$ optimize the information gain (IG), $\psi^* \triangleq \arg\max_\psi \mathrm{IG}(\psi; D_n)$. Given some ψ splitting the node data $D_n = D_{n_L}(\psi) \coprod D_{n_R}(\psi)$ between left and right children, $\mathrm{IG}(\psi; D_n)$ is given by:

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

with

$$p_{n_\epsilon}(c; \psi) \triangleq |\{i \in \mathcal{I}_{n_\epsilon}(\psi) | c_i = c\}| \,/\, |D_{n_\epsilon}(\psi)| \tag{2}$$

the empirical data distribution (*i.e.* the class histogram). After proper quantization of thresholds τ_n, the optimum ψ^* is found by exhaustive search. The left and right class predictors are set to the observed node data distribution, $p_{n_\epsilon}^* = p_{n_\epsilon}(\cdot; \psi^*)$, $\epsilon = L, R$.

Finally, random forests introduce randomization in the training of each tree via *bagging*.

Feature 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, 19]. This is a practical form of model averaging: across trees, a range of marginally suboptimal features $\boldsymbol{\theta} \in \Theta$ are selected, which would otherwise be discarded in favour of a single, marginally score-maximizing feature.

Data bagging. For the t-th tree and at node $n \in \mathcal{V}_t$, only a random subset $D'_n \subsetneq D_n$ of training examples sampled with(out) replacement is used [5]. This reduces the variance of learned predictors and brings drastic computational gains.

Class rebalancing. Large class imbalance often induces classifier bias in favor of the more frequent class. To correct for this, training samples can be weighted according to the relative frequency of their class whenever summing over training examples. Of course class rebalancing induces the opposite bias against more prevalent classes, which can be inextricable in a multilabel setting. Section 5 discusses alternative strategies to naturally correct for distribution imbalance.

Segmentation Forests. k-way image segmentation can be seen as a voxelwise k-way classification task. Let $I = \{I_j\}_{j=1..J}$ the set of input channels in a multichannel image and x a pixel. We define $\mathbf{x} = \{x, I\}$ as the feature vector of x, and proceed as above. Examples of weak learners are given in appendix A. At training time subsets of voxels in multiple annotated images are used. 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), each voxel on the image grid is sent through the forest for its label to be predicted.

4. Lifting Decision Forests: Hold-Out Error Minimization

We show that at a given node, information gain maximization w.r.t. (feature, threshold) parameters $\psi = (\boldsymbol{\theta}, \tau)$ can be reinterpreted as a joint maximum likelihood estimation (MLE) of $\phi \triangleq (\psi, p_L, p_R)$, the node parameters and children' class predictors. We discuss MLE limitations and propose a principled alternative based on hold-out estimates of generalization. We follow the notations of section 3 but drop the node index n for convenience.

4.1. Information Gain as Maximum Likelihood

Let $(T, \mathbf{f}, \mathbf{p})$ be a decision tree. We assume the distribution of class labels c at two distinct leaf nodes to be independent given their closest ancestral node' parameters (consistency w.r.t. the tree topology), and the data at a given node

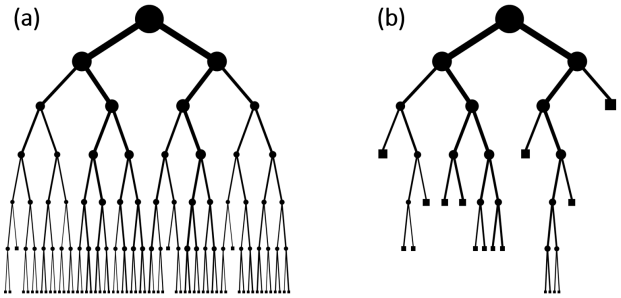

Figure 1. Typical topology of decision trees (a) trained by maximum likelihood estimation (b) trained by hold-out score maximization. In the case of MLE, trees are fully grown to the maximum allowed depth, save for leaf nodes reached by too few training samples.

to be i.i.d. knowing node parameters. Then the likelihood factorizes over leaf nodes given their parent nodes, and a given node can be singled out for greedy optimization as described in section 3. We show that decision trees are usually grown by greedily, recursively splitting leaf nodes by likelihood maximization:

$$\phi^* = \arg\max_{\phi} \; \mathcal{CF}(D;\phi) \triangleq \frac{p(D;\phi)}{p^*(D)} \,. \tag{3}$$

In Eq. (3), 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.

Let $\mathcal{L}^L(D) \triangleq \log p^*(D)$ and $\mathcal{L}^S(D;\phi) \triangleq \log p(D;\phi)$. The log likelihood $\mathcal{L}^S(D;\phi)$ under the split model factorizes over children nodes into $\log p_L(D_L(\psi)) + \log p_R(D_R(\psi))$. Expanding over data points, the log likelihood ratio $\mathcal{L}^S(D;\phi) - \mathcal{L}^L(D)$ rewrites as:

$$\sum_{\epsilon\in\{L,R\}} \sum_{i\in\mathcal{I}_\epsilon(\psi)} \log p_\epsilon(c_i|\mathbf{x}_i) - \sum_{i\in\mathcal{I}} \log p^*(c_i|\mathbf{x}_i) \,, \tag{4}$$

where $\mathcal{L}^L(D)$, the second term, is constant w.r.t. ϕ. Under a categorical probability model $p_\epsilon(c|\mathbf{x}) = p_\epsilon(c)$, we obtain:

$$\begin{aligned} \sum_{\epsilon\in\{L,R\}} \frac{|D_\epsilon(\psi)|}{|D|} \sum_{c\in\mathcal{C}} p_\epsilon(c;\psi) \log p_\epsilon(c) \\ - \sum_{c\in\mathcal{C}} p(c) \log p^*(c) \,, \end{aligned} \tag{5}$$

with $p_\epsilon(c;\psi)$ and $p(c)$ denoting empirical data distributions (class histograms) as before. For any ψ, $p^*_\epsilon = p_\epsilon(\cdot;\psi)$ can be shown to maximize Eq. (5), and ML maximization w.r.t. ϕ comes down to maximization of the IG of Eq. (1) w.r.t. ψ after setting the class predictors to observed empirical distributions.

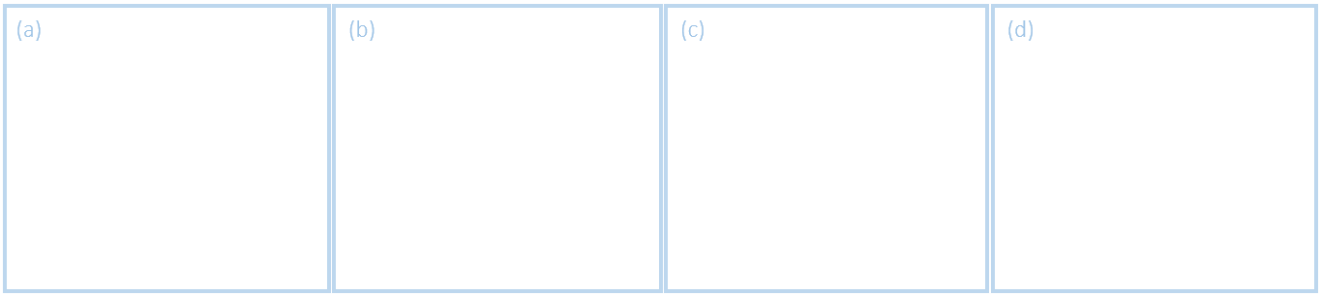

Figure 2. Comparison of MLE and CVE. (a) Accuracy vs. depth: 1 tree, 10 trees, 100 trees. Averaged over trees (just train 500 trees), one dataset 10 train/10 test if easier. Graph plot. (b) Effect of the training set size. (30 to 50 images total. Average/statistics over several training/test sets). Is there a saturation effect when most of the variability is seen in the training data? Graph plot w/ quantiles. (c) Effect of the # of candidate features. Is MLE more sensitive? Bar/box plot (say: 10, 100, 1000 features). (d) Accuracy vs. number of nodes (avg/tree at different depths) MLE vs CVE greedy pruning. Emphasizes point about accuracy & compactness. Graph plot. (e) Accuracy vs. number of nodes for individual trees (again, do this over different depths). MLE vs CVE w/ pruning.

4.2. Risk of overfitting

The risk is twofold: at a node-level, known limitations of MLE (emphasized when using several weak learners of varying complexity, cf. Fig. 3) and at a tree-level, lack of principled control of the model capacity. High accuracy vision tasks such as medical image segmentation 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). Model averaging across randomized trees is insufficient to balance overfitting.

Eq. (3) mistakenly appears to provide a natural mechanism to control tree growth, such as by splitting a node if and only if the likelihood ratio $\mathcal{CF}(D;\phi^*) > 1$. However the condition holds as long as training samples remain at both leaves – *i.e.* $D_\epsilon(\psi^*) \neq \varnothing$ for $\epsilon = L,R$ – and the leaf distribution is not pure. Hence the tree complexity grows unbounded regardless of the amount of training data. Therefore, artificial rules that impose a minimum number of training samples per node are sometimes used instead. They are highly sensitive to the learning rate at each node and to the overall size of the training set. As a result trees generally grow to the maximum allowed depth (Fig. 1), with little control over generalization.

Figure 3. Bar plots showing the feature %use for MLE and CVE, 10, 100, 1000 candidate features. To review depending on interpretability.

As an efficient alternative to MLE that can be directly used to control tree growth and generalization, we now propose to maximize the predictive score as obtained from *cross-validation* estimates.

4.3. 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:

$$\begin{aligned} \phi^* &= \arg\max_{\phi} \frac{p(D^{\text{CV}};\boldsymbol{\theta})}{p^*(D^{\text{CV}})}\,, \quad \text{s.t.} \\ \big(\tau_{\boldsymbol{\theta}}, p_\epsilon(\cdot;\boldsymbol{\theta})\big) &= \arg\max_{\tau, p_\epsilon} p_\epsilon(D^{\text{T}}_\epsilon;\phi) \end{aligned} \tag{6}$$

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. (6), the *hold-out* procedure has the benefit of efficiency and added randomness.

For classification, the CVE of the feature $\boldsymbol{\theta}$ is easily seen to achieve minimum cross-entropy between tuning and validation empirical distributions at children nodes $\epsilon = L, R$ (weighted by the number of points at each node). Equivalently it minimizes:

$$\begin{aligned} \sum_{\epsilon\in\{L,R\}} \frac{|D^{\text{CV}}_\epsilon(\psi_{\boldsymbol{\theta}})|}{|D^{\text{CV}}|} \sum_{c\in\mathcal{C}} p^{\text{CV}}_\epsilon(c;\psi_{\boldsymbol{\theta}}) \log p^{\text{T}}_\epsilon(c;\psi_{\boldsymbol{\theta}}) \\ - \sum_{c\in\mathcal{C}} p^{\text{CV}}(c) \log p^*(c)\,, \end{aligned} \tag{7}$$

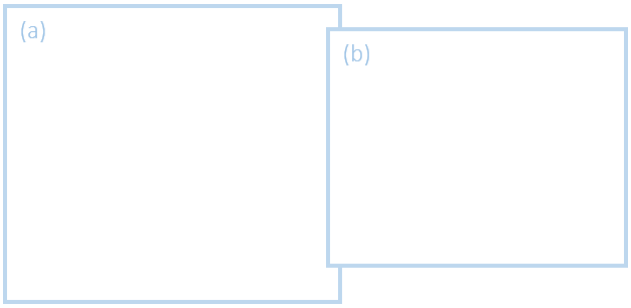

Figure 4. (a) Accuracy vs #nodes. MLE vs. CVE vs. CVE greedy growing vs CVE greedy pruning (50 or 100 trees, whatever).point cloud for individual tree accuracies (do this over different depths) (b) Smaller: number of nodes vs. depth for the three variants. Bar plot, stacking the three variants? Two types of features? Not necessary, it just changes absolute values.

where $\psi_{\boldsymbol{\theta}} = (\boldsymbol{\theta}, \tau_{\boldsymbol{\theta}})$. Eq. (6) and (7) now incorporate an explicit, assumption-free mechanism to select weak learners with good predictive power. In our experiments, this empirically leads to good predictive power of the decision ensembles (Fig. 4).

Algorithmics. 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 score of Eq. (7) is computed and returned.

The computational burden in training classification trees is dominated by the aggregation of sufficient statistics for node training. As we merely replace the ML feature score by a CV score computed from subsets of readily available data, the computational complexity remains unchanged.

Data fit vs. model complexity. Key to the proposed approach is that Eq. (7) takes negative values whenever no candidate split yields superior generalization to the current leaf node model. Based on this remark, we consider two schemes to control the tree complexity.

Greedy growing: Whenever the current node training fails to return a feature with positive score, the node is not split and the tree branch stops growing.

Greedy pruning: The tree is trained down to some preset depth, and branches that do not increase the score are pruned as post-processing. Scores are accumulated in a single bottom-up pass over the tree. Here the likelihood is computed over all of the node dataset[1] for consistency, although individual nodes may have been trained over distinct, random data subsets.

[1]In our implementation this is in fact a byproduct of the greedy tree training stage.

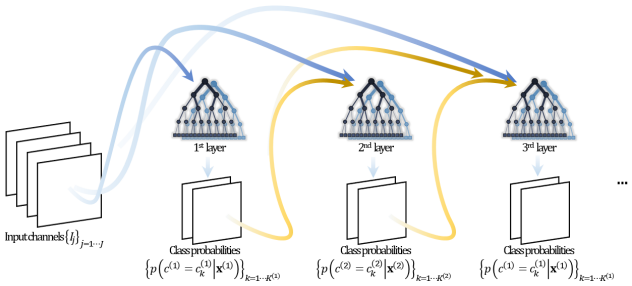

Figure 5. Auto-Context Segmentation Forests. In this schematic example, layer 2 solves a segmentation task distinct from that of layers 1, 3 but the interleaving allows to exploit joint dependencies.

Greedy pruning is found to be significantly more accurate while returning compact trees (Fig. 4). It allows temporary suboptimal node splits to be compensated for by later returns along a given branch, which seems to be empirically useful in multiclass or otherwise challenging applications.

Tree growing and greedy pruning can be performed in multiple stages to train very deep, close to optimal trees without being burdened by the exponential growth of the number of nodes. During each stage, leaf nodes that are not post-pruned are added to a training queue. The queue initially contains a single root node. The queued nodes are then expanded into fixed-depth ($d \ll$ max depth) subtrees then post-pruned, until the maximum depth is reached or the queue is empty.

5. Application to Semantic Image Segmentation

We experiment within a meta-architecture of cascaded DFs. Classifiers are built as 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 intertwine 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. Since DFs rely on generic context-sensitive features that disregard the exact nature of input channels (cf. examples in Appendix A), we simply proceed by augmenting the set of input channels for subsequent layers with output posterior maps from previous layers.

Layers are trained sequentially, one at a time in a greedy

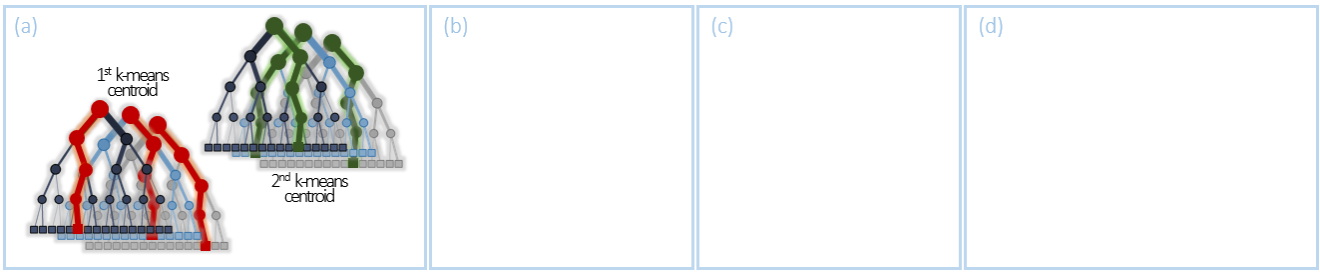

Figure 6. Clustering forest decision pathways. (a) Illustration of two centroids returned by the k-means scheme. Each centroid is a collection of paths (one per decision tree). (b) Visualization of the cluster assignments of voxels for an example image. Each colour corresponds to a different cluster. Clusters closely relate to intuitions of *uncertain boundary*, *inside* and *background*. (c) Example segmentation without (left) and with (right) the recourse to clustering. (d) Effect of ROI refinement and clustering. Three layers of DFs are used. We experiment with two ROI refinement settings, and for each setting and layer the accuracy with and without clustering is displayed. The baseline does not use ROI refinement.

manner (following section 3, 4.3). Alternative strategies are reviewed and discussed in section 6. The main appeal of the proposed approach is that it is practical, modular, fast and reliable. We emphasize the following points: a) the ease of adjusting the layer design, of training and tuning individual layers; b) the opportunity to mix DF layers with other forms of processing; and c) the benefit of dividing a challenging hierarchical task into simpler subtasks addressed independently albeit communicating with each other.

For instance, it is common for labels to define *nested* structures. Classically these labels would be restructured into mutually exclusive classes so as to recast the task as one of multiway classification. Instead the proposed framework allows to directly use and learn these hierarchical dependencies by interweaving binary DFs.

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.

5.2. Exposing Latent Semantics in Decision Forests

Datasets commonly contain several distinct "clusters" of data, *e.g.* two subtypes of anomalies under identical labels, or subgroups of images with varying acquisition protocols. In that case cluster proportions in the training data are often arbitrary, similarly to the proportions between background and foreground labels. While boosting may partially address the issue, its simplest variants are sensitive to mislabelling in the training data [24, 25] (as commonly occurs in manually annotated datasets) whereas noise-tolerant alternatives remain more intricate and costly.

Instead 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):

$$d_{\text{DP}}^{\mathcal{T}}(\mathbf{x}_i, \mathbf{x}_j) \triangleq \sum_{t=1}^{|\mathcal{T}|} \left(\frac{1}{2}\right)^{\text{depth}_t^{\mathcal{T}}(\mathbf{x}_i, \mathbf{x}_j)} , \qquad (8)$$

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. Relation to Other Work

To the authors' knowledge, cross-validation estimates of the generalization error were not previously proposed as a node-level cost to optimize. Cross-validation and out-of-bag (OOB) estimates have been an integral component of bagging classifiers, for the purpose of *monitoring* error and other key statistics, dating back to *e.g.* Wolpert and Macready [43] and Breiman [6]. In addition they have a long history for the purpose of optimal pruning of decision

trees, as in the early work of Breiman et al. [7] (see *e.g.* [13] for a review and analysis of such methods). Various sophisticated techniques such as minimum description length [32, 28] and other information-theoretic methods [14] have also been employed solely for the purpose of pruning fully-grown trees. Our proposed approach differs in that the primary goal is to select better weak learners at each node, preventing trees from overfitting regardless of the technicalities controlling tree growth. The pruning strategy that ensues is closely related to that of Quinlan [31] and comes free of additional computations.

Foundational research on random forest models remains active [34, 33, 3] to close the persistent gap between simplified, idealized models and actual algorithms. Biau [2] points out how crucially the behaviours of *single* tree models and *infinite* DF models differ in terms of generalization w.r.t. tree depth. With deep, fully grown trees, model averaging is one of the key mechanisms that induces consistent estimates. In practice data and computational resources are limited. Finite DF models are trained within the budget limits, so that the need for a principled control of model capacity node- and tree-wise is evident. Shotton et al. [36] give empirical cues as to the impact of tree depth and width on generalization. Note that *decision jungles*' node merging capabilities fully stem from the fixed-width of the directed acyclic graph. The node splitting criterion introduced in the present work may constitute a worthy alternative to induce that mechanism without explicitely constraining width.

The technical hurdles in experimenting with DFs and tuning them, in real-world applications, may explain in part why relatively little work has been done with regard to auto-context segmentation forests. Montillo et al. [29] investigate an alternative where semantic context is built within a single DF layer, by designing entanglement features that can access ancestral nodes' label predictions at neighbouring voxels (cf. also [20]). The drawback is that a given tree can only access its own auto-context, which may be significantly weaker than the whole forest's. In the medical imaging literature, related work includes that of Zografos et al. [46], who intertwine a hierarchical supervoxel representation with two cascaded layers of gradient-boosted forests; and that of Gauriau et al. [16] with a two-layer global-local DF-based framework for multi-organ localization. In contrast we cascade forests at will, by capitalizing on the ease of training each *lifted* DF in the cascade.

The motivation for clustering decision pathways is similar to that for the *guided bagging* scheme proposed for Laplacian forests [23] with the main difference that Lombaert et al. cluster *whole* images based on a predefined metric, whereas the proposed approach clusters image *voxels* based on a metric induced by supervised classification forests. For instance in the multi-organ segmentation setting, the main clusters in our approach relate to individual organs over which downstream cluster-specific DFs effectively specialize. It is also related in spirit to the spatially-localized random forests of Zhang et al. [44]. Lastly, DFs have also been proposed specifically for tasks of clustering [21, 8]. Here, rather than introducing a separate clustering step, latent semantics captured by the classification DF's pathways are directly reused to guide classification in subsequent layers.

7. Experiments & Results

7.1. Experimental setting

The proposed approach is implemented within the DF framework described in [45]. All experiments were performed on a 3.6GHz Intel Xeon processor system with 16GB RAM running Microsoft Windows 10. Windowing details, etc.

7.2. 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 Auto-GlioS baseline + to 1-layer CVE forests. Report predicted accuracy (training).

7.3. 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] Y. Amit and D. Geman. Shape quantization and recognition with randomized trees. *Neural computation*, 9(7):1545–1588, 1997. 1, 3

[2] G. Biau. Analysis of a random forests model. *Journal of Machine Learning Research*, 13(Apr):1063–1095, 2012. 7

[3] G. Biau and E. Scornet. A random forest guided tour. *Test*, 25(2):197–227, 2016. 7

[4] A. Bosch, A. Zisserman, and X. Munoz. Image classification using random forests and ferns. In *IEEE International Conference on Computer Vision*, 2007. 1

[5] L. Breiman. Bagging predictors. *Machine learning*, 24(2):123–140, 1996. 3

[6] L. Breiman. Random forests. *Machine learning*, 45(1):5–32, 2001. 1, 6, 9

[7] L. Breiman, J. Friedman, C. J. Stone, and R. A. Olshen. *Classification and regression trees*. CRC press, 1984. 7

[8] S. Conjeti, A. Katouzian, A. Kazi, S. Mesbah, D. Beymer, T. F. Syeda-Mahmood, and N. Navab. Metric hashing forests. *Medical Image Analysis*, 2016. 7

[9] P.-H. Conze, F. Rousseau, V. Noblet, F. Heitz, R. Memeo, and P. Pessaux. Semi-automatic liver tumor segmentation in dynamic contrast-enhanced CT scans using random forests and supervoxels. In *International Workshop on Machine Learning in Medical Imaging*, pages 212–219. Springer, 2015. 1

[10] A. Criminisi, D. Robertson, E. Konukoglu, J. Shotton, S. Pathak, S. White, and K. Siddiqui. Regression forests for efficient anatomy detection and localization in computed tomography scans. *Medical image analysis*, 17(8):1293–1303, 2013. 1, 9

[11] A. Criminisi and J. Shotton. *Decision forests for computer vision and medical image analysis*. Springer Science & Business Media, 2013. 1

[12] R. Cuingnet, R. Prevost, D. Lesage, L. D. Cohen, B. Mory, and R. Ardon. Automatic detection and segmentation of kidneys in 3D CT images using random forests. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pages 66–74. Springer, 2012. 1

[13] F. Esposito, D. Malerba, G. Semeraro, and J. Kay. A comparative analysis of methods for pruning decision trees. *IEEE transactions on pattern analysis and machine intelligence*, 19(5):476–491, 1997. 7

[14] R. S. Forsyth, D. D. Clarke, and R. L. Wright. Overfitting revisited: an information-theoretic approach to simplifying discrimination trees. *Journal of Experimental & Theoretical Artificial Intelligence*, 6(3):289–302, 1994. 7

[15] J. Gall, A. Yao, N. Razavi, L. Van Gool, and V. Lempitsky. Hough forests for object detection, tracking, and action recognition. *IEEE transactions on pattern analysis and machine intelligence*, 33(11):2188–2202, 2011. 9

[16] R. Gauriau, R. Cuingnet, D. Lesage, and I. Bloch. Multi-organ localization with cascaded global-to-local regression and shape prior. *Medical image analysis*, 23(1):70–83, 2015. 7

[17] E. Geremia, O. Clatz, B. H. Menze, E. Konukoglu, A. Criminisi, and N. Ayache. Spatial decision forests for MS lesion segmentation in multi-channel magnetic resonance images. *NeuroImage*, 57(2):378–390, 2011. 1

[18] E. Geremia, B. H. Menze, and N. Ayache. Spatially adaptive random forests. In *2013 IEEE 10th International Symposium on Biomedical Imaging*, pages 1344–1347. IEEE, 2013. 1

[19] T. K. Ho. The random subspace method for constructing decision forests. *IEEE transactions on pattern analysis and machine intelligence*, 20(8):832–844, 1998. 1, 3

[20] P. Kontschieder, S. R. Bulò, A. Criminisi, P. Kohli, M. Pelillo, and H. Bischof. Context-sensitive decision forests for object detection. In *Advances in neural information processing systems*, pages 431–439, 2012. 7

[21] E. Konukoglu, B. Glocker, D. Zikic, and A. Criminisi. Neighbourhood approximation using randomized forests. *Medical image analysis*, 17(7):790–804, 2013. 7

[22] V. Lempitsky, M. Verhoek, J. A. Noble, and A. Blake. Random forest classification for automatic delineation of myocardium in real-time 3D echocardiography. In *Proceedings of the 5th International Conference on Functional Imaging and Modeling of the Heart*, FIMH '09, pages 447–456, Berlin, Heidelberg, 2009. Springer-Verlag. 1

[23] H. Lombaert, D. Zikic, A. Criminisi, and N. Ayache. Laplacian forests: semantic image segmentation by guided bagging. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pages 496–504. Springer, 2014. 7

[24] P. Long and R. Servedio. Adaptive martingale boosting. In *Advances in Neural Information Processing Systems*, pages 977–984, 2009. 6

[25] P. M. Long and R. A. Servedio. Random classification noise defeats all convex potential boosters. *Machine Learning*, 78(3):287–304, 2010. 6

[26] R. C. Lozoya, J. Margeta, L. Le Folgoc, Y. Komatsu, B. Berte, J. S. Relan, H. Cochet, M. Haïssaguerre, P. Jaïs, N. Ayache, et al. Local late gadolinium enhancement features to identify the electrophysiological substrate of post-infarction ventricular tachycardia: a machine learning approach. *Journal of Cardiovascular Magnetic Resonance*, 17(1):1, 2015. 1

[27] D. Mahapatra and J. M. Buhmann. Prostate MRI segmentation using learned semantic knowledge and graph cuts. *IEEE Transactions on Biomedical Engineering*, 61(3):756–764, 2014. 1

[28] M. Mehta, J. Rissanen, and R. Agrawal. MDL-based decision tree pruning. In *Proceedings of the International Conference on Knowledge Discovery and Data Mining (KDD'95)*, pages 216–221, August 1995. 7

[29] A. Montillo, J. Shotton, J. Winn, J. E. Iglesias, D. Metaxas, and A. Criminisi. Entangled decision forests and their application for semantic segmentation of CT images. In *Biennial International Conference on Information Processing in Medical Imaging*, pages 184–196. Springer, 2011. 7

[30] O. Pauly, B. Glocker, A. Criminisi, D. Mateus, A. M. Möller, S. Nekolla, and N. Navab. Fast multiple organ detection and localization in whole-body mr dixon sequences. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pages 239–247. Springer, 2011. 1

[31] J. R. Quinlan. Simplifying decision trees. *International journal of man-machine studies*, 27(3):221–234, 1987. 7

[32] J. R. Quinlan and R. L. Rivest. Inferring decision trees using the minimum description length principle. *Inf. Comput.*, 80(3):227–248, March 1989. 7

[33] E. Scornet. On the asymptotics of random forests. *Journal of Multivariate Analysis*, 146:72–83, 2016. 7

[34] E. Scornet, G. Biau, J.-P. Vert, et al. Consistency of random forests. *The Annals of Statistics*, 43(4):1716–1741, 2015. 7

[35] J. Shotton, M. Johnson, and R. Cipolla. Semantic texton forests for image categorization and segmentation. In *Computer vision and pattern recognition, 2008. CVPR 2008. IEEE Conference on*, pages 1–8. IEEE, 2008. 2

[36] J. Shotton, T. Sharp, P. Kohli, S. Nowozin, J. Winn, and A. Criminisi. Decision jungles: Compact and rich models for classification. In *Advances in Neural Information Processing Systems*, pages 234–242, 2013. 7

[37] J. Shotton, J. Winn, C. Rother, and A. Criminisi. Textonboost for image understanding: Multi-class object recognition and segmentation by jointly modeling texture, layout, and context. *International Journal of Computer Vision*, 81(1):2–23, 2009. 9

[38] Z. Tu. Auto-context and its application to high-level vision tasks. In *Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on*, pages 1–8. IEEE, 2008. 1

[39] Z. Tu and X. Bai. 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. 1

[40] N. Tustison, M. Wintermark, C. Durst, and B. Avants. Ants andarboles. *Multimodal Brain Tumor Segmentation*, page 47, 2013. 1

[41] N. J. Tustison, K. Shrinidhi, M. Wintermark, C. R. Durst, B. M. Kandel, J. C. Gee, M. C. Grossman, and B. B. Avants. Optimal symmetric multimodal templates and concatenated random forests for supervised brain tumor segmentation (simplified) with antsr. *Neuroinformatics*, 13(2):209–225, 2015. 1

[42] P. Viola and M. Jones. 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*, volume 1, pages I–511. IEEE, 2001. 9

[43] D. H. Wolpert and W. G. Macready. An efficient method to estimate bagging's generalization error. *Machine Learning*, 35(1):41–55, 1999. 6

[44] L. Zhang, Q. Wang, Y. Gao, G. Wu, and D. Shen. Concatenated spatially-localized random forests for hippocampus labeling in adult and infant MR brain images. *Neurocomputing*, 2016. 7

[45] D. Zikic, B. Glocker, E. Konukoglu, A. Criminisi, C. Demiralp, J. Shotton, O. Thomas, T. Das, R. Jena, and S. Price. 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*, pages 369–376. Springer, 2012. 1, 7, 10

[46] V. Zografos, A. Valentinitsch, M. Rempfler, F. Tombari, and B. Menze. Hierarchical multi-organ segmentation without registration in 3D abdominal CT images. In *MCV workshop (MICCAI 2015)*, 2015. 7

A. Fast scale-space context-sensitive features

$3D$ medical image segmentation demands scalable, robust, high precision solutions. We show that scale-space representation offers a simple alternative to the popular integral or Haar-like features [42] to craft fast, expressive, compactly parametrized features.

Background: integral features. Integral features are based on intensity averages within anisotropic cuboids offset from the point of interest [37, 15, 10]. Cuboid averages are computed in constant time by probing the value of a precomputed integral map at the cuboid vertices [42]. 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 strong weak-learners [6] to be reachable within the budget allocated to feature sampling and optimization. Hence reducing the feature parametrization while

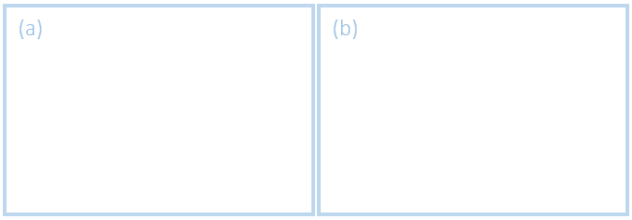

Figure 7. (a) Accuracy vs. choice of feature for CVE (bar plot): baseline (Haar features) vs. baseline + line response vs. scale-space vs. scale-space + rotation-invariant. Mention total running time for each setting (either number or chart). (b) Average computation time for different feature types.

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 $j_\epsilon \bmod s$ 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. Fig. 7 compares the feature impact on speed and accuracy to that of a directional line response introduced in [45].