Entangled Decision Forests and their Application for Semantic Segmentation of CT Images

Albert Montillo[1,2], Jamie Shotton[2], John Winn[2], Juan Eugenio Iglesias[2,3],
Dimitri Metaxas[4], and Antonio Criminisi[2]

[1] GE Global Research Center, Niskayuna, NY, USA montillo@ge.com,
[2] Microsoft Research, Cambridge, UK
{jamie.shotton, jwinn, antcrim}@microsoft.com,
[3] University of California, Los Angeles, USA jeiglesias@ucla.edu,
[4] Rutgers Univeristy, Piscataway, NJ USA dnm@rutgers.edu

Abstract. This work addresses the challenging problem of simultaneously segmenting multiple anatomical structures in highly varied CT scans. We propose the entangled decision forest (EDF) as a new discriminative classifier which augments the state of the art decision forest, resulting in higher prediction accuracy and shortened decision time. Our main contribution is two-fold. First, we propose *entangling* the binary tests applied at each tree node in the forest, such that the test result can depend on the result of tests applied earlier in the same tree and at image points offset from the voxel to be classified. This is demonstrated to improve accuracy and capture long-range semantic context. Second, during training, we propose injecting randomness in a guided way, in which node feature types and parameters are randomly drawn from a learned (non-uniform) distribution. This further improves classification accuracy. We assess our probabilistic anatomy segmentation technique using a labeled database of CT image volumes of 250 different patients from various scan protocols and scanner vendors. In each volume, 12 anatomical structures have been manually segmented. The database comprises highly varied body shapes and sizes, a wide array of pathologies, scan resolutions, and diverse contrast agents. Quantitative comparisons with state of the art algorithms demonstrate both superior test accuracy and computational efficiency.

Keywords: Entanglement, auto-context, decision forests, CT, segmentation.

1 Introduction

This paper addresses the challenging problem of automatically parsing a 3D Computed Tomography (CT) scan into its basic components. Specifically, we wish to recognize and segment organs and anatomical structures as varied as the aorta, pelvis, and the lungs, simultaneously and fully automatically. This task is cast as a voxel classification problem and is addressed via novel modifications to the popular decision forest classifier [1,2].

Background. The decision forest is experiencing rapid adoption in a wide array of information processing applications [3-8]. It can be used for clustering, regression, and as in this paper, for classification. The classifier has many attractive qualities that make it well suited for practical problems and close to an ideal universal learner [9]. It scales well computationally to large training sets, handles multi-class classification in a natural way, and the knowledge it has learned can be inspected and interpreted. In a typical image classification task [5], each pixel is classified separately. To improve segmentation results, constraints in the form of local consistency or semantic (e.g. anatomical) context are applied, but this requires either a separate random field [10] or multi-pass processing [5,11].

Our contributions. In this paper, we extend the decision forest classifier to directly enforce local consistency and semantic context without applying additional methods or passes. We show how this extension also speeds training and improves test accuracy. The two main contributions are as follows. First, to construct a tree node, *n*, at level, *L*, in the forest, we design new *entanglement* features which exploits the uncertain partial semantic information learned (or at test time, inferred) by the previous *L-1* levels of the forest about the classification of voxels in a neighborhood. Since the nodes in the resulting classifier share information with each other, we call the new classifier an *entangled* decision forest. Second, during training we randomly sample feature types and parameters from a learned, non-uniform *proposal distribution* rather than from the uniform distribution used (implicitly) in previous decision forest research [1,2,5,6,7,14,20]. The random draws select, with greater probability, the feature types and parameters that tend to be relevant for classification, allowing higher accuracy for the same number of features tested. We show how these two contributions allow faster training and prediction, more accurate prediction, and how the combination of these contributions yields best performance.

Further relevant literature. In [5], a separate decision forest is grown in each successive round of classifier training. The forest grown in each round uses semantic information learned during a previous round encoded in the form of a bag of textons that characterize decision paths down the tree of the previous round's forest. Similarly, in [11], a separate probabilistic boosting tree (PBT) [12] is constructed in each successive round. The classifier grown in each round uses semantic information from a previous round encoded as the most likely class label in a spatially offset region. These works inspired our development of EDF **to use the semantic information learned *in previous levels, in a single round* of classifier construction**. This yields a simpler, more efficient classifier than sequences of forests or PBTs and enables higher accuracy, faster training and prediction, and requires less memory.

Problem statement. We demonstrate the utility of our solution to segment 12 anatomical structures in large field of view CT. We are given density and ground truth labels at each voxel in a set of training volumes. Our goal is to infer the probability of each organ label for each voxel of unseen test scans. The task is challenging due to the extremely large variations of both healthy structures and pathology in the abdominal-thoracic region. Variations include organ location and shape, contrast agent presence/absence, scanner field of view, and image resolution.

The most closely related work for high-speed CT segmentation is non-rigid marginal space learning (MSL) [13] which uses a boosted classifier to predict the parameters of an active shape model. MSL can segment the liver boundary in 10 seconds; in contrast, our method requires only 12 seconds to segment 12 organs simultaneously. The active shape model of MSL offers resilience to noise; our method

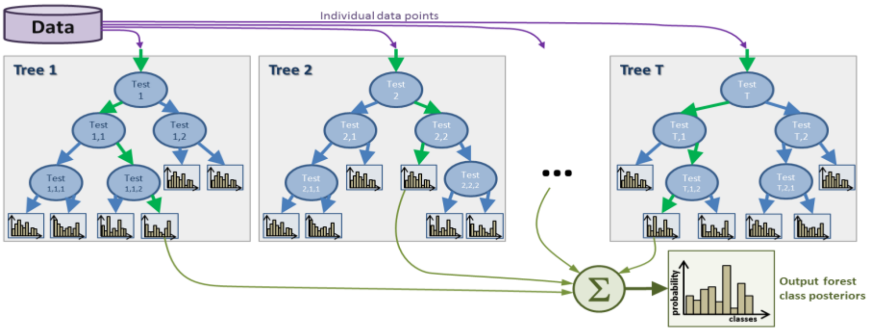


Fig. 1. Decision forest overview. During training, multiple trees are grown, using all training data for each tree. During testing, to classify a voxel it is initialized at the root node of each tree, and recursively sent left or right (red arrows) according to binary tests stored at each node. The voxel is classified using the average of the T posterior label distributions, with one coming from the leaf reached in each of the T trees.

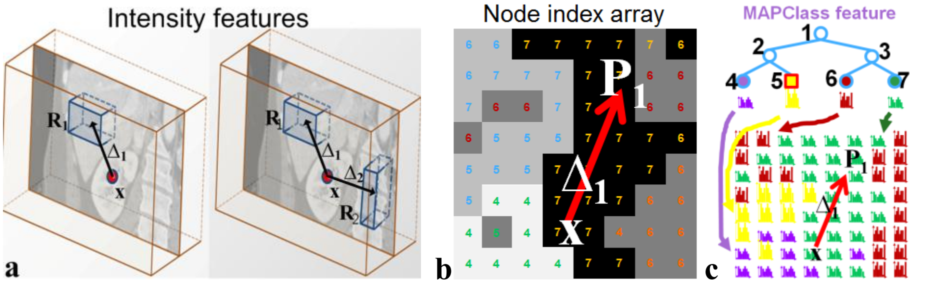
also offers flexibility to handle organs only partially visible within the scanner field of view or which have topological changes due to pathology (cists, tumors) as demonstrated in the results section.

The EDF is a new discriminative classifier which improves the accuracy and speed of the state of the art decision forest for image analysis. Our methodology may be used to improve results for other applications that reply upon the decision forest classifier, including MS lesion segmentation [8], brain segmentation [6], myocardium delineation [7], and beyond these medical applications for broad applicability in the field of computer vision, such as for object recognition [10].

2 Methods

2.1 Decision forest background

We begin with a brief review of randomized decision forests [1,2]. A decision forest is an ensemble of T decision trees. During *training*, the data (Fig. 1), consists of the set of data points from all training images, $S=\{v_i,l_i\}_1^N$. Each data point, s_i, consists of the voxel position, v_i, and its label, l_i. Tree t_i, receives the full set S and its root node selects a test to split S into two subsets to maximize information gain. A test consists of a feature (e.g. an image feature) and a feature response threshold. The left and right child nodes receive their respective subsets of S and the process is repeated at each child node to grow the next level of the tree. Growth stops when one or more stopping criteria, such as minimal information gain or a maximum tree depth occur. Each tree is unique because each tree node selects a random subset of the features and thresholds to try. During *testing*, the data (Fig. 1) consists of the voxel positions in a test image. The voxels are routed to one leaf in each tree by applying the test (selected during training) which is stored in each node. The test is applied to the voxel in the test image. The test result guides the voxel to the left or right child node, and this is



6	6	7	7	7	7	7	6
6	7	7	7	7	7	6	6
7	6	6	7	7		6	6
6	5	5	5	7	7	7	6
5	5	5	7		7	7	6
5	4	4	7	7	7	6	6
4	5	4	7	7	4	6	6
4	4	4	4	7	7	6	6

Fig. 2. Intensity and `MAPClass` features. (a) Intensity features measure image information from regions offset from the voxel to classify at **x**. (b) `MAPClass` feature retrieves the label that the classifier currently predicts at location $\mathbf{P}_1$ offset from **x**. Implementation-wise, we maintain a node index array which associates with each voxel the current tree node ID (represented by the number in each voxel). (c, top) This allows us to determine the current label posterior in the tree for the voxel at location $\mathbf{P}_1$. (c, bottom) Conceptually, the tree induces a vector image of class posteriors which we used when developing the `MAPClass` and `TopNClasses` features.

repeated until a leaf node is reached. An empirical distribution over classes learned from the training data is stored at each leaf. The voxel is classified by averaging the class distributions from the set of leaves it reached. The following section describes the features we use to define the node tests of our decision forest.

2.2 Context rich, long-range visual features

It has been shown [22] that to classify a voxel at a given location anatomical context from regions up to 200mm away are often very helpful. Therefore we do not use traditional features such as Haar wavelets [15] whose range is too short. Instead we construct two types of long-range, context-rich features. The first capture "appearance context", the later capture "semantic context". This will be explained next.

Appearance features. We construct intensity features that are spatially defined by (1) their position, **x**, centered on the voxel to be labeled (Fig. 2a), and (2) one or two rectangular probe regions, $\mathbf{R}_1$ and $\mathbf{R}_2$, offset from **x** by displacements Δ_1 and Δ_2 which can be up to 200mm in each dimension (x,y,z). We construct two categories of intensity features. The first category consists of the mean CT intensity at a probed region, $\mathbf{R}_1$ (Fig 2a, left), while the second consists of the difference in the mean intensity at probed regions, $\mathbf{R}_1$ and $\mathbf{R}_2$ (Fig 2a, right). These are defined as follows:

$$f_{Intensity}\left(\mathbf{x};\boldsymbol{\Delta}_1,\mathbf{R}_1\right)=\bar{I}\left(\mathbf{R}_1\left(\mathbf{x}+\boldsymbol{\Delta}_1\right)\right) \tag{1}$$

$$f_{IntensityDiff}\left(\mathbf{x};\boldsymbol{\Delta}_1,\mathbf{R}_1,\boldsymbol{\Delta}_2,\mathbf{R}_2\right)=\bar{I}\left(\mathbf{R}_1\left(\mathbf{x}+\boldsymbol{\Delta}_1\right)\right)-\bar{I}\left(\mathbf{R}_2\left(\mathbf{x}+\boldsymbol{\Delta}_2\right)\right) \tag{2}$$

During training, the features to try at each node are parameterized by dimensions of $\mathbf{R}_1$ and $\mathbf{R}_2$, offsets Δ_1 and Δ_2 and an intensity threshold α. These parameters are chosen randomly to define the intensity test: $f(.)>\alpha$. Once training has finished, the max information gain node test along with its optimal features are frozen and stored within the node for later use during testing.

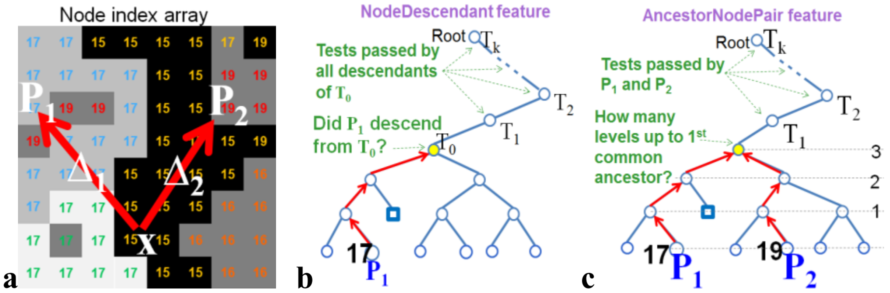


Fig. 3. Additional entanglement features. (a) Node index array associates voxels with intensity and tree node indices (same format as Fig. 2b but for a deeper tree level). (b) NodeDescendant feature tests whether probe voxel at $\mathbf{P_1}$ descends from a node (T_0 in this case). (c) AncestorNodePair feature tests whether the nodes of voxels $\mathbf{P_1}$ and $\mathbf{P_2}$ have a common ancestor $< \tau$ levels away.

Semantic context entanglement features. We now describe the first contribution of our paper. The basic idea is that during testing on novel images, we exploit the confident voxel label predictions (peaked distributions) that can be found using just early levels of the forest to aid the labelling of nearby voxels. This provides semantic context similar to auto-context [5,11], but does so within the same forest. We define four types of long range entanglement features to help train the node currently being grown using knowledge learned in already trained split nodes of the forest. Two features (`MAPClass` and `TopNClasses`) are based on the posterior class distribution of the nodes corresponding to probed voxels, and two (`NodeDescendant` and `AncestorNodePair`) are based on the location of the nodes within the trees.

MAPClass entanglement features. As the name suggests, this type of feature uses the maximum a posteriori label of a neighboring voxel at $\mathbf{P_1}$ in order to reduce uncertainty about the label at $\mathbf{x}$ (Fig 2b). When such semantic context is helpful to classify the voxel at $\mathbf{x}$ the feature yields high information gain and may become the winning feature for the node during tree growth. `MAPClass` tests whether the MAP class in the posterior of a probed voxel $\mathbf{P_1} = \mathbf{x} + \mathbf{\Delta_1}$ is equal to a particular class, C:

$$f_{MAPClass}(\mathbf{x}; \mathbf{\Delta}_1, \mathbf{P}_1, C) = \begin{cases} \arg\max\limits_{c} p(c; n_{\mathbf{P}_1}) = C & 1 \\ \quad \text{otherwise} & 0 \end{cases} \tag{3}$$

where $p(c; n_{\mathbf{P}_1})$ is the posterior class distribution of the node of $\mathbf{P_1}$. This posterior can be retrieved from the tree because we (1) train and test voxels in breadth first fashion and (2) maintain an association between voxels and the tree node ID at which they reside while moving down the tree. This association is a node index array (Fig 2b).

TopNClasses entanglement features. Similarly we define features, called `TopNClasses`, where $N \in \{2,3,4\}$ that generalize the `MAPClass` feature. A `TopNClass` feature tests whether a particular class C is in the top N classes of the posterior class distribution of the probe voxel at $\mathbf{P_1} = \mathbf{x} + \mathbf{\Delta_1}$. The feature is defined as:

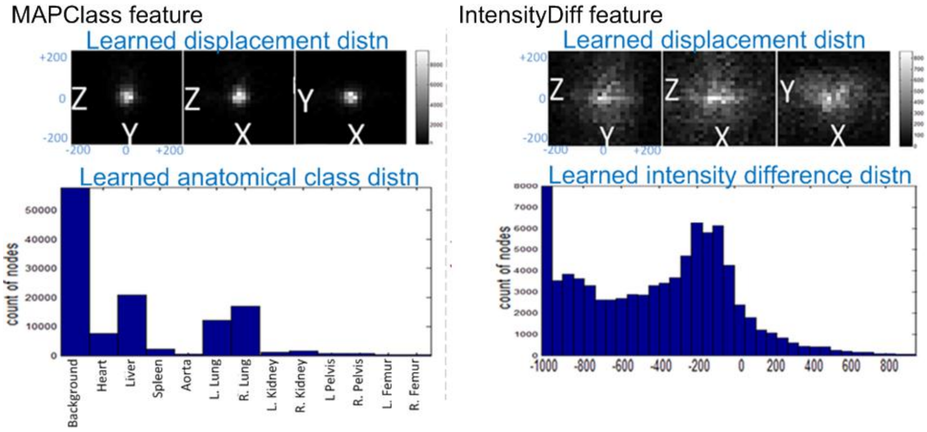


Fig. 4. Learned parameter distributions are clearly non-uniform. (left) Learned displacement and anatomical class distributions for `MAPClass` feature. (right) Displacement and intensity difference distributions for `IntensityDiff` feature.

$$f_{TopNClasses}(\mathbf{x};\boldsymbol{\Delta}_1,\mathbf{P}_1,N,C)=\begin{cases}C\in\text{top N classes of }p(c;n_{\mathbf{p}_1}) & 1\\ \text{otherwise} & 0\end{cases} \tag{4}$$

NodeDescendant entanglement features. This type of feature tests whether a region near voxel **x** has a particular appearance. The neighboring region is centered at voxel $\mathbf{P}_1$ (Fig. 3a,b). The test is whether the node currently corresponding to $\mathbf{P}_1$ descends from a particular tree node, T_0. If it does, then we know $\mathbf{P}_1$ has satisfied the appearance test ($T_1 \ldots T_k$) above T_0 in the tree in a particular way to arrive at T_0.

AncestorNodePair entanglement features. This type of feature tests whether two regions near voxel **x** have passed similar appearance and semantic tests. The neighboring regions are centered at voxels $\mathbf{P}_1$ and $\mathbf{P}_2$ (Fig 3a). The test is whether the nodes currently corresponding to $\mathbf{P}_1$ and $\mathbf{P}_2$ have their first common ancestor < τ tree levels above the current level (Fig. 3c). The threshold controls the required degree of similarity: the lower τ, the greater the required appearance and context similarity needed to pass the test, because the lower τ, the greater the number of tests, ($T_1 \ldots T_k$), above the common ancestor.

2.3 Feature selection is guided by learned proposal distributions

This section describes the second contribution of our paper. We match the distribution of feature types and their parameters proposed at each tree node during training to the ones that tend to be most useful for training. The decision forest still chooses the winning feature, but each node chooses from features sets that are likely to be useful based on prior experience. The basic idea is to help the classifier explore more of the sweet spot of feature space and hopefully find superior features. Since our features contain several parameters, the joint feature space is too large to search exhaustively. However, only a small subset of feature space tends to be relevant. Therefore, rather than drawing feature parameters from a uniform distribution over parameter space, we draw from a learned distribution. Specifically, we train an initial decision forest, F_{temp}, on our training data and record the distribution of accepted (winning) feature

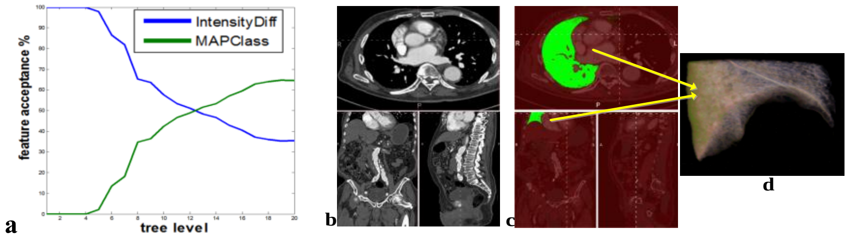


Fig. 5. (a) Example of learned feature type distribution by level. (b-d) **See results section**: (b) right lung is only partially visible in the scan. (c) EDF correctly assigns high posterior probability and (d) segments visible portion (3D rendering)

parameters and feature types across all its tree nodes. F_{temp} is then discarded, and we then use parameter distributions as the proposal distributions in a subsequent training of a final decision forest. While this requires additional training, it imposes no time penalty for prediction.

Parameter proposal distribution. The learned distribution of displacements tends to be Gaussian distributed and centered on the voxel to be labeled (Fig. 4 top row). Acceptance distributions of the remaining parameters also have non-uniform distributions (Fig. 4 bottom row). We draw feature parameters from these distributions during training. This tends to provide the forest with more useful features to choose from at each node and can improve final classifier accuracy.

Feature type proposal distributions. Similarly, the distribution of feature types for each tree level is learned. Drawing feature types from this distribution can also improve classifier accuracy. Fig. 5a shows how the ratio of feature types varies with tree depth if we construct a forest using just `MAPClass` and `IntensityDiff` features. Early in tree growth appearance features dominate, while entanglement features dominate deeper levels. As more information gets inferred from intensity, entanglement features exploit semantic context and neighborhood consistency.

3 Results

3.1 Experimental setup

We evaluate our EDF model on a database which consists of 250 large field of view CT scans in which each voxel has an intensity and is assigned one of 12 labels from a set of very diverse anatomical structures {*heart, liver, spleen, aorta, left/right lung, left/right femur, left/right pelvis, left/right kidney*} or the *background* class label. This database was chosen because it was designed to include wide variations in patient health and scan protocol. We randomly selected 200 volumes for training and 50 for testing.

3.2 Qualitative results

The EDF achieves a visually accurate segmentation of organs throughout the 50 test volumes. Example segmentations are shown in Fig. 6a where the first column is the ground truth organ segmentation, and the second column is the EDF segmentation

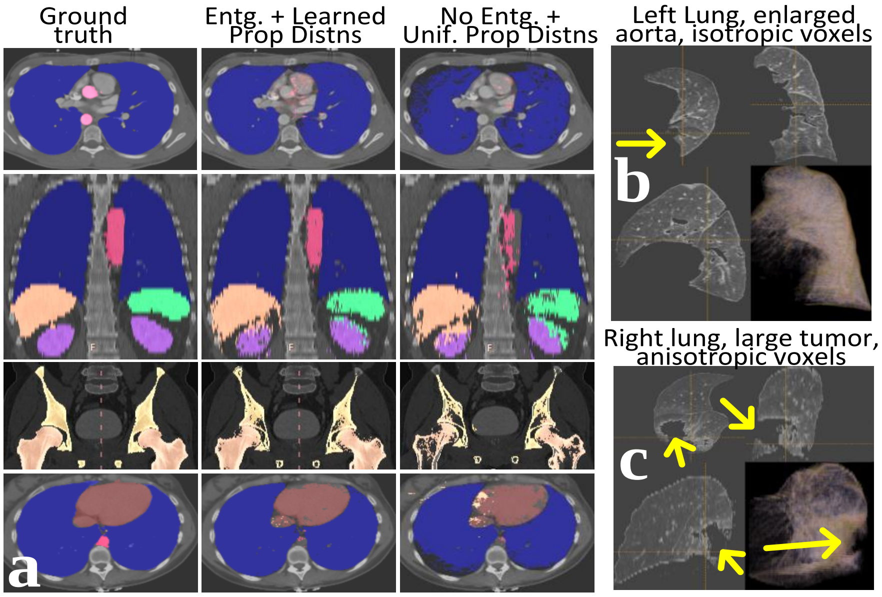

Ground truth
Entg. + Learned Prop Distns
No Entg. + Unif. Prop Distns
Left Lung, enlarged aorta, isotropic voxels
b
Right lung, large tumor, anisotropic voxels
c
a

Fig. 6. Qualitative segmentation results. (a) Entanglement and learned proposal distributions (column 2) provide marked improvement compared to not using them (column 3). Four different subjects shown with axial slices (rows 1,4), and coronal (rows 2,3). (b) 2x2 panel showing intensities of voxels comprising an EDF segmented left lung distorted by enlarged aorta; volume rendering in lower right quadrant (c) EDF accurately segments despite severe anomaly and voxel anisotropy.

result. We see good agreement for the lungs (blue) shown in rows 1, 2 and 4, for the liver (orange), spleen (green), and kidneys (purple) shown in row 2, for the femur bones (tan) in row 3, and for the heart (dark brown) in row 4. Column 3 shows the result using our decision forest but without entanglement features and without the learned proposal distributions. Node entanglement noticeably improve the lungs in row 1, the aorta (red), kidneys, and spleen in row 2, the femurs in row 3 and the lungs and heart in row 4.

The algorithm handles many complexities commonly found in the clinic. Fig. 6b shows how our algorithm correctly segmented the lung (physician verified) despite the fact that the patient had a severely enlarged aorta which caused a distortion (see yellow arrow). Fig. 6c shows how EDF accurately segments despite a large tumor (arrows) and severe anisotropy in the voxel dimensions. Fig. 5b shows a case in which only a portion of the patient's lungs were in the scanner's field of view. EDF correctly assigns high posterior probability to lung pixels (see right lung in Fig. 5c) and properly segments the portion in the scanner.

3.3 Quantitative results including the impact of each of our contributions

Accuracy measure. For a quantitative analysis we measured the EDF segmentation accuracy across all 50 test scans using the average class Jaccard similarity coefficient [16]. The metric is the ratio of the intersection size (ground truth and predicted labels)

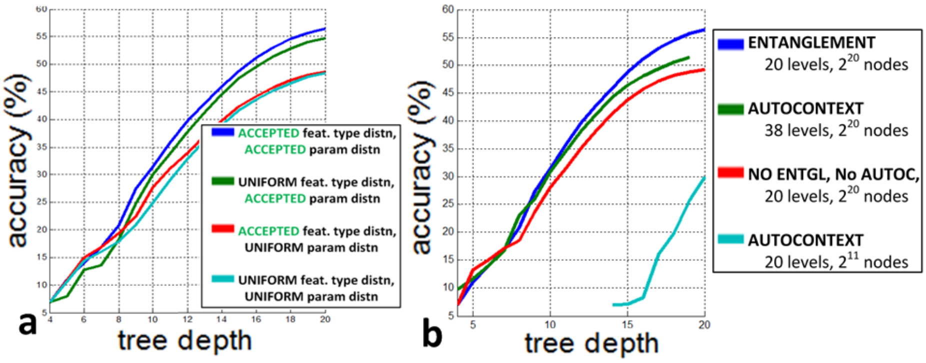


Fig. 7. Quantitative impact of each contribution. (a) Learning both proposal distributions increases accuracy. (b). Entanglement (blue) provides greater accuracy and prediction speed than auto-context (green). Note: green curve should be plotted at depths 20-38, but for comparison we plot it at depths 1-19.

divided by the size of their union. While the EDF achieves >97% average voxel accuracy throughout the volumes in our database, we use the Jaccard metric in this section, because we feel it is a more honest and reliable metric for segmentation accuracy and is not unduly influenced by the background class.

Measuring the impact of learned proposal distributions. To understand the impact of using the acceptance distribution as proposal distributions (section 2.3), we trained the decision forest in four different ways: (1) using uniform feature type and uniform feature parameter distributions for baseline performance (light blue curve, Fig. 7a), (2) using learned (i.e. accepted) feature type distribution with uniform feature parameter distributions (red curve), (3) using uniform feature type distributions with learned feature parameter distributions (green curve), (4) using learned feature type and learned parameters distributions (dark blue curve). Learning only the feature type distribution yields a negligible improvement to baseline (red vs light blue). Learning feature parameter distribution boosts accuracy significantly (green vs red). Learning both distributions yields the best performance without penalty at lower depths (dark blue vs green) and boosts accuracy over baseline by 8% (dark blue vs light blue).

Comparing Entanglement and Auto-context. We compared our method to auto-context [5, 11], a state of the art approach which has yielded some of the best accuracy and speed for multi-structure segmentation. Specifically, we define the same auto-context features as [11] for our decision forest. Auto-context requires multiple complete decision forests to be constructed. The auto-context feature defines semantic context to help classify a voxel at location **x** by examining the class predicted for a probe voxel by a previous decision forest. For our comparison we conducted four experiments. *First,* we trained our decision forest 20 levels deep without entanglement and without auto-context for a baseline performance (red curve, Fig. 7b). *Second*, we trained a two-round, auto-context decision forest (ADF) using 20 total levels (light blue curve). Here we constructed a sequence of two decision forests with the same total number of levels as the baseline classifier, in order to achieve the same prediction *time*. Specifically, we used the output from the first 10 levels of the baseline as the input to the second round, 10 level forest. The second round forest uses the prediction from the first round to form auto-context features and also uses our intensity based features. *Third,* we trained another ADF, but this time with an equal *modeling capacity* to the baseline, (i.e. we trained the same number of tree nodes, requiring roughly the same amount of memory and training time). For this test, we used the final output from the first 19 levels of the baseline classifier as the input to train a second round, 19 level forest, for a total of 38 levels in the ADF. In this way, the ADF consists of $2*2^{19}=2^{20}$ maximum possible nodes. *Fourth*, we trained the proposed EDF method as a single, 20 level deep forest using entanglement (dark blue curve). When the ADF is constrained to give its prediction in the same time as the baseline classifier, it yields much lower accuracy (light blue vs red). When the ADF is allowed more time for prediction using 38 levels, it beats the baseline (green versus red). However, we find *considerably better accuracy* using the EDF method (dark blue curve vs green). In addition to beating the performance of ADF, it reduces the prediction time by 47% since the EDF requires 18 fewer levels (20 vs 38).

In separate tests, we varied the test:train ratio. We found only minor degradation in accuracy. Using 50 images for test and 195 for training, accuracy = 56%; using 75 test and 170 train, accuracy = 56%; using 100 test and 145 train, accuracy = 54%.

Efficiency considerations. With a parallel tree implementation, EDF segments novel volumes in just 12 seconds per volume (a typical volume is 512x512x424) using a standard Intel Xeon 2.4GHz computer (8 core) with 16GB RAM running Win7 x64. A very good, coarse labeling (at 8x downsampling) can be achieved in <1 second. Training on the 200 volumes, which need only be done once, requires about 8 hours.

4 Discussion

Practical impact. To the best of our knowledge, EDF segments volumetric CT at a speed equal to or better than state of the art methods. For example nonrigid marginal space learning (MSL) [13] can segment the outer liver surface in 10 seconds; EDF simultaneously segments 12 organs, including the liver, in 12 seconds.

Our existing implementation of the EDF could be used to automatically measure organ properties (e.g. volume, mean density). It could also be used to initialize interactive segmentation methods [17, 18] or to identify biologically homologous structures that guide non-rigid registration [19].

The EDF is a reusable algorithm. Applying it to segment abdominal-thoracic organs requires no specialization for a particular organ, nor any image alignment; we only assume the patient is supine. Applying it to CT merely requires normalized intensities (i.e. Hounsfield units). This suggests that EDF could be used to segment other organs, or to segment other modalities. Our formulations of node entanglement and the learning of proposal distribution are generic. These methods amplify the value of many hand-crafted, image-based features that have been defined in the literature for specific classification problems. EDF could be directly used to improve the results of other applications [5,6,8,7,] or combined with complementary methods [21] to improve CT image segmentation using decision forests.

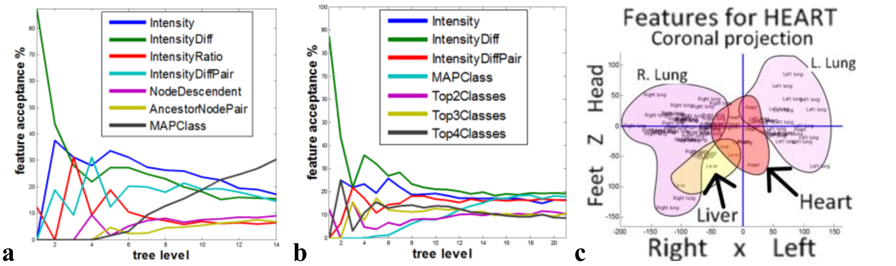


Fig. 8 EDF reveals how and what it learns. (a, b) relative importance of feature types at each level of forest growth. (c) Location and organ class of the top 50 features used to identify heart voxels. The hand-drawn regions here group these locations for different `MAPClass` classes *C*.

Theoretical impact. Compared to black-box learning methods (e.g. neural networks), one can query the EDF to understand what it has learned. For example in our EDF experiments, we queried the EDF to reveals what features it is using to learn at each level of growth. Our tests show that `NodeDescendant` entanglement (purple) achieves peak utilization before `AncestorNodePair` entanglement (tan) shown in Fig. 8a, while `MAPClass` (black) enjoys an ever increasing utilization rate with increasing depth. When we compared `MAPClass` to `TopNClasses` (Fig 8b) we found that `Top4Classes` (black) peaks, then `Top3Classes` (tan), and finally `MAPClass` peaks (light blue).

The EDF can also reveal the anatomical context that it has learned for each structure. By rendering a *scatter plot* of the top contributing features for a target structure, we can visualize the contextual information learned for that structure. For example, Fig. 8c shows how the `MAPClass` feature learns to segment a heart voxel, located at the blue cross-hair intersection. To find the top contributing features, we express information gain (5) as a sum of the information gain from each class:

$$G(F,A,B)=\sum_{c}\left(-p(F_c)\log p(F_c)-\left[-\frac{\sum A_{c_1}}{\sum F_{c_1}}p(A_c)\log p(A_c)-\frac{\sum B_{c_2}}{\sum F_{c_2}}p(B_c)\log p(B_c)\right]\right) \quad (5)$$

where *F* is the set of voxels being split into partitions *A* and *B,* and *c* is the index over classes. This enables us to rank learned node features based on how much they contributed to identifying the voxels of a given class by increasing the information gain for that class. Fig 8c shows a projection of the 3D scatter plot onto a coronal plane. The semantic context that favors classifying a voxel as heart includes other heart pixels nearby (red region), lungs to the right and left (purple regions), and liver below the right lung (yellow region). All of this is learned by the EDF automatically.

5 Conclusions

This paper has proposed the entangled decision forest (EDF) as a new discriminative classifier which achieves higher prediction accuracy and shortened decision time. Our first contribution is to entangle the tests applied at each tree node with other nodes in the forest. This propagates knowledge from one part of the forest to another which speeds learning, improves classifier generalization and captures long range-semantic context. Our second contribution is to inject randomness in a guided way through the random selection of feature types and parameters drawn from learned distributions. Our contributions are an intrinsic improvement to the underlying classifier methodology and augment features defined in the literature.

We demonstrated EDF effectiveness on the very challenging task of simultaneously segmenting 12 organs in large field of view CT scans. The EDF achieves accurate voxel-level segmentation in 12 seconds per volume. The method handles large population variation and protocol variations. We suggest the method may be useful in other body regions and modalities.

References

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

[2] Breiman, L.: Random Forests. *Machine Learning*, 45(1):5–32, 2001.

[3] Menze, B.H., Kelm, B.M., Masuch, R., Himmelreich, U., Petrich, W., Hamprecht, F.A.: A comparison of random forest and its Gini importance with standard chemometric methods for the feature selection and classification of spectral data. *BMC Bioinformatics* 10, 213, 2009.

[4] Andres, B., Kothe, U., Helmstaedter, M., Denk, W., Hamprecht, F.A.: Segmentation of SBFSEM volume data of neural tissue by hierarchical classification. In: *DAGM-Symposium*. 142–152, 2008.

[5] Shotton, J., Johnson, M., and Cipolla, R.: Semantic texton forests for image categorization and segmentation. In *Proc. of CVPR*, 2008.

[6] Yi, Z., Criminisi, A., Shotton, J., Blake, A.: Discriminative, semantic segmentation of brain tissue in MR images. In: *Proc. of MICCAI*, 558–565, 2009.

[7] Lempitsky, V.S., Verhoek, M., Noble, J.A., Blake, A.: Random forest classification for automatic delineation of myocardium in real-time 3D echocardiography. In: *FIMH*, 447–456, 2009.

[8] Geremia, E., Menze, B., Claz, O., Konukoglu, E., Criminisi, and Ayache, N.: Spatial Decision Forests for MS Lesion Segmentation in Multi-Channel MR Images, In *Proc. of MICCAI*, 2010.

[9] Hastie, T., Tibshirani, R., Friedman, J.H.: The Elements of Statistical Learning, Springer, 2009.

[10] Shotton, J., Winn, J.M., Rother, C., Criminisi, A.: Textonboost for image understanding: Multi-class object recognition and segmentation by jointly modeling texture, layout, and context. *Int. J. Comp. Vision* 81(1), 2–23, 2009.

[11] Tu, Z., and Bai, X.: Auto-context and Its Application to High-Level Vision Tasks and 3D Brain Image Segmentation. *PAMI*, 2009.

[12] Tu, Z.: Probabilistic boosting tree: Learning discriminative models for classification, recognition, and clustering. In *Proc. of ICCV*, 1589–1596, 2005.

[13] Zheng, Y., Georgescu, B., Comaniciu, D.: Marginal Space Learning for Efficient Detection of 2D/3D Anatomical Structures in Medical Images, In *IPMI*, Williamsburg, VA, 2009

[14] Geurts, P., Ernst, D., Wehenkel, L.: Extremely randomized trees. *Machine Learning*, 36(1):3-42, 2006.

[15] Viola, P., Jones, M.J.: Robust Real-Time Face Detection. *Int. J. Comp. Vision*, 57(2):137–154, 2004.

[16] Everingham, M., Van Gool, L., Williams, C. K. I., Winn, J. and Zisserman, A. *Int. J. Comp. Vision*, 88(2), 303-338, 2010.

[17] Rother, C., Kolmogorov, V., and Blake, A.: GrabCut -Interactive Foreground Extraction using Iterated Graph Cuts. In *SIGGRAPH*, August 2004

[18] Criminisi, A., Sharp, T., and Blake, A.: GeoS: Geodesic Image Segmentation, In *Proc.of ECCV*, Springer, 2008.

[19] Konukoglu, E., Criminisi, A., Pathak, S., Robertson, D., White, S., and Siddiqui, K.: Robust Linear Registration of CT Images using Random Regression Forests, In *SPIE Medical Imaging*, February 2011.

[20] Criminisi, A., Shotton, J., Bucciarelli, S.: Decision forests with long-range spatial context for organ localization in CT volumes. In: *Proc. of MICCAI-PMMIA*, 2009.

[21] Iglesias, J., Konukoglu, E., Montillo, A., Tu, Z., Criminisi, A.: Combining Generative & Discriminative Models for Semantic Segmentation of CT Scans via Active Learning. IPMI 2011, accepted.

[22] Criminisi, A., Shotton, J., Robertson, D., Konukoglu, E. Regression Forests for Efficient Anatomy Detection and Localization in CT Scans, MICCAI-MCV Workshop 2010.