Structured Class-Labels in Random Forests for Semantic Image Labelling

Peter Kontschieder*, Samuel Rota Bulò△, Horst Bischof* and Marcello Pelillo△

*Institute for Computer Graphics and Vision
Graz University of Technology - Austria
{kontschieder,bischof}@icg.tugraz.at

△Dipartimento di Scienze Ambientali,
Informatica e Statistica
Università Ca' Foscari Venezia - Italy
{srotabul,pelillo}@dsi.unive.it

Abstract

In this paper we propose a simple and effective way to integrate structural information in random forests for semantic image labelling. By structural information we refer to the inherently available, topological distribution of object classes in a given image. Different object class labels will not be randomly distributed over an image but usually form coherently labelled regions. In this work we provide a way to incorporate this topological information in the popular random forest framework for performing low-level, unary classification. Our paper has several contributions: First, we show how random forests can be augmented with structured label information. In the second part, we introduce a novel data splitting function that exploits the joint distributions observed in the structured label space for learning typical label transitions between object classes. Finally, we provide two possibilities for integrating the structured output predictions into concise, semantic labellings. In our experiments on the challenging MSRC and CamVid databases, we compare our method to standard random forest and conditional random field classification results.

1. Introduction

The field of visual object classification has received great attention and evolved in a remarkable manner during the past couple of years. Besides major progresses in the development of new image representations, a large variety of novel machine learning algorithms have been developed and applied to problems in the computer vision domain like object detection, classification, tracking, or action recognition. In this work we present a novel classification algorithm based on random forests, customized to the application of semantic image labelling [27, 9], *i.e.* a per-pixel classification of an image.

Using supervised learning algorithms for semantic image labelling typically requires a large amount of densely labelled training data. A label image corresponding to a natural scene exhibits structured information, respecting the topology shown in the scene. For instance, a typical street scene might result in coherently labelled regions of road, car, bicyclist and so on. Structured learning [30] provides ideas to take this form of additional, structural information into account and therefore intuitively fits to the needs of semantic image labelling or segmentation. Considering our example of the street scene, structured learning allows to integrate the actual label topology in the training process, *e.g.* a car should be driving on a road, but not on top of a building. However, exploiting this form of topological structure of the label images directly in machine learning algorithms for computer vision problems is still widely ignored.

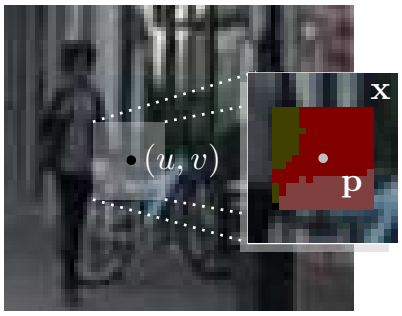


Figure 1. Training data example, as used in our proposed structured learning random forest. While standard random forests associate only the center label at position (u, v) to an image patch $\mathbf{x}$, we incorporate the topology of the local label neighborhood $\mathbf{p}$ and therefore learn valid labelling transitions among adjacent object categories. Here: person, building and bicycle.

For the task of semantic image labelling, state-of-the-art approaches [16, 17, 13, 24] are typically using complementary features at different levels within random field models [18]. Low-level features are mostly calculated on a per-pixel basis and incorporate local color or texture statistics, while mid-level features operate on regions or superpixels to provide shape, continuity or symmetry information. Motivated by perceptual psychology [3], high-level features introduce global image statistics and information about inter-

object or contextual relations, seeking for proper scene configurations on the image level. In such a way, structural information is mostly incorporated on the highest, semantic level. Recently, [31] presented a way to effectively learn a contextual model named Auto-context. A boosting classifier is trained by iteratively learning from appearance and contextual information, collecting typical structures from the images. However, the learning phase is computationally very demanding. Other approaches like boosted random fields [29] or SpatialBoost [2] share both the disadvantage of significant computational complexity when considering contextual beliefs as weak learners. The work in [4] introduces a generalization of a support vector machine (SVM) for structured output regression, used to predict bounding boxes for the task of object localization. Finally, we refer to [23], which gives a comprehensive tutorial on structured learning and prediction in computer vision.

In this paper, we provide a simple but effective way to incorporate structural information in the popular random forest [8, 1, 12] learning algorithm which is considered to be competitive to other state-of-the-art learning techniques like boosting or SVMs. Inspired by ideas of structured learning we provide a novel way to incorporate joint statistics about the local label neighborhood in the random forest framework for learning typical labelling transitions among object class categories, as illustrated in Figure 1. In contrast to standard classification, which can only deal with a single (atomic) label per training sample during the training process, we take structured labelling information of the label neighborhood into account. Including this information at the classification level drastically improves the results and simultaneously counteracts the assignment of meaningless label configurations, as experienced when using standard random forests. Our proposed method is easy to implement and we show superior results on all our conducted experiments for the task of semantic image labelling on the challenging CamVid [9] and MSRCv2 [27] databases in comparison to standard random forests. To sum up, using our proposed structured learning method possesses several advantages when used for semantic image labelling:

- Including the label topology in the training stage yields a classification stage, that respects the label configurations observed during training
- Using structured label information in the classification avoids assigning implausible label transitions

The major drawback of our method is the need for densely labelled training data. However, this problem is shared with state-of-the-art image labelling algorithms and the results of our experiments on the MSRCv2 database indicate that also non-completely labelled training data are well handled by our method.

2. Randomized Decision Forests

We start by providing a brief review of the randomized decision forests [1, 12] and introducing some notations which will be used in the subsequent sections. Randomized decision forests exhibit several appealing properties: They are extremely fast for training and classification, can be easily parallelized [25], are inherently multi-class capable, tend not to overfit and are robust to label noise [8].

A (binary) *decision tree* is a tree-structured classifier which makes a prediction by routing a feature sample $\mathbf{x} \in \mathcal{X}$ through the tree to a leaf, where the actual classification is taking place. A *leaf* $\textsc{Lf}(\pi) \in \mathbb{T}$ is the simplest form of a decision tree and is able to cast a class prediction $\pi \in \mathcal{Y}$ for any sample it is reached by. In all other cases, a decision tree is a *node* $\textsc{Nd}(\psi, t_l, t_r) \in \mathbb{T}$, which is characterized by a binary test (or split) function $\psi(\mathbf{x}) : \mathcal{X} \to \{0, 1\}$, a left decision sub-tree $t_l \in \mathbb{T}$ and a right decision sub-tree $t_r \in \mathbb{T}$. The role of the test function is to decide whether a sample feature $\mathbf{x}$ reaching the node should be forwarded to its left decision sub-tree t_l if $\psi(\mathbf{x}) = 0$, or to its right decision sub-tree t_r if $\psi(\mathbf{x}) = 1$.

A (binary) *decision forest* is an ensemble $F \subseteq \mathbb{T}$ of (binary) decision trees which makes a prediction about a sample feature by averaging over the single predictions collected from the trees in the ensemble.

Class prediction. A class prediction for a sample $\mathbf{x} \in \mathcal{X}$ can be obtained from a tree $t \in \mathbb{T}$ by recursively branching the sample down the tree until a leaf is reached. Formally, we write the tree prediction function $h(\mathbf{x} \,|\, t) : \mathcal{X} \to \mathcal{Y}$ for a decision tree $t \in \mathbb{T}$ recursively as

$$h(\mathbf{x} \,|\, \textsc{Nd}(\psi, t_l, t_r)) = \begin{cases} h(\mathbf{x} \,|\, t_l) & \text{if } \psi(\mathbf{x}) = 0\,, \\ h(\mathbf{x} \,|\, t_r) & \text{if } \psi(\mathbf{x}) = 1\,, \end{cases}$$
$$h(\mathbf{x} \,|\, \textsc{Lf}(\pi)) = \pi\,.$$

The class prediction of a sample $\mathbf{x} \in \mathcal{X}$ given a forest F can then be obtained from the individual decision tree predictions as the one receiving the majority of the votes, *i.e.*,

$$y^* = \arg\max_{y \in \mathcal{Y}} \sum_{t \in F} [h(\mathbf{x}|t) = y]\,. \tag{1}$$

where $[Q]$ is the Iverson bracket which gives 1 if proposition Q is true and 0 otherwise. Combining the outputs of multiple decision trees into a single classifier supports the ability to generalize and mitigates the risk of overfitting, which may affect single decision trees.

Randomized training. We train the binary decision forest according to the extremely randomized trees algorithm [12]. Each tree in a forest is trained independently on a random subset of the training set $\mathcal{D} \subseteq \mathcal{X} \times \mathcal{Y}$ according to a

recursive learning procedure. If $\mathcal{D}$ is smaller than a minimum size or if the entropy of its class distribution $E(\mathcal{D})$ is below a given threshold, a leaf $\textsc{Lf}(\pi)$ is grown where the class prediction π is set to the most represented class in the training data $\mathcal{D}$, *i.e.*,

$$\pi \in \arg\max_{z \in \mathcal{Y}} \sum_{(\mathbf{x},y)\in\mathcal{D}} [y = z] \,. \tag{2}$$

Otherwise a node $\textsc{Nd}(\psi, t_l, t_r)$ is grown, where ψ is a test function selected from a randomly generated set Ψ, maximizing the expected information gain about the label distribution due to the split $\{\mathcal{D}_l^\psi, \mathcal{D}_r^\psi\}$ of the training data, which has been induced by ψ [21]:

$$\begin{aligned}\psi &= \arg\max_{\psi'\in\Psi} \{E(\mathcal{D}) - E(\mathcal{D}\,;\,\psi')\} = \arg\min_{\psi'\in\Psi} E(\mathcal{D}\,;\,\psi') \\ &= \arg\min_{\psi'\in\Psi} \left\{ \frac{|\mathcal{D}_l^{\psi'}|}{|\mathcal{D}|} E(\mathcal{D}_l^{\psi'}) + \frac{|\mathcal{D}_r^{\psi'}|}{|\mathcal{D}|} E(\mathcal{D}_r^{\psi'}) \right\} .\end{aligned}$$

Finally, the trees t_l and t_r are recursively grown with their respective training data $\mathcal{D}_l^\psi$ and $\mathcal{D}_r^\psi$.

In case of unbalanced training data among the different classes to be learned, the tree classifiers can be trained by weighting each label $z \in \mathcal{Y}$ according to the inverse class frequencies observed in the training data $\mathcal{D}$, *i.e.*, $\omega_z = \left(\sum_{(\mathbf{x},y)\in\mathcal{D}} [y = z]\right)^{-1}$. The weights are also considered in the computation of the expected (weighted) information gain, which determines the selection of the best test function during the training procedure. This allows to reduce the class average prediction error.

3. Random Forests in Computer Vision

Recently, random forests were customized for a large variety of tasks in computer vision [5, 11, 20, 19, 21, 22]. Typically, computer vision applications have used random forests for classification tasks in the image domain, where the feature space is anchored to a pixel grid topology. They are trained on a specific feature space $\mathcal{X}$, which consists of a set of $d \times d$ patches extracted from a set of multi-channel images $\mathcal{I}$, where channels may include color features such as gradients, filter banks, *etc*.

More formally, a multi-channel training image is a 3-dimensional matrix I and $I_{(u,v,c)}$ denotes the value at pixel (u, v) and channel c in the image. A patch is simply a triplet $(u, v, I) \in \mathcal{X}$, representing the coordinates (u, v) of the patch center in image $I \in \mathcal{I}$. The label space $\mathcal{Y} = \{1, \ldots, k\}$ is given by the set of k object classes we are going to find in the images.

Different types of test functions for a patch $\mathbf{x} = (u, v, I) \in \mathcal{X}$ have been investigated for the classification task. The following are the most commonly used ones:

$$\begin{aligned}\psi^{(1)}(\mathbf{x}\,|\,\theta_1, \tau) &= [I_{(u,v,0)+\theta_1} > \tau] \,, \\ \psi^{(2)}(\mathbf{x}\,|\,\theta_1, \theta_2, \tau) &= [I_{(u,v,0)+\theta_1} - I_{(u,v,0)+\theta_2} > \tau] \,, \\ \psi^{(3)}(\mathbf{x}\,|\,\theta_1, \theta_2, \tau) &= [I_{(u,v,0)+\theta_1} + I_{(u,v,0)+\theta_2} > \tau] \,, \\ \psi^{(4)}(\mathbf{x}\,|\,\theta_1, \theta_2, \tau) &= \left[\left|I_{(u,v,0)+\theta_1} - I_{(u,v,0)+\theta_2}\right| > \tau\right] \,,\end{aligned}$$

where $\theta_i = (\delta u_i, \delta v_i, c_i)$, $i = 1, 2$, are displacement parameters relative to the patch center used to index a point in the patch, and $\tau \in \mathbb{R}$ is a threshold. Note that test functions of random type and with randomly generated parameters are drawn during the training procedure to form the sets Ψ of split functions in each node of the decision trees.

Once a random forest F has been trained, the classification of a test image can be naively obtained by labelling each pixel with the most probable class predicted by the forest, centered on the $d \times d$ patch.

4. Structured Learning in Random Forests

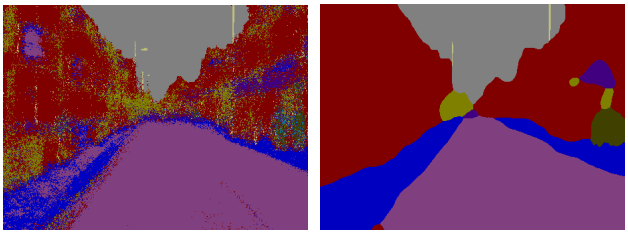
In traditional classification approaches like the one presented in the previous section, input data samples are assigned to single, *atomic* class labels, acting as arbitrary identifiers without any dependencies among them. For many computer vision problems however, this model is limited because the label space of a classification task does exhibit an inherently topological structure, which renders the class labels explicitly interdependent. Although this structured label space is already present in the training data, it remains largely unexploited by standard classification approaches, like the random forests introduced in the previous sections. Consequently, when applying standard random forest classifiers for semantic image labelling, the obtained results are quite noisy (*e.g.*, see Figure 2(c)). Indeed, a random patch extracted from the labelled image will likely show a configuration which never appeared in the ground-truth classification used to train the classifiers.

To overcome this limitation, we propose a novel way of enriching the standard random forest classifiers by rendering them aware of the local topological structure of the output label space. Towards this end, we depart from the traditional classification paradigm and address the problem from a structured learning perspective [30] within the random forest framework.

4.1. Structured Label Space

Our structured label space $\mathcal{P}$ consists of $d' \times d'$ patches of object class labels, *i.e.*, $\mathcal{P} = \mathcal{Y}^{d'\times d'}$. With $p_{ij} \in \mathcal{Y}$ we denote the ij-entry of the label patch $\mathbf{p}$. Additionally, we index the entries in a way that index $(0, 0)$ takes the central position. To distinguish between a patch $\mathbf{x}$ from the feature space $\mathcal{X}$ (see, Section 3) and a patch $\mathbf{p}$ from the

(a) Original (b) Ground truth



(c) Random Forest (d) Our method

Figure 2. Examples of object class segmentations using unary classifiers. Best viewed in color.

structured label space $\mathcal{P}$, we refer to them as *feature patch* and *label patch*, respectively. Each training feature patch $\mathbf{x} = (u, v, I)$ has an associated label patch $\mathbf{p}$ which holds the labels of all pixels of image I within a $d' \times d'$ neighborhood of (u, v) (see, Figure 1). In other words, p_{ij} is the label of pixel $(u + i, v + j)$ in image I. Please note that the size d' of the label patch may differ from the size d of the feature patch.

In the next subsection we show how the split function selection strategy in the nodes of the random forest will be adapted to account for the new label space. However, for the moment we will simply assume that the training patches from $\mathcal{D} \subseteq \mathcal{X} \times \mathcal{P}$ have been routed through the tree to the leafs. Consider now a leaf t and let $\mathcal{P}_t \subseteq \mathcal{P}$ be the set of label patches present in the training data used to grow the leaf (see Figure 8). Then, the class label $\boldsymbol{\pi}$ parametrizing the leaf is now a structured label of size $d' \times d'$ from $\mathcal{P}$ and not just an atomic label from $\mathcal{Y}$ as in the standard random forest. A good selection for the structured class label should represent a mode of the joint distribution of the label patches in $\mathcal{P}_t$.

In order to keep the complexity of this step low, we compute the joint probability by making a pixel independence assumption as

$$\Pr(\mathbf{p}|\mathcal{P}_t) = \prod_{i,j} \Pr^{(i,j)}(p_{ij}|\mathcal{P}_t)\,, \tag{3}$$

where $\Pr^{(i,j)}(c|\mathcal{P}_t)$ represents the marginal class distribution over all the label patches of pixel position (i, j). The label patch $\boldsymbol{\pi}$ is finally selected for leaf t as the one in $\mathcal{P}_t$ maximizing the joint probability:

$$\boldsymbol{\pi} = \arg\max_{\mathbf{p} \in \mathcal{P}_t} \Pr(\mathbf{p}|\mathcal{P}_t)\,. \tag{4}$$

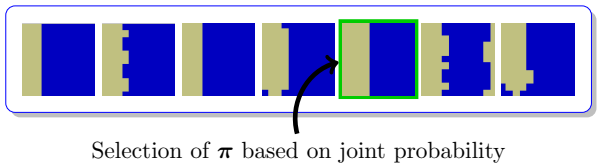


Figure 3. Example of label patches reaching a leaf during training. Based on the joint probability distribution of labels in the leaf a label patch $\boldsymbol{\pi}$ is selected.

4.2. Test Function Selection for Structured Labels

The change introduced in the label space should be coupled with an adaptation of the way a test function is selected in each node of the random forest during the training procedure in order to account for the additional information carried by the structured labels. One naive solution is to port the test selection criterion actually used in the standard random forest to our context, *e.g.*, by simply associating each patch with the label we find in the center of the associated label patch $\mathbf{p}$. This, however, results in a split of the training set which is identical to what the standard random forest implementation does, without thus properly exploiting the label topology.

In order to take advantage of the new label space, we propose to select the best split function at each node based on the information gain with respect to a two-label joint distribution. Specifically, we associate each training pair $(\mathbf{x}, \mathbf{p})$ with two labels: One label is provided by the patch central pixel label p_{00}, whereas the second one is given by p_{ij}, where (i, j) is a patch label position which has been uniformly drawn (once per node). By adopting this new test function selection criterion, all entries of a label patch have the chance to actively influence the way a feature patch is branched through the tree during the training procedure.

One drawback of this new test selection method is the increased complexity deriving from the evaluation of the 2-label joint distribution ($|\mathcal{Y}|^2$ elements) instead of the simple, single label distribution ($|\mathcal{Y}|$ elements). To overcome this we consider also a different test function selection method, which consists in associating each training pair $(\mathbf{x}, \mathbf{p})$ with just one label, but instead of considering label p_{00} of the central pixel, we consider a label p_{ij} from a random position (i, j), which is generated once per node. By so doing, we still have the effect that all entries of the label patch may influence the learning procedure, but at no higher computational cost.

4.3. Structured Label Predictions

The structured predictions gathered from the trees of a forest have to be combined into a single label patch prediction. To this end, we follow a procedure which is similar to the one adopted in order to select the label patch $\boldsymbol{\pi}$ in a leaf (see Section 4.1).

Consider a trained forest F, a test patch $\mathbf{x} = (u, v, I)$. and let $\mathcal{P}_F$ be the set of predictions for $\mathbf{x}$ gathered from each tree $t \in F$:

$$\mathcal{P}_F = \{h(\mathbf{x}|t) \in \mathcal{P} \,:\, t \in F\}\,. \tag{5}$$

Similarly to (4), the label patch prediction of the forest F for feature patch $\mathbf{x}$ is given by the one maximizing the patch label joint probability estimated from $\mathcal{P}_F$, *i.e.*,

$$\mathbf{p}^* = \arg\max_{\mathbf{p}\in\mathcal{P}_F} \Pr(\mathbf{p}\,|\,\mathcal{P}_F)\,, \tag{6}$$

where $\Pr(\mathbf{p}|\mathcal{P}_F)$ is defined as in (3).

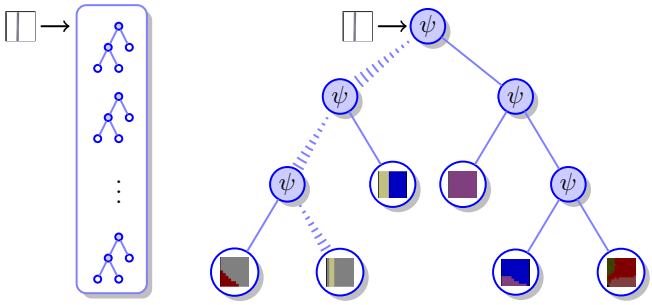


Figure 4. Prediction of the structured label of a feature patch in a random forest. The feature patch is routed through each tree in the forest according to the test functions ψ in the tree nodes until a leaf is reached, holding the learned label transitions between color-coded classes. The structured label in the leaf is then assigned to the feature patch. Best viewed in color.

4.4. Simple Fusion of Structured Predictions

As opposed to standard classification algorithms which, given a test image I, directly assign an object class label to a each pixel, our classifiers cast a prediction for each pixel, involving also the neighboring ones. Indeed, if $\mathbf{p} \in \mathcal{P}$ is the patch label predicted for pixel (u, v) in a test image then a neighbor $(u + i, v + j)$ in a $d' \times d'$ neighborhood of (u, v) could be classified as $p_{ij} \in \mathcal{Y}$. Hence, for each test pixel we collect $d' \times d'$ class predictions, which have to be integrated into a single class prediction. A simple way of performing this operation consists in selecting the most voted class per pixel. This process is illustrated in Figure 5.

The outcome of this fusion step is a labelling $\boldsymbol{\ell}$ from the set $\mathcal{L}$ of all possible labellings for the image, $\ell_{uv} \in \mathcal{Y}$ being the class label associated with pixel (u, v).

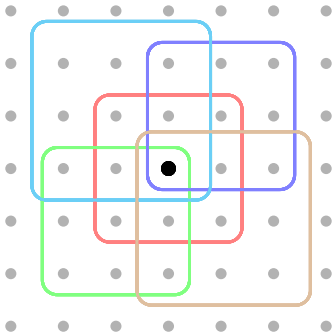


Figure 5. Fusion of structured predictions. Each pixel collects class hypotheses from the structured labels predicted for itself and neighboring pixels, which have to be fused into a single class prediction. For clarity reasons, only 5/9 label patches are drawn. Best viewed in color.

4.5. Optimizing the Label Patch Selection

A different and more principled approach to the computation of the final labelling can be obtained by optimizing the label patch selection with respect to a given labelling rather than solely taking (6) for each pixel. This allows to better exploit the label patch diversity in the set of predictions $\mathcal{P}_F$ obtained from (5).

We define the *agreement* of an individual label patch $\mathbf{p}$ located at $(i, j) \in I$ with a given labelling $\ell \in \mathcal{L}$ as the number of corresponding pixels sharing the same label, *i.e.*

$$\phi^{(i,j)}(\mathbf{p}, \boldsymbol{\ell}) = \sum_{(u,v)\in I} \left[p_{(u-i)(v-j)} = \ell_{uv}\right]. \tag{7}$$

Furthermore, let $\mathbf{z} \in \mathcal{Z}_I$ be an assignment of label patches to pixels in I, $z_{uv} \in \mathcal{P}_F$ being a label patch for pixel (u, v) taken from (5), where $\mathcal{Z}_I$ denotes the set of all such assignments for image I. Then, for a particular configuration $\mathbf{z} \in \mathcal{Z}_I$ and a labelling $\boldsymbol{\ell} \in \mathcal{L}$, the total agreement $\Phi(\mathbf{z}, \boldsymbol{\ell})$ is defined as the sum of agreements of each label patch in $\mathbf{z}$ with the labelling $\boldsymbol{\ell}$ according to

$$\Phi(\mathbf{z}, \boldsymbol{\ell}) = \sum_{(u,v)\in I} \phi^{(u,v)}(z_{uv}, \boldsymbol{\ell})\,. \tag{8}$$

As we want to find the label patch configuration that leads to the maximum total agreement with the labelling of a test image I, we can write the optimal solution as a pair $(\mathbf{z}^*, \boldsymbol{\ell}^*) \in \mathcal{Z}_I \times \mathcal{L}$, where

$$(\mathbf{z}^*, \boldsymbol{\ell}^*) \in \arg\max_{(\mathbf{z},\boldsymbol{\ell})} \{\Phi(\mathbf{z}, \boldsymbol{\ell}) \,|\, (\mathbf{z}, \boldsymbol{\ell}) \in \mathcal{Z}_I \times \mathcal{L}\}\,. \tag{9}$$

To solve (9), we use a simple, iterative optimization method that alternates between selecting the best agreeing label patch per pixel and producing a new labelling as described in Section 4.4. For a more detailed description, we refer the interested reader to our recent work [15].

5. Experiments

In this section we evaluate our proposed structured learning random forest algorithm on the challenging CamVid [9] and MSRCv2 [27] databases for the task of semantic image labelling. For performance reasons, we implemented our method in C++ and ran all experiments on a standard desktop computer with 2.9 GHz and 2 GB RAM.

In all our experiments we show a comparison to a standard random forest implementation (denoted as 'Our Baseline RF'), which is actually a special instance of our method with a label patch size of 1×1 and a fixed label center position. Where available, we list results of state-of-the-art methods [27, 9, 14] that are also using random forests (but not the same features), in order to show that our baseline random forest implementation achieves state-of-the-art performance. Additionally, we compare to the results obtained when minimizing the energy term of a pairwise, conditional random field (CRF) model with graph cuts, using the publicly available GCO [7] implementation[1]. To this end, we provide the class label statistics of the baseline random forest as unary or data terms and use the standard, contrast-sensitive Potts model as suggested in [6] for the pairwise or smoothness term.

To show the impact of the respective stages of our method, we evaluate different training ['Structure' / 'Full'] and classification ['Simple Fusion' / 'Optimized Selection'] procedures as follows: 'Structure' considers the structured label patches but only takes one random label position, *i.e.* a single label distribution into account for training. 'Full' considers structured label patches and the two-label joint distribution in the split functions (see Section 4.2 for more details). 'Simple Fusion' and 'Optimized Selection' refer to the fusion methods of the structured output predictions as described in Sections 4.4 and 4.5, respectively.

We used the same, primitive low-level features for training both, our baseline and our novel structured learning random forests, since our primary intention is to show the improvement when the label topology is taken into account: CIELab raw channel intensities, first and second order derivatives as well as HOG-like features, computed on the L-Channel. In all experiments we fixed the feature patch size to 24×24 and trained 10 trees, using 500 iterations for the node tests and stopping when less then 5 samples per leaf were available.

We list the scores of our experiments according to the same evaluation criteria as used in [27, 9, 14] and additionally include the more strict average intersection vs. union score as *e.g.* used in the PASCAL VOC challenges [10]. In particular, '*Global*' refers to the percentage of all pixels that were correctly classified, '*Avg(Class)*'[2] expresses the average recall over all classes and '*Avg(Pascal)*'[3] denotes the average intersection vs. union score.

Method	*Global*	*Avg(Class)*	*Avg(Pascal)*
RF using Motion and Structure cues [9]	61.8	43.6	-
Our Baseline RF	69.9	42.2	30.6
Our Baseline RF + CRF	74.5	45.4	33.8
Our method (Structure + Simple Fusion)	74.8	45.0	34.1
Our method (Full + Simple Fusion)	76.8	46.1	35.4
Our method (Full + Optimized Selection)	79.2	46.0	36.2

Table 1. Classification results on CamVid database for label patch size 13×13.

5.1. CamVid Database Experiments

The Cambridge-driving Labeled Video Database (CamVid) [9] is a collection of videos captured on road driving scenes. It consists of more than 10 minutes of high quality (970×720), 30 Hz footage and is divided into four sequences. Three sequences were taken during daylight and one at dusk. A subset of 711 images is almost entirely annotated into 32 categories, but we used only the 11 commonly used categories with the same splits for training and testing as presented in [9, 28].

We resized the training images by a factor of 0.5 and randomly collected training samples on a regular lattice with a stride of 10, resulting in approximately 850k training samples. The training time per tree is 23 minutes when using the single label test and 30 minutes with the joint label test. For the experiment where we only consider the labelling transitions, we reduced the stride to 8. In order to correct the imbalance among samples of different classes, we applied an inverse frequency weighting as mentioned in Section 2.

CamVid - 11 Classes. The standard protocol for evaluating on the CamVid database considers the following 11 object categories, forming a majority of the overall labelled pixels (89.16%): ROAD, BUILDING, SKY, TREE, SIDEWALK, CAR, COLUMN_POLE, SIGN-SYMBOL, FENCE, PEDESTRIAN and BICYCLIST. In Table 1 we list our results using a label patch size of 13×13, clearly indicating the performance boost when using our proposed structured learning method over the standard random forest. We can achieve comparable results to the CRF implementation with the Simple Fusion approach and significantly increase the scores using the Optimized Selection. We explain this by the fact that our method is restricted to pick from a candidate set of semantically plausible label patches provided by the trees, rather than propagating arbitray label configurations in the associated graph.

In Figure 6 we show the influence of the label patch size,

[1] http://vision.csd.uwo.ca/code/

[2] $\frac{\text{True Positives}}{\text{True Positives + False Negatives}}$

[3] $\frac{\text{True Positives}}{\text{True Positives + False Negatives + False Positives}}$

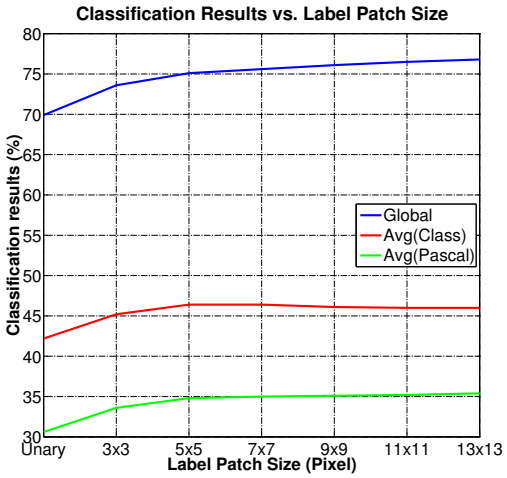


Figure 6. Classification results on CamVid database as a function of the label patch size using Simple Fusion.

Method	*Global*	*Avg(Class)*	*Avg(Pascal)*
Our Baseline RF	63.8	44.2	29.8
Our Baseline RF + CRF	68.2	48.2	33.3
Our method (Structure + Simple Fusion)	69.9	50.4	35.0
Our method (Full + Simple Fusion)	71.6	50.1	35.8
Our method (Full + Optimized Selection)	72.5	51.4	36.4

Table 2. Classification results for labelling transitions on the CamVid database for label patch size 11×11.

i.e. the size of the considered label topology during training and classification using the configuration 'Full + Simple Fusion'. It is clearly shown that even a small neighborhood ($\geq 5 \times 5$) leads to a significant boost in the classification stage.

Labelling Transition Evaluation. In this experiment we evaluate only the transitions between object classes to demonstrate the impact of structured predictions on the label border classification results. To perform this experiment, we discarded all labels in the ground truth information when they were outside a radius of 24 pixels to a transition between two or more classes. This results in a drop to 41.9% of the original amount of labelled pixels. In Table 2, the corresponding results are listed when using a label neighborhood of 11×11. Although the global score has slightly dropped compared to the previous experiment, we obtain improvements on the (stricter) *Avg(Class)* and *Avg(Pascal)* criteria. This strengthens our assumptions that the proposed framework yields to superior results, especially when classifying transitions between object classes.

5.2. MSRCv2 Database Experiments

To show that our method also yields to an improvement when the images are not entirely labelled, we performed another experiment on the MSRCv2 Database [27].

Method	*Global*	*Avg(Class)*	*Avg(Pascal)*
Texton forests naïve (supervised) [26]	49.7	34.5	-
RF using covariance features [14]	55.8	42.2	-
Our Baseline RF	54.8	43.4	28.3
Our Baseline RF + CRF	61.0	52.8	35.1
Our method (Structure + Simple Fusion)	60.8	51.0	33.8
Our method (Full + Simple Fusion)	60.8	51.1	33.9
Our method (Full + Optimized Selection)	63.9	55.6	37.6

Table 3. Classification results on MSRCv2 database for label patch size 11×11.

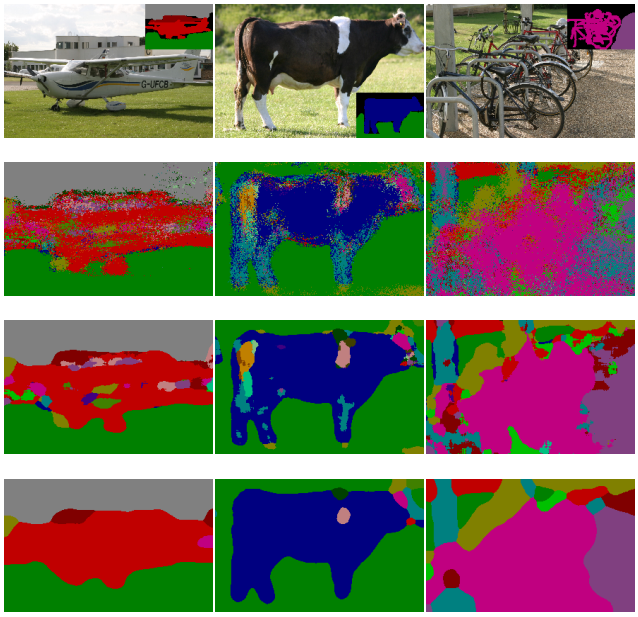


Figure 7. Qualitative labelling results on images of the MSRCv2 database. Top row: Original images with ground truth annotations. Second row: Labelling using our baseline random forest classifier. Third row: Full + Simple Fusion. Last row: Full + Optimized Selection. Best viewed in color.

This database consists of 532 images containing 21 object classes and predefined splits into 276 training and 256 test images. We collected the training samples on a regular lattice with a stride of 5, leading to approximately 500k training samples and training times of 13 and 17 minutes per tree using single or joint label distributions, respectively. In contrast to the almost completely labelled CamVid database, the labellings for MSRCv2 are only available for 71.9% of the pixels, hence more roughly sketching the object classes of interest. In Figure 7 we show some qualitative results and in Table 3, we provide the scores for a label neighborhood size of 11×11 and again find an improvement with our structured learning algorithm. The gain of using the joint statistics over the single label distribution seems to vanish in the Simple Fusion approach, however, we explain this by the fact that our algorithm does not see enough properly labelled transitions between different classes.

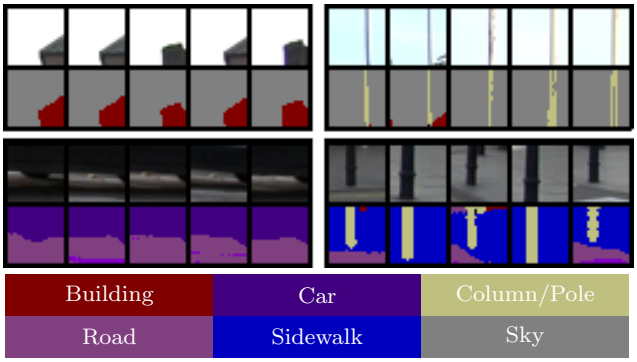


Figure 8. Illustration of feature patches with corresponding label patches, collected from different leaf nodes when trained on CamVid database. Bottom rows: Label sets and associated colors. Best viewed in color.

6. Conclusions

In this paper we presented a simple and effective way to integrate ideas from structured learning into the popular random forest framework for the task of semantic image labelling. In particular, we incorporated the topology of the local label neighborhood in the training process and therefore intuitively learned valid labelling transitions among adjacent object categories. During the tree construction, we used topological joint label statistics of the training data in the node split functions for exploring the structured label space. For classification, we provided two possibilities for fusing the structured label predictions: A simple method using overlapping predictions and a more principled approach, selecting most compatible label patches in the neighborhood. We provided several experiments on the callenging CamVid and MSRCv2 databases and found superior results when compared to standard random forest or conditional random field (using pairwise potentials) classification results. In our future work we will investigate how the output of our classifiers can be used as higher-order potential generator in a CRF.

Acknowledgements. We acknowledge the financial support of the Austrian Science Fund (FWF) from project 'Fibermorph' with number P22261-N22 and the Future and Emerging Technology (FET) Programme within the Seventh Framework Programme for Research of the European Commission, under FET-Open project 'SIMBAD', grant no. 213250.

References

[1] Y. Amit and D. Geman. Shape quantization and recognition with randomized trees. *Neural Computation*, 1997. 2

[2] S. Avidan. Spatialboost: Adding spatial reasoning to adaboost. In *(ECCV)*, 2006. 2

[3] I. Biederman. Perceiving real-world scenes. *Science*, 1972. 1

[4] M. Blaschko and C. Lampert. Learning to localize objects with structured output regression. In *(ECCV)*, 2008. 2

[5] A. Bosch, A. Zisserman, and X. Muñoz. Image classification using random forests and ferns. In *(ICCV)*, 2007. 3

[6] Y. Boykov and M. P. Jolly. Interactive graph cuts for optimal boundary and region segmentation of objects in N-D images. In *(ICCV)*, 2001. 6

[7] Y. Boykov, O. Veksler, and R. Zabih. Fast approximate energy minimization via graph cuts. *(PAMI)*, 2001. 6

[8] L. Breiman. Random forests. In *Machine Learning*, 2001. 2

[9] G. J. Brostow, J. Shotton, J. Fauqueur, and R. Cipolla. Segmentation and recognition using structure from motion point clouds. In *(ECCV)*, 2008. 1, 2, 6

[10] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman. The pascal visual object classes (VOC) challenge. *(IJCV)*, 2010. 6

[11] J. Gall, A. Yao, N. Razavi, L. Van Gool, and V. Lempitsky. Hough forests for object detection, tracking, and action recognition. *(PAMI)*, 2011. 3

[12] P. Geurts, D. Ernst, and L. Wehenkel. Extremely randomized trees. *Machine Learning*, 2006. 2

[13] J. M. Gonfaus, X. Boix, J. van de Weijer, A. D. Bagdanov, J. Serrat, and J. Gonzalez. Harmony potentials for joint classification and segmentation. In *(CVPR)*, 2010. 1

[14] S. Kluckner, T. Mauthner, P. M. Roth, and H. Bischof. Semantic image classification using consistent regions and individual context. In *(BMVC)*, 2009. 6, 7

[15] P. Kontschieder, S. Rota Bulò, M. Donoser, M. Pelillo, and H. Bischof. Semantic image labelling as a label puzzle game. In *(BMVC)*, 2011. 5

[16] L. Ladicky, C. Russell, P. Kohli, and P. Torr. Graph cut based inference with co-occurrence statistics. In *(ECCV)*, 2010. 1

[17] L. Ladicky, P. Sturgess, K. Alahari, C. Russell, and P. Torr. What, where & how many? combining object detectors and CRFs. In *(ECCV)*, 2010. 1

[18] J. Lafferty, A. McCallum, and F. Pereira. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *(ICML)*, 2001. 1

[19] C. Leistner, A. Saffari, and H. Bischof. MIForests: Multiple-instance learning with randomized trees. In *(ECCV)*, 2010. 3

[20] C. Leistner, A. Saffari, J. Santner, and H. Bischof. Semi-supervised random forests. In *(ICCV)*, 2009. 3

[21] V. Lepetit, P. Lagger, and P. Fua. Randomized trees for real-time keypoint recognition. In *(CVPR)*, 2005. 3

[22] F. Moosmann, B. Triggs, and F. Jurie. Fast discriminative visual codebooks using randomized clustering forests. In *(NIPS)*, 2006. 3

[23] S. Nowozin and C. Lampert. Structured learning and prediction in computer vision. In *Foundations and Trends in Computer Graphics and Vision*, 2011. 2

[24] A. Rabinovich, A. Vedaldi, C. Galleguillos, E. Wiewiora, and S. Belongie. Objects in context. In *(ICCV)*, 2007. 1

[25] T. Sharp. Implementing decision trees and forests on a GPU. In *(ECCV)*, 2008. 2

[26] J. Shotton, M. Johnson, and R. Cipolla. Semantic texton forests for image categorization and segmentation. In *(CVPR)*, 2008. 7

[27] J. Shotton, J. Winn, C. Rother, and A. Criminisi. Textonboost: Joint appearance, shape and context modeling for multi-class object recognition and segmentation. In *(ECCV)*, 2006. 1, 2, 6, 7

[28] P. Sturgess, K. Alahari, L. Ladicky, and P. Torr. Combining appearance and structure from motion features for road scene understanding. In *(BMVC)*, 2009. 6

[29] A. Torralba, K. P. Murphy, and W. T. Freeman. Contextual models for object detection using boosted random fields. In *(NIPS)*. 2005. 2

[30] I. Tsochantaridis, T. Hofmann, T. Joachims, and Y. Altun. Support vector learning for interdependent and structured output spaces. In *(ICML)*, 2004. 1, 3

[31] Z. Tu. Auto-context and its application to high-level vision tasks. In *(CVPR)*, 2008. 2