Structured Local Predictors for Image Labelling

Samuel Rota Bulò$^\triangle$, Peter Kontschieder*, Marcello Pelillo$^\triangle$ and Horst Bischof*

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

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

Abstract

In this paper we introduce Structured Local Predictors (SLP) – A new formulation that considers the image labelling problem from a structured learning point of view. SLP are locally operating models, which provide a per-pixel labelling by exploiting contextual relations, learned from complex interactions between labels and a customizable intermediate representation of the image data. Our first key contribution is to handle flexible configurations of pairwise interactions between image pixels while allowing them to be made arbitrarily dependent on the image data. Moreover, we pose the parameter learning process as a convex, structured-learning problem, which can be efficiently solved in a globally optimal way due to the introduction of a continuous, structured output space. Finally, we provide an interface to our model by means of a quantization space, allowing to define task-specific intermediate representations for the input data. In our experiments we demonstrate the broad applicability of our model for tasks like inpainting and semantic labelling.

1. Introduction

In many computer vision problems, random field models are used to perform image labelling tasks, with the goal of predicting labels for each pixel in a given input image. Typical examples are foreground-background segmentation [2, 21], semantic image labelling [22, 7, 10] or depth image estimation [15]. With random field models, the joint (or posterior) distribution of labels and input data can be factorized into products of local interactions. Markov random fields (MRF) provide a posterior label distribution by combining a per-pixel likelihood function with a pairwise consistency potential. Moreover, with the advent of conditional random fields (CRF) [14], more elaborate terms for modelling the smoothness costs and contextual relationships between classes could be made dependent on the input data. Such models are commonly solved by performing maximum a posteriori (MAP) inference.

A simple CRF model is composed of unary and pairwise potentials, modeling local and neighboring relations, respectively. While the unary or *data term* mostly uses classifier outputs to model the local label distribution on a per-pixel basis, the pairwise potential or *smoothing term* enforces adjacent pixels to take on the same labels when indicated. For the task of semantic image labelling, some approaches introduced hierarchical layers or higher-order potentials [11, 12, 8] defined over segments, for improving segmentation results. However, despite the rapid developments made for some computer vision tasks, many CRF models still suffer from substantial limitations: The pairwise potentials are often restricted to have a simple, parametric form and need to obey a certain neighborhood structure. This hinders both, the capability of modelling long range and more direct interactions with the image data.

Recently, a new graphical model named Decision Tree Field (DTF) [19] was introduced to overcome some of the above mentioned shortcomings. DTF combines and generalizes random forests and random fields by using the structure of decision trees for defining interaction variables in the potential functions of the model. Doing so enables to express all potentials in a non-parametric way. For modelling pairwise or higher-order terms over larger and thus more flexible neighborhoods, DTF uses decision trees trained on respective, combinatorial combinations of their label sets. Once the trees are trained, the model parameters can be efficiently learned by maximizing a convex surrogate likelihood function using standard optimization tools. For inference, DTF uses Gibbs sampling in combination with simulated annealing to heuristically approach the MAP solution.

In this work we propose a novel way to tackle the image labelling task from a structured learning [23, 18] perspective, overcoming the aforementioned problems of simple random field models, while providing a principled approach of defining potential functions and their interactions, which differs from [19]. To this end, we introduce *Structured Local Predictors* (SLP), which are functions providing a label prediction for each individual pixel by exploiting contextual information expressed in terms of both, labels of neighboring pixels and local appearance. Intuitively, each pixel in our model can predict the label of neighboring pixels, based on information like relative position, respective labels and local image appearance. The final labelling is delivered by maximizing the agreement between the label assignments and these structured local predictions. One major advantage of the resulting method is that both prediction and training can be cast as convex optimization problems.

Differently from standard literature, our model works on abstract image representations, which are obtained by so-called quantization functions. A *quantization function* is an application-dependent function that maps pixels to quantization classes, which summarize *e.g*. the local appearance, shape, texture, *etc*. of the original image. For instance, random decision trees, which are commonly used classifiers in computer vision can be used as quantization functions by considering their leaves as the quantization classes.

Using a quantization space as an intermediate representation provides several appealing properties: First, it can be seen as a compressed representation of the image data during training and testing. Moreover, it helps to control the number of model parameters that substantially decide whether a model is computationally tractable or not. In addition, the quantization function can be customized to specific problems and may integrate (partial) solutions from several sources, *e.g*. the output of several classifier stages, prior knowledge, *etc*. With the quantization function as interface, we think that many computer vision problems could potentially benefit from using our proposed approach. Please note that the way we use quantization here should not be confused with quantization in terms of image segmentation granularity, *i.e*. pixels or superpixels, and it should also not be considered as a simple means of remapping samples in a new feature space. To sum up, our structured learning method possesses several advantages when used for the task of image labelling:

- We can handle a large number of interactions among variables which are not restricted to obey a fixed neighborhood structure;
- Our approach operates on an intermediate representation (Quantization space) of the image data, allowing to describe non-parametric interactions among the variables in a customized way;
- Our model can be efficiently solved in a globally optimal way for training and prediction using off-the-shelf algorithms from convex optimization.

The rest of our paper is structured as follows. In Section 2 we discuss related work before we introduce the terminology and notation for our approach in Section 3. Section 4 introduces the general concept of structured learning and our definition of the auxiliary output space we are using to efficiently learn the parameters for our novel SLP, described in 5. In Section 6 we explain our approach with the help of several experiments before concluding in Section 7.

2. Related Work

Learning image-dependent potential functions for random field models does not have a long history in computer vision. Instead, as outlined also by [19], most approaches [1, 17] learn class-to-class energy tables in a direct way without explicitly modelling dependence on the image content.

In [9], the authors learned a multi-class logistic regression classifier to model dependencies on features like length and orientation of region boundaries for pairwise terms in a CRF. Recently, [5] proposed a method to adapt image priors according to the underlying low-level, local structures as well as mid-level texture cues for the task of image restoration. The approach in [16] proposes learning of structured prediction models for the task of interactive image labelling. However, despite the similarity in the name to our title, [16] refers only to the structured dependencies among image labels. In particular, they use a CRF where the pairwise potentials are defined by scalar parameters for each joint state of the corresponding labels, independent of the image input.

Our paper is most directly related and inspired by the recent work of Nowozin *et al*. [19]. As already outlined, they provide a way of expressing dependencies between image data and potential functions in a graphical model by using decision trees [3]. However, their work is restricted to model the image dependency in terms of decision trees, as their energy functional depends on the tree structure for the parameter learning step. Moreover, in our work we can use decision trees as a special instance of a quantizer (as we also demonstrate in one of our experiments in Section 6). In this sense our model also generalizes decision trees since in the most simple case, when we do not consider any neighborhood relations with other pixels, our model obtains exactly the single decision tree classification result (whereas using multiple quantizations from several trees straightforwardly generalizes to the result of random forests). Despite the positive properties exhibited by decision trees (scalability *w.r.t.* training data, efficient and parallel training), it is easy to construct examples where the dependency on the image data can be more appropriately expressed using alternative representations (see quantization rule for the snake example in 6). Therefore, we consider providing the quantization space as powerful and flexible interface for interacting with our model with potential impact on many labelling problems in computer vision.

3. Definitions and Notations

We model an *image* as a collection $I = \{(u_i, \phi_i)\}_{i=1}^n$ of n pixels. For simplicity, we use I also to denote the set of pixel indices of image I. Each pixel $i \in I$ has a *position* denoted by u_i and an associated *feature vector* $\phi_i \in \Phi$. The set of images with pixels taking values in a feature space Φ is denoted by $\mathcal{I}^\Phi$. Given two pixels $i, j \in I$, we write Δ_{ij} for the displacement vector between i and j, *i.e.* $\Delta_{ij} = u_i - u_j \in \mathbb{Z}^2$.

A *neighborhood system* for an image I is a function associating each pixel in I with a set of neighbors. We denote by $\mathcal{N}_i$ the set of pixels that are neighbors of pixel i.

Let $L = \{1, \dots, k\}$ be a set of labels. A *labelling* for an image I is a collection $\boldsymbol{\ell} = \{\ell_i\}_{i \in I}$ assigning a label $\ell_i \in L$ to each pixel $i \in I$. We denote by $\mathcal{L}$ the set of labellings.

A *quantization function* or simply *quantizer* is a function $q : \mathcal{I}^\Phi \to \mathcal{I}^\mathcal{Q}$ that provides a compressed representation of an image I by associating each pixel $i \in I$ with an element of the *quantization space* $\mathcal{Q}$, which is a discrete finite set. The elements of $\mathcal{Q}$ are called *quantization classes*. For simplicity, we denote by $q_i \in \mathcal{Q}$ the quantization class associated to pixel $i \in I$ by means of quantizer q.

We denote by $\mathbf{e}$ the column vector of all 1s, and by $\mathbf{e}^k$ the column vector having 1 in the kth position and 0 elsewhere.

Given a proposition P, we denote by 1_P the truth value of proposition P expressed as 1 (true), or 0 (false).

4. Structured Prediction

In structured learning theory, a *structured predictor* is a function $f : \mathcal{X} \to \mathcal{Y}$, mapping elements from an input domain $\mathcal{X}$ to a structured output domain $\mathcal{Y}$ defined as

$$f(x) \in \operatorname*{arg\,min}_{y \in \mathcal{Y}} g(x, y) \,, \tag{1}$$

for some auxiliary function $g : \mathcal{X} \times \mathcal{Y} \to \mathbb{R}$. In many computer vision problems the structured output space is discrete, leading to a combinatorial optimization problem in (1), which is typically intractable to solve. To overcome this limitation, we introduce a different perspective for structured learning that includes the traditional one as a special case, and allows us to define a structured predictor in terms of an auxiliary function, working with an arbitrary output space.

Structured prediction with auxiliary output space. Our alternative definition of a structured predictor requires the specification of an *auxiliary output space*, denoted by $\tilde{\mathcal{Y}}$, and an onto-mapping $\pi_{\tilde{\mathcal{Y}} \to \mathcal{Y}}$ which allows to map points from the auxiliary output space to the original one. Additionally, the latter mapping has to admit a left-inverse function, denoted by $\pi_{\mathcal{Y} \to \tilde{\mathcal{Y}}}$, which is a function such that $\pi_{\mathcal{Y} \to \tilde{\mathcal{Y}}} \circ \pi_{\tilde{\mathcal{Y}} \to \mathcal{Y}}$ is the identity on $\mathcal{Y}$. Given the new ingredients, a structured predictor becomes a function $f : \mathcal{X} \to \mathcal{Y}$ defined as

$$f(x) \in \pi_{\tilde{\mathcal{Y}} \to \mathcal{Y}} \left(\operatorname*{arg\,min}_{\tilde{y} \in \tilde{\mathcal{Y}}} h(x, \tilde{y}) \right) \,, \tag{2}$$

where $h : \mathcal{X} \times \tilde{\mathcal{Y}} \to \mathbb{R}$ is the counterpart of g, working on the new structured output space. Note that in (2), the output of arg min is a set $\tilde{Y} \subseteq \tilde{\mathcal{Y}}$, and we understand $\pi_{\tilde{\mathcal{Y}} \to \mathcal{Y}}(\tilde{Y}) = \{\pi_{\tilde{\mathcal{Y}} \to \mathcal{Y}}(\tilde{y}) : \tilde{y} \in \tilde{Y}\}$.

With this formulation we are free to select a continuous, auxiliary output space and make use of powerful tools from continuous optimization theory for solving the minimization problem in (2). Additionally and as mentioned before, if $\tilde{\mathcal{Y}} = \mathcal{Y}$ and $\pi_{\mathcal{Y} \to \mathcal{Y}}$ is the identity map on $\mathcal{Y}$ then (2) is equivalent to (1), *i.e.* our new formulation in (2) becomes the traditional one in (1).

5. Structured Local Predictors

In this section we introduce a structured prediction function for image labelling, which is defined according to (2). Our model relies on an intermediate image abstraction delivered by a quantization function $q : \mathcal{I}^\Phi \to \mathcal{I}^\mathcal{Q}$, according to which pixels of an image are mapped to quantization classes in $\mathcal{Q}$. The input space of our structured prediction function is thus given by $\mathcal{I}^\mathcal{Q}$, while the output space is given by the set of labellings $\mathcal{L}$. As for the auxiliary output space, denoted by $\tilde{\mathcal{L}}$, we replace the discrete labels in L with $|L|$-dimensional real vectors with components summing up to 1, *i.e.* elements of the set $\tilde{L} = \{z \in \mathbb{R}^{|L|} : \sum_j z_j = 1\}$. A labelling in the auxiliary output space is thus given by $\mathbf{y} = \{y_i\}_{i \in I}$, where $y_i \in \tilde{L}$. The kth-component of y_i is denoted by y_{ik}. Labels from L are mapped to labels in $\tilde{L}$, and vice versa, according to the following functions:

$$\pi_{L \to \tilde{L}}(k) = \mathbf{e}^k \,, \tag{3}$$

$$\pi_{\tilde{L} \to L}(z) \in \operatorname*{arg\,max}_{k} z_k \,. \tag{4}$$

In words, given a discrete label $\ell_i \in L$ we map it to a vector $y_i \in \tilde{L}$ satisfying $y_{ik} = 1_{(k=\ell_i)}$. In the opposite direction, given a label $y_i \in \tilde{L}$ we map it to an index of y_i yielding the maximum value. The functions $\pi_{\mathcal{L} \to \tilde{\mathcal{L}}}$ and $\pi_{\tilde{\mathcal{L}} \to \mathcal{L}}$, linking the original structured output space and the auxiliary one, can be obtained by applying pixel-wise (3) and (4). After defining the mappings between original and auxiliary output label spaces, we are now ready to introduce our new predictor function.

A *structured local predictor* is a function $h_i(I, \mathbf{y}, \Theta)$ providing a label from the auxiliary output space $\tilde{\mathcal{L}}$ for a pixel i of image I by exploiting contextual information in terms of both, labels *and* the quantization classes of neighboring pixels. Our structured prediction function f, defined according to (2), tries to find an agreement between the label prediction derived from h_i and the actual label y_i of pixel i, *i.e.* it finds a solution (in the least-squares sense) to the following system of equations:

$$y_i = h_i(I, \mathbf{y}, \Theta), \qquad \forall i \in I .$$

Accordingly, the final form of the auxiliary function h in (2) is

$$h(I, \mathbf{y}, \Theta) = \sum_{i \in I} \| y_i - h_i(I, \mathbf{y}, \Theta) \|^2 . \tag{5}$$

The way h_i is defined is inspired by the idea that any pixel i can receive a label prediction from a pixel j, of which i is a neighbor. This prediction can be either *class-dependent* or *class-indepedent*. Both types depend on the quantization class q_j of pixel j and on the relative position Δ_{ij} of pixel i with respect to j, whereas the class-dependent one depends additionally on the label y_j of pixel j. We impose two different neighborhood systems for the class-independent and class-dependent predictions, which are denoted by $\mathcal{N}^{(1)}$ and $\mathcal{N}^{(2)}$, respectively. Additionally, we define two nonnegative constants $\alpha^{(1)}$ and $\alpha^{(2)}$, satisfying $\alpha^{(1)} + \alpha^{(2)} = 1$, that control the relative importance of class-independent predictions over class-dependent ones. The predictions are delivered by means of a set of parameters that have to be learned. We denote by $\mathbf{w}^{(r,\Delta)} \in \mathbb{R}^{|L|}$ the class-independent parameters that we have for a given pair of quantization class $r \in \mathcal{Q}$ and relative position $\Delta \in \mathbb{Z}^2$, whereas we write $\mathbf{W}^{(r,\Delta)} \in \mathbb{R}^{|L| \times |L|}$ for the class-dependent parameters, which form a matrix due to the additional dependence on the label (see, Figure 1). Intuitively, given pixels j and $i \in \mathcal{N}_j^{(1)}$, we have that $w_k^{(q_j, \Delta_{ij})}$ represents a weight proportional to the hypothesis that pixel i should be labelled $k \in L$ from the viewpoint of pixel j, independently from the label of j. Similarly, given pixels j and $i \in \mathcal{N}_j^{(2)}$, we have that $W_{kk'}^{(q_j, \Delta_{ij})}$ represents a weight proportional to the hypothesis that pixel i should be labelled $k \in L$ according to pixel j, given that the latter has label $k' \in L$.

By combining all predictions for pixel i cast by the neighboring pixels we obtain the following definition for function h_i:

$$\begin{aligned} h_i(I, \mathbf{y}, \Theta) = & \sum_{j \in I} \nu_{ij}^{(1)} W^{(q_j, \Delta_{ij})} y_j \\ & + \sum_{j \in I} \nu_{ij}^{(2)} w^{(q_j, \Delta_{ij})} . \end{aligned} \tag{6}$$

where $\nu_{ij}^{(1,2)} = \alpha^{(1,2)} 1_{i \in \mathcal{N}_j^{(1,2)}} / \sum_h 1_{i \in \mathcal{N}_h^{(1,2)}}$ and $\Theta = (\mathbf{w}, \mathbf{W})$. Since h_i is a linear function with respect to the labelling $\mathbf{y}$, the minimization problem in (2) is convex.

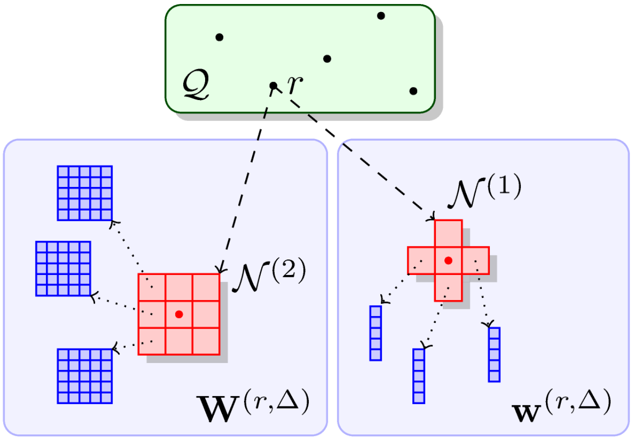


Figure 1. The parameters of our model depend on the quantization space $\mathcal{Q}$ and on the neighborhood systems $\mathcal{N}^{(1)}$ and $\mathcal{N}^{(2)}$.

Learning. We learn the parameters Θ of our model from a training set $\left\{ \left(I^{(t)}, \boldsymbol{\ell}^{(t)} \right) \right\}_{t=1}^{N}$ by solving the following minimization problem:

$$\min_{\Theta} \frac{\lambda}{2} \|\Theta\|^2 + \sum_{t=1}^{N} h^{(\gamma)} \left(I^{(t)}, \pi_{\mathcal{L} \to \tilde{\mathcal{L}}}(\boldsymbol{\ell}^{(t)}), \Theta \right) . \tag{7}$$

The first term is the ℓ_2-regularization, and h_γ is a parametrization of (5) defined as follows

$$h^{(\gamma)} (I, \mathbf{y}, \Theta) = \sum_{i \in I} d_{M_\gamma(\ell_i)} \left(y_i, h_i(I, \mathbf{y}, \Theta) \right)^2 .$$

Here, $\ell_i = \pi_{\tilde{L} \to L}(y_i)$, $d_{M_\gamma(\ell_i)}(\cdot, \cdot)$ is the Mahalanobis distance with precision matrix $M_\gamma(\ell_i)$ given by

$$M_\gamma(\ell_i) = \sum_k \pi_{L \to \tilde{L}}(k) D \left[\gamma(\mathbf{e} - \mathbf{e}^{\ell_i}) + \mathbf{e}^{\ell_i} \right] \pi_{L \to \tilde{L}}(k)^\top , \tag{8}$$

where $D[z]$ denotes a diagonal matrix with diagonal z. Note that, by sticking to (3) and (4), the precision matrix $M_\gamma(\ell_i)$ reduces to

$$M_\gamma(\ell_i) = D \left[\gamma(\mathbf{e} - \mathbf{e}^{\ell_i}) + \mathbf{e}^{\ell_i} \right] .$$

Moreover, h is equivalent to $h^{(\gamma)}$ when $\gamma = 1$, *i.e.* $h(I, \mathbf{y}, \Theta) = h^{(1)}(I, \mathbf{y}, \Theta)$. Intuitively, the parameter $\gamma > 0$ controls the degree of orthogonality of h_i with respect to the subspace spanned by wrong labels in $\{\pi_{\mathcal{L} \to \tilde{\mathcal{L}}}(k) : k \in L \setminus \{\ell_i\}\}$.

The optimization in (7) is a *convex*, ℓ_2-regularized, sparse least-squares problem in the parameters Θ, which can be solved efficiently, *e.g.* by using LSQR [20] or LSMR [6]. The number of variables involved in the optimization is given by the number of pairs in $\mathcal{Q} \times L$ that are effectively observed in the training data times the average number of entries of the matrices $W^{(q_j, \Delta_{ij})}$, which is given by the per-pixel average neighborhood size times the number of classes.

The additional parameters $\alpha^{(1,2)}$ and γ are tuned in such a way as to minimize the Hamming loss function on the training data.

Inference. Inference on an image I takes place according to (2) by solving a least-squares problem, which is *convex* due to the linearity of (5) in the variables y_i's and the convexity of $\tilde{\mathcal{L}}$. As in the case of learning, a solution to the optimization problem can be found efficiently, *e.g.* by using LSQR [20] or LSMR [6]. The number of variables to optimize in this case is simply given by the number of pixels times the number of classes.

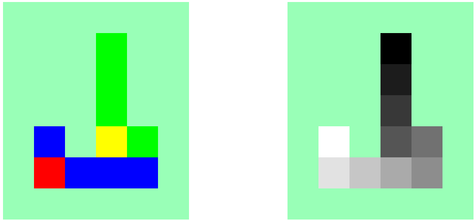
6. Experiments

In this section we provide experimental results on different datasets, demonstrating practicability and efficiency of our approach. We consider [19] as the most related work from literature and therefore reproduced two of their experiments. First, we implemented their *snake toy example*, to demonstrate the general concept of our method and its capability for learning conditional interactions. The second experiment we reproduced from [19] aims at learning *calligraphy properties for reconstruction/inpainting tasks* in occluded regions of handwritten Chinese characters. This experiment shows that inpainting results can be considerably improved when applying rules learned from conditional interactions in the neighborhood. To further demonstrate the practicability of our approach, we evaluate on the CamVid dataset [4] for the task of *semantic image labelling*. This dataset is especially challenging since it exhibits large variation and complex interactions among several object classes in street scene images.

To demonstrate the flexibility of our proposed model, we use different quantization functions in each experiment. Please note that the design of proper quantization functions can be customized to individual applications. Finally, we report the used parameters as well as running times for training and prediction, when executing our non-optimized C++ implementation on a single core in a standard desktop computer.

6.1. Snake Toy Example

The snake example illustrates that even very simple tasks can only be solved in a satisfying way when conditional interactions are learned. A *snake* consists of exactly ten pixels, sequentially arranged with connections only in a four-neighbourhood. Each position in the snake is associated with one label, starting from head (black) to tail (white) as illustrated in Figure 2(b). During training and prediction, only the color-coded quantizations as shown in Figure 2(a) are given, which encode the direction of the next label. More specifically, red means *go up*, yellow means *go right*, blue means *go left* and green means *go down*. When the quantization directs to a background pixel (cyan), the end of the snake is reached.



(a) Snake training sample.

(b) Snake ground truth labelling

Figure 2. Snake experiment. A snake is a sequence of 10 pixels connected according to a 4-neighborhood system. Each pixel is labelled with its index in the snake sequence (right). A training sample is a direction-based encoding of the snake (left), from which we want to recover the original snake sequence.

	RF	Unary	MRF	DTF [19]	**SLP**
Accuracy	90.3	90.9	91.9	99.4	**100**
Accuracy (tail)	100	100	100	100	100
Accuracy (mid)	28	28	38	95	**100**

Table 1. Results obtained on the snake experiment.

Clearly, the use of unary classifiers alone would be insufficient to infer the correct labelling as there is only little local evidence in the input images. However, when the quantization rules can be recovered as conditional interactions from training data, inference can reconstruct each labelling by propagating the respective information from tail to head.

For this experiment we impose a 1-neighborhood in the pixel itself (represented as $\cdot$) for class-independent interactions, and a 4-neighborhood ($\uparrow$, $\leftarrow$, $\rightarrow$, $\downarrow$) for the class-dependent ones. The label space is given by $L = \{0, \ldots, 10\}$, where label 0 corresponds to the background and labels $1-10$ identify the snake sequence. We adopt the following values for our additional parameters: $\alpha = 0.5$, $\gamma = 1$, and $\lambda = 0$.

In Table 1 we report the results obtained on a test set of 100 snakes. As we can see, we achieve a perfect reconstruction of the snakes with a full score of 100%, outperforming all competing approaches among which we find unary classifiers, MRF and the recent DTF [19]. Interestingly, our training procedure is based on just 20 training samples as opposed to the 200 ones used in [19]. The reason why such a small training set is enough for our approach to learn the snake rules, can be evinced by estimating the quantity of information, expressed as pairs quantization class / groundtruth label, that is effectively observed in the training sam-

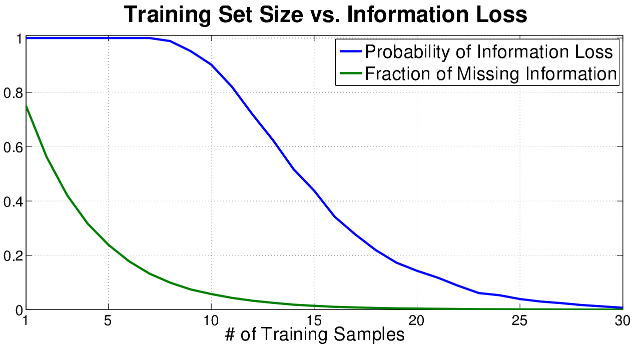


Figure 3. Missing information quantity (quantization-label co-occurrence) as a function of training examples in a repeated Bernoulli process (1000 repetitions).

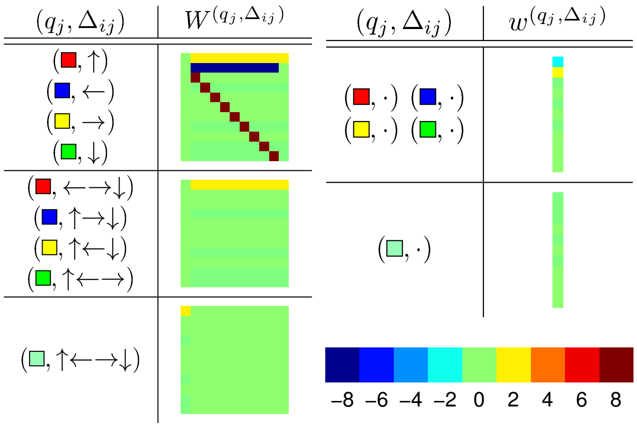


Figure 4. Parameters $\Theta = (\mathbf{W}, \mathbf{w})$ learned for the snakes experiments using 20 training samples. For a detailed description see Section 6.1. The color bar indicates the values of the matrix/vector entries.

ples. Specifically, Figure 3 shows the empirically estimated probability of information loss in the training set, *i.e.* the probability of not observing all admissible combinations of quantization class and ground-truth label in the training set, as a function of the training set size (blue curve). Additionally, we plot the fraction of information lost, *i.e.* the fraction of admissible combinations of quantization class and ground-truth label not present in the training set, as a function of the training set size (green curve). All statistics are taken with respect to 1000 trials. As we can see, with 20 training samples the probability of not gathering all the information about the snakes is around 15%. The fraction of missing information however is small enough for our method to close the gap and achieve a perfect reconstruction.

In Figure 4 we report the parameters Θ that have been leaned by our approach. The table on the left reports the class-dependent parameters. The first column shows pairs of quantization class and displacement vector, while the second one shows the entries of the corresponding $|L|^2$-matrix $W^{(q_j,\Delta_{ij})}$. The table on the right reports the class-independent parameters. Similarly, the first column shows pairs of quantization class and displacement vector, while the second one shows the entries of the corresponding $|L|$-vector $w^{(q_j,\Delta_{ij})}$.

In order to decode the meaning of the learnt parameters, we focus on labelling a pixel i based on the observations deriving from a pixel j, of which i is a neighbor. Starting from the class-independent prediction in the right table first row, we can see that if i has a color not corresponding to background, then the label of pixel i is pushed towards 1 and away from 0 (background), whereas if pixel i is cyan (second row) then no action is taken. More interesting is what happens once the class-dependent parameters in the left table are taken into account. The first row shows all cases where the displacement vector Δ_{ij} follows the underlying direction encoded in the color q_j of pixel j, *e.g.* color blue corresponds to the snake developing in the direction ←. As we can see, the learned matrix of parameters $W^{(q_j,\Delta_{ij})}$ in these cases coincides. Intuitively, the matrix shows that in these cases whenever the class label of pixel j is in the range $k \in \{1, \dots, 9\}$, pixel i is pushed towards label $k+1$ and away from label 1, which was the "default" choice of the class-independent predictions. This rule can be clearly evinced from the brown and blue entries of the matrix. If pixel j has label 10, then i should take background (*i.e.* label 0; see last column of the matrix) since a snake terminates in a background pixel by definition, and if j is background then no information is propagated to the neighbors (first column of the matrix). In the second row, we see all cases of displacement vectors not corresponding to the color's direction. In this case, if j is background no information is propagated, whereas if j takes any other label, then i is pushed towards the background label 0. Finally, the last row shows that if the color of pixel j is cyan, *i.e.* background, and the label of pixel j is background, then pixel i is pushed to take label 0 as well, otherwise no action is taken. The combination of those rules in the structured local prediction (6) allows our method to perfectly reconstruct any given test snake image.

6.2. KAIST Hanja2 Dataset

In our second experiment we demonstrate that the incorporation of neighborhood information can be used to learn calligraphy properties for reconstructing occluded regions in handwritten, Chinese characters of the KAIST Hanja2 Dataset[1]. We used the original training (300 images) and testing data (100 images) of [19] and their respective, randomly generated occlusions for both, the *small* and *large* occlusion datasets. For quantization we use a single, ran-

[1] http://ai.kaist.ac.kr/Resource/dbase/Hanja/HanjaDB2.htm

Single Decision Tree (Avg)	Tree Ensemble (RF)	MRF	DTF [19]	**SLP**
68.52	74.95	75.18	76.01	**78.07**

Table 2. Reconstruction results for KAIST Hanja2 dataset in [%] for occluded regions.

domly trained decision tree with a maximum depth of 10 and simply take the resulting leaf node indices as quantization classes. In such a way, the quantization space has a maximum cardinality of $2^{10} = 1024$. During training of the decision tree, we used 2000 iterations per node and simple pixel difference tests on the gray values, which were allowed to look at most 80 pixels away. As class-dependent parameters for our model we used a densely connected 8-neighborhood and additionally a sparse set of neighbors at $\{(-3, 0), (3, 0), (0, -3), (0, 3)\}$. The class-independent location was fixed to the center position of the pixel.

In Table 2, we compare to the classification results for the small-occlusion dataset when using only our single, baseline decision tree, a tree ensemble of 10 trees and the MRF and DTF results taken from [19]. Our method boosts the initial classification score of the single decision tree by almost 10%, outperforming sophisticated methods like MRF and DTF. Additionally our method is extremely fast, *i.e.* once the decision tree is trained ($\approx$ 30s), our model takes only 3.2s including training on all available training samples *and* evaluation on the entire test set while DTF reports a total training time of approximately one hour.

In Figure 5, we show some qualitative results obtained on the large occlusion dataset and compare to plain decision tree classification results. Here, we trained our model on all densely connected neighbors in a 5×5 neighborhood and learned parameters for 24 pixel pairs, while fixing the class-independent parameter to the center position of the respective pixel. Please note how our approach produces meaningful reconstructions by automatically learning typical calligraphic strokes from the data while vanilla classification with the single tree (expectedly) produces noisy inpainting results.

6.3. Semantic Segmentation on CamVid Dataset

In this experiment we demonstrate the performance of our model when using it for the task of semantic image labelling on the challenging CamVid dataset [4]. This dataset is a collection of several driving scene videos, where a subset of 711 images is almost entirely annotated into 32 object categories. The standard protocol for evaluating on this dataset considers 11 categories [4, 13] and a split into 367 images for training and 233 for testing. As our baseline we used the unary classifications obtained from the publicly available automatic labelling environment (ALE) software [13][2], rescaling the images by a factor of three and intentionally ignoring additional segmentations and object detector information. Assigning the class label with maximum probability yields a considerable global classification score of 77.39%, which we use as a baseline for our approach.

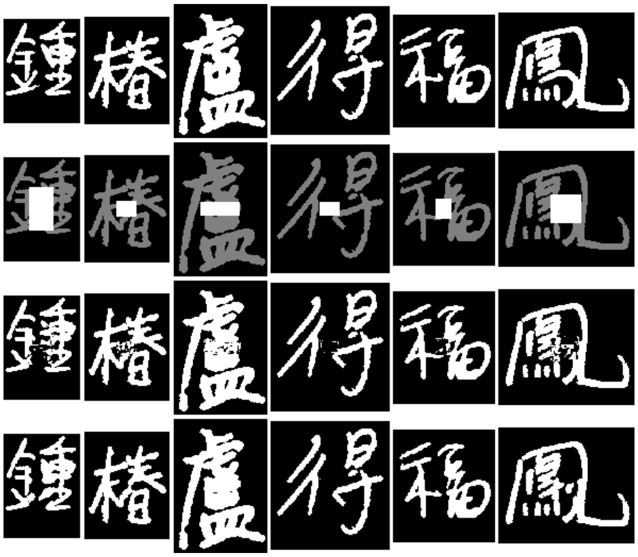


Figure 5. Qualitative reconstruction results on large occlusion areas of KAIST Hanja DB2 dataset. Each column illustrates one example, where first row shows ground truth, second row the occlusion area, third row the single decision tree reconstruction and final row our obtained results.

We designed a simple quantization function that interrelates gradient direction and magnitude strength of the corresponding intensity images with the chosen unary class label. In particular, we were binning the magnitude strength into intervals $\{[0], (0, 0.05], (0.05, 0.15], (0.15, 0.25], (0.25, 1]\}$ and the angles into 8 uniform partitions over 2π, resulting in maximally $(5 * 8 * 11) = 440$ quantization classes. We investigated several configurations for the class-dependent neighborhood settings for this experiment but obtained the best results by selecting a densely connected 8-neighborhood with an additional, small and sparse set of neighbors at $\{(-3, 0), (3, 0), (0, -3), (0, 3)\}$ while fixing the class-independent location to the center position. Training our model takes approximately 2.5 hours, using 20% of the available training data while prediction takes several seconds per image. In Figure 6 we show the comparison of classification scores on a per-image basis for the Day-Scene test data (171 images). On average, we improve the results by 4.77% per test image. Considering the whole dataset, *i.e.* including also the Dusk Scene (62 images), we improve

[2] http://cms.brookes.ac.uk/staff/PhilipTorr/ale.htm

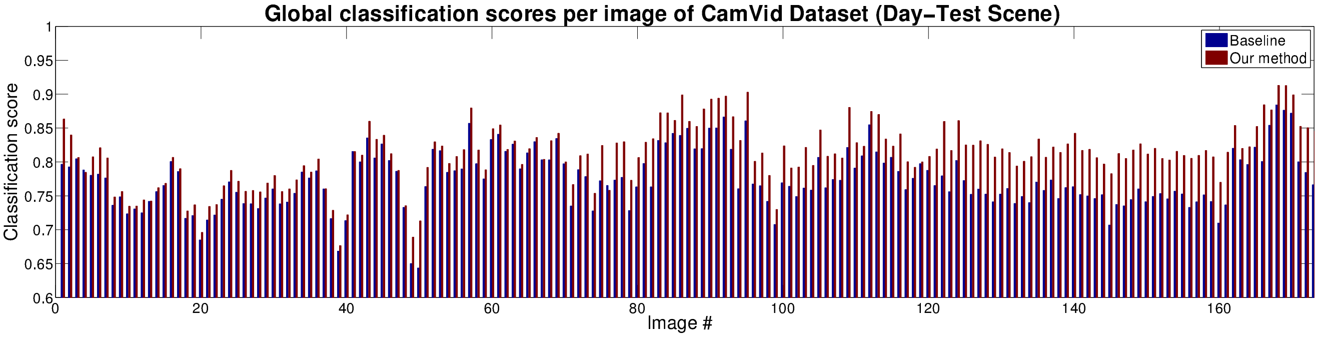


Figure 6. Classification scores on Day-Test scene of CamVid dataset (171 images) for used baseline (blue) and our obtained result (red).

the global pixel accuracy scores to 81.50% (+4.11%). Our score is slightly lower compared to 83.80% reported in [13], however, we are neither using a hierarchical model nor additional object detector information.

7. Conclusions

In this work we have presented *Structured Local Predictors* (SLP), *i.e.* a novel model that is applicable to image labelling problems. SLP are functions that provide labels for a pixel in an image by exploiting contextual information from labels *and* so-called quantization classes. The context is defined using flexibly configurable, (pairwise) neighborhood relations. We cast the parameter learning process into a structured-learning problem, which can be efficiently solved due to the introduction of an auxiliary continuous, structured output space. Moreover, our model is convex in training and prediction which guarantees to find the global solution by employing efficient off-the-shelf algorithms from convex optimization. Another core contribution of our model is to allow interaction via custom-made quantizer functions, *i.e.* discrete intermediate representations of the image data which can be designed in a task-specific manner. In our experiments we demonstrated broad applicability on computer vision tasks like inpainting and semantic labelling, obtaining competitive results when compared to state-of-the-art labelling approaches.

Acknowledgements. Peter Kontschieder acknowledges financial support of the Austrian Science Fund (FWF) from project ’Fibermorph’ with number P22261-N22.

References

[1] D. Batra, R. Sukthankar, and T. Chen. Learning class-specific affinities for image labelling. In *(CVPR)*, 2008.

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

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

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

[5] T. S. Cho, N. Joshi, C. L. Zitnick, S. B. Kang, R. Szeliski, and W. T. Freeman. A content-aware image prior. In *(CVPR)*, 2010.

[6] D. C.-L. Fong and M. A. Saunders. LSMR: An iterative algorithm for sparse least-squares problems. *SIAM J. Sci. Comput.*, 33(5):2950–2971, 2011.

[7] B. Fulkerson, A. Vedaldi, and S. Soatto. Class segmentation and object localization with superpixel neighborhoods. In *(ICCV)*, 2009.

[8] 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.

[9] S. Gould, R. Fulton, and D. Koller. Decomposing a scene into geometric and semantically consistent regions. In *(ICCV)*, 2009.

[10] S. Gould, J. Rodgers, D. Cohen, G. Elidan, and D. Koller. Multi-class segmentation with relative location prior. *(IJCV)*, 2008.

[11] P. Kohli, L. Ladicky, and P. H. S. Torr. Robust higher order potentials for enforcing label consistency. *(IJCV)*, 2009.

[12] L. Ladicky, C. Russell, P. Kohli, and P. Torr. Associative hierarchical CRFs for object class image segmentation. In *(ICCV)*, 2009.

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

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

[15] B. Liu, S. Gould, and D. Koller. Single image depth estimation from predicted semantic labels. In *(CVPR)*, 2010.

[16] T. Mensink, J. Verbeek, and G. Csurka. Learning structured prediction models for interactive image labeling. In *(CVPR)*, 2011.

[17] S. Nowozin and C. H. Lampert. Global connectivity potentials for random field models. In *(CVPR)*, 2008.

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

[19] S. Nowozin, C. Rother, S. Bagon, T. Sharp, B. Yao, and P. Kohli. Decision tree fields. In *(ICCV)*, 2011.

[20] C. C. Paige and M. A. Saunders. LSQR: An algorithm for sparse linear equations and sparse least squares. *TOMS*, 8(1):43–71, 1982.

[21] C. Rother, V. Kolmogorov, and A. Blake. Grabcut - interactive foreground extraction using iterated graph cuts. *(SIGGRAPH)*, 2004.

[22] 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. *(IJCV)*, 81, 2007.

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