Perceptually Inspired Layout-aware Losses for Image Segmentation

Anton Osokin[1] and Pushmeet Kohli[2]

[1] Moscow State University, Moscow, Russia
[2] Microsoft Research, Cambridge, UK

Abstract. Interactive image segmentation is an important computer vision problem that has numerous real world applications. Models for image segmentation are generally trained to minimize the Hamming error in pixel labeling. The Hamming loss does not ensure that the topology/structure of the object being segmented is preserved and therefore is not a strong indicator of the quality of the segmentation as perceived by users. However, it is still ubiquitously used for training models because it decomposes over pixels and thus enables efficient learning. In this paper, we propose the use of a novel family of higher-order loss functions that encourage segmentations whose layout is similar to the ground-truth segmentation. Unlike the Hamming loss, these loss functions do not decompose over pixels and therefore cannot be directly used for loss-augmented inference. We show how our loss functions can be transformed to allow efficient learning and demonstrate the effectiveness of our method on a challenging segmentation dataset and validate the results using a user study. Our experimental results reveal that training with our layout-aware loss functions results in better segmentations that are preferred by users over segmentations obtained using conventional loss functions.

Keywords: structured prediction, image segmentation, loss-based learning, max-margin learning, perceptual error metrics

1 Introduction

Interactive image segmentation is an important problem in Computer Vision that involves separating an object (foreground 'fg') of interest, specified by some user provided seeds, from the rest of the image (background 'bg'). Like many other problems in Computer Vision, and Machine learning in general, fg-bg segmentation can be formulated in terms of learning a prediction function $f : \mathcal{X} \to \mathcal{Y}$ that maps a set of inputs $x \in \mathcal{X}$ (e.g. images or features) to some desired outputs $y \in \mathcal{Y}$ (eg. locations or pixel-wise segmentations of one or more objects in the image). The prediction function for image segmentation is typically assumed to take the form of a minimization of a low order (typically pairwise) energy function. Such prediction functions have become very popular in computer vision because certain classes of pairwise energies can be efficiently minimized (either exactly or approximately) using Min-cut/Max-flow algorithms [14].

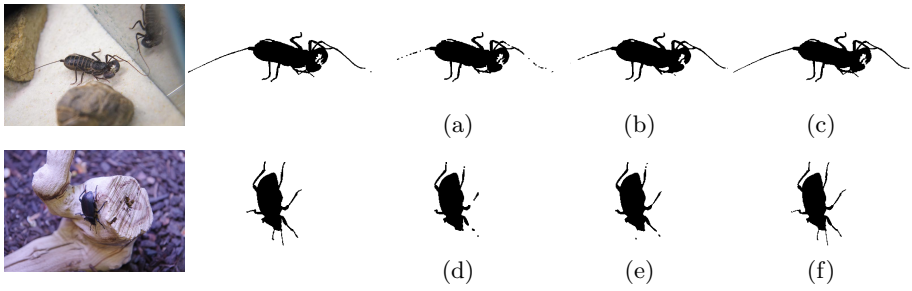


Fig. 1. *The left column shows* 2 images from "twigs and legs" dataset [10]. *The second left column* shows the zoomed ground-truth labelings. *The other columns* correspond to different distortions of the ground truth: *(a)* and *(d)* – morphological opening of the ground truth; *(b)* and *(e)* – averaging over 5x5 window; *(c)* and *(f)* – thinning of "fat" parts of the ground truth. Table 1 shows 4 different loss functions computed for these segmentations

Table 1. Different losses computed for segmentations (a)-(f) in figure 1 (the lower the loss is the better the segmentation is w.r.t. it)

Loss	1a	1b	1c	1d	1e	1f
Hamming	**0.19**	0.20	0.24	0.11	**0.10**	0.12
Jaccard	**3.91**	4.18	4.95	5.14	**4.91**	5.83
Area	0.20	**0.05**	0.21	0.11	**0.06**	0.12
Skeleton	3.72	0.71	**0.45**	2.90	1.53	**0.59**

Most discriminative methods for learning prediction functions such as Max-Margin Markov Networks (M^3N) [24], and Structured Support Vector Machines (SSVMs) [25] are based on the principle of Empirical Risk Minimization (ERM). This technique has gained wide-spread acceptance in Computer Vision by producing impressive results for many vision problems including image segmentation [19,22] and object detection [3,6]. ERM requires choosing the prediction function that makes the best predictions under some metric or *loss function*. More formally, given a family of prediction functions $f_{\mathbf{w}} : \mathcal{X} \to \mathcal{Y}$ parameterized by $\mathbf{w}$ and a training set of examples of input/output pairs $\mathcal{D} = \{(\mathbf{x}_n, \mathbf{y}_n); n \in \{1, ..., N\}, \mathbf{x}_n \in \mathcal{X}, \mathbf{y}_n \in \mathcal{Y}\}$, the learning algorithm tries to find the optimal weight vector $\mathbf{w}^*$ that minimizes the total task-dependent loss as:

$$\mathbf{w}^* = \arg\min_{\mathbf{w}} L(\mathbf{w}) = \arg\min_{\mathbf{w}} \frac{1}{N} \sum_{n=1}^{N} \Delta(\mathbf{y}_n, f_{\mathbf{w}}(\mathbf{x}_n)) \tag{1}$$

where loss function $\Delta(\mathbf{y}_n, \hat{\mathbf{y}}_n)$ denotes the cost of predicting output $\hat{\mathbf{y}}_n$ when the correct prediction is $\mathbf{y}_n$. The Hamming distance (number of mislabelled pixels) between the prediction and the ground truth is widely used as the loss for training models for image segmentation. One of the reasons for this choice is

the fact that Hamming distance decomposes over pixel variables and thus results in a tractable learning problem (see sec. 2). Although the choice of the Hamming loss allows efficient learning, it also poses a problem. The Hamming loss is not a strong indicator of the quality of the segmentation results as perceived by users. This point is illustrated in figure 1 and table 1.

A number of loss functions have been proposed in the literature as a replacement for the Hamming loss. Lempitsky et al. [16] balance costs for false positives and false negatives in order to compensate for differences in prior probabilities. The PASCAL VOC segmentation challenge [5] uses Jaccard distance (the intersection/union metric). Pletscher and Kohli [19] use a loss that encourages segmentations to be of the correct size (area loss). All the above-mentioned losses can be computed using the confusion matrix of the mislabeled pixels. These loss functions consider how many errors were made but ignore *where* these errors were made, i.e. they do not care if the topology/structure of the object being segmented is preserved.

In this paper, we address the problem of computing layout preserving segmentations. We propose a skeleton-based loss function that allows to learn a low order model using higher order loss functions that penalize segmentations that differ from the ground truth in terms of the layout. This approach can be combined for further improvement with recently proposed high-order models (e.g. [17,1]) or densely-connected pairwise models [15]. Our work generalizes the loss functions used in [19] to a much larger family of layout-aware losses that still allow efficient training. We propose two layout-aware loss functions: the Row-Column loss and the Skeleton loss. Both losses take into account the spatial relationships between image pixels. Figure 1 and table 1 show examples of segmentations that are ranked differently using Hamming, Jaccard, Area, and Skeleton losses. It can be observed that only the skeleton loss selects the segmentation with the preserved structure as the best.

We evaluate the proposed loss functions together with several other baseline losses on a challenging interactive image segmentation dataset. We perform a user study to quantify the difference between the human notion of similarity of image segmentations and the similarities defined by our loss functions. Experimental results reveal that the skeleton loss leads to better segmentation results that are considered better by users.

2 Loss-based training

In this section we introduce our notation and review the loss-based max-margin approach [25,24]. For a given input $\mathbf{x} \in \mathcal{X}$ we consider a model that predicts output $\mathbf{y} \in \mathcal{Y}$[3] by maximizing a linearly parameterized score function:

$$f_{\mathbf{w}}(\mathbf{x}) = \arg\max_{\mathbf{y} \in \mathcal{Y}} \langle \mathbf{w}, \varphi(\mathbf{x}, \mathbf{y}) \rangle \tag{2}$$

[3] Formally the set of possible labelings $\mathcal{Y}$ itself depends on the object $\mathbf{x}$, e.g. the number of variables in each $\mathbf{y}$ depends on the resolution on image $\mathbf{x}$. In this paper we omit the dependency of $\mathcal{Y}$ on $\mathbf{x}$ to lighten the notation.

where $\mathbf{w}$ is a vector of the model parameters and $\varphi(\mathbf{x}, \mathbf{y})$ is a mapping of a joint input/output space to a space of so-called generalized features. Generalized features are usually predefined and depend on the nature of the task.

In computer vision, problems are very often formulated using graphical models which represent the factorized form of a score function. We consider a pairwise model with structure specified by graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, where $\mathcal{V}$ is a set of nodes (each node $i \in \mathcal{V}$ corresponds to a variable y_i in output space $\mathcal{Y}$) and $\mathcal{E}$ is a set of edges that correspond to direct dependencies between the variables. Each variable y_i can take values from set $\mathcal{K} = \{0, 1\}$. The score function (negative energy) of the model takes the following form:

$$\langle \mathbf{w}, \varphi(\mathbf{x}, \mathbf{y}) \rangle = \sum_{i \in \mathcal{V}} \psi_i(y_i, \mathbf{x}, \mathbf{w}^u) + \sum_{(ij) \in \mathcal{E}} \psi_{ij}(y_i, y_j, \mathbf{x}, \mathbf{w}^p) \tag{3}$$

where ψ_i and ψ_{ij} are linear function w.r.t. model parameters $\mathbf{w}$. For convenience the set of model parameters is separated into unary ($\mathbf{w}^u$) and pairwise ($\mathbf{w}^p$) parameters. It is well known that if the score function is supermodular w.r.t. variables $\mathbf{y}$ than it can be maximized efficiently over labelings $\mathbf{y}$ using min-cut/max-flow algorithms [14].

Having defined the form of the predictor function we formulate the problem of learning the model parameters $\mathbf{w}$ given the set of input/output pairs

$$\left\{(\mathbf{x}^1, \mathbf{y}^1), \dots, (\mathbf{x}^N, \mathbf{y}^N)\right\}.$$

The margin-rescaled version of the max-margin approach (also referred to as structured SVM, SVM-struct) formulates a convex upper bound on the regularized empirical risk in the following way:

$$\frac{\lambda}{2}\|\mathbf{w}\|^2 + \frac{1}{N}\sum_{n=1}^{N}\left(\max_{\mathbf{y} \in \mathcal{Y}} \left(\langle \mathbf{w}, \varphi(\mathbf{x}^n, \mathbf{y}) \rangle + \Delta(\mathbf{y}, \mathbf{y}^n)\right) - \langle \mathbf{w}, \varphi(\mathbf{x}^n, \mathbf{y}^n) \rangle\right). \tag{4}$$

Here $\Delta(\mathbf{y}, \mathbf{y}^n)$ is a loss function, specifying the penalty for prediction $\mathbf{y}$ in place of ground-truth labeling $\mathbf{y}^n$.

The traditional approach to minimize (4) is to formulate it as a quadratic program (QP) with exponentially many constraints,

$$\min_{\mathbf{w}\xi} \quad \frac{\lambda}{2}\|\mathbf{w}\|^2 + \sum_{n=1}^{N} \xi^n \tag{5}$$

$$\text{s.t.} \quad \langle \mathbf{w}, \varphi(\mathbf{x}^n, \mathbf{y}) \rangle + \Delta(\mathbf{y}, \mathbf{y}^n) - \langle \mathbf{w}, \varphi(\mathbf{x}^n, \mathbf{y}^n) \rangle \leq \xi^n,$$
$$\forall n = 1, \dots, N, \quad \forall y \in \mathcal{Y}, \tag{6}$$

and to solve it via a cutting plane algorithm [25]. Algorithm 1 states the simplest version of this approach.

3 Choosing the loss function

The choice of loss function $\Delta(\mathbf{y}, \mathbf{y}^n)$ is a crucial component of the max-margin formulation (4). The only requirement that algorithm 1 imposes on the loss

Algorithm 1 Cutting plane algorithm to solve SSVM problem (5)-(6) [25].

Input: $\{(\mathbf{x}^1, \mathbf{y}^1), \ldots, (\mathbf{x}^N, \mathbf{y}^N)\}$, λ, ε.
Output: $\mathbf{w}^*$ – the solution of 5-6;
1: $S^n = \emptyset$, $\forall n = 1, \ldots, N$;
2: **repeat**
3: **for** $n = 1, \ldots, N$ **do**
4: $H(\mathbf{y}) := \Delta(\mathbf{y}, \mathbf{y}^n) + \langle \mathbf{w}, \varphi(\mathbf{x}^n, \mathbf{y}) - \varphi(\mathbf{x}^n, \mathbf{y}^n) \rangle$;
5: $\hat{\mathbf{y}} = \arg\max_{\mathbf{y} \in \mathcal{Y}} H(\mathbf{y})$; *// find the most violated constraint*
6: $\xi^n = \max\{0, \max_{\mathbf{y} \in S^n} H(\mathbf{y})\}$; *// compute the current slack*
7: **if** $H(\hat{\mathbf{y}}) > \xi^n + \varepsilon$ **then**
8: $S^n := S^n \cup \{\hat{\mathbf{y}}\}$;
9: $\mathbf{w} \leftarrow$ optimize (5) with constraints defined by $\bigcup_n S^n$
10: **until** no S^n has changed during iteration

function is that it should allow the solution of the loss-augmented MAP inference problem in step 5:

$$\max_{\mathbf{y} \in \mathcal{Y}} \Big(\langle \mathbf{w}, \varphi(\mathbf{x}^n, \mathbf{y}) \rangle + \Delta(\mathbf{y}, \mathbf{y}^n) \Big). \tag{7}$$

In this section we review the loss functions typically used to evaluate segmentation quality and describe new layout-aware losses.

3.1 Decomposable loss functions

From a computational stand-point, the simplest loss functions are the ones that can be represented as the sum of the unary potentials w.r.t. labeling $\mathbf{y}$. In this case, the loss-augmented inference problem (7) is as hard as the vanilla MAP-inference problem (2). Arguably the simplest and most often used decomposable loss function is the Hamming distance to the ground-truth labeling:

$$\Delta_{\mathrm{H}}(\mathbf{y}, \mathbf{y}^n) = \frac{1}{|\mathcal{V}|} \sum_{i \in \mathcal{V}} [y_i \neq y_i^n]. \tag{8}$$

Here $[\cdot]$ is the Iverson bracket notation: $[A]$ equals 1 if a logical expression A is true and 0 otherwise.

A more flexible version of Hamming loss can be obtained by associating weights to all terms in (8) as follows:

$$\sum_{i \in \mathcal{V}} \sum_{k \in \mathcal{K}} c_{ik} [y_i = k] [y_i^n \neq k]. \tag{9}$$

In this formulation weights c_{ik} can be conditioned on the ground-truth labeling $\mathbf{y}^n$ without making optimization w.r.t. $\mathbf{y}$ harder. This manipulation allows us to construct the more general weighted Hamming loss (wH) function.

The weighted Hamming loss follows the intuition that the pixels at the boundary of the object are harder to segment correctly [12] and thus are more valuable: $c_{ik} = 1 + A \exp(-d_i(\mathbf{y}^n)/B)$, where $d_i(\mathbf{y}^n)$ is the distance between pixel i and the closest point on the foreground-background boundary in the labeling $\mathbf{y}^n$, and A, B are distance parameters. In our experiments we use $A = 10$ and $B = 7$.

3.2 High-order loss functions

The Hamming loss (and even its weighted variant) do not represent the "perceptual quality" of the segmentation well. This has led researchers to adopt other metrics for evaluating results. For instance, the PASCAL VOC segmentation challenge [5] uses the Jaccard distance (also known as "the intersection/union metric") between the sets of pixels belonging to the object according to the predicted and the ground-truth segmentations. This metric is defined as follows:

$$\Delta_J(\mathbf{y}, \mathbf{y}^n) = 1 - \frac{TP}{TP + FP + FN} = 1 - \frac{\sum_{i \in \mathcal{V}} y_i y_i^n}{\sum_{i \in \mathcal{V}} y_i y_i^n + y_i(1 - y_i^n) + (1 - y_i) y_i^n} \tag{10}$$

where TP, FP, FN denote the number of True Positives, False Positives, and False Negatives, correspondingly. Jaccard loss (10) is not decomposable which makes the loss-augmented MAP-inference (7) problem different from MAP inference. Tarlow and Zemel [23] proposed a message passing scheme that can approximately solve the loss-augmented MAP-inference problem with the Jaccard loss function. Nowozin [18] studied the process of decision making given the probabilistic model of unknown variables and the Jaccard loss.

3.3 Supermodular loss functions

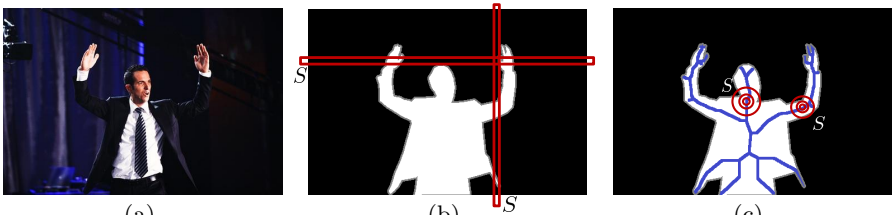
Pletscher and Kohli [19] make the observation that certain high-order loss functions $\Delta(\mathbf{y}, \mathbf{y}^n)$ are supermodular and can thus be maximized in polynomial time. They used a supermodular higher-order loss function that penalized segmentations that differed in terms of the area of the foreground segment:

$$\Delta_{\mathrm{A}}(\mathbf{y}, \mathbf{y}^n) = \frac{1}{|\mathcal{V}|} \left| \sum_{i \in \mathcal{V}} y_i - \sum_{i \in \mathcal{V}} y_i^n \right|. \tag{11}$$

They showed that the loss-augmented MAP-inference problem can be solved by solving an equivalent st-mincut/maxflow problem.

3.4 Layout dependent losses

High-order losses (11), (10) and the decomposable Hamming loss (8) view the segmentation as a sequence of pixels and are unaware of the spatial layout of the pixels. Weighted Hamming loss (9) is in some sense smarter and tries to introduce the notion that for "good" segmentation not all pixels are equally important: the closer the pixel is to the ground-truth boundary the more important it is. We believe that for human perception, layout and topology of the segmentation are very important criteria for the segmentation to be "good". In this section we present two new higher-order loss functions that take the spatial layout of pixels into account.



(a) (b) (c)

Fig. 2. The initial image – *(a)*; pixel sets that form groups for the row-column loss – *(b)*, and skeleton loss – *(c)*.

Row-Column loss. One way to characterize a segmentation is through its silhouette: an orthogonal projection of the 2-D set of pixels in a certain direction. The silhouette-based loss requires that for each line in a certain direction the number of pixels that belong to the object should be as close to the number of pixels of the ground-truth segmentation as possible. Formally, this loss can be expressed as follows:

$$\Delta_{\text{RC}}(\mathbf{y}, \mathbf{y}^n) = \sum_{d \in \mathcal{D}} \sum_{S \in \mathcal{S}_d} \left| \sum_{i \in S} y_i - \sum_{i \in S} y_i^n \right|. \tag{12}$$

Here $\mathcal{S}_d$ is a set of all lines in direction d that intersect the image domain and $\mathcal{D}$ is a set of directions considered. In our experiments we restrict ourselves to 2 directions: horizontal and vertical. Figure 2b illustrates the pixel sets included in one group for the row-column loss for the case where $\mathcal{D} = \{\text{Horizontal, Vertical}\}$.

Skeleton loss. A skeleton of the object segment is an important morphological characteristic for human perception. The notion of skeleton is well-known in the image processing and mathematical morphology communities (see e.g. [8] for a review).

Our skeleton-based loss function is motivated from the intuition that pixels around the skeleton of the ground-truth segmentation are very important and should be segmented correctly. We achieve this by using the following loss function:

$$\Delta_{\text{s}}(\mathbf{y}, \mathbf{y}^n) = \sum_{d \in \mathcal{D}} \sum_{S \in \mathcal{S}_d} a_S^d \left| \sum_{i \in S} y_i - \sum_{i \in S} y_i^n \right| \tag{13}$$

where $\mathcal{D}$ is a set of all points on the skeleton, $\mathcal{S}_d$ is a set of all pixel sets associated with pixel d of the skeleton, a_S^d are the weighting coefficients.

For each pixel on the skeleton, we define the following three sets: a circle centered at the pixel with a radius 25% larger than the distance to the boundary, and circles with the radius equal to 50% and 25% of the radius of the first circle. To make the number of sets smaller, we sub-sample the points on the skeleton such that the loss only considers 25% of the points. We set the weighting coefficients in such a way that all the sets have equal impact regardless of their

size: $a_S^d = 1/\sum_{i \in S} y_i^n$. Figure 2c illustrates the pixel sets included in one group for the skeleton loss.

As defined in equation 13, the skeleton loss does not penalize segmentations for mislabeling pixels that are not contained in any of the circles. To compensate for this bias instead of the pure skeleton loss we use a composite loss which is a weighted sum of Skeleton and Hamming losses and is defined as:

$$\Delta_{\text{skel}}(\mathbf{y}, \mathbf{y}^n) = \alpha \Delta_{\text{s}}(\mathbf{y}, \mathbf{y}^n) + (1 - \alpha) \Delta_{\text{H}}(\mathbf{y}, \mathbf{y}^n) \tag{14}$$

where α is a mixing coefficient. In our experiments we use $\alpha = 0.5$ and normalize both losses in such a way that the maximum possible loss equals 1.

4 Inference with high-order losses

We now describe how the loss-augmented inference problem can be solved for the layout-aware loss functions. Both skeleton (14) and row-column (12) losses can be written down in the following form:

$$\Delta(\mathbf{y}, \hat{\mathbf{y}}) = \sum_{S \in \mathcal{S}} a_S \left| \sum_{i \in S} y_i - b_S \right| \tag{15}$$

where set $\mathcal{S}$ is a set of subsets of nodes: $\mathcal{S} \subset 2^{\mathcal{V}}$, $a_S \geq 0$, b_S are the coefficients. Note that this form is very general. In fact, decomposable losses like the Hamming loss as well as higher-order losses such as the area loss (11) can be viewed as special cases of this general form. It is also easy to see that this general form is supermodular and thus can be maximized in polynomial time. However, the worst case complexity of supermodular maximization is still quite high which prevents their application to large scale maximization problems such as the ones encountered in this paper.

Instead, we follow the approach used in [12,19,11] to transform the problem to the pairwise one:

$$\begin{aligned} \left| \sum_{i \in S} y_i - b_S \right| &= \max_{z_S \in \{0,1\}} \left(z_S \left(\sum_{i \in S} y_i - b_S \right) + (1 - z_S) \left(b_S - \sum_{i \in S} y_i \right) \right) \\ &= \max_{z_S \in \{0,1\}} \left((1 - 2z_S) \left(b_S - \sum_{i \in S} y_i \right) \right). \end{aligned} \tag{16}$$

The function in the r.h.s. is supermodular and pairwise w.r.t. variables z_S and y_i. Such functions can be maximized efficiently using standard max-flow/min-cut algorithms [14].

5 Image segmentation model

In this section we provide the details of the pairwise random field model (3) for image segmentation that was trained using the different loss functions. Following

the approach of [19], we have one variable in $\mathcal{V}$ for each pixel of the image. We define the set of edges $\mathcal{E}$ using the 8-connected pixel grid. For each node and edge in the model we compute a set of unary and pairwise features respectively. Afterwards, we combine corresponding unary features of each pixel to form a vector of unary generalized features (which is later multiplied by unary weights $\mathbf{w}^U$). The pairwise features define the Potts pairwise potentials combined into the two groups for each pairwise feature: diagonal and horizontal/vertical edges.

Unary potentials. For every node in the model we compute the following 51 unary features: 3 RGB channels, the likelihood from 5-component Gaussian mixture models fitted to the predefined foreground and background seeds in RGB and CIELUV spaces independently, 3 CIELUV color channels, 40 distance transform features, and 1 constant bias feature. Every feature $f_k(i)$ is used to define a unary potential:

$$\psi_i(y_i, \mathbf{x}, \mathbf{w}^u) = \sum_k \Big(w^u_k f_k(i)[y_i = \text{`}fg\text{'}] - w^u_k f_k(i)[y_i = \text{`}bg\text{'}] \Big). \tag{17}$$

All the features are normalized in such a way that the maximum value equals 1, and the minimum value – 0. We use the approach from [9] to learn the GMM-based appearance models from pixels that belong to user-specified "seed" regions.

To construct the distance transform features we use generalized distance transforms to background and foreground seeds. Following [9] we define the distance from a pixel to the seed region as the minimum length of a discrete path that leads form the pixel to the region, where the length of each path is defined as follows:

$$L(\Gamma) = \sum_{i=0}^{n-1} \sqrt{(1-\gamma) d(\Gamma^i, \Gamma^{i+1})^2 + \gamma \|\Delta I(\Gamma^i, \Gamma^{i+1})\|^2}. \tag{18}$$

Here $\Gamma = \{\Gamma^i\}_{i=0}^n$ is a discrete path connecting point Γ_0 and Γ_n. All neighboring elements of the path are connected in 8-connectivity sense and $d(\cdot,\cdot)$ is the Euclidian distance between them. $\Delta I(\cdot,\cdot)$ is a vector in color space equal to the color difference of the corresponding points; γ is a weight factor between Euclidian and color differences. We construct features using distance transforms to both foreground and background seeds with the following parameters: $\gamma = 0$ (Euclidian distance); I – CIELUV color channels, $\gamma = 1$; I – RGB color channels, $\gamma = 1$; I – appearance model response channel, $\gamma = 1$. Two feature responses for an example image are shown in figure 3e and 3f. In addition we threshold each of the distances at 5 levels uniformly distributed between the minimum and the maximum values. In total we get 40 distance transform features.

Pairwise potentials. We use one constant pairwise feature and 5 contrast dependent features. All the pairwise features define Potts pairwise potentials with separate weights for diagonal and horizontal/vertical edges (12 pairwise generalized

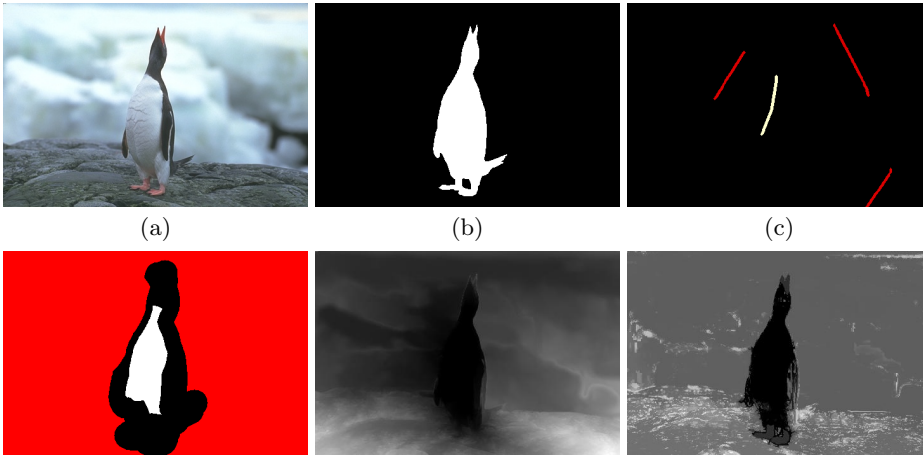


Fig. 3. *(a)* – the initial image from the dataset; *(b)* – the ground truth; *(c)* – the initial seed provided by [9]; *(d)* – the extended seeds that we use in out experiments; geodesic distance from the foreground seeds in the *(e)* RGB channels; *(f)* appearance model response channel.

features in total). Formally, the pairwise potentials of the image segmentation model are defined as follows:

$$\psi_{ij}(y_i, y_j, \mathbf{x}, \mathbf{w}^p) = \sum_k \exp\left(-c_k \beta \|I_i - I_j\|^2\right)[y_i \neq y_j] \tag{19}$$

where I_i is the color of the i-th pixel in the RGB color space, β is the inverse average difference between the neighboring pixels on the current image (as suggested in [20]), c_k is a weighting coefficient that we varied to get different features; we used the following values of c_k: 0, 0.1, 0.3, 1, 3, 10.

6 Experiments

In this section we present the details of our experimental evaluation including the dataset, the implementation details and the results.

Dataset. We perform our experiments on a subset of the dataset provided by [9], which is an extended version of the GrabCut dataset [2]. The dataset contains 151 images, their ground-truth segmentations, and user-defined object/background seeds. Prior to all experiments we've chosen 60 images which seemed to be hard for the model and on which different losses seemed to show different properties: objects with thin and long structure, mixed color statistics, etc. Figure 3 shows an image from the data set (a), the corresponding ground truth (b), and the predefined seeds (c). To avoid model mis-specification we've enlarged the user defined seeds, making them more similar to tight trimaps [13] (see fig. 3d). To

do this we apply the following morphological operations. We dilate the user defined seeds by 50% of the maximum distance to segment boundary, intersect the obtained region with 10 pixel eroded segment, union the result with 20 pixel eroded segment.

Implementation details. We implemented algorithm 1 in Matlab. In all our experiments, finding the most violated constraints (step 5) was the computational bottleneck. We used the interior-point method provided by MATLAB Optimization Toolbox to solve the QP (step 9) and called the QP-solver after adding each individual constraint. To solve the min-cut/max-flow problem we used the C++ code connected to Matlab via the MEX interface. Following [19] we use the IBFS algorithm [7][4] instead of the popular Boykov-Kolmogorov [4] algorithm. The latter is slower for denser graph structures. For the majority of image processing operations mentioned in section 5 we use the software provided by [9][5] and [19].[6] All the losses in our experiments are normalized to $[0, 1]$ segment. Although the worst possible configuration w.r.t. the skeleton and the row-column losses can be non-trivial we can always obtain it by solving the loss-augmented inference (7) with weights $\mathbf{w}$ fixed to 0.

Experimental evaluation. We split our dataset half and half 8 times into train and test sets. We train the segmentation algorithm (2) on each training set with Hamming loss (8), weighted Hamming loss (9), area loss (11), row-column loss (12), and skeleton loss (14). For each loss we select a value of regularization parameter λ using the cross-validation over the generated 8 sets.

For each combination of train and test losses we report the test-loss value averaged over all generated train/test datasets and images within one dataset (table 2). As a baseline we add the GrabCut method [2] implemented in the OpenCV library. Note that the best performance with respect to a certain loss is achieved not necessarily with training with the same loss. Figure 4 shows some qualitative results of images achieved via training with different losses.

User study. To evaluate the quality of the proposed loss functions, we conducted a user study. The user study consisted of 2 stages. In the first stage, a participant not associated with the project was shown all segmentations and asked to select images in the test set (selected randomly prior to the study) where the difference in the segmentation results was most significant. The person was not told about the objective of the research. The resulting 15 images were used for the second stage of the user study.

In the second stage of the user study each of the 18 participants for each image selected in the first stage was shown a page containing the following images: the original image, the ground-truth segmentation, and the results obtained from the models trained with 4 different loss functions (Hamming, area, row-column,

[4] http://www.cs.tau.ac.il/~sagihed/ibfs/
[5] http://www.robots.ox.ac.uk/~vgg/software/iseg/
[6] https://github.com/ppletscher/hol

Table 2. Values of different losses on Train and Test sets when the model is trained using different losses at the Training stage

	Hamming		HammingW		Area		Row-Column		Skeleton	
Training Stage	Train	Test	Train	Test	Train	Test	Train	Test	Train	Test
Hamming	7.62	9.56	10.56	12.86	6.26	8.31	10.06	12.81	20.65	23.32
HammingW	7.75	9.99	10.54	13.17	6.91	9.60	10.41	13.67	21.47	24.35
Area	7.43	**9.32**	10.57	12.86	**3.54**	**7.07**	9.22	**12.12**	17.63	19.57
Row-Column	7.22	9.40	**10.27**	**12.82**	4.39	7.24	**9.07**	12.30	17.79	20.37
Skeleton	**7.09**	9.61	10.36	13.33	4.65	7.43	9.11	12.58	**13.30**	**15.20**
GrabCut	13.14	13.14	15.97	15.97	12.08	12.08	18.17	18.17	28.04	28.04

Table 3. Results of the user study. For each training loss (Hamming, Area, Row-Column, and Skeleton) we report (a) the percentage of times users gave the particular loss the best rank (these numbers do not sum up to one because participants were allowed to assign equal ranks to multiple segmentation results); (b) the mean ranks given by participants (the lower, the better)

	Hamming	Area	Row-Col.	Skeleton
Best vote	20%	17%	12%	**64%**
Mean rank	2.62	2.66	2.76	**1.96**

and skeleton) in a randomized order (see fig. 5 for an example). To avoid framing biases, no annotations were provided with the segmentations. We asked the participants to rank the 4 segmentation results according to the "similarity" to the ground-truth segmentation and to also specify the magnitude of the differences – on a scale of 1 to 3 (1 being most significant).

The summary of the user-study results is presented in table 3. For each loss function we report the percentage of times the prediction from the model trained using the loss was marked as the best segmentation (among significant results) and the average rank. We also applied the nonparametric Friedman's test together with the Tukey-Kramer multiple comparison procedure [21]. The analysis showed that the skeleton loss performed best[7] and the other losses did not have significant differences. Figure 6 shows the correlations between the quality of segmentation results as perceived by users and as measured by the weighted combination of loss functions.

7 Discussion and Conclusions

In this paper we have proposed the use of higher-order layout-aware loss functions for learning conventional pairwise random field models for the problem

[7] 95% confidence intervals for the difference between the ranks of Hamming/Area/Row-Column loss and Skeleton loss are $[0.39, 0.92]$, $[0.43, 0.96]$, $[0.53, 1.06]$, correspondingly.

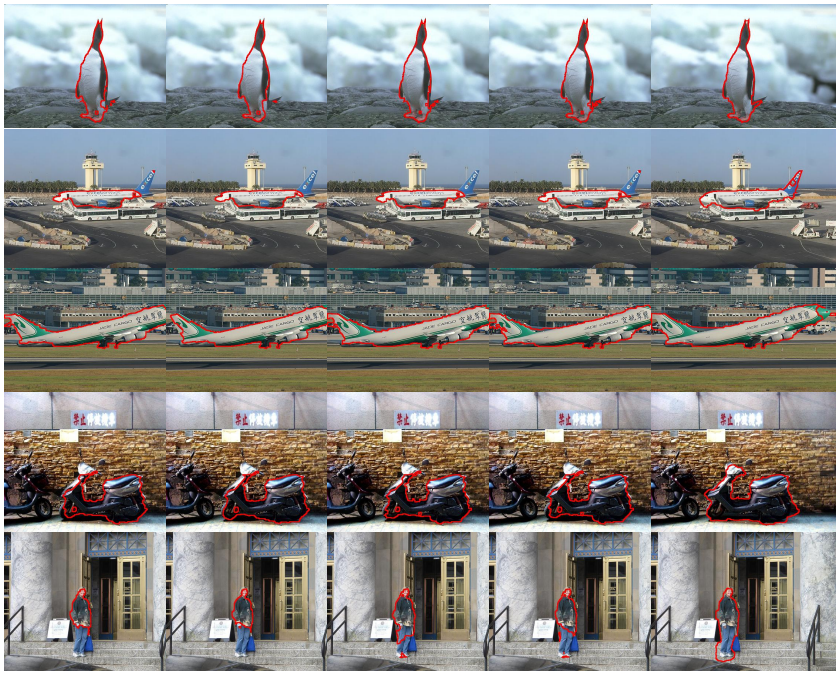


Fig. 4. Results of the prediction function trained by SSVM with *(a)* Hamming loss; *(b)* weighted Hamming loss; *(c)* area loss; *(d)* row-column loss; *(e)* skeleton loss.

of interactive image segmentation. We have shown that models trained using layout-aware loss functions produce segmentations that are consistent with the layout of the ground truth while still allowing efficient MAP inference. Further, the segmentation results produced by these models are considered perceptually closer to the ground truth by human judges compared to the results from the models trained using the conventional Hamming loss.

Our work throws up a number of directions for future work. The effect of using different low- and high-order loss functions for problems such as image denoising, optical flow, image compression are all important topics that could be investigated. A characterization of higher-order loss functions that allow efficient learning is another important direction.

In our experiments we have observed that the best results on the test set according to a certain loss are not necessarily achieved when training with the convex upper bound (4) of the same loss. This effect might be the consequence of the fact that such convex uppers bounds are not tight and the correlation between the bound and the target empirical risk (1) near the minimum of (4) is not high enough. Investigation of these effects is another important direction for future work.

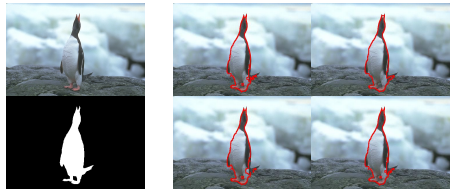


Fig. 5. A sample slide of the conducted user study. The slide contains the initial image, the ground truth, 4 segmentations produced by maximizing the score function trained using different losses (Hamming, area, row-column, and skeleton). The order of segmentations is random

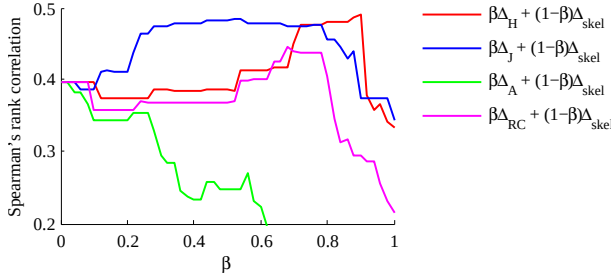


Fig. 6. The analysis of the user study results. We show Spearman's rank correlation coefficient between the weighted combination of the Skeleton and the Hamming/Jaccard/Area/Row-Column losses and rankings produced by the users

Acknowledgments. This work was supported by Microsoft – Moscow State University Joint Research Center (RPD# 1053945) and Russian Foundation for Basic Research (project 12-01-33085).

References

1. Alexe, B., Deselaers, T., Ferrari, V.: What is an object? In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR). pp. 73–80. IEEE (2010) 3
2. Blake, A., Rother, C., Brown, M., Pérez, P., Torr, P.H.S.: Interactive image segmentation using an adaptive GMMRF model. In: European Conference on Computer Vision (ECCV). pp. 428–441 (2004) 10, 11
3. Blaschko, M.B., Lampert, C.H.: Learning to localize objects with structured output regression. In: European Conference on Computer Vision (ECCV). pp. 2–15 (2008) 2
4. Boykov, Y., Kolmogorov, V.: An experimental comparison of Min-Cut/Max-Flow algorithms for energy minimization in vision. IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI) 26(9), 1124–1137 (2004) 11
5. Everingham, M., Van Gool, L., Williams, C.K.I., Winn, J., Zisserman, A.: The PASCAL visual object classes (VOC) challenge. IJCV 88(2), 303–338 (2010) 3, 6
6. Felzenszwalb, P.F., Girshick, R.B., McAllester, D.A., Ramanan, D.: Object detection with discriminatively trained part-based models. IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI) 32(9), 1627–1645 (2010) 2

7. Goldberg, A.V., Hed, S., Kaplan, H., Tarjan, R.E., Werneck, R.F.: Maximum flows by incremental breadth-first search. In: European Symposium on Algorithms (ESA). pp. 457–468 (2011) 11
8. Gonzalez, R., Woods, R.: Digital image processing. Prentice Hall (2002) 7
9. Gulshan, V., Rother, C., Criminisi, A., Blake, A., Zisserman, A.: Geodesic star convexity for interactive image segmentation. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2010) 9, 10, 11
10. Jegelka, S., Bilmes, J.: Submodularity beyond submodular energies: Coupling edges in graph cuts. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2011) 2
11. Kohli, P., Kumar, M.P.: Energy minimization for linear envelope MRFs. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2010) 8
12. Kohli, P., Ladickỳ, L., Torr, P.: Robust higher order potentials for enforcing label consistency. International Journal of Computer Vision (IJCV) 82(3), 302–324 (2009) 5, 8
13. Kohli, P., Nickish, H., Rother, C., Rhemann, C.: User-centric learning and evaluation of interactive segmentation systems. International Journal of Computer Vision (IJCV) 100(3), 261–274 (2012) 10
14. Kolmogorov, V., Zabih, R.: What energy functions can be minimized via graph cuts? IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI) 26(2), 147–159 (2004) 1, 4, 8
15. Krahenbuhl, P., Koltun, V.: Efficient inference in fully connected CRFs with gaussian edge potentials. In: Advances in Neural Information Processing Systems (NIPS) (2011) 3
16. Lempitsky, V., Vedaldi, A., Zisserman, A.: A Pylon model for semantic segmentation. In: Advances in Neural Information Processing Systems (NIPS) (2011) 3
17. Nowozin, S., Lampert, C.H.: Global interactions in random field models: A potential function ensuring connectedness. SIAM Journal on Imaging Sciences (SIIMS) 3(4) (2010) 3
18. Nowozin, S.: Optimal decisions from probabilistic models: the intersection-over-union case. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2014) 6
19. Pletscher, P., Kohli, P.: Learning low-order models for enforcing high-order statistics. In: International Conference on Artificial Intelligence and Statistics (AISTATS) (2012) 2, 3, 6, 8, 9, 11
20. Shotton, J., Winn, J., Rother, C., Criminisi, A.: Textonboost for image understanding: Multi-class object recognition and segmentation by jointly modeling texture, layout, and context. International Journal of Computer Vision 81(1), 2–23 (2009) 10
21. Sprent, P., Smeeton, N.: Applied Nonparametric Statistical Methods. Chapman & Hall/CRC (2001) 12
22. Szummer, M., Kohli, P., Hoiem, D.: Learning CRFs using graph cuts. In: European Conference on Computer Vision (ECCV). pp. 582–595 (2008) 2
23. Tarlow, D., Zemel, R.: Structured output learning with high order loss functions. In: International Conference on Artificial Intelligence and Statistics (AISTATS) (2012) 6
24. Taskar, B., Guestrin, C., Koller, D.: Max-Margin Markov networks. In: Advances in Neural Information Processing Systems (NIPS) (2003) 2, 3
25. Tsochantaridis, I., Joachims, T., Hofmann, T., Altun, Y.: Large margin methods for structured and interdependent output variables. Journal of Machine Learning Research (JMLR) 6(9), 1453–1484 (2005) 2, 3, 4, 5