Optimizing Edge Detection for Image Segmentation with Multicut Penalties

Steffen Jung, [1] **Sebastian Ziegler,** [2] **Amirhossein Kardoost,** [2] **Margret Keuper** [1,3]

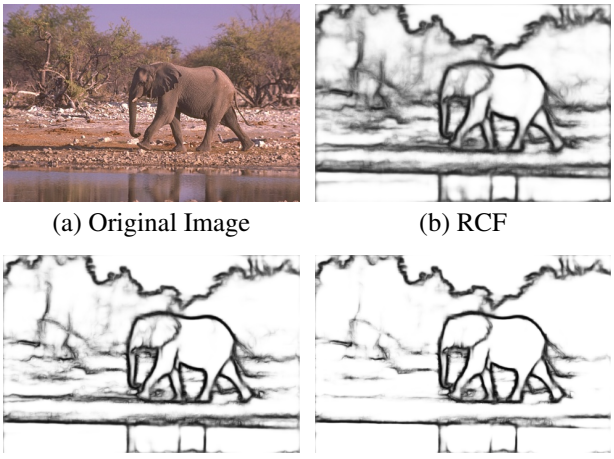
[1] Max Planck Institute for Informatics, Saarland Informatics Campus,
[2] University of Mannheim, [3] University of Siegen
steffen.jung@mpi-inf.mpg.de, sziegler@mail.uni-mannheim.de, kardoostamirhossein@gmail.com,
margret.keuper@uni-siegen.de

Abstract

The Minimum Cost Multicut Problem (MP) is a popular way for obtaining a graph decomposition by optimizing binary edge labels over edge costs. While the formulation of a MP from independently estimated costs per edge is highly flexible and intuitive, solving the MP is NP-hard and time-expensive. As a remedy, recent work proposed to predict edge probabilities with awareness to potential conflicts by incorporating cycle constraints in the prediction process. We argue that such formulation, while providing a first step towards end-to-end learnable edge weights, is suboptimal, since it is built upon a loose relaxation of the MP. We therefore propose an adaptive CRF that allows to progressively consider more violated constraints and, in consequence, to issue solutions with higher validity. Experiments on the BSDS500 benchmark for natural image segmentation as well as on electron microscopic recordings show that our approach yields more precise edge detection and image segmentation.

Introduction

Image Segmentation, one of the fundamental problems in Computer Vision, is the task of partitioning an image into multiple disjoint components such that each component is a meaningful part of the image. It is a low-level technique that assigns a label to every pixel in the image, and plays an important role in many areas like medical image analysis (Xian et al. 2018), for example localizing tumors (Litjens et al. 2017) or aneurysms (de Bruijne et al. 2004), and in other fields like satellite imagery and forensics (Ghosh et al. 2019). Former successful image segmentation approaches are graph-based, i.e., they map image elements, for example pixels or superpixels, onto a graph. While there are several different approaches to obtain a decomposition from a graph, the Minimum Cost Multicut Problem (MP) (Chopra and Rao 1993; Deza, Laurent, and Weismantel 1997), also called Correlation Clustering (Bansal, Blum, and Chawla 2004), is a well-known approach for image segmentation. Here, the number of components is unknown beforehand and no bias is assumed in terms of component sizes. The resulting segmentation is only determined by the input graph (Keuper et al. 2015), for which edge features can be generated by an edge detector such as Convolutional Neural Networks (CNNs) that predict edge probabilities (see Fig. 1 for examples).



(a) Original Image (b) RCF (c) RCF-basic-CRF (ours) (d) RCF-adaptive-CRF (ours)

Figure 1: (a) Example BSDS500 (Arbelaez et al. 2010) test image, (b) the edge map produced with the RCF edge detector (Liu et al. 2019), (c) the edge map by RCF-CRF using the basic CRF by Song et al. (2019), and (d) the edge map from RCF-CRF using our adaptive CRF. The adaptive CRF promotes closed contours and removes trailing edges.

A feasible solution to the MP decomposes the graph into disjoint subgraphs via a *binary* edge labeling. The decomposition is enforced by cycle consistency constraints in general graphs. If a path exists between two nodes where the direct edge between both is cut, then this constraint is violated. Song et al. (2019) introduced relaxed cycle constraints, i.e. constraints evaluated on non-binary network predictions, as higher-order potentials in a Conditional Random Field (CRF) to allow for end-to-end training of graph-based human pose estimation. By doing so, cycle constraint violations become a supervision signal and can be reduced when training the feature extractor and the CRF jointly. We transfer this approach to image segmentation, where such constraints enforce that object contours are closed. Yet, we observe that optimizing non-binary network predictions instead of binary edge labels only leads to few additionally closed contours. This is consistent with prior works on linear program multicut solvers which show that relaxations of

the cycle constraints to non-binary edge labels are too loose in practice (Kappes et al. 2011). In the CRF formulation, we propose to alleviate this issue by enforcing more binary (i.e. closer to 0 or 1) edge predictions that lead to less violated cycle constraints. Our contributions are twofold. We are the first to address boundary driven image segmentation using multicut constraints. To this end, we combine a CRF with the neural edge detection model Richer Convolutional Features (RCF) by Liu et al. (2019) to design an end-to-end trainable architecture that inputs the original image and produces a graph with edge probabilities optimized by the CRF. Secondly, we propose an approach that progressively uses "closer to binary" boundary estimates in the optimization of the CRF model by Song et al. (2019) and thus resolves progressively more boundary conflicts. In consequence, the end-to-end trained network yields more and more certain predictions throughout the training process while reducing the number of violated constraints. We show that this improved certainty yields improved results for edge detection and segmentation on the BSDS500 benchmark (Arbelaez et al. 2010) and the ISBI 2012 neuronal structure segmentation (Arganda-Carreras et al. 2015). An example result is given in Fig. 1. Compared to the plain RCF architecture as well as to the RCF with the CRF by Song et al. (2019), our approach issues cleaner, less cluttered edge maps with closed contours.

Related Work

In the following, we first review the related work on edge detection and minimum cost multicuts in the context of image segmentation.

Edge detection in the context of image segmentation is usually based on learning-driven approaches that facilitate to learn to discriminate between object boundary and other sources of brightness change such as textures. Structured random forests have been employed in Dollár and Zitnick (2013) to train an edge detector on local image patches. Xie and Tu (2015) proposed a CNN-based approach called holistically nested edge detection (HED) that intrinsincally leverages multiple edge map resolutions. Similarly, Kokkinos (2017) propose an end-to-end CNN for low-, mid- and high-level vision tasks such as boundary detection, semantic segmentation, region proposal generation, and object detection in a single network based on multi-scale learning. Convolutional oriented boundaries (COB) (Maninis et al. 2016) compute multiscale oriented contours and region hierarchies in a single forward pass and provide boundary orientation estimates. Such boundary orientations are needed as input along with the edge maps in order to compute hierarchical image segmentations in frameworks such as MCG (Arbeláez et al. 2014; Pont-Tuset et al. 2015) or gpb-owt-ucm (Arbelaez et al. 2010). If not estimated by the network, they have to be approximated, for example using filter-based approaches (Dollár and Zitnick 2013). In He et al. (2020), a bi-directional cascade network (BDCN) is proposed for edge detection of objects in different scales, where individual layers of a CNN model are trained by labeled edges at a specific scale. Similarly, to address the edge detection in multiple scales and aspect ratios, Liu et al. (2019) provide an edge detector using richer convolutional features (RCF) by exploiting multiscale and multilevel information of objects. Although BDCN provides slightly better edge detection accuracy on the BSDS dataset (Arbelaez et al. 2010), we base our approach on the RCF edge detection framework because of its more generic training procedure.

The multicut approach has been extensively used for image segmentation, for example in Kappes et al. (2015, 2011); Keuper et al. (2015); Arbelaez et al. (2010); Beier, Hamprecht, and Kappes (2015); Andres et al. (2011). Due to the NP-hardness of the MP, segmentation has often been addressed on pre-defined superpixels (Andres et al. 2013; Kappes et al. 2011, 2013; Beier, Hamprecht, and Kappes 2015). While Kappes et al. (2015) utilize multicuts as a method for discretizing a grid graph defined on the image pixels, where the local connectivity of the edges define the join/cut decisions and the nodes represent the image pixels, Keuper et al. (2015) proposed long-range terms in the objective function of the multicut problem defined on the pixel grid. Such terms are efficient to deal with image decomposition problems where there is no clear cut or join signal along blurry edges. An iterative fusion algorithm for the MP has been proposed in Beier, Hamprecht, and Kappes (2015) to decompose the graph. Andres et al. (2011) propose a graphical model for probabilistic image segmentation and globally optimal inference on the objective function with higher orders. A similar higher order approach is proposed also in Kappes et al. (2013); Kim, Yoo, and Nowozin (2014) for image segmentation.

End-to-end Learning of Edge Weights for Graph Decompositions

Cycle constraints in the Multicut Problem

The MP is based on a graph $G = (V, E)$, where every pixel (or superpixel) is represented by an individual node or vertex $v \in V$. Edges $e \in E$ encode whether two pixels belong to the same component or not. The goal is to assign every node to a cluster by labeling the edges between all nodes as either "cut" or "join" in an optimal way based on edge costs c_e. One of the main advantages of this approach is that the number of components is not fixed beforehand, contrary to other clustering algorithms, and is determined by the input graph instead. Since the number of segments in an image cannot be foreseen, the MP is a well-suited approach. The MP can be formulated as integer linear program (Chopra and Rao 1993; Deza, Laurent, and Weismantel 1997) with objective function $c : E \to \mathbb{R}$ as follows:

$$\min_{y \in \{0,1\}^E} \sum_{e \in E} c_e y_e \tag{1}$$

subject to

$$\forall C \in cc(G) \forall e \in C : y_e \leq \sum_{e' \in C \setminus \{e\}} y_{e'}, \tag{2}$$

where y_e is the binary labeling of an edge e that can be either 0 (join) or 1 (cut), and $cc(G)$ represents the set of all chordless cycles in the graph. If the cycle inequality constraint in

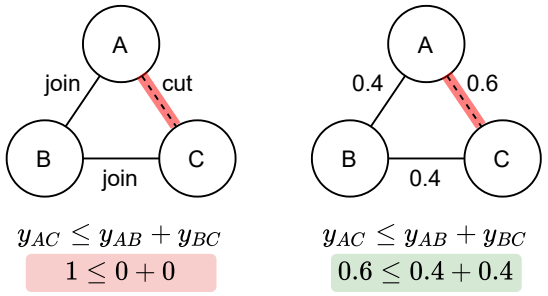


Figure 2: Invalid solution to the Multicut Problem. (left) Covered by cycle inequality constraints given the integer solution. (right) Not covered given a relaxed solution that results in (left) when rounded.

Eq. (2) is satisfied, the MP solution results in a decomposition of the graph and therefore in a segmentation of the image. Informally, cycle inequality constraints ensure that there cannot be exactly one cut edge in any chordless cycle. However, computing an exact solution is not tractable in many cases due to the NP-hardness of the problem. Relaxing the integrality constraints such that $y \in [0,1]^E$ can improve tractability, however, valid edge label configurations are not guaranteed in this case. An example can be seen in Fig. 2, where node B is supposed to be in the same component as A and C, however, A and C are considered being in different components. Infeasible solutions have to be repaired in order to obtain a meaningful segmentation of the input image. This can be achieved by using heuristics like in Beier et al. (2014); Kardoost and Keuper (2018); Keuper et al. (2015); Pape et al. (2017).

Incorporating cycle constraints into a CRF

To improve the validity of relaxed solutions, Song et al. (2019) reformulate the MP as an end-to-end trainable CRF based on a formulation as Recurrent Neural Network (RNN) by Zheng et al. (2015). By doing so, they are able to impose costs for violations of cycle inequality constraints during training. This is accomplished by first transforming the MP into a binary cubic problem, considering all triangles in the graph:

$$\min_{y \in \{0,1\}^E} \sum_{e \in E} c_e y_e + \gamma \sum_{\{u,v,w\} \in \binom{V}{3}} \begin{aligned}(&y_{uv}\overline{y}_{vw}\overline{y}_{uw} + \\ &\overline{y}_{uv}y_{vw}\overline{y}_{uw} + \\ &\overline{y}_{uv}\overline{y}_{vw}y_{uw}).\end{aligned} \quad (3)$$

This formulation moves cycle inequalities into the objective function by incurring a large cost γ whenever there is an invalid edge label configuration like $(cut, join, join)$ in a clique (as shown in Fig. 2 – all other orders implied). The binary cubic problem is then transformed to a CRF by defining a random field over the edge variables $y = (y_1, y_2, ..., y_{|E|})$, which is conditioned on the image I. The cycle inequality constraints are incorporated in the form of higher-order potentials as they always consider three edge variables. The combination of unary and third-order potentials yields the following energy function building the CRF:

$$E(y|I) = \sum_i \psi_i^U(y_i) + \sum_c \psi_c^{Cycle}(y_c) \quad (4)$$

The energy $E(y|I)$ is the sum of the unary potentials and the higher-order potentials. For the latter (Song et al. 2019) used pattern-based potentials proposed by Komodakis and Paragios (2009):

$$\psi_c^{Cycle}(y_c) = \begin{cases} \gamma_{y_c} & \text{if } y_c \in P_c \\ \gamma_{max} & \text{otherwise} \end{cases} \quad (5)$$

P_c represents the set of all valid edge label configurations, which are $(join, join, join)$, $(join, cut, cut)$, and (cut, cut, cut). The invalid edge label configuration is $(cut, join, join)$. The potential assigns a high cost γ_{max} to an invalid labeling and a low cost γ_{y_c} to a valid labeling.

Such CRFs can be made end-to-end trainable using Mean-Field Updates, as proposed by Zheng et al. (2015). This approach computes an auxiliary distribution Q over y such as to minimize the Kullback-Leibler Divergence (KL-Divergence) (Csiszár 1975; Csiszár, Katona, and Tardos 2007) between Q and the true posterior distribution of y. This step of optimizing $Q(y_i)$ instead of y_i can be interpreted as relaxation. Instead of considering violated constraints on binary edge variables (see Eq. (1)), we optimize probabilities of y_i taking a certain label l and optimize $Q(y_i = l)$ which admits values in the interval $[0, 1]$.

Zheng et al. (2015) reformulate the update steps as individual CNN layers and then repeat this stack multiple times in order to compute multiple mean-field iterations. The repetition of the CNN layer stack is treated equally to an RNN and remains fully trainable via backpropagation. Vineet, Warrell, and Torr (2014) extends this idea by incorporating higher-order potentials in the form of pattern-based potentials and co-occurrence potentials. For the CRF by Song et al. (2019) the corresponding update rule becomes:

$$Q_i^t(y_i = l) = \frac{1}{Z_i} exp\Bigg\{ -\sum_{c \in C} \Bigg(\overbrace{\sum_{p \in P_{c|y_i=l}} \Bigg(\prod_{j \in c, j \neq i} Q_j^{t-1}(y_j = p_j) \Bigg)}^{\text{valid labeling case gets low costs}} \gamma_p + \gamma_{max} \underbrace{\Bigg(1 - \Bigg(\sum_{p \in P_{c|y_i=l}} \Bigg(\prod_{j \in c, j \neq i} Q_j^{t-1}(y_j = p_j) \Bigg) \Bigg) \Bigg)}_{\text{inverse of the valid labeling case gets high costs}} \Bigg) \Bigg\}, \quad (6)$$

where $Q_i^t(y_i = l) \in [0, 1]$ is the Q value of one edge variable at mean-field iteration t with fixed edge label l (either cut or $join$). $P_{c|y_i=l}$ represents the set of valid edge label configurations according to Eq. (2), where the considered edge label l is fixed. Looking at the case where $y_i = 1$ (cut), possible valid configurations are (y_i,0,1), (y_i,1,0), and (y_i,1,1). For all valid labelings the other two variables $y_{j \neq i}$ in clique c are taken into account. Their multiplied label

probabilities from iteration t–1 for every valid label set, are summed and then multiplied with the cost for valid labelings γ_p. The inverse of the previous result is then multiplied with γ_{max}. Taking the inverse is equal to computing the same as in the valid case but with all possible invalid labelings for the considered edge variable. The results of all cliques C, which the variable y_i is part of, are summed. The updated $Q(y_i = l)$ are projected onto $[0, 1]$ using softmax. Costs γ_p and γ_{max} are considered as trainable parameters, and the update rule is differentiable with regard to the input $Q(y_i = l)$ (a network estimate of the likelihood of y_i taking label l) and the cost parameters. Since non-binary values for $Q(y_i = l)$ are optimized, invalid configurations are assigned lower costs when they become uncertain, i.e., as $Q(y_i = l) \rightarrow 0.5$.

Cooling Mean-Field Updates

There have been various attempts tightening the relaxation of the MP. For example Swoboda and Andres (2017) incorporate odd-wheel inequalities and Kappes et al. (2013) use additional terminal cycle constraints. While these approaches can achieve tighter solution bounds, they involve constraints defined on a larger number of edges. A formulation of such constraints in the context of higher-order CRFs, requiring at least an order of 4, is intractable. Therefore, we choose an alternative, more straight forward approach – we push the network predictions $Q(y)$, which we interpret as relaxed edge labels, progressively closer to 0 or 1 in the mean-field update by introducing a cooling scheme. This not only issues solutions closer to the integer solution, but also provides a better training signal to the CRF, where cycle constraints penalize non-valid solutions more consistently. For this, we substitute $Q_i^{t-1}(y_i = p_j)$ in Eq. (6) by $\phi\left(Q_j^{t-1}(y_j = p_j), k\right)$, where ϕ is the newly introduced function modifying a probability q as follows:

$$\phi(q, k) = \begin{cases} 1 - (1 - q)^k & \text{if } q >= 0.5 \\ q^k & \text{otherwise.} \end{cases} \quad (7)$$

Here, k is an exponent that can be adapted during training, and q is the edge probability. This function pushes values larger or equal to 0.5 closer to 1 and values below 0.5 closer to 0. Using this transformation in the mean-field iterations reduces the intregrality gap and therefore enforces the MP cycle constraints more strictly. This effect is amplified the larger the exponent k becomes.

Choosing a large k from the start hampers network training as the edge detection parameters would need to adapt too drastically for close-to-binary edges. We thus aim at adapting k throughout the training process such as to first penalize violated cycle constraints on the most confident network predictions and progressive addressing violated constraints on less confident (i.e. "less binary") estimates. We therefore propose a cooling schedule, which defines criteria upon which the parameter k is increased, as follows:

$$k = \begin{cases} k + 0.05 & \text{if } N(C_{inv}) < a \\ k & \text{otherwise,} \end{cases} \quad (8)$$

where $N(C_{inv})$ is the average number of invalid cycles (that do not adhere to the cycle inequality constraints) across all images and the hyperparameter a is the number of allowed cycle constraint violations before increasing the exponent k. The exponent therefore is increased after every epoch in which the number of invalid (relaxed) cycles has been decreased below the threshold a.

Leveraging the mean-field update for image segmentation requires an edge detection network that provides values for the CRF potentials. A possible learning-based approach that provides high quality edge estimates is the RCF (Liu et al. 2019) architecture. In practice, the edge detection network is first pre-trained until convergence, and then fine-tuned with the CRF. Parameter k is initialized as 1 and updated using Eq. (8) during training.

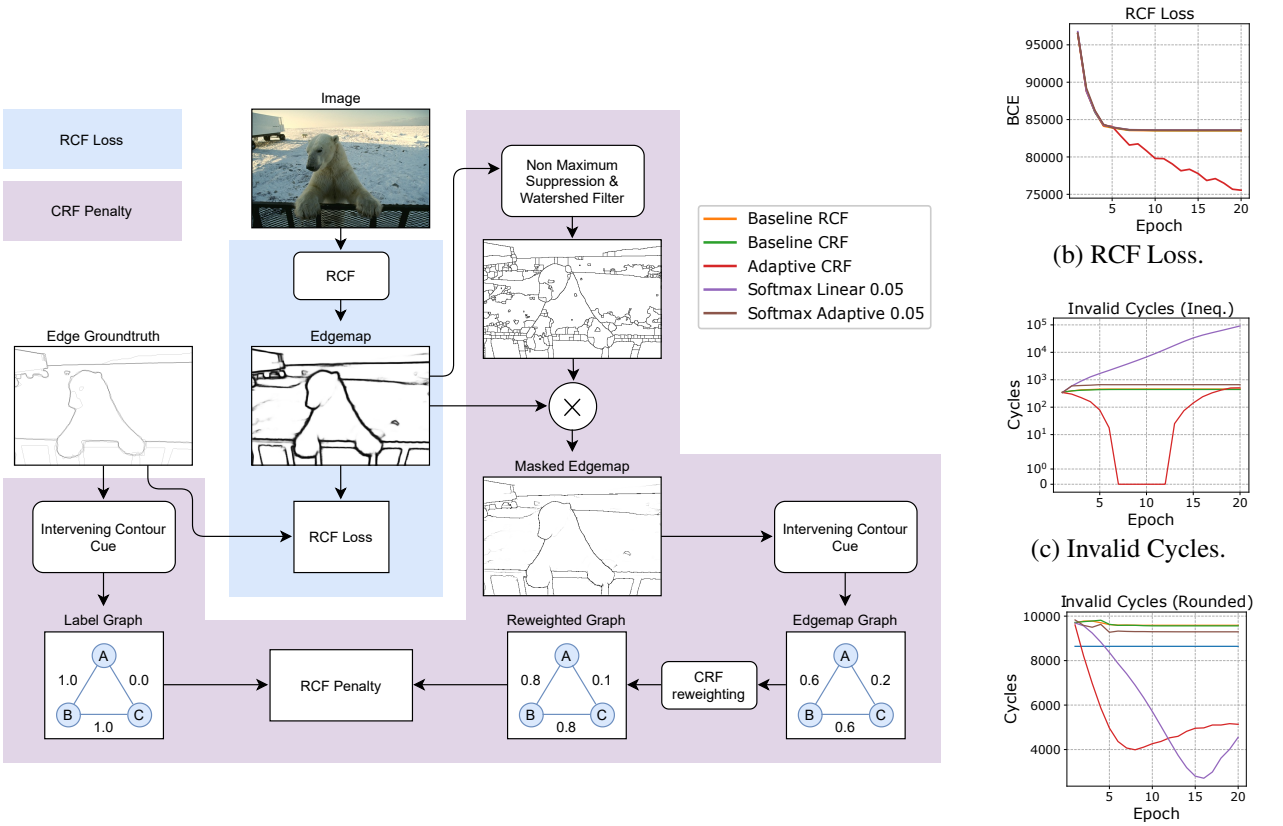
Richer Convolutional Features - CRF

The Richer Convolutional Features (RCF) architecture for edge detection has been recently developed by Liu et al. (2019). Their main idea is based on Holistically-nested edge detection (HED) (Xie and Tu 2015), where an image classification architecture like VGG16 (Simonyan and Zisserman 2014) is divided into five stages. Each stage produces a side output while and fully connected layers are removed. These side outputs are trained with an individual loss function and combined by a weighted fusion layer that learns to combine them. In contrast to HED, which only considers the last convolutional layer from each stage, the RCF architecture uses all layer information. Hence, the features used become "richer". Every side output is transformed via sigmoid activation, creating edge probability maps.

RCF-CRF Architecture

Fig. 3 shows the full proposed architecture where the CRF is combined with the RCF to optimize for consistent boundary predictions. In a first step edge maps Pb from the input image are computed by the RCF, applying the RCF Loss during training. To apply the CRF Penalty, we then need to efficiently generate a pixel-graph that represents boundaries as edge weights. For every pixel, edges of the 8-connectivity are inserted for all distances in the range of 2 to 8 pixels yielding a significant amount of cycles. For efficient weight computation, we compute Watershed boundaries on the non-maximum suppressed RCF output of the pretrained RCF model *once*. We can use those to mask subsequent RCF predictions and efficiently retrieve graph weights as they are updated by the network. As edge weights, we consider the Intervening Contour Cue (ICC) Leung and Malik (1998) which computes the probability of an edge between pixels i and j to be cut as $W_{i,j} = max(Pb(x)_{x \in L_{i,j}})$, where x represents a coordinate in the image, $Pb(x)$ is the edge probability at x and $L_{i,j}$ indicates the set of all coordinates on the line between i and j including themselves. From the Watershed masks, potential locations of the maximum can be pre-computed for efficiency.

Edge ground truth labels are created on the fly similar to the ICC by checking if there is an edge on the line between two pixels in the ground truth edge probability map to determine their cut/join label. The RCF loss is based on the cross entropy used in the original RCF (Liu et al. 2019), which ignores controversial edge points that have been annotated as



(a) The RCF-CRF training process.

(b) RCF Loss.

(c) Invalid Cycles.

(d) Invalid Cycles (rounded).

Figure 3: (a) The RCF-CRF training process is depicted, where blue highlights the RCF loss and purple highlights the additional CRF penalty. (b) Training progress in terms of RCF loss when training in different settings. Only Adaptive CRF is able to reduce the RCF loss further, while all other settings provide insufficient training signal to further improve on the basic RCF model. (c) Number of invalid cycle inequalities during training. Adaptive CRF significantly outperforms all other training settings. (d) Number of invalid cycle inequalities after rounding the solution during training. Adaptive CRF and Softmax CRF are able to improve on baselines significantly.

an edge by less than half of the annotators but at least one. They are summed to obtain the final RCF-CRF loss. After training, the model segmentations can be computed using hierarchical approaches such as MCG (Arbeláez et al. 2014) or multicut solvers such as (Keuper et al. 2015).

Experiments

We evaluate our approach in two different image segmentation applications. First, we show experiments and results on the BSDS500 (Arbelaez et al. 2010) dataset for edge detection and image segmentation. Second, we consider the segmentation of bio-medical data, in particular, electron microscopic recordings of neuronal structures of fruit flies (Arganda-Carreras et al. 2015).

Berkeley Segmentation Dataset and Benchmark

BSDS500 (Arbelaez et al. 2010) contains 200 train, 100 validation and 200 test images that show colored natural photography often depicting animals, landscapes, buildings, or humans. Due to its large variety in objects with different surfaces and different lightning conditions, it is generally considered a difficult task for edge detection as well as image segmentation. Several human annotations are given per image. The RCF is pre-trained on the augmented BSDS500 data used by Liu et al. (2019) and Xie and Tu (2015). After convergence it is fine-tuned with our CRF using only the non-augmented BSDS images to reduce training time. We evaluate the impact of fine-tuning the CRF in different training settings. These are: (Baseline RCF) further training of the RCF without CRF, (Baseline CRF) fine-tuning the RCF network without cooling scheme, and (Adaptive CRF) fine-tuning with cooling scheme. Additionally, we consider two settings where we enforce more binary solutions by introducing temperature decay in the softmax (Hinton, Vinyals, and Dean 2015) that is applied after each mean field iteration. Here, we consider two cases: (Softmax Linear) where we decay the temperature by 0.05 after each epoch, and (Softmax Adaptive), where we consider the same adaption process as in in Eq. (8).

Figure 4: Example BSDS500 (Arbelaez et al. 2010) test image. Top row depicts original image, Baseline RCF and Adaptive CRF edge maps. Bottom row depicts the respective UCMs. The edge map optimized with Adaptive CRF is less cluttered while accurately localizing the contours.

Invalid Cycles Fig. 3(c) depicts the number of violated cycle inequalities during fine-tuning of the RCF network in different settings. There, it can be seen that Baseline CRF is able to reduce the number of violations rapidly to around 500 invalid cycles per image on average. During training, it constantly stays at this level and does not significantly change anymore. Due to this observation we set parameter a in the cooling scheme of Adaptive CRF and softmax temperature decay to 500. When training Adaptive CRF with this parameter setting, the number of violations further decreases to zero. The impact of reduced uncertainty can furthermore be seen when looking at the number of invalid cycles after rounding edge probabilities to binary edge labels. As Fig. 3(d) shows, only the settings Adaptive CRF and Softmax Linear are able to reduce the number of invalid cycles after rounding. However, Softmax Linear is not able to reduce the number of violations before rounding in contrast to Adaptive CRF. This indicates that the cooling scheme of Adaptive CRF provides a better training signal for the RCF, which is also confirmed considering RCF training loss depicted in Fig. 3(b). Only Adaptive CRF is able to provide sufficiently strong training signals such that the RCF network can further decrease its training loss. Interestingly, the number of invalid rounded cycles as well as constraint violations start increasing again after some training iterations for the adaptive CRF. This shows the trade-off between the RCF loss and the penalization provided by the CRF.

Edge Detection Tab. 1 shows evaluation scores on the BSDS500 test set. The F-measure at the optimal dataset scale (ODS) and the optimal image scale (OIS) as well as the Average Precision (AP) are reported. The multiscale version (MS) is computed similar to (Liu et al. 2019) with scales 0.5, 1.0 and 1.5. Adaptive CRF achieves significant improvements compared in terms of ODS and OIS to all other training settings. The respective precision recall curves are given in the Appendix. AP decreases slightly with the CRF models, which is expected since the CRF removes uncertain edges that do not form closed components and therefore affects the high recall regime. A qualitative example is shown in Fig. 4 and in the Appendix.

Table 1: Edge Detection Results on the BSDS500 test set. All RCFs are based on VGG16. Results reported for Baseline RCF are computed for a model trained by ourselves and are slightly worse than the originally reported scores in (Liu et al. 2019).

Models	**ODS**	**OIS**	**AP**
Baseline RCF	0.811	0.827	0.815
Baseline RCF (MS)	0.812	0.830	**0.836**
Baseline CRF (ours)	0.810	0.827	0.815
Baseline CRF (MS) (ours)	0.812	0.831	**0.836**
Softmax Linear (ours)	0.810	0.826	0.815
Softmax Linear (MS) (ours)	0.812	0.831	**0.836**
Adaptive CRF (ours)	0.815	0.830	0.812
Adaptive CRF (MS) (ours)	**0.817**	**0.835**	0.833

Table 2: Segmentation results on the BSDS500 test set. All RCFs are based on VGG16. Results reported for Baseline RCF are computed for a model trained by ourselves and are slightly better than the scores reported in (Liu et al. 2019).

Models	**ODS**	**OIS**	**AP**
Baseline RCF	0.803	0.829	0.832
Baseline RCF (MS)	0.808	0.832	**0.849**
Baseline CRF (ours)	0.804	0.828	0.831
Baseline CRF (MS) (ours)	0.808	0.831	**0.849**
Softmax Linear (ours)	0.803	0.829	0.831
Softmax Linear (MS) (ours)	0.808	0.831	**0.849**
Adaptive CRF (ours)	0.808	0.830	0.828
Adaptive CRF (MS) (ours)	**0.813**	**0.834**	0.847

Image Segmentation To obtain a hierarchical segmentation using the predicted edge maps we compute MCG (Arbeláez et al. 2014) based Ultrametric Contour Maps (UCM) (Arbelaez et al. 2010) that generate hierarchical segmentations based on different edge probability thresholds. Edge orientations needed for MCG were computed using the standard filter operations. In contrast to (Liu et al. 2019) we do not use the COB framework but use pure MCG segmentations to allow for a more direct assessment of the proposed approach. Results for all training settings are reported in Tab. 2. Again, the Adaptive CRF models outperforms all other models in ODS, while AP is only slightly affected. Multi-scale (MS) information additionally improves results.

Fig. 5 depicts the segmentation PR curves comparing the RCF based methods to other standard models. Similar to the edge map evaluation (see Appendix) the curves are steeper for the CRF based model compared with the plain RCF, thereby following the bias of human annotations, i.e., approaching the green marker in Fig. 5. Depending on when the curves start to tilt the corresponding F-measure can be slightly lower as it is the case for the basic CRF. Adaptive CRF, however, also yields a considerably higher F-score improving over the baseline from 0.808 to 0.813. This result shows that employing cycle information is generally beneficial to estimate closed boundaries.

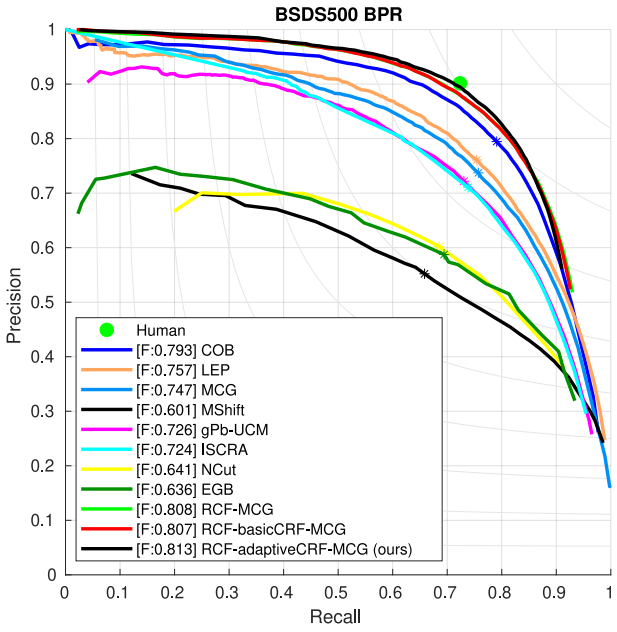


Figure 5: Precision recall curves for segmentation on the BSDS500 test set. The proposed RCF with Adaptive CRF yields the highest ODS score and operates at a higher precision level than other models.

Neuronal Structure Segmentation

Next, we conduct experiments on the segmentation of neuronal structures (Arganda-Carreras et al. 2015). The data was obtained from a serial section Transmission Electron Microscopy dataset of the Drosophila first instar larva ventral nerve cord (Cardona et al. 2010, 2012). This technique captures images of the Drosophila brain with a volume of 2 $\times$ 2 $\times$ 1.5μ and a resolution of 4 $\times$ 4 $\times$ 50 nm/pixel. The volumes are anisotropic, i.e. while the x- and y-directions have a high resolution the z-direction has a rather low resolution (Arganda-Carreras et al. 2015). Both, training and test set consist of a stack of 30 consecutive grayscale images. The goal of the challenge is to produce a binary map that corresponds to the membranes of cells in the image.

Beier et al. (2017) have applied a multicut approach to this application. Their pipeline first produced an edge map from the original images by using either a cascaded random forest or a CNN. In order to reduce the complexity they aggregated individual pixels to superpixels by using the distance transform watershed technique (Acharjya et al. 2013). Based on these superpixels they solve the multicut and the lifted multicut problem using the fusion moves algorithm (Beier, Hamprecht, and Kappes 2015).

In this context, we apply Adaptive CRF as a post-processing method to an existing graph without training the underlying edge detection. Edge weights for the test set are computed using a random forest in the simple (not lifted) multicut pipeline from (Beier et al. 2017) and define a graph. We optimize these graph weights with the proposed approach and subsequently decompose the graph using the fusion move algorithm as in (Beier et al. 2017). The number of mean-field iterations was set to 20 and for the adaptive CRF the update threshold a was set to 100. Since the CRF is not trained but used only for optimizing the graph once, the update threshold is evaluated after every mean-field iteration rather than every epoch.

Table 3: Results of the ISBI challenge on the test set.

Models	$\mathbf{V}^{\mathrm{Rand}}$	$\mathbf{V}^{\mathrm{Info}}$
Baseline (Beier et al. 2017)	0.9753	0.9874
Basic-CRF-optimized (ours)	0.9784	**0.9889**
Adaptive-CRF-optimized (ours)	**0.9808**	0.9887

The graph obtained from Beier et al. (2017) for the test set contained 74 485 cycles in total. Before applying the CRF, 61 099 of them violated the cycle inequality constraints and 38 283 cycles were invalid after rounding. Afterwards, both cycle counts were close to zero. Tab. 3 contains the results obtained for the not-modified edge weights (Baseline), the edge weights optimized with Baseline CRF and the edge weights optimized with Adaptive CRF. For evaluation we also refer to the ISBI challenge (Arganda-Carreras et al. 2015) that indicates two measures: the foreground-restricted Rand Scoring after border thinning V^{Rand} and the foreground-restricted Information Theoretic Scoring after border thinning V^{Info}. Both CRF models were able to improve the segmentation result in both evaluation metrics. While the differences in V^{Info}-score are rather small, Adaptive CRF increased the V^{Rand}-score by 0.005 compared to the baseline. Taking into account that the baseline is already very close to human performance, this is a very good result. Comparing the two CRF models it can be seen that the V^{Info}-score is almost the same for both approaches. In terms of V^{Rand}-score, Adaptive CRF improves stronger over the baseline than Baseline CRF. Overall, this experiment shows that applying the CRF is beneficial for image segmentation even without training the edge extraction. Accordingly, our approach can be applied even as a post-processing step without increasing training time.

Conclusion

We introduce an adaptive higher-order CRF that can be optimized jointly with an edge detection network and encourages edge maps that comply with the cycle constraints from the Minimum Cost Multicut Problem. Combining the CRF with the RCF model (Liu et al. 2019) for edge detection yields much sharper edge maps and promotes closed contours on BSDS500. PR curves show that the CRF based model yields steeper curves having a higher precision level. Similarly, the resulting segmentations show that the approach is able to generate more accurate and valid solutions. Moreover, the CRF can be used as post-processing to optimize a graph for cycle constraints as shown on the electron microscopy data. It has shown considerable improvement in the evaluation metrics without increasing training time.

References

Acharjya, P.; Sinha, A.; Sarkar, S.; Dey, S.; and Ghosh, S. 2013. A new approach of watershed algorithm using distance transform applied to image segmentation. *International Journal of Innovative Research in Computer and Communication Engineering*, 1(2): 185–189.

Andres, B.; Kappes, J. H.; Beier, T.; Köthe, U.; and Hamprecht, F. A. 2011. Probabilistic Image Segmentation with Closedness Constraints. In *ICCV*.

Andres, B.; Yarkony, J.; Manjunath, B. S.; Kirchhoff, S.; Turetken, E.; Fowlkes, C.; and Pfister, H. 2013. Segmenting Planar Superpixel Adjacency Graphs w.r.t. Non-planar Superpixel Affinity Graphs. In *EMMCVPR*.

Arbelaez, P.; Maire, M.; Fowlkes, C.; and Malik, J. 2010. Contour detection and hierarchical image segmentation. *IEEE transactions on pattern analysis and machine intelligence*, 33(5): 898–916.

Arbeláez, P.; Pont-Tuset, J.; Barron, J.; Marques, F.; and Malik, J. 2014. Multiscale Combinatorial Grouping. In *Computer Vision and Pattern Recognition*.

Arganda-Carreras, I.; Turaga, S. C.; Berger, D. R.; Cireşan, D.; Giusti, A.; Gambardella, L. M.; Schmidhuber, J.; Laptev, D.; Dwivedi, S.; Buhmann, J. M.; et al. 2015. Crowdsourcing the creation of image segmentation algorithms for connectomics. *Frontiers in neuroanatomy*, 9: 142.

Bansal, N.; Blum, A.; and Chawla, S. 2004. Correlation clustering. *Machine learning*, 56(1-3): 89–113.

Beier, T.; Hamprecht, F. A.; and Kappes, J. H. 2015. Fusion moves for correlation clustering. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 3507–3516.

Beier, T.; Kroeger, T.; Kappes, J. H.; Kothe, U.; and Hamprecht, F. A. 2014. Cut, glue & cut: A fast, approximate solver for multicut partitioning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 73–80.

Beier, T.; Pape, C.; Rahaman, N.; Prange, T.; Berg, S.; Bock, D. D.; Cardona, A.; Knott, G. W.; Plaza, S. M.; Scheffer, L. K.; et al. 2017. Multicut brings automated neurite segmentation closer to human performance. *Nature methods*, 14(2): 101–102.

Cardona, A.; Saalfeld, S.; Preibisch, S.; Schmid, B.; Cheng, A.; Pulokas, J.; Tomancak, P.; and Hartenstein, V. 2010. An integrated micro-and macroarchitectural analysis of the Drosophila brain by computer-assisted serial section electron microscopy. *PLoS Biol*, 8(10): e1000502.

Cardona, A.; Saalfeld, S.; Schindelin, J.; Arganda-Carreras, I.; Preibisch, S.; Longair, M.; Tomancak, P.; Hartenstein, V.; and Douglas, R. J. 2012. TrakEM2 software for neural circuit reconstruction. *PloS one*, 7(6): e38011.

Chopra, S.; and Rao, M. R. 1993. The partition problem. *Mathematical Programming*, 59(1-3): 87–115.

Csiszár, I. 1975. I-divergence geometry of probability distributions and minimization problems. *The annals of probability*, 146–158.

Csiszár, I.; Katona, G. O.; and Tardos, G. 2007. *Entropy, search, complexity*, volume 16. Springer Science & Business Media.

de Bruijne, M.; van Ginneken, B.; Viergever, M. A.; and Niessen, W. J. 2004. Interactive segmentation of abdominal aortic aneurysms in CTA images. *Medical Image Analysis*, 8(2): 127–138.

Deza, M.; Laurent, M.; and Weismantel, R. 1997. Geometry of cuts and metrics. *Mathematical Methods of Operations Research-ZOR*, 46(3): 282–283.

Dollár, P.; and Zitnick, C. L. 2013. Structured Forests for Fast Edge Detection. In *ICCV*.

Ghosh, S.; Das, N.; Das, I.; and Maulik, U. 2019. Understanding deep learning techniques for image segmentation. *ACM Computing Surveys (CSUR)*, 52(4): 1–35.

He, J.; Zhang, S.; Yang, M.; Shan, Y.; and Huang, T. 2020. BDCN: Bi-Directional Cascade Network for Perceptual Edge Detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 1–1.

Hinton, G.; Vinyals, O.; and Dean, J. 2015. Distilling the Knowledge in a Neural Network. arXiv:1503.02531.

Kappes, J. H.; Speth, M.; Andres, B.; Reinelt, G.; and Schnörr, C. 2011. Globally Optimal Image Partitioning by Multicuts. In *EMMCVPR*.

Kappes, J. H.; Speth, M.; Reinelt, G.; and Schnörr, C. 2013. Higher-order Segmentation via Multicuts. *CoRR*, abs/1305.6387.

Kappes, J. H.; Swoboda, P.; Savchynskyy, B.; Hazan, T.; and Schnörr, C. 2015. Probabilistic Correlation Clustering and Image Partitioning Using Perturbed Multicuts. In *SSVM*.

Kardoost, A.; and Keuper, M. 2018. Solving minimum cost lifted multicut problems by node agglomeration. In *Asian Conference on Computer Vision*, 74–89. Springer.

Keuper, M.; Levinkov, E.; Bonneel, N.; Lavoué, G.; Brox, T.; and Andres, B. 2015. Efficient decomposition of image and mesh graphs by lifted multicuts. In *Proceedings of the IEEE International Conference on Computer Vision*, 1751–1759.

Kim, S.; Yoo, C. D.; and Nowozin, S. 2014. Image Segmentation Using Higher-Order Correlation Clustering. *IEEE TPAMI*, 36(9): 1761–1774.

Kokkinos, I. 2017. Ubernet: Training a Universal Convolutional Neural Network for Low-, Mid-, and High-Level Vision Using Diverse Datasets and Limited Memory. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Komodakis, N.; and Paragios, N. 2009. Beyond pairwise energies: Efficient optimization for higher-order MRFs. In *2009 IEEE Conference on Computer Vision and Pattern Recognition*, 2985–2992. IEEE.

Leung, T.; and Malik, J. 1998. Contour continuity in region based image segmentation. In *European Conference on Computer Vision*, 544–559. Springer.

Litjens, G.; Kooi, T.; Bejnordi, B. E.; Setio, A. A. A.; Ciompi, F.; Ghafoorian, M.; Van Der Laak, J. A.; Van Ginneken, B.; and Sánchez, C. I. 2017. A survey on deep learning in medical image analysis. *Medical image analysis*, 42: 60–88.

Liu, Y.; Cheng, M.-M.; Hu, X.; Bian, J.-W.; Zhang, L.; Bai, X.; and Tang, J. 2019. Richer Convolutional Features for Edge Detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 41(8): 1939–1946.

Maninis, K.-K.; Pont-Tuset, J.; Arbeláez, P.; and Van Gool, L. 2016. Convolutional oriented boundaries. In *European Conference on Computer Vision*, 580–596. Springer.

Pape, C.; Beier, T.; Li, P.; Jain, V.; Bock, D. D.; and Kreshuk, A. 2017. Solving large multicut problems for connectomics via domain decomposition. In *Proceedings of the IEEE International Conference on Computer Vision Workshops*, 1–10.

Pont-Tuset, J.; Arbeláez, P.; Barron, J.; Marques, F.; and Malik, J. 2015. Multiscale Combinatorial Grouping for Image Segmentation and Object Proposal Generation. In *arXiv:1503.00848*.

Simonyan, K.; and Zisserman, A. 2014. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*.

Song, J.; Andres, B.; Black, M. J.; Hilliges, O.; and Tang, S. 2019. End-to-end Learning for Graph Decomposition. In *Proceedings of the IEEE International Conference on Computer Vision*, 10093–10102.

Swoboda, P.; and Andres, B. 2017. A message passing algorithm for the minimum cost multicut problem. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 1617–1626.

Vineet, V.; Warrell, J.; and Torr, P. H. 2014. Filter-based mean-field inference for random fields with higher-order terms and product label-spaces. *International Journal of Computer Vision*, 110(3): 290–307.

Xian, M.; Zhang, Y.; Cheng, H.-D.; Xu, F.; Zhang, B.; and Ding, J. 2018. Automatic breast ultrasound image segmentation: A survey. *Pattern Recognition*, 79: 340–355.

Xie, S.; and Tu, Z. 2015. Holistically-nested edge detection. In *Proceedings of the IEEE international conference on computer vision*, 1395–1403.

Zheng, S.; Jayasumana, S.; Romera-Paredes, B.; Vineet, V.; Su, Z.; Du, D.; Huang, C.; and Torr, P. H. 2015. Conditional random fields as recurrent neural networks. In *Proceedings of the IEEE international conference on computer vision*, 1529–1537.

Optimizing Edge Detection for Image Segmentation with Multicut Penalties: Supplementary Material

Appendix A: Training Details

Adaptive CRF By reformulating mean field updates from Eq. (6) with our adaptive function ϕ (see Eq. (7)), we get:

$$Q_i^t(y_i = l) = \frac{1}{Z_i} exp\Bigg\{ -\sum_{c \in C} \Bigg(\sum_{p \in P_{c|y_i=l}} \Big(\prod_{j \in c, j \neq i} \phi\left(Q_j^{t-1}(y_j = p_j), k\right) \Big) \gamma_p + \Big(1 - \Big(\sum_{p \in P_{c|x_i=l}} \Big(\prod_{j \in c, j \neq i} \phi\left(Q_j^{t-1}(y_j = p_j), k\right) \Big) \Big) \Big) \gamma_{max} \Bigg) \Bigg\}, \quad (9)$$

where function ϕ modifies probabilities such that their values are pushed closer to 0 or 1, depending on their current value. Fig. 6(a) shows a plot of this function with different values of k. During training, k is initialized with 1.0 and then increased based on the cooling scheme defined in Eq. (8). When training Adaptive CRF on BSDS500 for 20 epochs, we observe that k is increased each epoch with the exception of the last one (see Fig. 6(b)).

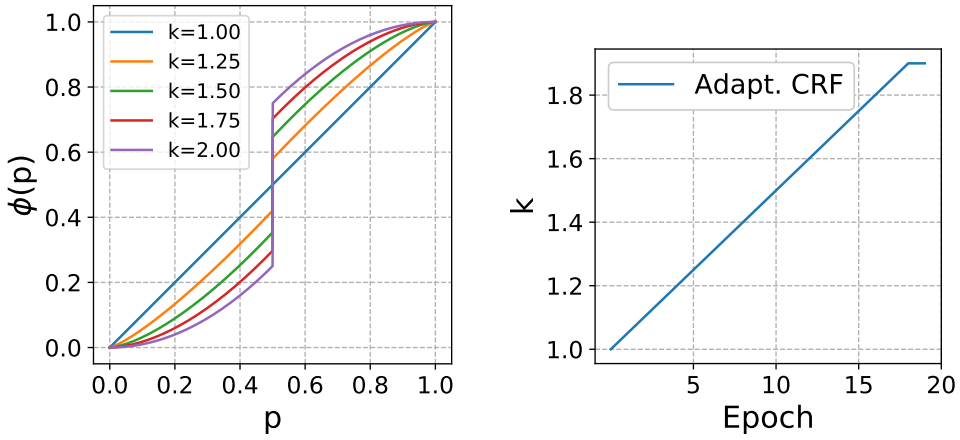


Figure 6: (left) Function ϕ plotted for different values of k. (right) Value of k during 20 epochs of training Adaptive CRF.

Linear Softmax and Adaptive Softmax The purpose of ϕ is to push values towards 1 or 0, and therefore enforce multicut constraints more strictly. However, this should also be possible by considering temperature decay in the softmax function that is called after each mean field update iteration. Hence, instead of

$$Q_i^t(y_i = l) = \frac{\exp\{\tilde{Q}_i^t(y_i = l)\}}{\sum_l \exp\{\tilde{Q}_i^t(y_i = l)\}} \quad (10)$$

we compute

$$Q_i^t(y_i = l) = \frac{\exp\{\tilde{Q}_i^t(y_i = l)/t\}}{\sum_l \exp\{\tilde{Q}_i^t(y_i = l)/t\}}, \quad (11)$$

where we initialize $t = 1.0$ and decay each epoch in the case of Linear Softmax by 0.05. In the case of Adaptive Softmax we use the same condition as for Adaptive CRF (see Eq. (8)), where t is only decayed by 0.05 if the average number of violated cycle inequalities is ≤ 500. Fig. 7 shows the temperature decaying schemes during training of 20 epochs on BSDS500.

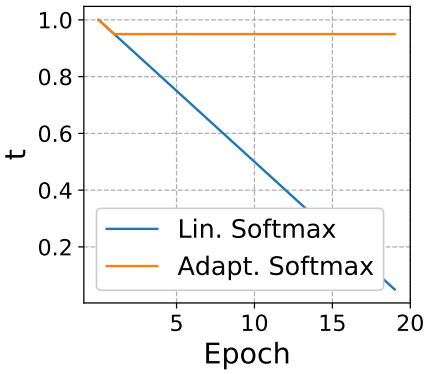


Figure 7: Values of t during 20 epochs of training Linear Softmax and Adaptive Softmax.

Training Runs We train each model on one RTX8000 GPU with batch size 10, and optimize with SGD with learning rate 10^{-6}, momentum 0.9, weight decay $2 \cdot 10^{-4}$, stepsize 3, gamma 0.1 for 20 epochs. Code is provided with the supplementary submission.

Appendix B: Quantitative Edge Detection Results on BSDS500

Fig. 8 shows the precision recall curves for edge detection on the BSDS500 test set. The proposed Adaptive CRF yields the highest ODS score and operates at a higher precision level than other models.

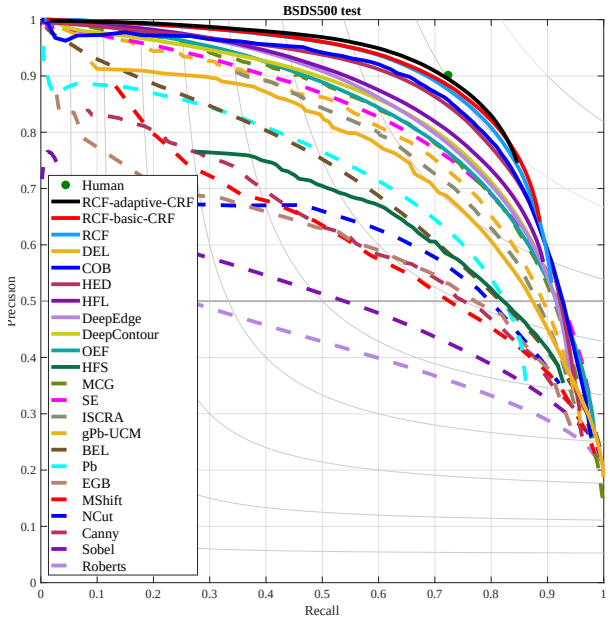
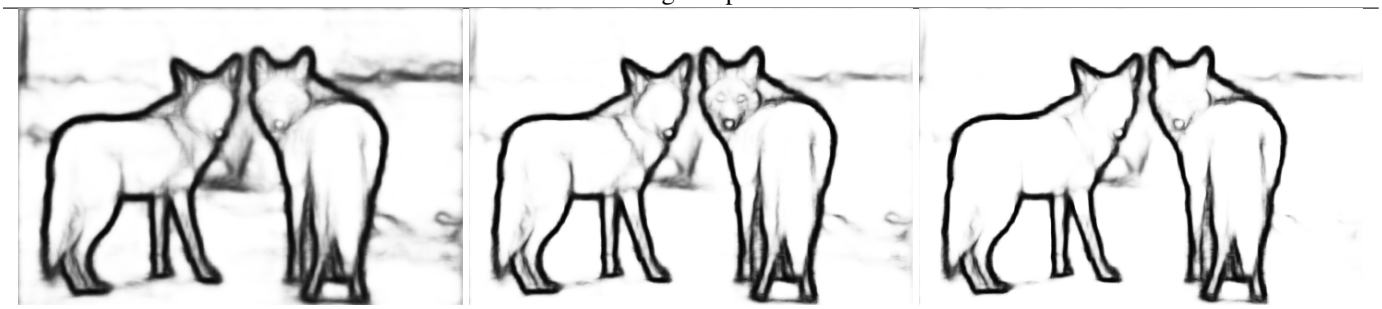


Figure 8: Edge Detection precision recall curves for the BSDS500 test set. RCF models all have the VGG backbone. Models that were optimized with the CRF yield steeper curves.

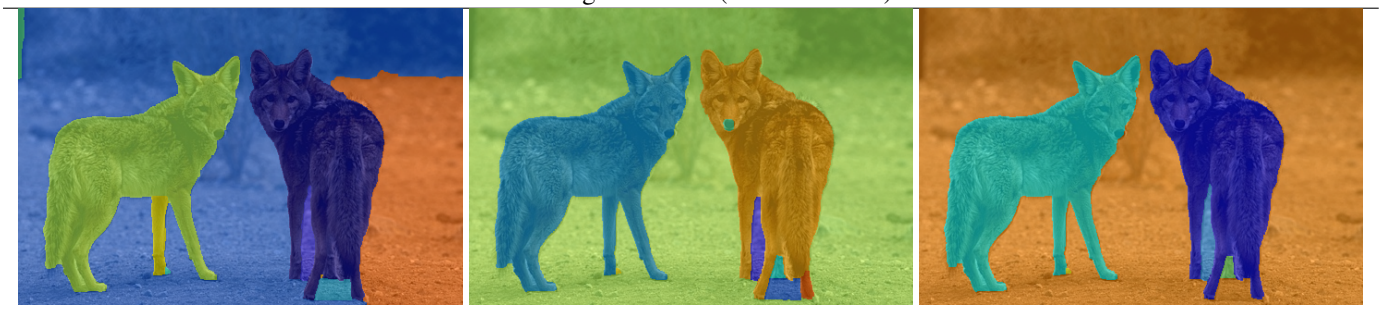
Appendix C: Qualitative Results on BSDS500

In the supplementary material we provide additional examples of BSDS500 images, their respective edge maps as well as the corresponding UCM segmentation. We compare plain RCF results with models optimized with the basic CRF and the adaptive CRF which both encourage valid cycles that comply with the cycle inequality constraints of the Minimum Cost Multicut Problem. It can be seen that applying the CRFs during training results in cleaner edge maps where trailing edges are removed and contours are more likely to be closed. The effect is amplified for the adaptive CRF. The corresponding segmentations suffer less from oversegmented backgrounds and individual components are more precise compared to the plain RCF. The additional examples illustrate that promoting closed contours when learning edge maps yields more accurate segmentations.

Edge maps

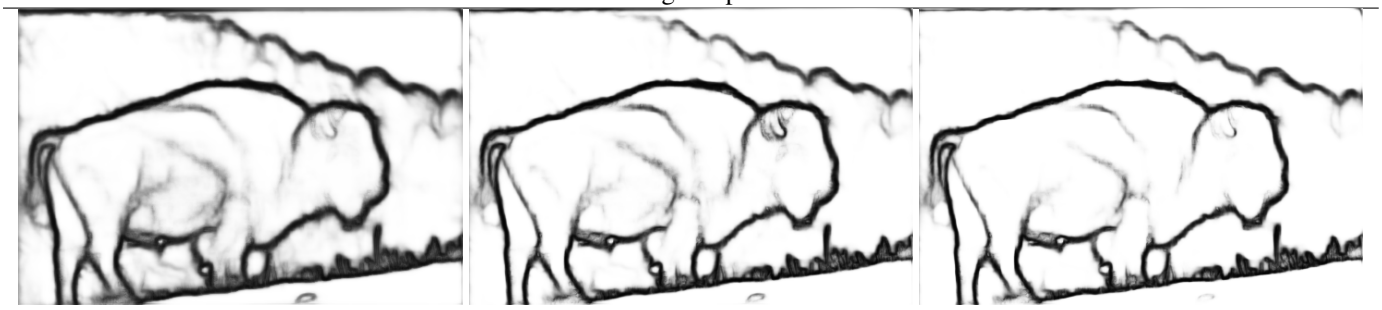


UCM Segmentations (Threshold=0.5)

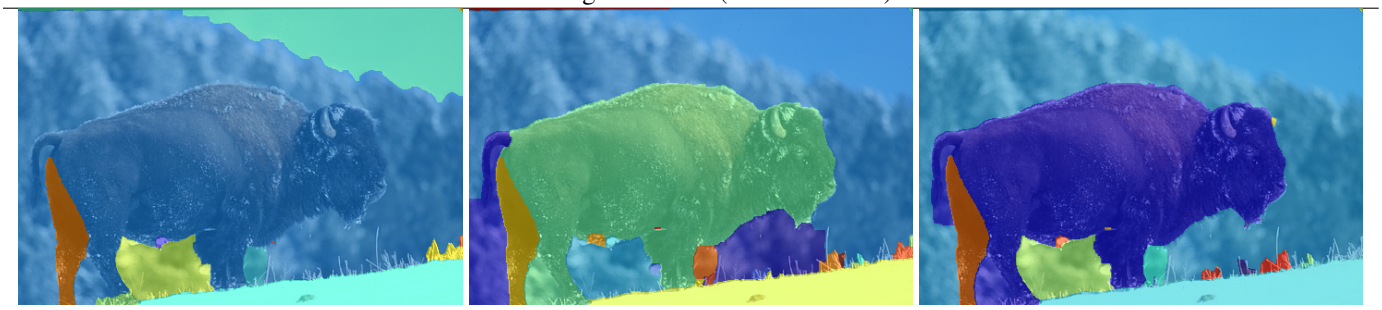


Baseline RCF | Baseline CRF | Adaptive CRF

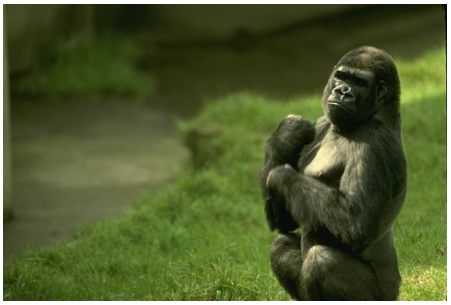
Edge maps



UCM Segmentations (Threshold=0.5)

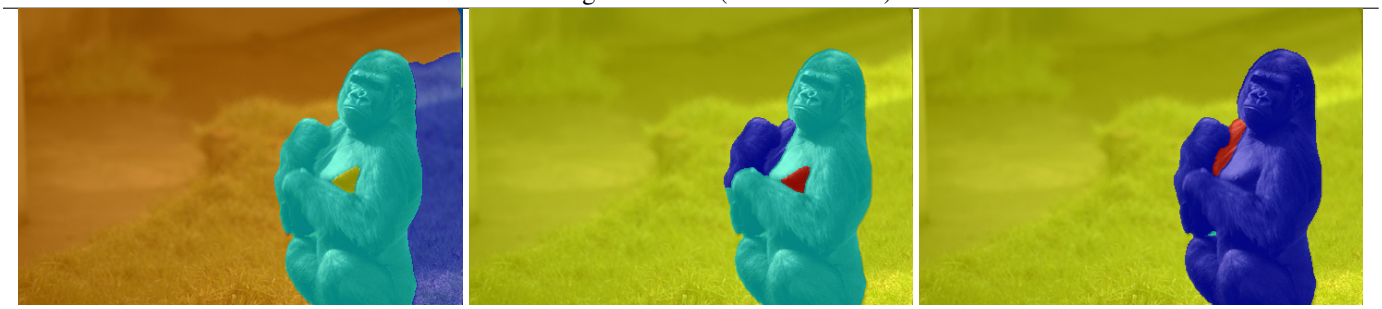


Baseline RCF | Baseline CRF | Adaptive CRF



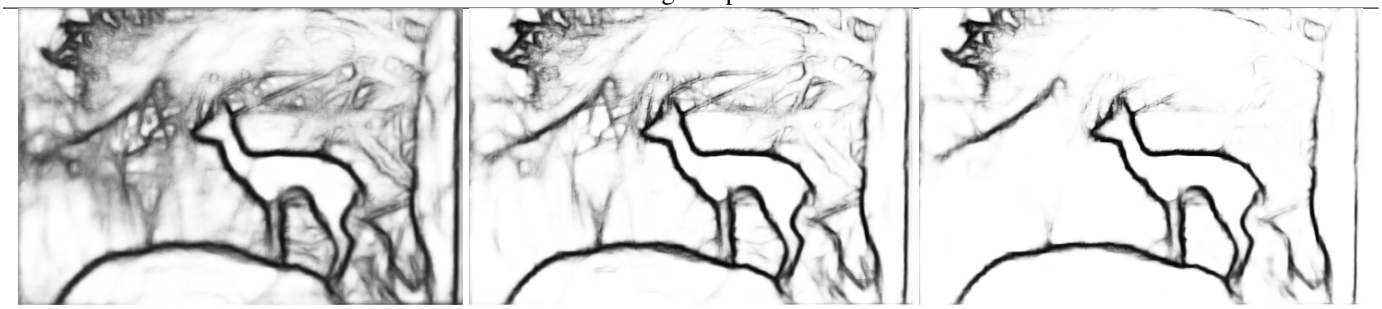
Edge maps

UCM Segmentations (Threshold=0.5)

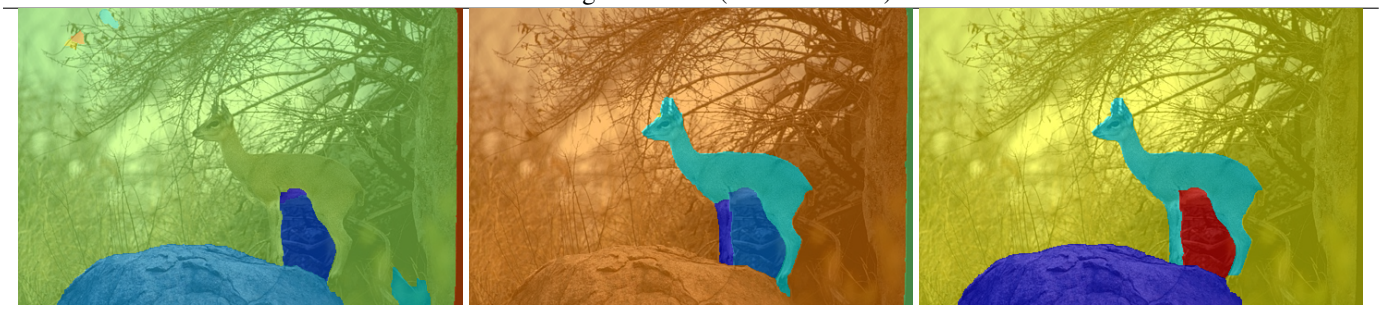


Baseline RCF | Baseline CRF | Adaptive CRF

Edge maps



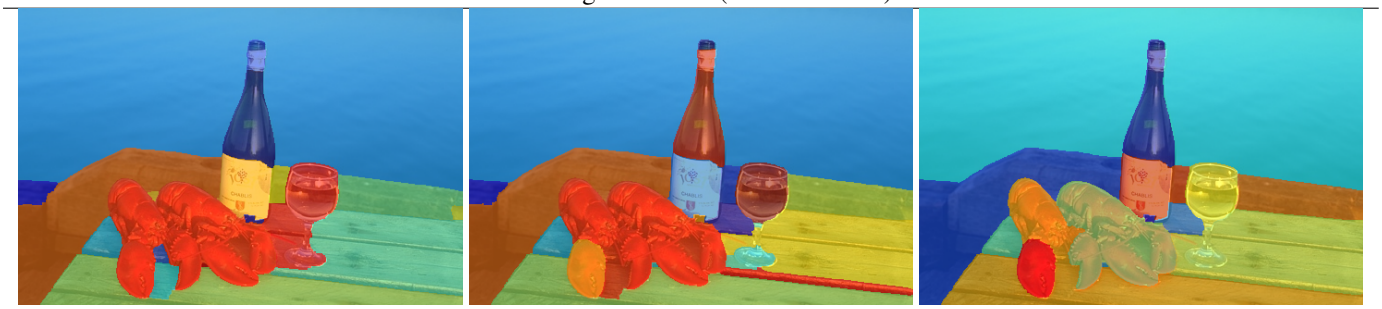
UCM Segmentations (Threshold=0.5)



Baseline RCF | Baseline CRF | Adaptive CRF

Edge maps

UCM Segmentations (Threshold=0.5)



Baseline RCF | Baseline CRF | Adaptive CRF