22^{nd} *Computer Vision Winter Workshop*
Nicole M. Artner, Ines Janusch, Walter G. Kropatsch (eds.)
Retz, Austria, February 6–8, 2017

End-to-End Training of Hybrid CNN-CRF Models for Semantic Segmentation using Structured Learning

Aleksander Colovic[†]
aleksander.colovic@student.tugraz.at

Patrick Knöbelreiter[†]
knoebelreiter@icg.tugraz.at

Alexander Shekhovtsov[†]
shekhovtsov@icg.tugraz.at

Thomas Pock[†,‡]
pock@icg.tugraz.at

[†]Institute for Computer Graphics and Vision
Graz University of Technology

[‡] Digital Safety & Security Department
AIT Austrian Institute of Technology

Abstract. *In this work we tackle the problem of semantic image segmentation with a combination of convolutional neural networks (CNNs) and conditional random fields (CRFs). The CRF takes contrast sensitive weights in a local neighborhood as input (pairwise interactions) to encourage consistency (smoothness) within the prediction and align our segmentation boundaries with visual edges. We model unary terms with a CNN which outperforms non data driven models. We approximate the CRF inference with a fixed number of iterations of a linear-programming relaxation based approach. We experiment with training the combined model end-to-end using a discriminative formulation (structured support vector machine) and applying stochastic subgradient descend to it.*

Our proposed model achieves an intersection over union score of 62.4 in the test set of the cityscapes pixel-level semantic labeling task which is comparable to state-of-the-art models.

1. Introduction

The task of semantic segmentation is to compute a class label for each pixel in the image. Classes can be e.g., *car*, *street*, *pedestrian*, *etc*. Solving this problem with hand-crafted methods is hard. However, recent improvements in computer hardware and the seminal work of Krizhevsky *et al.* [14] made it possible to train deep convolutional neural networks (CNNs) with millions of parameters on graphic processing units (GPUs). Since then, CNN-based approaches significantly improved the performance compared to hand-crafted methods in various computer vision tasks.

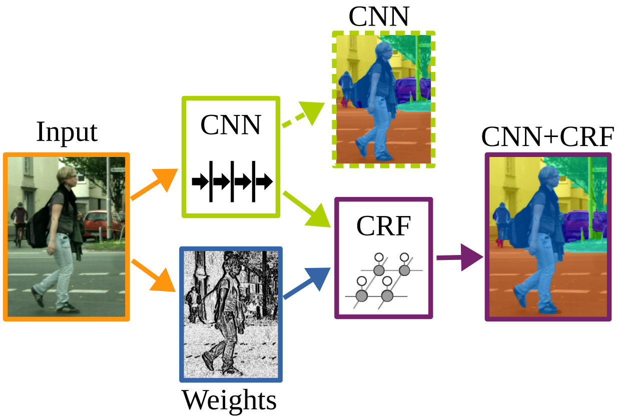


Figure 1. Abstract representation of our model containing its building blocks. The input is used to compute weights for the pairwise interactions in the CRF. Simultaneously a CNN computes unary potentials which are used in the CRF to produce the final output. No further pre- or post-processing is performed.

State of the art CNN architectures commonly consist of pooling layers. The advantages of pooling layers are i) the computation of descriptions of the image content at different levels of abstraction and ii) the introduced invariance to spatial deformations. However, exactly these pooling layers make it more difficult to apply these network architectures to problems requiring dense outputs such as semantic segmentation. Simply upscaling the coarse solution is often visually not pleasing due to blurred edges. Therefore these models are extended to keep spatial information [18] or to directly learn a refinement [6].

CNNs are known to be very good in extracting features from images and conditional random fields

(CRF) are often used to incorporate prior knowledge in terms of a regularizer. In this work we investigate a hybrid CNN+CRF architecture to combine the best of both worlds. Instead of using the CRF as a post-processing step for the output of the CNN, we are tackling the challenging problem of training this CNN+CRF combination jointly. We want to emphasize that we do not use the well known fully connected CRF of [12], but the linear-programming relaxation based approach of [25], due to its efficiency. We show how to compute a gradient for training our hybrid model jointly using a structured output support vector machine (SSVM) approach in a non-linear setting.

2. Related Work

The present work uses the hybrid architecture and the training method similar to [11], where authors applied it to the stereo reconstruction problem. The setting in this paper is somewhat different in the CRF model interactions (classification versus depth), CNN architecture (deep model versus shallow in [11]) and the final application loss (intersection over union versus truncated l_1).

Our model consists of two main building blocks. First, we use a CNN to extract suitable features for semantic segmentation from data. The output of the CNN is then used as *unary* costs in a CRF formulation. In a final step, the CRF optimizes the joint energy of the data-cost and a consistency-enforcing smoothness prior.

Convolutional Neural Networks CNNs are among the top performing models for computer vision tasks like image classification [14, 26], object detection [22, 7, 17] and semantic segmentation [16, 3, 18, 32] for the last couple of years. Backed by steadily increasing computational power learning models with millions of parameters became manageable. In image classification, striding or pooling is used to create abstract, low resolution representations of images to obtain a single label per input image in the end. In contrast to classification, the difficulty in semantic segmentation is to assign a label to every pixel in an image to obtain a so called *dense prediction*. Losing spatial information throughout the model makes correct predictions at object boundaries difficult. Nonetheless, abstract representations are important and many approaches to semantic segmentation use a network trained for classification such as [26] as a starting point [6, 16, 3, 31, 18, 19, 30, 32].

Deep CNNs mostly use pooling or striding at consecutive layers of the network to obtain abstract and spatial invariant representations. For dense predictions, it is a common approach to use upsampling or deconvolution to reconstruct high resolution predictions [18]. In contrast to that, *Atrous convolutions* (or later called dilated convolutions [31]) were originally developed for efficient computation of the undecimated wavelet transform [9]. By filling filters with holes (*trous* in french) this enables large receptive fields without increasing the number of parameters. This approach was used by [24, 20, 8] to obtain dense features via a CNN. Using multi-scale context, [3, 31] applied spatial pyramid pooling to further increase the performance of such models.

Conditional Random Fields CRFs are probabilistic models that incorporate relations between nodes in a graph [15]. In the most general case the graph is fully connected. The nodes represent discrete random variables, defined over the set of possible labels, that are conditioned on the input image. The edges in the graph (pairwise terms) model label consistency over node pairs. In the context of semantic segmentation, nodes typically correspond to image pixel or super-pixel.

Finding the best label for each node can be interpreted as an energy minimization problem.

To solve this problem for a fully connected graph, [12] reformulates the computation of the model as high dimensional filtering. This formulation is advantageous because it allows the filtering scheme [1] to be applied to the problem. This approach is often used as a post processing method, *e.g.*, for semantic segmentation [3].

Reducing the edges in the graph to a four-connected neighborhood, [25] provides a heavily parallelized GPU implementation. The methods inference is fast enough to be performed during the training stage instead of as post processing afterwards. We use [25] to incorporate the CRF into the learning procedure and train our model in an end-to-end fashion. By doing so, the preceding CNN can adopt to the capabilities of the CRF and hence provide richer feature representations.

End-to-End Learning The hybrid training of CNNs and CRFs for structured predictions has been

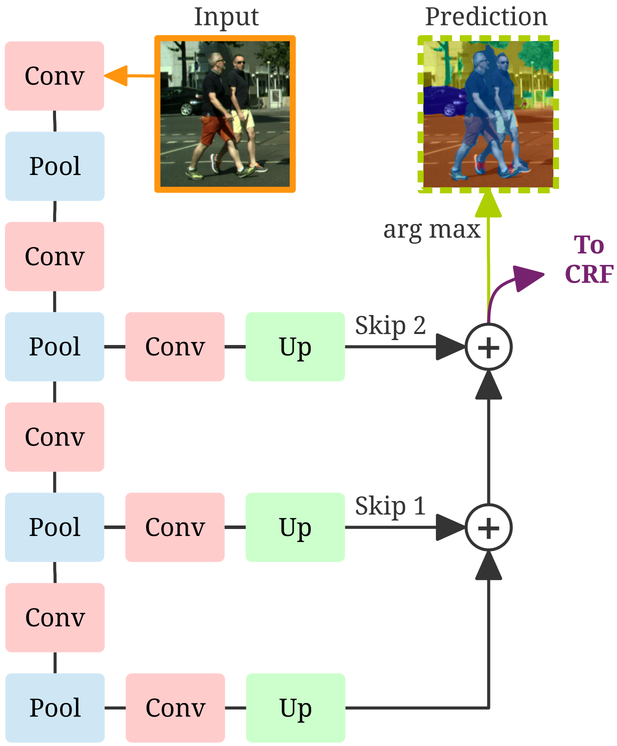


Figure 2. Simplified visualization of the unary network. The model produces a predicted label assignment as well as unary input to the subsequent CRF. *Conv* boxes represent multiple convolutions while *Pool* indicates max-pooling. *Up* are upsampling stages that rescale the predictions to the original image resolution.

explored by others as well in recent history. Lin *et al.* [16] approximate the gradient of a low resolution CRF using the piecewise training method [27]. In piecewise training, the marginal probability modeled in a CRF is approximated with the product of individual potentials which allows for a simpler gradient computation.

Zheng *et al.* [32] unrolled the mean field iterations of [12] in a recurrent neural network. They did this by expressing a single iteration in the mean field inference of their CRF as a sequence of standard convolutions. This approach requires the pairwise terms to be modeled with high dimensional Gaussian filters but in turn allows for training their model with the exact gradient through back-propagation. This is in contrast to our approach where we use a SSVM formulation in a nonlinear setting to approximate the gradient of the CRF.

3. Model

We combine deep CNNs with CRFs to pair good semantic reasoning with sharp prediction-borders. Furthermore we refrain from using a CRF as a post processing method and rather include it during training in an end-to-end fashion. The CRF takes unary and pairwise terms as its input. The unary terms consist of per-pixel independent data-costs and the pairwise terms guide the smoothing of the final output.

The main idea is that in a final learning phase, the CNN is *not* trained to predict labels directly but receives error gradients from the CRF which does the labeling instead. As a result, the CNN can move some of its responsibilities regarding semantic segmentation (*e.g.*, sharp boundaries) to the CRF. This procedure allows the CNN to deviate from a model that predicts a labeling directly towards providing richer features for the CRF hence increasing the overall performance.

Figure 1 shows an abstraction of our used model. The input to the CRF is composed of unary- and pairwise terms. The unary input is generated through a fully convolutional network [18]. The pairwise interactions are modeled with a nonlinear transformation of the image gradients (contrast sensitive weighting). These pairwise interactions encourage label jumps at strong object boundaries and discourage them within objects. The resulting segmentation is then more likely to align with edges in the image.

3.1. Unary Network

We build on the CNN architecture of [18]. We call this network *unary network*, because it computes the unary- or data costs in our CRF formulation (defined in (2)). The network is based on the *VGG16* net [26] and includes additional skip-connections and upsampling layers in order to achieve fine grained predictions. To make this section self-contained, we briefly review the architecture of this network and its building blocks.

Figure 2 shows the architecture of the unary network. *Conv* blocks indicate convolutional layers which are at the core of CNNs. They are used to extract learnable feature-maps from the given input. The output of each convolution layer is activated using the *ReLu* function. Stacking multiple layers together allows to learn features of different abstraction. Convolution can be seen as filtering the data with a (typically small) kernel. As a result, each output neuron only sees a small subset of the input called *receptive field*. *Pool* blocks indicate max-pooling layers. They reduce the spatial resolution by aggregating information of nearby pixels. This allows the network to learn hierarchical features at different levels of detail. The representation at the end of the lowest branch has passed the most pooling layers and therefore has the lowest spatial resolution. By upsampling that representation using an *up*-layer we obtain a full scale prediction with poor segmentation borders but high certainty within objects. Following [18] we add skip branches to the network. As seen in figure 2, these branches forgo pooling operations to preserve finer details at the cost of having a less abstract (and therefor less noise resistant) representation of the data. They guide the low resolution prediction along object boundaries.

3.2. Conditional Random Fields

While in a fully convolutional network the relation between two output neurons is given by the overlap in their receptive fields, CRFs allow to model this relation directly using pairwise interactions.

With the energy $f(x|I,\theta)$, which is defined over a set of variables x conditioned on an input image I and parameters θ, the CRF is a Gibbs distribution of the form

$$P(x|I,\theta) = \frac{1}{Z(I,\theta)} \exp\left(-f(x|I,\theta)\right), \quad (1)$$

where $Z(I,\theta)$ is the partition function [15]. Each variable represents a label from the set $\mathcal{L} = \{l_1, l_2, \dots, l_N\}$. The energy of a label assignment $x \in \mathcal{L}^{|\mathcal{V}|}$ is composed of unary terms ψ_i and pairwise interactions $\psi_{i,j}$ and can be written as

$$f(x|I,\theta) = \sum_{i\in\mathcal{V}} \psi_i(x_i) + \sum_{i,j\in\mathcal{E}} \psi_{i,j}(x_i, x_j), \quad (2)$$

where $\mathcal{V}$ is the set of variables in the input I and $\mathcal{E}$ is the set of edges between variables. Note that we use a four connected grid for $\mathcal{E}$ here, but in general an arbitrary set of connections is possible.

Finding the label assignment with the highest probability is equivalent to minimizing the CRF energy, which can be written as

$$\bar{x} \in \arg\min_x f(x|I,\theta). \quad (3)$$

In the energy model (2), ψ_i are *unary* terms which model each pixel individually. The unary terms are constructed from a probability distribution over all possible labels which is provided though the unary CNN[1]. With $p_i(x_i)$ being the x_i-th element of that probability distribution at location i, we define

$$\psi_i(x_i) = -p_i(x_i). \quad (4)$$

$\psi_{i,j}$ are *pairwise terms* that penalize label jumps. Using the Iverson bracket $[\![\cdot]\!]$ notation, we define the locally weighted Potts model [21] as

$$\psi_{i,j}(x_i, x_j) = w_{i,j}[\![x_i \neq x_j]\!], \quad (5)$$

where weights $w_{i,j}$ depend on the image I. If x_i and x_j are assigned the same label, $\psi_{i,j}(x_i, x_j)$ is zero and $w_{i,j}$ otherwise. We choose weights that discourage label jumps in homogeneous regions (high weight) and encourage them along borders (low weight). As a result, the solution of equation (3) is more likely to align its segmentation boundaries with visual edges in the input image I. In that sense we define *contrast sensitive weights* [2] as

$$w_{i,j} = \lambda \exp\left(-\alpha \|I_j - I_i\|^\beta\right), \quad (6)$$

where λ weights unary against pairwise terms and α and β are parameters of the model. Furthermore, we restrict the weights to be symmetric *i.e.*, $w_{i,j} = w_{j,i}$.

3.2.1 Gradient Estimation

In oder to train the full model, we need to propagate gradients of the loss functions though the CRF to the CNN. Despite our final evaluation criterion is intersection over union, we start by optimizing the Hamming loss. Given the predicted solution $\bar{x}$ and the ground truth labeling x^*, the Hamming loss is

$$l(\bar{x}, x^*) = \sum_{i\in\mathcal{V}} [\![\bar{x}_i \neq x_i^*]\!]. \quad (7)$$

The goal is to optimize this loss in parameters subject to $\bar{x}$ being a minimizer, *i.e.* satisfies (3). In a discrete setting, minimizing w.r.t. x is non-trivial since the solution only changes if the parameters pass certain breakpoints. Therefore the gradient is zero almost everywhere which makes training impossible. Instead we relax the problem to an upper bound of the loss and minimize that instead. This approach is known as structured support vector machine with margin rescaling [23, 29] and will be reviewed next.

Overview of SSVM Without further restrictions we consider a weighted loss $\gamma l(\bar{x}, x^*)$ hereafter.

The predicted label assignment $\bar{x}$ has the lowest energy $f(\bar{x})$ and is therefore either equal to the energy of the true label $f(x^*)$ (in the case where

[1] Note that this is *not* the distribution of equation (1)

$\bar{x} = x^*$) or smaller than that energy. Keeping that in mind, we obtain an upper bound $\bar{l}(x^*)$ from

$$\gamma l(\bar{x}, x^*) \leq \max_{x:f(x)\leq f(x^*)} \gamma l(x, x^*) \tag{8a}$$

$$\leq \max_{x:f(x)\leq f(x^*)} \gamma l(x, x^*) + f(x^*) - f(x) \tag{8b}$$

$$\leq \max_{x} \gamma l(x, x^*) + f(x^*) - f(x) \tag{8c}$$

$$= \bar{l}(x^*). \tag{8d}$$

The value $\bar{l}(x^*)$ can be also equivalently written as finding the minimum ξ such that

$$\forall x : \gamma l(x, x^*) + f(x^*) - f(x) \leq \xi, \tag{9}$$

which reveals the margin property, *i.e.*, in the separable case the energy $f(x^*)$ should be smaller than any other energy by at least $\gamma l(x, x^*)$. If this is not possible, then the statement is relaxed by the slack variable ξ. Here, γ weights the margin against the energy barrier.

A subgradient of $\bar{l}(x^*)$ w.r.t. the energy volume at location i, $f_i(x_i)$ is given by

$$\frac{\partial}{\partial f_i(x_i)} \bar{l}(x^*) = [\![x_i^* = x_i]\!] - [\![\hat{x}_i = x_i]\!], \tag{10}$$

where $\hat{x} \in \arg\min_x f(x) - \gamma l(x, x^*)$ is a solution (among possibly many) of the *loss augmented inference* problem. Since we can perform inference only approximately, we take $\hat{x}$ to be an approximate solution resulting after a fixed number of iterations.

As a result of minimizing the upper bound of the loss function, we eventually also lower the actual loss. Another way of interpreting this is that we repeatedly increase the energy of whichever label violates the statement in equation (9) the most while reducing the energy of the correct label at the same time.

4. Training

We use gradient descent to train our model. Therefore we estimate the gradient of a CRF and propagate it back though the network. The method described in section 3.2.1 gives a fixed magnitude gradient which means that the learning rate has to be small in order to reach good local minima in the loss function. As a result, we need to pretrain our unary model before we can jointly train the full model.

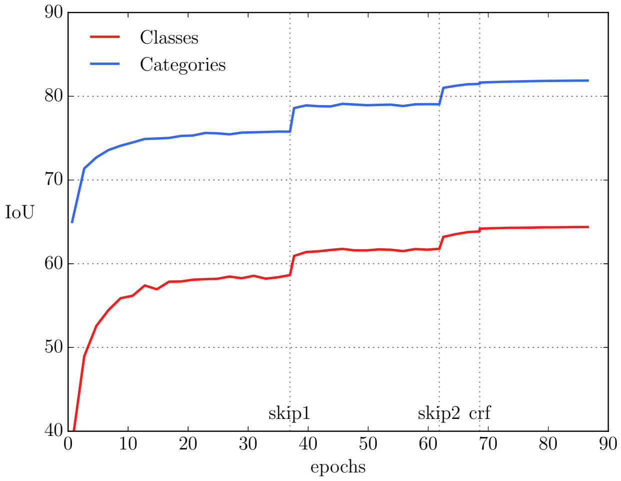


Figure 3. Training progress evaluated on the validation set. The IoU is printed for classes and categories (groups of classes). Vertical lines mark the beginning of a new training stage. *Skip* indicates that additional skip branches are added. *CRF* indicates the joint training of CNN and CRF.

4.1. Pretraining the Model

Due to the fixed magnitude of the gradients in equation (10) and the consequential slow convergence rate, training our model from scratch is impracticable. We therefore consider the prominent *VGG16* model of [26] as a reasonable initialization for our purpose. The model is then modified following [18] to be suitable for semantic segmentation. By doing so we transform the last fully connected layers to convolutional ones and add two *skip*-branches as indicated in figure 2. We use the training parameters of the *Pascal* model of [18] but adjust the unnormalized learning rate to match the larger images in the dataset. We then use stochastic gradient descend to pretrain our unary model (no pairwise interactions in the CRF) in three stages. At first we do not use any skip branches and train the low resolution prediction. Then we add one layer at a time and repeat the fine tuning. Note that we subsample the data by a factor of 2 due to hardware limitations.

In a final training stage, we jointly train CNN and CRF. To that end we use the gradient of section 3.2.1 and update parameters with a fixed and normalized learning rate of 1×10^{-10} to account for the previously outlined gradient magnitude.

Figure 3 shows the training progress by displaying the intersection over union (IoU; see *Score* in section 5.1) over the training epochs. The score is evaluated in the validation set (which was not used for learning) of the *Cityscapes* dataset (see section 5.1). Each stage is trained until convergence. The start of the next stage is marked as a vertical line in the plot.

4.2. Incorporating Pairwise Interactions

We use the linear programming relaxation based method of [25] (`Dual_MM`) for CRF inference. This method performs a dual block-coordinate descent algorithm whose main advantage is its massively parallel implementation[2]. With a small number of iterations (15 in our case), [25] provides results that are comparable to the best sequential algorithms.

To find an suitable parameter setting for equation (6), we perform an exhaustive grid search. Then our network is jointly trained by back-propagating the gradients (equation (10)). For this, the loss augmented inference problem has to be solved for each training iteration. We can see the slight performance gain at the vertical *crf* line in figure 3. Although the benefits might seem small, section 5 shows that this is still a valuable improvement over the CNN alone.

5. Experiments

Using the *Caffe* [10] and *Theano* framework [28] to implement our models, we investigate the impact of CRFs and joint training on semantic segmentation. We consider two models: our unary- and combined network. We fixate the following hyper-parameter of our model for all conducted experiments: $\alpha = 35$, $\beta = 0.9$, $\lambda = 2.5$ and $\gamma = 0.1$. In the following, we give a qualitative comparison of both models.

5.1. Benchmark

We use the *Cityscapes Dataset* [4] to train and evaluate our approach. This recently released database consists of 5.000 images with high quality fine annotations. The samples were taken in 50 different cities throughout the year to reduce biases in the data. The set is split into sets of size 2.950, 500, 1.250 for training, validation and test respectively. The 19 different classes in the database represent various things *e.g.*, *cars, humans, streets* or *vegetation*. Additionally the dataset provides 20.000 images with coarse annotations which are not used in this work.

	Cityscapes Dataset			Intersection over Union	
Method	Coarse	Stereo	Half Res.	Classes	Categories
CNN only			∗	*63.8*	*81.5*
CNN+CRF			∗	*64.1*	*81.2*
CNN+CRF (joint)			∗	*64.4*	*81.9*
Ours			∗	62.4	82.3
[19]	∗		∗	64.8	81.3
DeepLab [3]			∗	63.1	81.2
CRF as RNN [32]			∗	62.5	82.7
FCN 8s [18]			∗	61.9	-
LRR-4x [6]	∗			71.8	88.4
Adelaide [16]				71.6	51.7
DeepLab [3]				70.4	86.4
Dilation10 [31]				67.1	86.5
[13]		∗		66.3	85.0
FCN 8s [18]				65.3	85.7
[30]		∗		64.3	85.9

Table 1. Comparison of different approaches to the pixel level semantic segmentation task on the cityscapes dataset [4]. *Ours* indicates the score of the jointly trained CNN+CRF model in the test set. *Italicized* methods were evaluated in the validation set, other ones in the test set. The results for other models on this data were taken from their respective paper if available or from [4] otherwise. The score for *FCN 8s* [18] on half sized images was taken from the supplementary material of [4] where no IoU score for categories is listed. See section 5.1 for a detailed discussion of this table.

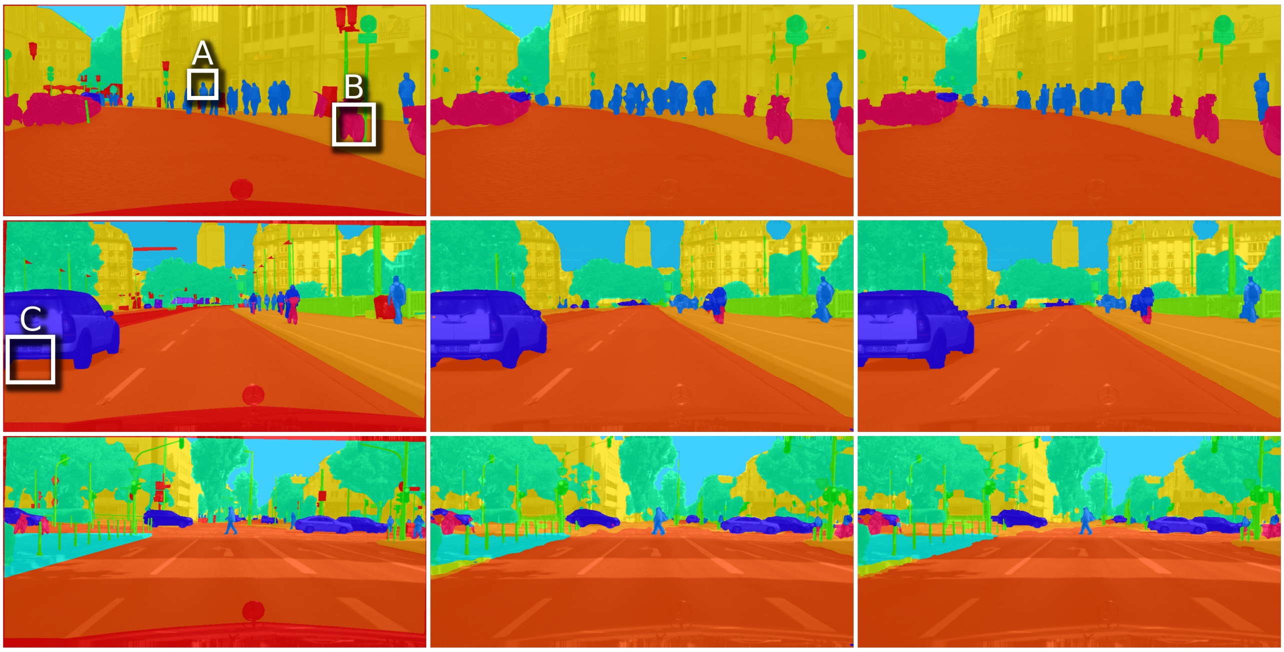
Score The primary measurement of prediction quality on the dataset is the PASCAL VOC intersection-over-union (IoU) [5] which is also known as the Jaccard index. The measure is defined with true-positive- (TP), false-positive- (FP) and false-negative (FN) pixel predictions for each class as

$$IoU = \frac{TP}{TP + FP + FN}. \tag{11}$$

The final score is then computed as the average of the individual class scores. This metric penalizes false-positive predictions *i.e.*, classifying all pixel as car, as opposed to only considering the average classification rate. Note that *IoU Classes* is computed as the average over individual classes while *IoU Categories* denotes the average over groups of classes *e.g.*, *vehicle* = $\{$*car, bus, bicycle, . . .*$\}$.

Results Table 1 shows the score of our approach on that dataset and compares it to current state-of-the-art models. *CNN only* is our unary network which disregards pairwise interactions. *CNN+CRF* indicates our

[2] Implementation provided by the authors of [25].



(a) Groundtruth

(b) CNN

(c) CNN + CRF

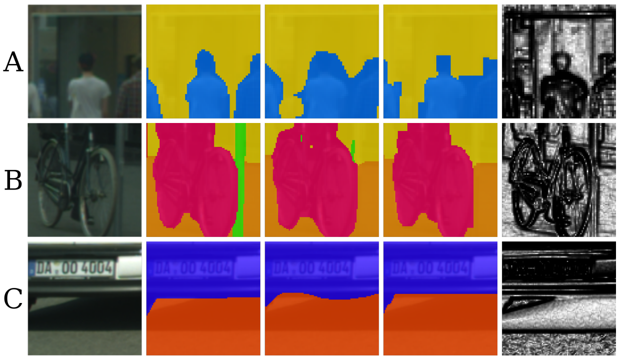
Figure 4. Comparison of the CNN only model and the combination of CNN and CRF. All label assignments are blended over the input image to better show the errors around object borders. Column (a) contains the ground-truth image (note that red indicates the ignore-label). Column (b) displays the results obtained without a CRF layer while the last column shows the results of the joint model (CNN followed by a CRF). For better visualization we marked four regions (A to C) and enlarged them in figure 5. Best viewed in color on a screen.

combined model of CNN and CRF. The lines containing *joint training* in brackets indicates that the model was jointly trained as compared to just using the CRF for post processing. Lines containing *validation* were evaluated locally on the validation set while the *test* line is taken from the public test servers of [4]. Columns *coarse* and *stereo* indicate if a model additionally used weakly labeled (coarse) or stereo data for training respectively. There is an obvious gap between models operating on half sized images and ones that use the full data resolution. We expect that retraining our model on full scale images will give a similar performance increase. We want to validate that in future work.

5.2. A Comparison of our Models

To illustrate the benefits of the combined model over the unary network, we provide a comparison of our models. In the first group of models in table 1, we compare the three stages of our approach (unary network, CRF for post processing, joint training) on the validation set. We can observe that the use of a CRF as well as the joint training steadily increase the score for the *IoU Classes* task.

In the following, we provide a qualitative comparison of the unary network and the trained combined model based on a few exemplary predictions. Figure 4 shows the desired groundtruth (a) as well as predictions of the unary network (b) and the combined model (c). We mark four regions (A to C) which are enlarged in figure 5 and further discussed next. In figure 5 we additionally print the pairwise interactions in column (e). Hereby, dark regions correspond to low energy which encourages label jumps while bright regions indicate high energy and therefore discourages label discontinuities. For instance the weights in row A show low cost for a label jump around the head of the person in the center. As a result, the CRF energy is lower in that region and the segmentation boundary is moved towards it. Note that the low cost region is wide enough to allow multiple boundaries. Due to the four connected neighborhood in our CRF implementation, the model tends to prefer grid-aligned solutions. We believe that by extending the neighborhood of the CRF to model diagonal or sparse long range connections can lower that tendency. Row B shows similar effects around the front wheel of the bicycle. Additionally it contains an example of how the CRF enforces consistency (smoothness) of the solution. We can observe



(a) Input (b) Groundtruth (c) CNN (d) CNN+CRF (e) Weights

Figure 5. Scaled versions of marked regions in figure 4. The first three columns contain the groundtruth labeling, the prediction of the CNN model and the results of our CNN+CRF model respectively. The last column shows the pairwise costs of neighboring pixel. Best viewed in color.

that small outliers (visible in the CNN prediction within the pink region) are removed because the corresponding pairwise costs are too high. Finally row C contains an example of how fattened foreground and curvaceous boundaries become aligned with the true objects in the combined model.

5.3. Comparison with the State-of-the-Art

To justify our usage of the locally connected CRF implementation of [25] over the commonly used fully connected CRF of [12], we compare our model against the work of [32]. The authors of [32] also use the FCN-8s network of [18] to provide unary terms to the fully connected CRF of [12] which is unrolled and interpreted as layers in a recurrent network. Additionally they also operate on half sized images. An advantage of that approach is that it allows to directly compute the exact gradient instead of an approximation. Their model is restricted to formulate the pairwise interactions as high dimensional Gaussian filters. In contrast to that, our model approximates the gradient of the CRF using a SSVM formulation which in turn allows for arbitrary definition of pairwise potentials. Table 1 shows that both approaches achieve similar improvements over the FCN-8s model and yield almost identical IoU scores, even though our model is simpler.

6. Discussion

In this paper, we showed how to train a combination of CNN and CRF for semantic image segmentation using a SSVM formulation. The joint training allowed the CNN to adapt to the CRFs capabilities and hence improved the quality of the output. Pairwise interactions encouraged label discontinuities across visual edges in the input image. The predicted segmentation boundaries were more likely to align with the true object borders because of that.

In future work we want to replace gradient-based pairwise weights with a second CNN. We believe that contrast sensitive weights have room for improvement and that moving pairwise interactions towards penalizing label jumps within objects is reasonable. Additionally we plan to extend the current Potts model to general label compatibilities such that *e.g.*, the label pair $\{car, street\}$ is more likely than $\{car, bicycle\}$. We also believe that changing the neighborhood for pairwise interactions in the CRF to model non-symmetric, diagonal and/or sparse long-range interactions is beneficial for semantic segmentation. Another interesting direction is to compare different training methods and different loss surrogates, possibly addressing IoU score more directly, within the SSVM approach.

References

[1] A. Adams, J. Baek, and M. A. Davis. Fast high-dimensional filtering using the permutohedral lattice. In *Computer Graphics Forum*, volume 29, pages 753–762. Wiley Online Library, 2010. 2

[2] Y. Y. Boykov and M.-P. Jolly. Interactive graph cuts for optimal boundary & region segmentation of objects in nd images. In *Computer Vision, 2001. ICCV 2001. Proceedings. Eighth IEEE International Conference on*, volume 1, pages 105–112. IEEE, 2001. 4

[3] L.-C. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and A. L. Yuille. Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs. *arXiv preprint arXiv:1606.00915*, 2016. 2, 6

[4] M. Cordts, M. Omran, S. Ramos, T. Rehfeld, M. Enzweiler, R. Benenson, U. Franke, S. Roth, and B. Schiele. The cityscapes dataset for semantic urban scene understanding. In *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016. 6, 7

[5] M. Everingham, S. A. Eslami, L. Van Gool, C. K. Williams, J. Winn, and A. Zisserman. The pascal visual object classes challenge: A retrospective. *International Journal of Computer Vision*, 111(1):98–136, 2015. 6

[6] G. Ghiasi and C. Fowlkes. Laplacian reconstruction and refinement for semantic segmentation. *arXiv preprint arXiv:1605.02264*, 2016. 1, 2, 6

[7] R. Girshick, J. Donahue, T. Darrell, and J. Malik. Region-based convolutional networks for accurate object detection and segmentation. *IEEE transactions on pattern analysis and machine intelligence*, 38(1):142–158, 2016. 2

[8] A. Giusti, D. C. Cireşan, J. Masci, L. M. Gambardella, and J. Schmidhuber. Fast image scanning with deep max-pooling convolutional neural networks. *arXiv preprint arXiv:1302.1700*, 2013. 2

[9] M. Holschneider, R. Kronland-Martinet, J. Morlet, and P. Tchamitchian. A real-time algorithm for signal analysis with the help of the wavelet transform. In *Wavelets*, pages 286–297. Springer, 1990. 2

[10] Y. Jia, E. Shelhamer, J. Donahue, S. Karayev, J. Long, R. Girshick, S. Guadarrama, and T. Darrell. Caffe: Convolutional architecture for fast feature embedding. *arXiv preprint arXiv:1408.5093*, 2014. 6

[11] P. Knöbelreiter, C. Reinbacher, A. Shekhovtsov, and T. Pock. End-to-end training of hybrid cnn-crf models for stereo. *arXiv preprint arXiv:1611.10229*, 2016. 2

[12] P. Krähenbühl and V. Koltun. Efficient inference in fully connected crfs with gaussian edge potentials. *Adv. Neural Inf. Process. Syst*, 2011. 2, 3, 8

[13] I. Krešo, D. Čaušević, J. Krapac, and S. Šegvić. Convolutional scale invariance for semantic segmentation. In *German Conference on Pattern Recognition*, pages 64–75. Springer, 2016. 6

[14] A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pages 1097–1105, 2012. 1, 2

[15] J. Lafferty, A. McCallum, and F. Pereira. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *Proceedings of the eighteenth international conference on machine learning, ICML*, volume 1, pages 282–289, 2001. 2, 4

[16] G. Lin, C. Shen, I. Reid, et al. Efficient piecewise training of deep structured models for semantic segmentation. *arXiv preprint arXiv:1504.01013*, 2015. 2, 3, 6

[17] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, and S. Reed. Ssd: Single shot multibox detector. *arXiv preprint arXiv:1512.02325*, 2015. 2

[18] J. Long, E. Shelhamer, and T. Darrell. Fully convolutional networks for semantic segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3431–3440, 2015. 1, 2, 3, 4, 5, 6, 8

[19] G. Papandreou, L.-C. Chen, K. P. Murphy, and A. L. Yuille. Weakly-and semi-supervised learning of a deep convolutional network for semantic image segmentation. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1742–1750, 2015. 2, 6

[20] G. Papandreou, I. Kokkinos, and P.-A. Savalle. Modeling local and global deformations in deep learning: Epitomic convolution, multiple instance learning, and sliding window detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 390–399, 2015. 2

[21] R. B. Potts. Some generalized order-disorder transformations. In *Mathematical proceedings of the cambridge philosophical society*, volume 48, pages 106–109. Cambridge Univ Press, 1952. 4

[22] S. Ren, K. He, R. Girshick, and J. Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. In *Advances in neural information processing systems*, pages 91–99, 2015. 2

[23] B. T. C. G. D. Roller. Max-margin markov networks. *Advances in neural information processing systems*, 16:25, 2004. 4

[24] P. Sermanet, D. Eigen, X. Zhang, M. Mathieu, R. Fergus, and Y. LeCun. Overfeat: Integrated recognition, localization and detection using convolutional networks. *arXiv preprint arXiv:1312.6229*, 2013. 2

[25] A. Shekhovtsov, C. Reinbacher, G. Graber, and T. Pock. Solving dense image matching in real-time using discrete-continuous optimization. *arXiv preprint arXiv:1601.06274*, 2016. 2, 6, 8

[26] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014. 2, 3, 5

[27] C. Sutton and A. McCallum. Piecewise training for undirected models. *arXiv preprint arXiv:1207.1409*, 2012. 3

[28] Theano Development Team. Theano: A Python framework for fast computation of mathematical expressions. *arXiv e-prints*, abs/1605.02688, May 2016. 6

[29] I. Tsochantaridis, T. Joachims, T. Hofmann, and Y. Altun. Large margin methods for structured and interdependent output variables. *Journal of Machine Learning Research*, 6(Sep):1453–1484, 2005. 4

[30] J. Uhrig, M. Cordts, U. Franke, and T. Brox. Pixel-level encoding and depth layering for instance-level semantic labeling. *arXiv preprint arXiv:1604.05096*, 2016. 2, 6

[31] F. Yu and V. Koltun. Multi-scale context aggregation by dilated convolutions. *arXiv preprint arXiv:1511.07122*, 2015. 2, 6

[32] S. Zheng, S. Jayasumana, B. Romera-Paredes, V. Vineet, Z. Su, D. Du, C. Huang, and P. H. Torr. Conditional random fields as recurrent neural networks. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1529–1537, 2015. 2, 3, 6, 8