Detection and Instance Segmentation of Neuroblastoma Cells using YOLO, UNet and Conditional Random Fields

Kristofer delas Peñas and Dominic Waithe

***Abstract*—The abstract goes here.**

***Index Terms*—IEEE, IEEEtran, journal, LaTeX, paper, template.**

I. Introduction

THIS demo file is intended to serve as a "starter file" for IEEE journal papers produced under LaTeX using IEEEtran.cls version 1.8b and later. I wish you the best of success.

II. Literature Review

A. Neuroblastoma

B. YOLO

C. UNet

D. Conditional Random Fields

Many learning problems can be described as a graphical model. In artificial intelligence, one common way to model these problems is a probabilistic approach wherein probability distributions Ψ are assigned over the random variables Y. Formally, the problem can be modelled as the product of all Ψ_i distributions

$$p(Y) = \frac{1}{Z}\prod_{1}^{A}\Psi_a(y_a) \tag{1}$$

for factors F = $\{\Psi_a\}$ that have $\Psi_a \geq 0$.

Markov networks and conditional random fields (CRF) are formulated similarly in this manner. The main difference is that the CRF learns the conditional probability $p(y|x)$ while Markov networks ultimately obtain the joint probability $p(y, x)$. With the joint probability $p(y, x)$, models like the Markov networks can describe the hypothesis space through the generation of all possible features x for all labels y. However, joint probability $p(y, x)$ involves prior knowledge on or estimate for $p(x)$, and the dependence (or independence) of the random variables. For general classification tasks, however, modelling of $p(x)$ is not needed as the concern is only on assigning labels to features, which is exactly what the conditional probability $p(y|x)$ gives.

Exact inference on conditional random fields is computationally expensive and usually impossible. If exact inference is possible, performing naively the sum of products of potentials (**message passing**) for all variables, can take a long time, especially for dense graphs. Current implementations of CRFs perform inference by approximation, with the speedup by employing dynamic programming. One inference algorithm is the mean field inference described in Algorithm 1.

Each iteration of the mean field inference described in Algorithm 1 performs a message passing step, compatibility transform, and local update and normalization. Each step, except the message passing, runs in linear time. Message passing is the computational bottleneck. For a naive solution, it requires summing up over all variables and runs in quadratic time. For this very reason, the conditional random fields gained the reputation of being nooriously slow and impractical for a lot of machine learning tasks, especially those involving dense graph representations like image segmentation.

Addressing the slow inference in CRFs, [1] showed that message passing can be approximated and expressed as a convolution with a Gaussian kernel as follows:

$$\widetilde{Q}_i^{(m)}(l) \leftarrow \sum_{j\neq i} k^{(m)}(f_i, f_j)Q_j(l) = [G_(m) \otimes Q(l)](f_i) - Q_i(l) \tag{2}$$

This approximation can be extended to higher dimensions, and with the permutohedral lattice data structure, efficient message passing can be done in $O(Nd)$ time where N is the number of variables and d the number of dimensions. Furthermore, [2] and [3] demonstrated how the mean field inference in CRFs can be written as recurrent neural network with learnable weights. This formulation allows the CRFs to be seamlessly integrated as part of a convolutional neural network model for tasks such as image segmentation.

With CRFs implemented as RNNs, several research has been done applying CNN-CRFs for general image segmentation task. [1] and [4] proposed systems using CRFs integrated in CNNs for semantic image segmentation. Both systems train the CRF by minimizing the Gibbs energy and in doing so, finding the Maximum A Posteriori labelling of the image pixels. The CRF for semantic image segmentation is formulated with the following energy function for assignment of the pixels to semantic classes,

$$E(X = x) = \sum_i U(x_i) + \sum_{i<j} P(x_i, x_j) \tag{3}$$

where U is the unary potential and P the pairwise potential. In [1], responses from the TextonBoost filter bank, color, histogram of oriented gradients (HOG) and pixel location features were used for the unary potential. On the other hand, [4] used the segmentation prediction from ResNet101 for the unary. Both systems used gaussian kernels for the pairwise potentials, given as:

$$k(f_i^I, f_j^I) = w^{(1)} G_{appearance} + w^{(2)} G_{smoothness} \tag{4}$$

$$G_{appearance} = exp\left(-\frac{|p_i - p_j|^2}{2\theta_\alpha^2} - \frac{|I_i - I_j|^2}{2\theta_\beta^2}\right) \tag{5}$$

$$G_{smoothness} = exp\left(-\frac{|p_i - p_j|^2}{2\theta_\gamma^2}\right) \tag{6}$$

The two terms in the pairwise potential encourages neighboring pairs of pixels that look similar to be assigned the same semantic label.

Extending from this, [5] and [6] introduced some modification to this energy function to be able to perform pixelwise instance segmentation. The systems described for this task works on an initial instance segmentation from a detection algorithm, where each detection has a corresponding prediction score. This instance segmentation is further refined by adding information from semantic segmentation and the gaussian pairwise potentials. To do this, They have broken down the unary potential to accommodate two distinct terms Ψ_{box} and Ψ_{global} as follows:

$$U(x_i) = -ln[w_1 \Psi_{box}(x_i) + w_2 \Psi_{global}(x_i)] \tag{7}$$

The Ψ_{box} term encourages the pixel to be assigned to the instance corresponding to the detection. This is proportional to the probability of the semantic class assigned to the pixel and the detection score. The Ψ_{global} term accounts for pixels that might have been misclassified to another instance label but belongs to the same semantic label. This addresses the problem in the initial instance segmentation that does not fully cover the entire extent of the individual instances.

III. Conclusion

The conclusion goes here.

Appendix A
Proof of the First Zonklar Equation

Appendix one text goes here.

Appendix B

Appendix two text goes here.

Acknowledgment

The authors would like to thank...

References

[1] P. Krähenbühl and V. Koltun, "Efficient inference in fully connected crfs with gaussian edge potentials," in *Advances in Neural Information Processing Systems 24*, J. Shawe-Taylor, R. S. Zemel, P. L. Bartlett, F. Pereira, and K. Q. Weinberger, Eds. Curran Associates, Inc., 2011, pp. 109–117. [Online]. Available: http://papers.nips.cc/paper/4296-efficient-inference-in-fully-connected-crfs-with-gaussian-edge-potentials.pdf

[2] S. Zheng, S. Jayasumana, B. Romera-Paredes, V. Vineet, Z. Su, D. Du, C. Huang, and P. H. S. Torr, "Conditional random fields as recurrent neural networks," in *International Conference on Computer Vision (ICCV)*, 2015.

[3] A. Arnab, S. Jayasumana, S. Zheng, and P. H. S. Torr, "Higher order conditional random fields in deep neural networks," in *European Conference on Computer Vision (ECCV)*, 2016.

[4] M. T. T. Teichmann and R. Cipolla, "Convolutional crfs for semantic segmentation," *CoRR*, vol. abs/1805.04777, 2018.

[5] A. Arnab and P. H. S. Torr, "Pixelwise instance segmentation with a dynamically instantiated network," *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 879–888, 2017.

[6] Q. Li, A. Arnab, and P. H. Torr, "Weakly- and semi-supervised panoptic segmentation," in *The European Conference on Computer Vision (ECCV)*, September 2018.

PLACE
PHOTO
HERE

Michael Shell Biography text here.

John Doe Biography text here.

Jane Doe Biography text here.

Algorithm 1 Mean Field Inference

```
Initialize $Q$
while not converged do
    $\widetilde{Q}_i^{(m)}(l) \leftarrow \sum_{j \neq i} k^{(m)}(f_i, f_j) Q_j(l)$ for all m        ▷ Message Passing
    $\hat{Q}_i(x_i) \leftarrow \sum_{l \in L} \mu^{(m)}(x_i, l) \sum_w^{(m)} \widetilde{Q}_i^{(m)}(l)$
    $Q_i(x_i) \leftarrow exp{-}\psi_u(x_i) - \hat{Q}_i(x_i)$
    normalize $Q_i(x_i)$
end while
```