Efficient Inference in Fully Connected CRFs with Gaussian Edge Potentials

Philipp Krähenbühl and Vladlen Koltun

Stanford University

Presenter: Yuan-Ting Hu

Conditional Random Field (CRF)

$$E(x|I) = \sum_i \phi_u(x_i|I) + \sum_i \sum_{j \in \partial i} \phi_p(x_i, x_j|I)$$

Unary term　　　　　　Pairwise term

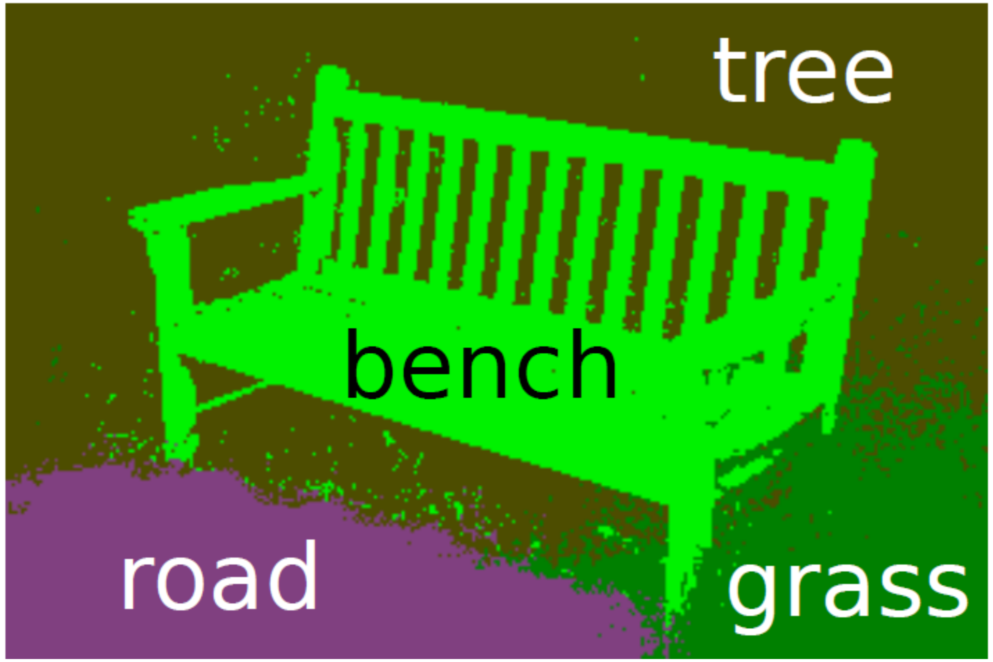
- X, I : random fields
- Application:
  - Image segmentation: achieve state-of-the-art performance (in 2011)

Image Segmentation

- Example: semantic image segmentation

Input

Output



CRF for Image Segmentation

$$E(x|I) = \underbrace{\sum_i \phi_u(x_i|I)}_{\text{Unary term}} + \underbrace{\sum_i \sum_{j \in \partial i} \phi_p(x_i, x_j|I)}_{\text{Pairwise term}}$$

- X: a random field defined over a set of variables $\{X_1, \dots, X_N\}$
  - Label of pixels (grass, bench, tree,..)
- I: a random field defined over a set of variables $\{I_1, \dots, I_N\}$
  - Image (observation)

CRF for Image Segmentation

$$E(x) = \sum_i \psi_u(x_i) + \sum_i \sum_{j \in \partial i} \psi_p(x_i, x_j)$$

Unary term Pairwise term

- Unary term
  - Trained on dataset

CRF for Image Segmentation

$$E(x) = \sum_i \psi_u(x_i) + \sum_i \sum_{j \in \partial i} \psi_p(x_i, x_j)$$

Unary term **Pairwise term**

- Pairwise term
  - Impose consistency of the labeling
  - Defined over neighboring pixels

$$\psi_p(x_i, x_j) = \mu(x_i, x_j) \sum_{m=1}^{K} w^{(m)} k^{(m)}(f_i, f_j)$$

CRF for Image Segmentation

$$E(x) = \sum_i \psi_u(x_i) + \sum_i \sum_{j \in \partial i} \psi_p(x_i, x_j)$$

Unary term **Pairwise term**

- Pairwise term

$$\psi_p(x_i, x_j) = \mu(x_i, x_j) \sum_{m=1}^{K} w^{(m)} k^{(m)}(f_i, f_j)$$

$k^{(m)}(f_i, f_j)$ is a Gaussian kernel

f_i, f_j is the feature vectors for pixel i and j, e.g., color intensities, …

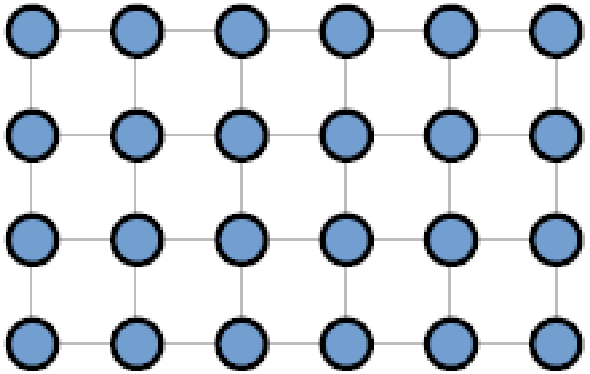
$w^{(m)}$ is the weight of the m-th kernel

$\mu(x_i, x_j)$ is the label compatibility function

CRF for Image Segmentation

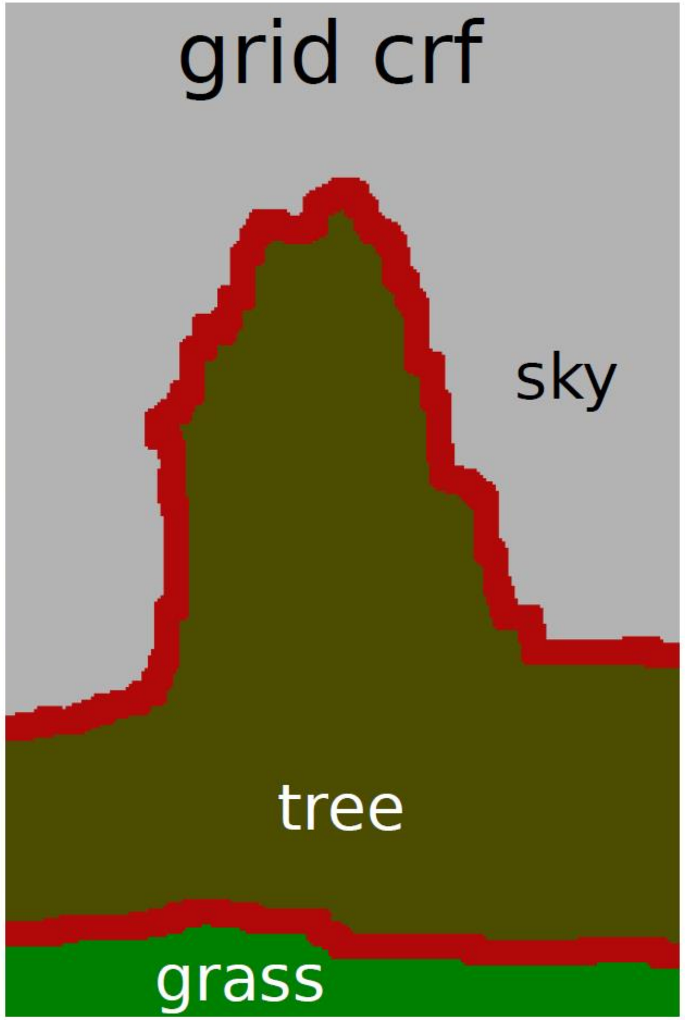
$$E(x|I) = \sum_i \phi_u(x_i|I) + \sum_i \sum_{j \in \partial i} \phi_p(x_i, x_j|I)$$

Unary term **Pairwise term**

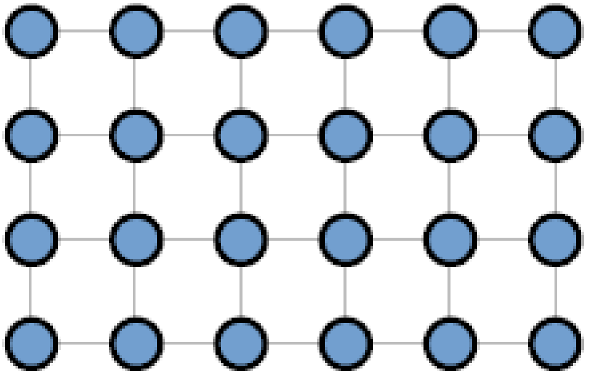


- Neighboring pixels
- Local connections
- May not capture the sharp boundaries

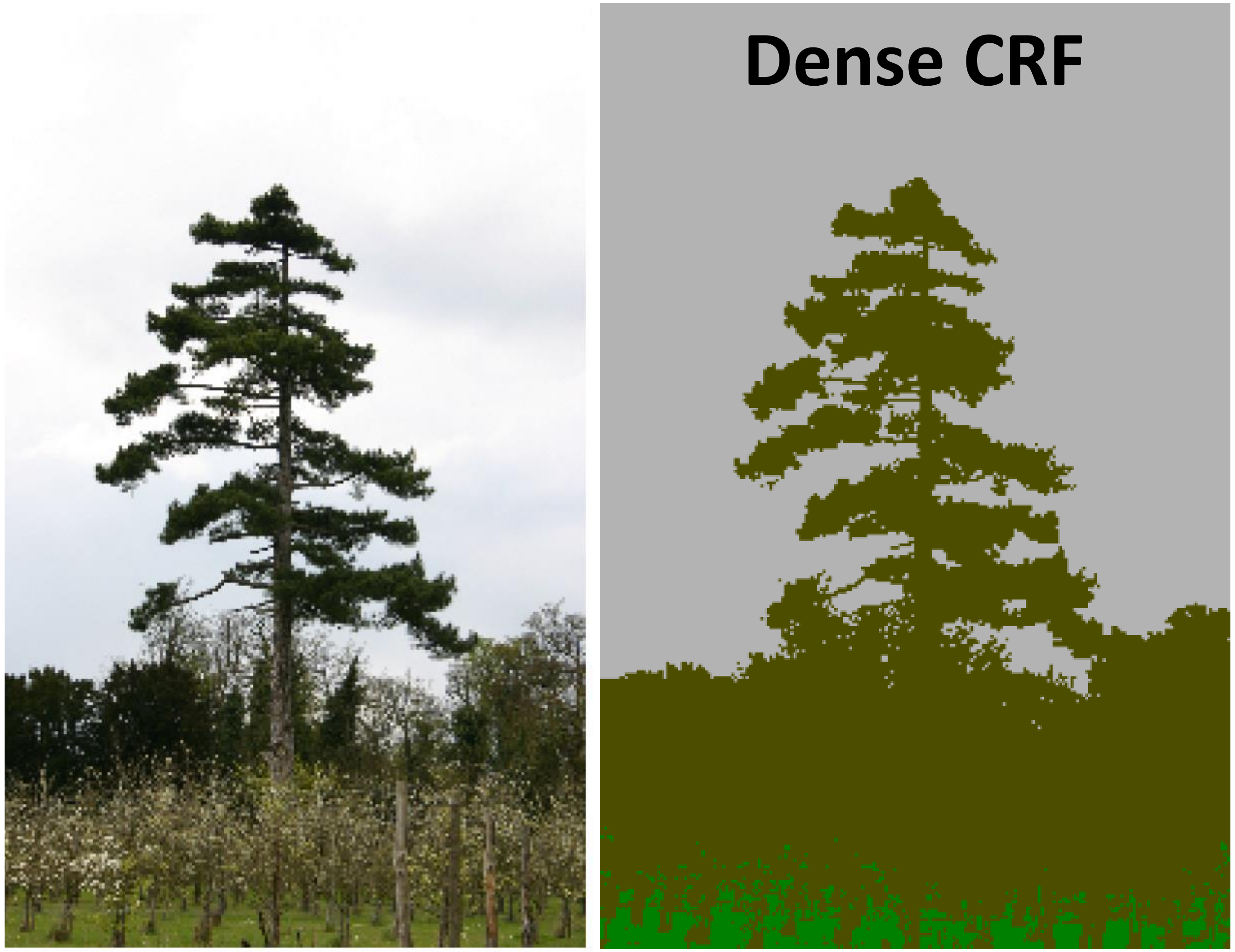
Grid CRF for Image Segmentation



- Local connections
- May not capture the sharp boundaries

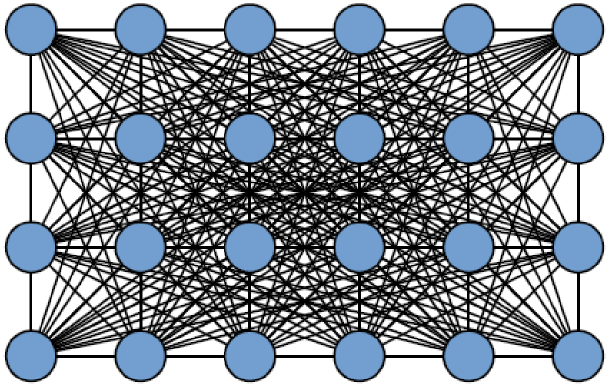


Fully connected CRF for Image Segmentation



- Fully connected CRF
  - Every node is connected to every other node

- MCMC inference, 36 hours!!



Efficient Inference on Fully connected CRF

- They propose an efficient approximate algorithm for inference on fully connected CRF
- Inference in 0.2 seconds
  - ~50,000 nodes (apply to pixel level segmentation)
- Based on a mean field approximation to the CRF distribution

Mean Field Approximation

- Mean field update rule for CRF

$$Q_i(x_i = l) = \frac{1}{Z_i}\exp\{-\psi_u(x_i) - \sum_{l' \in L}\mu(l, l')\sum_{m=1}^{K} w^{(m)}\sum_{j \neq i} k^{(m)}\left(f_i, f_j\right)Q_j(l')\}$$

Mean Field Approximation

$Q_i(x_i = l)$ =

$$\frac{1}{Z_i}\exp\left\{-\psi_u(x_i) - \sum_{l' \in L}\mu(l,l')\sum_{m=1}^{K} w^{(m)} \sum_{j \neq i} k^{(m)}(f_i, f_j) Q_j(l')\right\}$$

Algorithm

- Initialize Q : $Q_i(x_i) = \frac{1}{Z_i}\exp\{-\phi_u(x_i)\}$
- While not converged
    - Message passing: $\widetilde{Q_i^{(m)}}(l) = \sum_{j \neq i} k^{(m)}(f_i, f_j) Q_j(l')$

Mean Field Approximation

$$Q_i(x_i = l) = \frac{1}{Z_i} \exp\left\{ -\psi_u(x_i) - \sum_{l' \in L} \mu(l, l') \sum_{m=1}^{K} w^{(m)} \widetilde{Q_i^{(m)}}(l) \right\}$$

Algorithm

- Initialize Q : $Q_i(x_i) = \frac{1}{Z_i} \exp\{-\phi_u(x_i)\}$
- While not converged
  - Message passing: $\widetilde{Q_i^{(m)}} = \sum_{j \neq i} k^{(m)}(f_i, f_j) Q_j(l')$
  - Compatibility transform: $\widehat{Q_i}(x_i) = \sum_{l' \in L} \mu(l, l') \sum_{m=1}^{K} w^{(m)} \widetilde{Q_i^{(m)}}(l)$

Mean Field Approximation

$$Q_i(x_i = l) = \frac{1}{Z_i}\exp\{-\psi_u(x_i) - \widehat{Q_i}(x_i)\}$$

Algorithm

- Initialize Q : $Q_i(x_i) = \frac{1}{Z_i}\exp\{-\phi_u(x_i)\}$
- While not converged
  - Message passing: $\widetilde{Q_i^{(m)}} = \sum_{j\neq i} k^{(m)}(f_i, f_j) Q_j(l')$
  - Compatibility transform: $\widehat{Q_i}(x_i) = \sum_{l'\in L} \mu(l, l') \sum_{m=1}^{K} w^{(m)} \widetilde{Q_i^{(m)}}(l)$
  - Update to calculate $Q_i(x_i = l)$
  - Normalization

Mean Field Approximation

$$Q_i(x_i = l) = \frac{1}{Z_i} \exp\{-\psi_u(x_i) - \widehat{Q_i}(x_i)\}$$

Algorithm

- Initialize Q : $Q_i(x_i) = \frac{1}{Z_i} \exp\{-\phi_u(x_i)\}$
- While not converged
  - $\mathbf{O(N^2)}$ Message passing: $\widetilde{Q_i^{(m)}} = \sum_{j \neq i} k^{(m)}(f_i, f_j) Q_j(l')$
  - $\boldsymbol{O(N)}$ Compatibility transform: $\widehat{Q_i}(x_i) = \sum_{l' \in L} \mu(l, l') \sum_{m=1}^{K} w^{(m)} \widetilde{Q_i^{(m)}}(l)$
  - $\mathbf{O(N)}$ Update to calculate $Q_i(x_i = l)$
  - $\mathbf{O(N)}$ Normalization

Mean Field Approximation

$$Q_i(x_i = l) = \frac{1}{Z_i} \exp\{-\psi_u(x_i) - \widehat{Q_i}(x_i)\}$$

Algorithm

- Initialize Q : $Q_i(x_i) = \frac{1}{Z_i} \exp\{-\phi_u(x_i)\}$
- While not converged

$\mathbf{O(N^2)}$ • $\boxed{\textbf{Message passing: } \widetilde{\boldsymbol{Q}_i^{(m)}} = \sum_{j \neq i} \boldsymbol{k}^{(m)}(\boldsymbol{f}_i, \boldsymbol{f}_j) \boldsymbol{Q}_j(\boldsymbol{l}')}$

$\boldsymbol{O(N)}$ • Compatibility transform: $\widehat{Q_i}(x_i) = \sum_{l' \in L} \mu(l, l') \sum_{m=1}^{K} w^{(m)} \widetilde{Q_i^{(m)}}(l)$

$\mathbf{O(N)}$ • Update to calculate $Q_i(x_i = l)$

$\mathbf{O(N)}$ • Normalization

Efficient Message Passing

- Message passing

$$\widetilde{Q_i^{(m)}} = \sum_{j \neq i} k^{(m)}(f_i, f_j) Q_j(l')$$

  - Gaussian filter $k^{(m)}(f_i, f_j)$
  - Apply convolution to $Q_j(l')$

Efficient Message Passing

- Message passing

$$\widetilde{Q_i^{(m)}} = \sum_{j \neq i} \boldsymbol{k}^{(m)}(\boldsymbol{f}_i, \boldsymbol{f}_j) Q_j(l') = [\boldsymbol{G}^{(m)} \otimes \mathrm{Q(l)}] - \mathrm{Q_i(l)}$$

  - Gaussian filter $k^{(m)}(f_i, f_j)$
  - Apply convolution to $Q_j(l')$
  - Smooth, low-pass filter -> can be reconstructed by a set of samples (by sampling theorem)

Efficient Message Passing

- Message passing

$$\widetilde{Q_i^{(m)}} = \sum_{j \neq i} \boldsymbol{k}^{(\boldsymbol{m})}(\boldsymbol{f}_i, \boldsymbol{f}_j) Q_j(l') = [\boldsymbol{G}^{(\boldsymbol{m})} \otimes \mathrm{Q(l)}] - \mathrm{Q_i(l)}$$

  - Downsampling $Q_j(l')$
  - Blur the downsampled signal (apply convolution operator with kernel $k^{(m)}$)
  - Upsampling to reconstruct the filtered signal ~ $\widetilde{Q_i^{(m)}}$

- Reduce the time complexity to $\boldsymbol{O}(\boldsymbol{N})$

Mean Field Approximation

$$Q_i(x_i = l) = \frac{1}{Z_i}\exp\{-\psi_u(x_i) - \widehat{Q_i}(x_i)\}$$

Algorithm

- Initialize Q : $Q_i(x_i) = \frac{1}{Z_i}\exp\{-\phi_u(x_i)\}$
- While not converged

$\mathbf{O}(\boldsymbol{N})$
  - Message passing: $\widetilde{Q_i^{(m)}} = \sum_{j\neq i} k^{(m)}(f_i, f_j) Q_j(l')$

$\boldsymbol{O}(\boldsymbol{N})$
  - Compatibility transform: $\widehat{Q_i}(x_i) = \sum_{l'\in L}\mu(l, l')\sum_{m=1}^{K} w^{(m)}\,\widetilde{Q_i^{(m)}}(l)$

$\mathbf{O}(\mathbf{N})$
  - Update to calculate $Q_i(x_i = l)$

$\mathbf{O}(\mathbf{N})$
  - Normalization

Results

Image

Dense CRF Results

Results

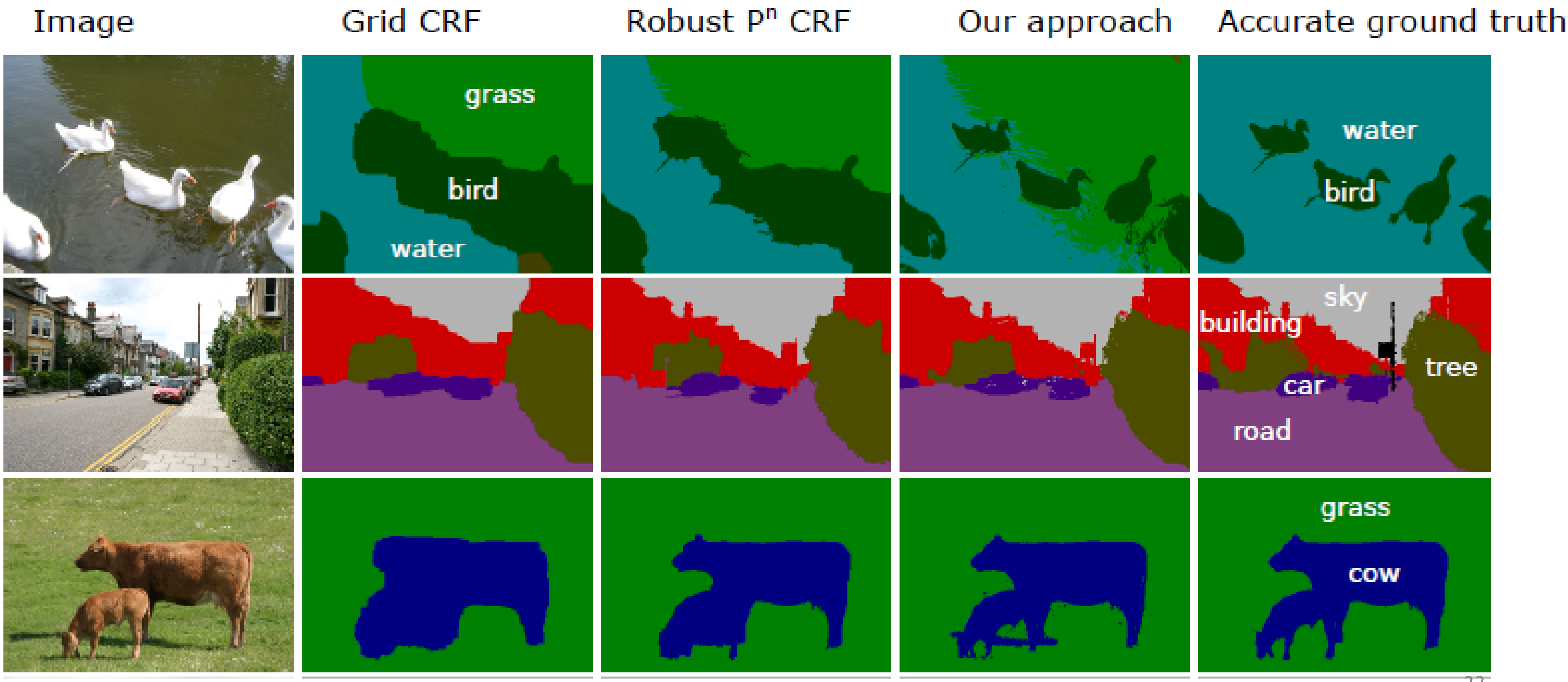
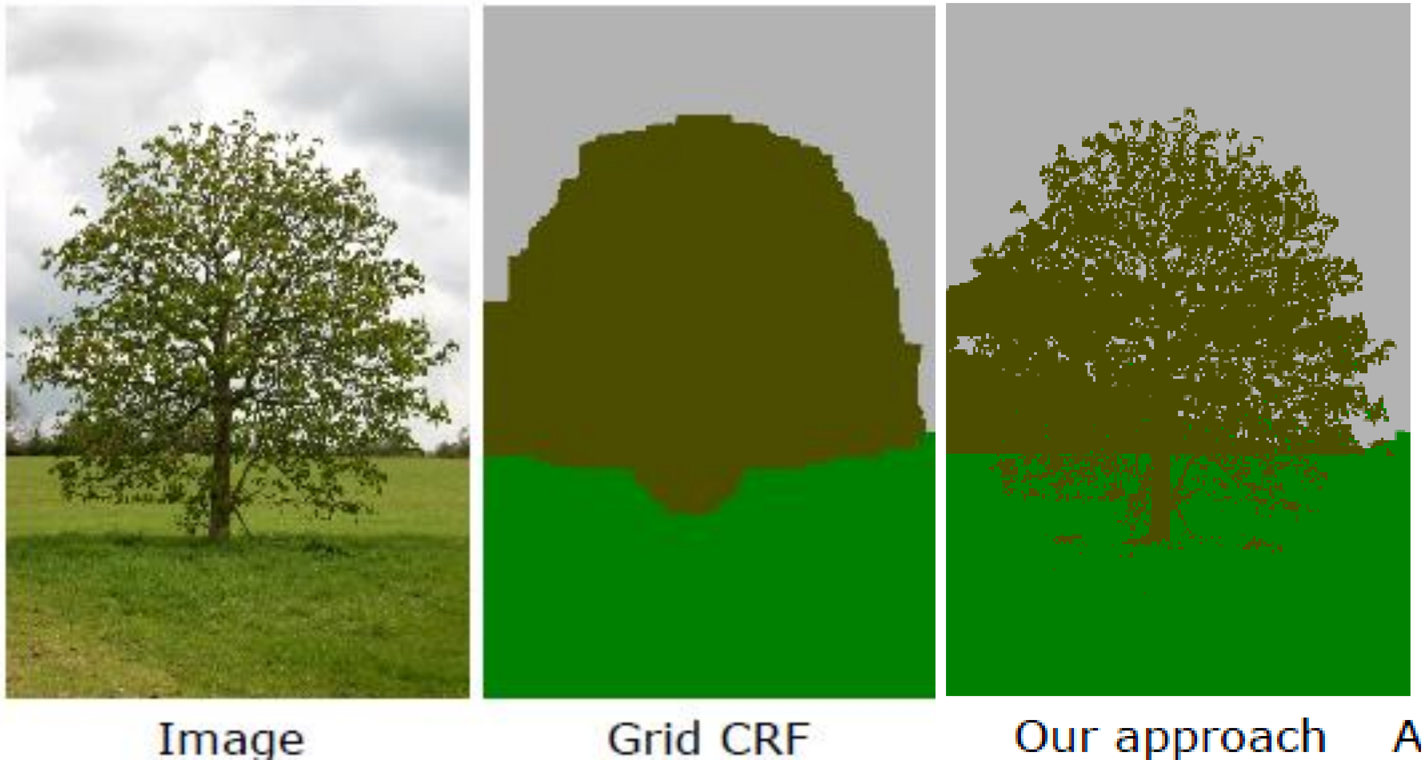


Image
Grid CRF
Robust P^n CRF
Our approach
Accurate ground truth
grass
bird
water
water
bird
sky
building
tree
car
road
grass
cow

Conclusion

- A fully connected CRF model for pixel level segmentation
- Efficient inference on the fully connected CRF
  - Linear in number of variables



Dense CRF as Post-processing

Semantic Image Segmentation with Deep Convolutional Nets and Fully Connected CRFs. Chen et al. ICLR'15

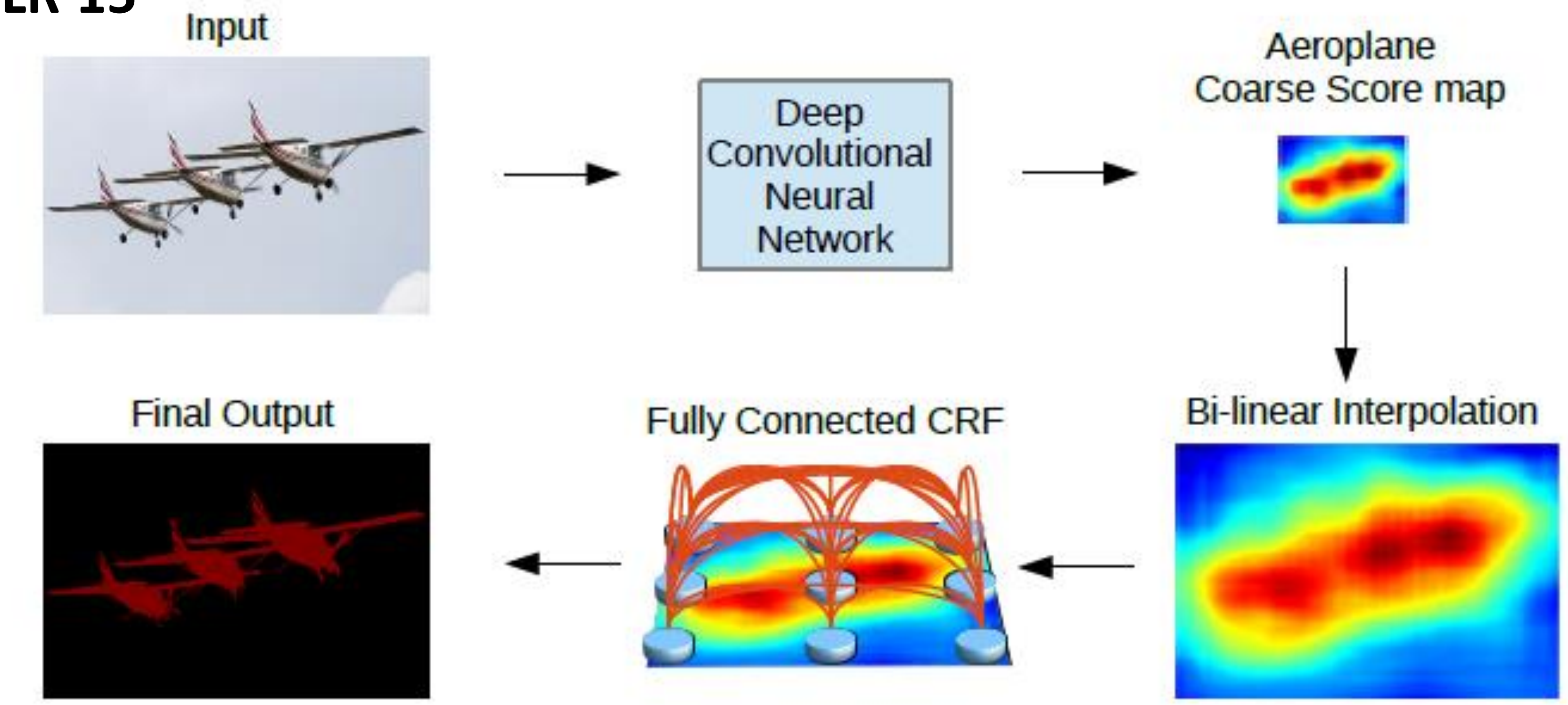


Figure 3: Model Illustration. The coarse score map from Deep Convolutional Neural Network (with fully convolutional layers) is upsampled by bi-linear interpolation. A fully connected CRF is applied to refine the segmentation result. Best viewed in color.

Convergent Inference

- Parameter Learning and Convergent Inference for Dense Random Fields. Philipp Krähenbühl and Vladlen Koltun. ICML’13.
  - A new efficient inference algorithm in dense CRF that is guaranteed to converge for some specific kernels and label compatibility functions.

Questions?

Pairwise Term in the Dense CRF Model

- Pairwise term

$$\psi_p(x_i, x_j) = \mu(x_i, x_j) \sum_{m=1}^{K} w^{(m)} k^{(m)}(f_i, f_j)$$

- They use

$$k(\mathbf{f}_i, \mathbf{f}_j) = \underbrace{w^{(1)} \exp\left(-\frac{|p_i - p_j|^2}{2\theta_\alpha^2} - \frac{|I_i - I_j|^2}{2\theta_\beta^2}\right)}_{\text{appearance kernel}} + \underbrace{w^{(2)} \exp\left(-\frac{|p_i - p_j|^2}{2\theta_\gamma^2}\right)}_{\text{smoothness kernel}}$$

p_i: position of pixel i
I_i: color intensity of pixel I
θ_*: hyper parameters