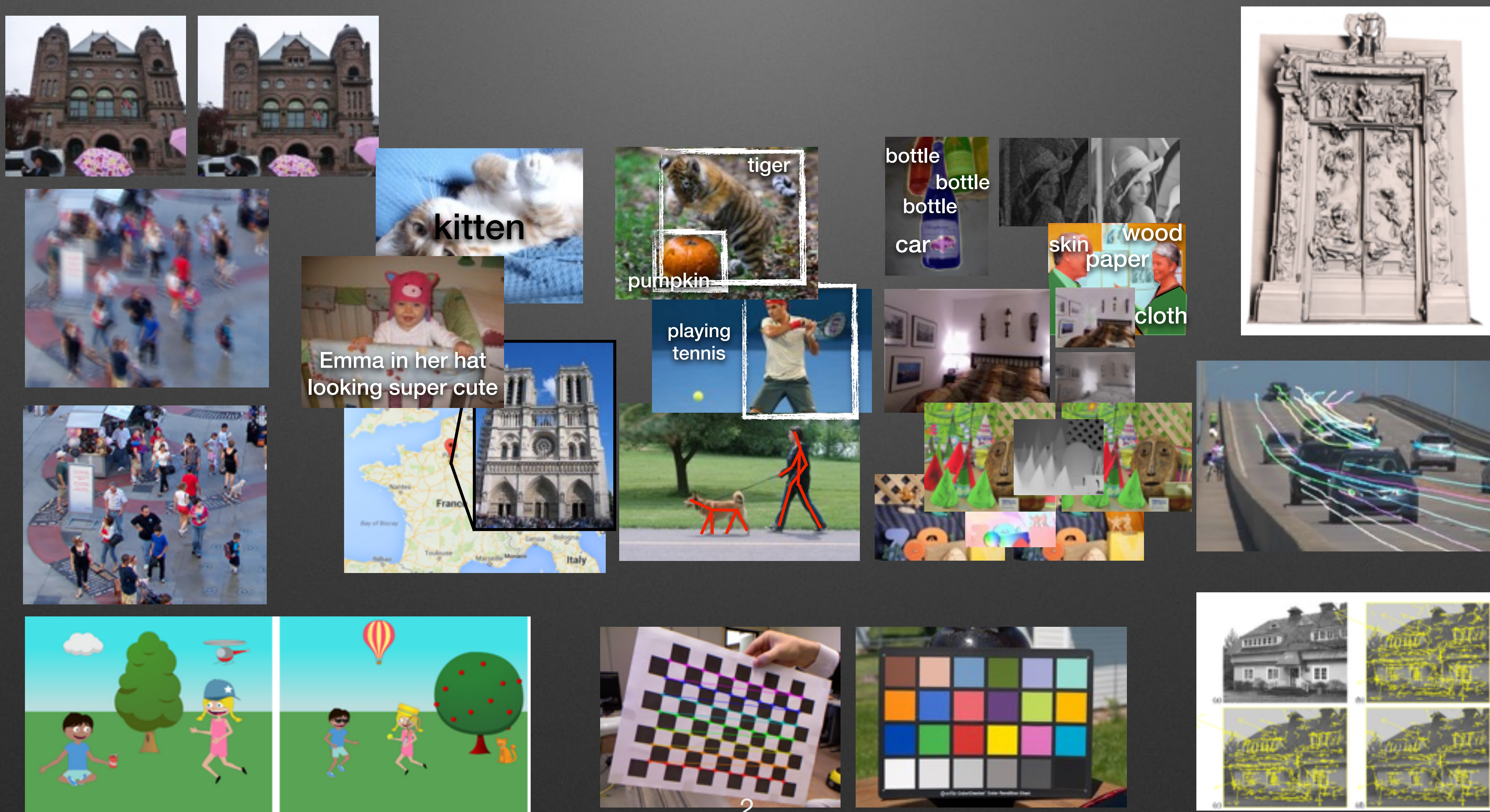
Dense Random Fields

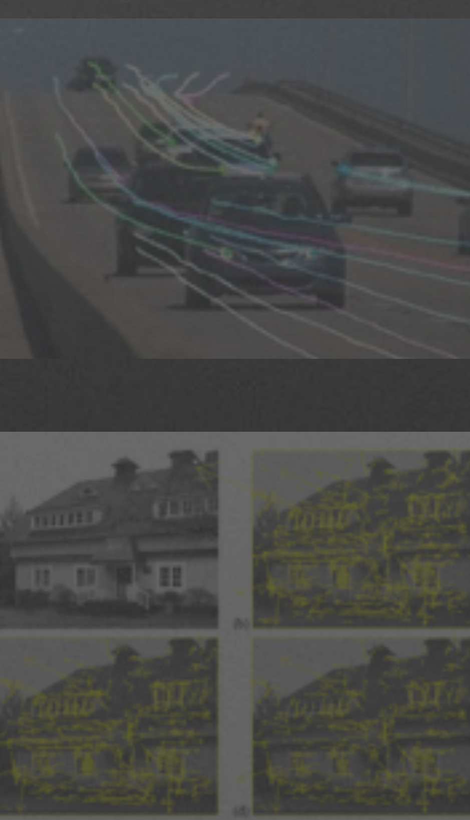
Philipp Krähenbühl
Stanford University

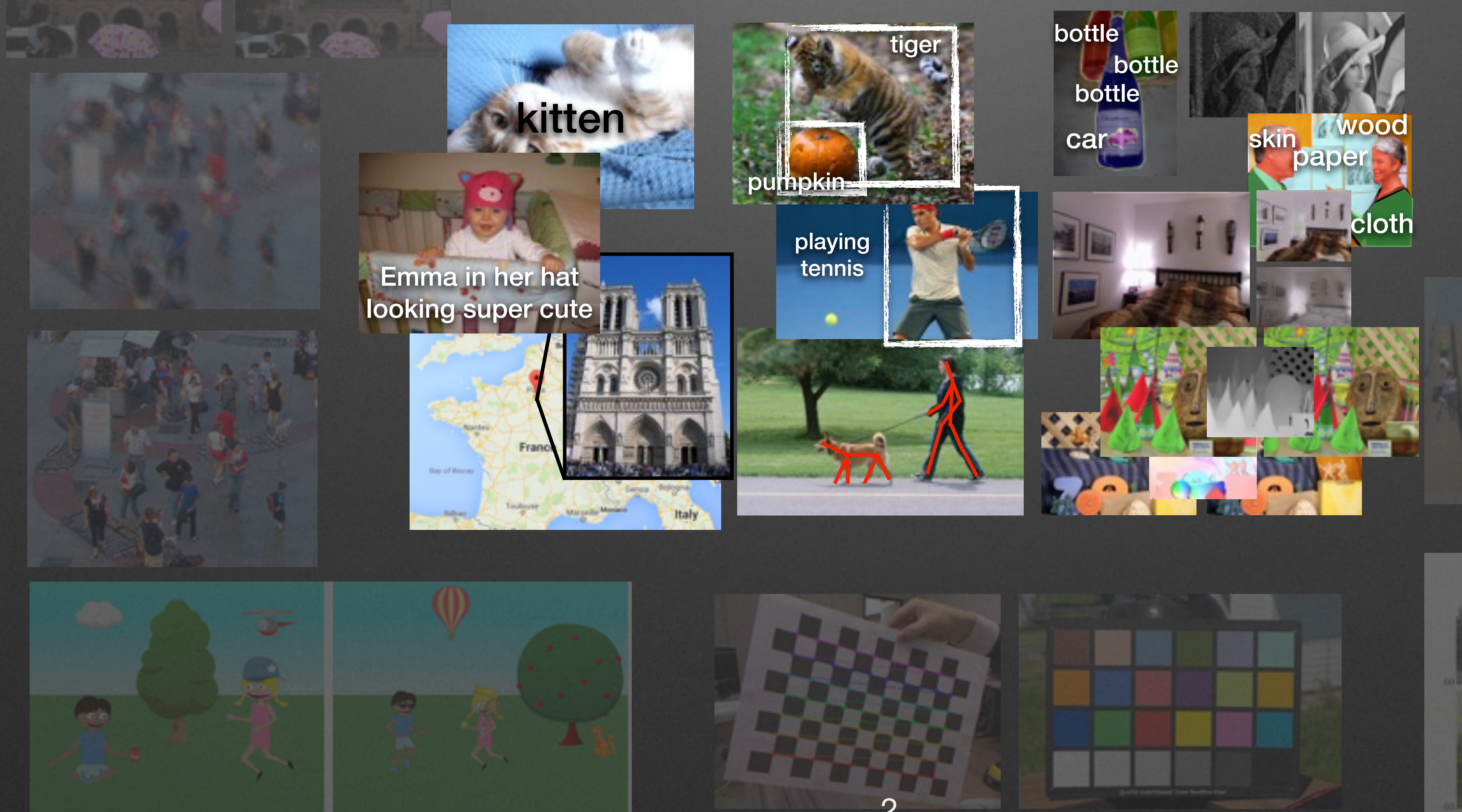
Zoo of computer vision problems



Zoo of computer vision problems

Labeling problems

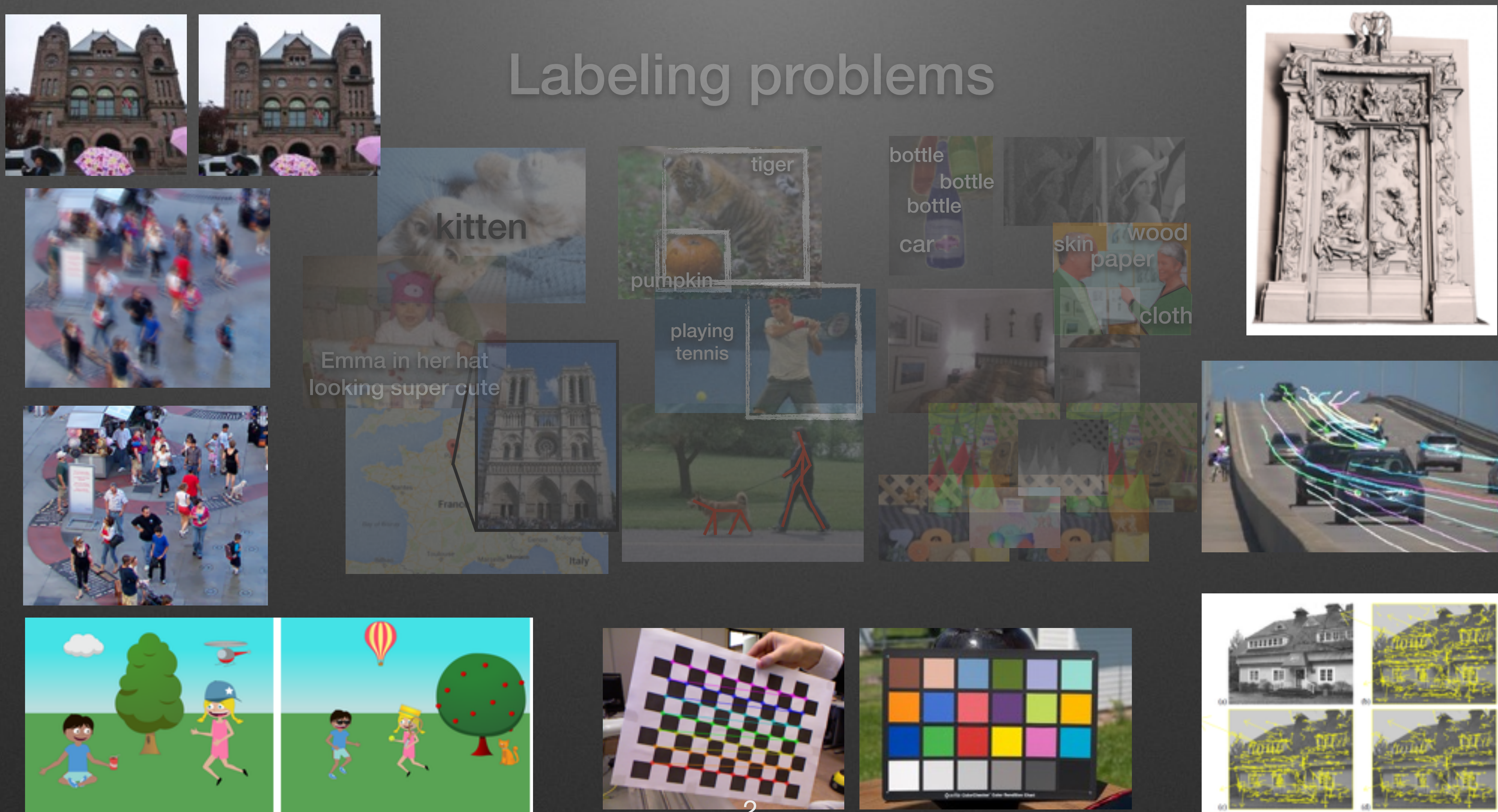




Zoo of computer vision problems

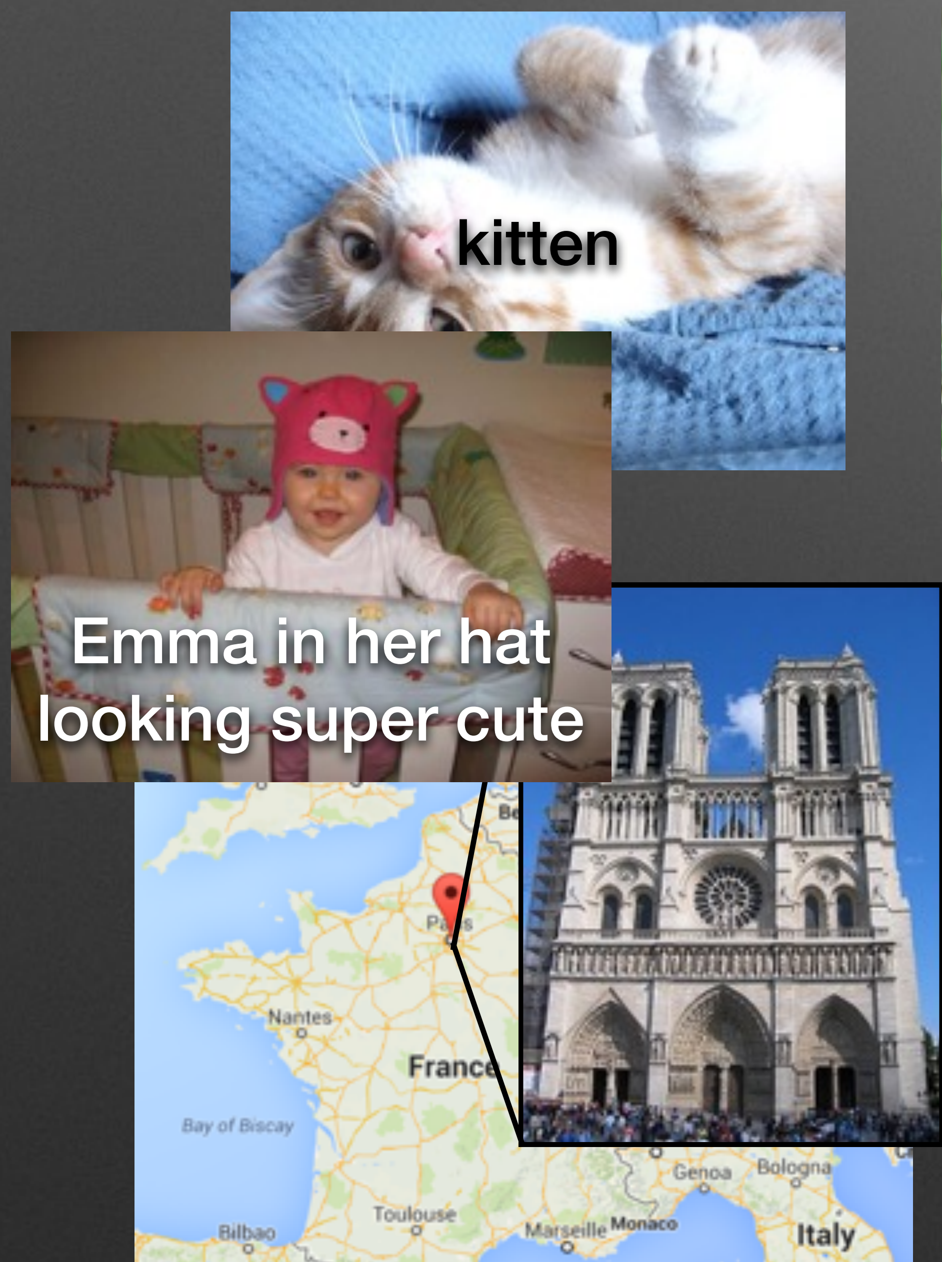
Labeling problems

Zoo of computer vision problems

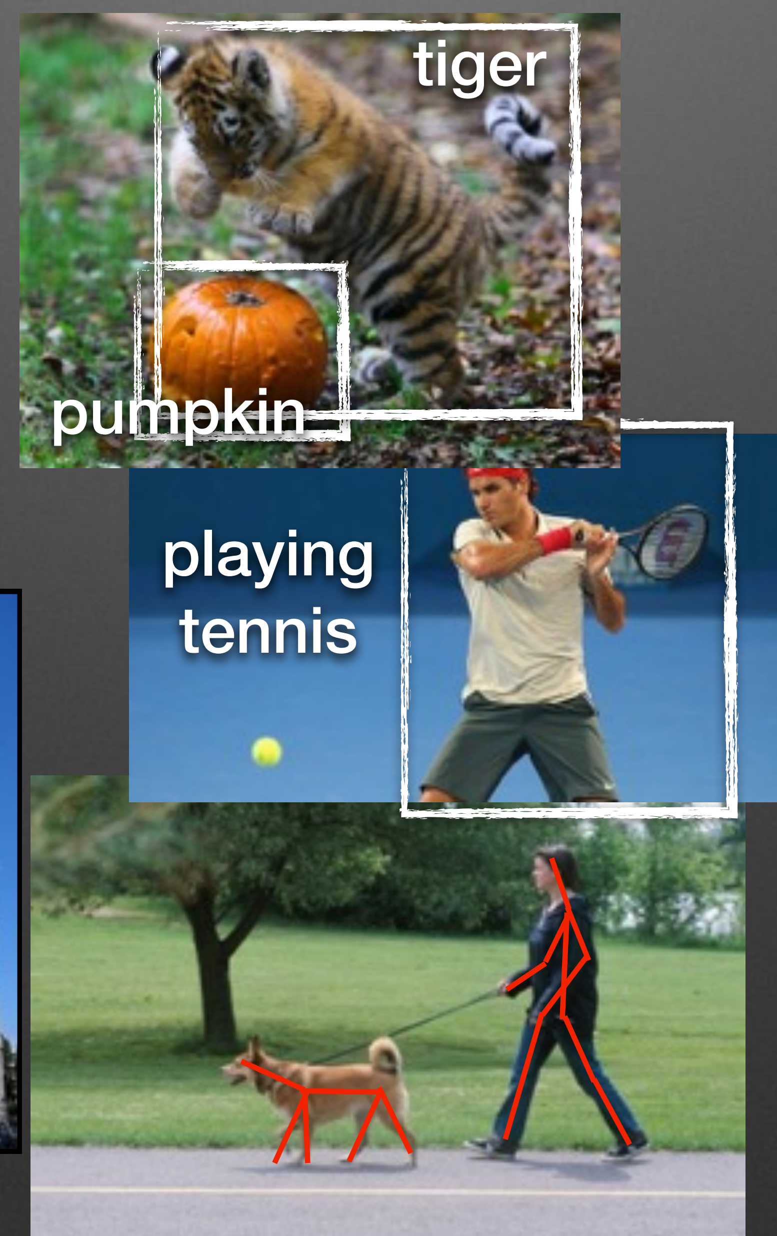


Labeling problems

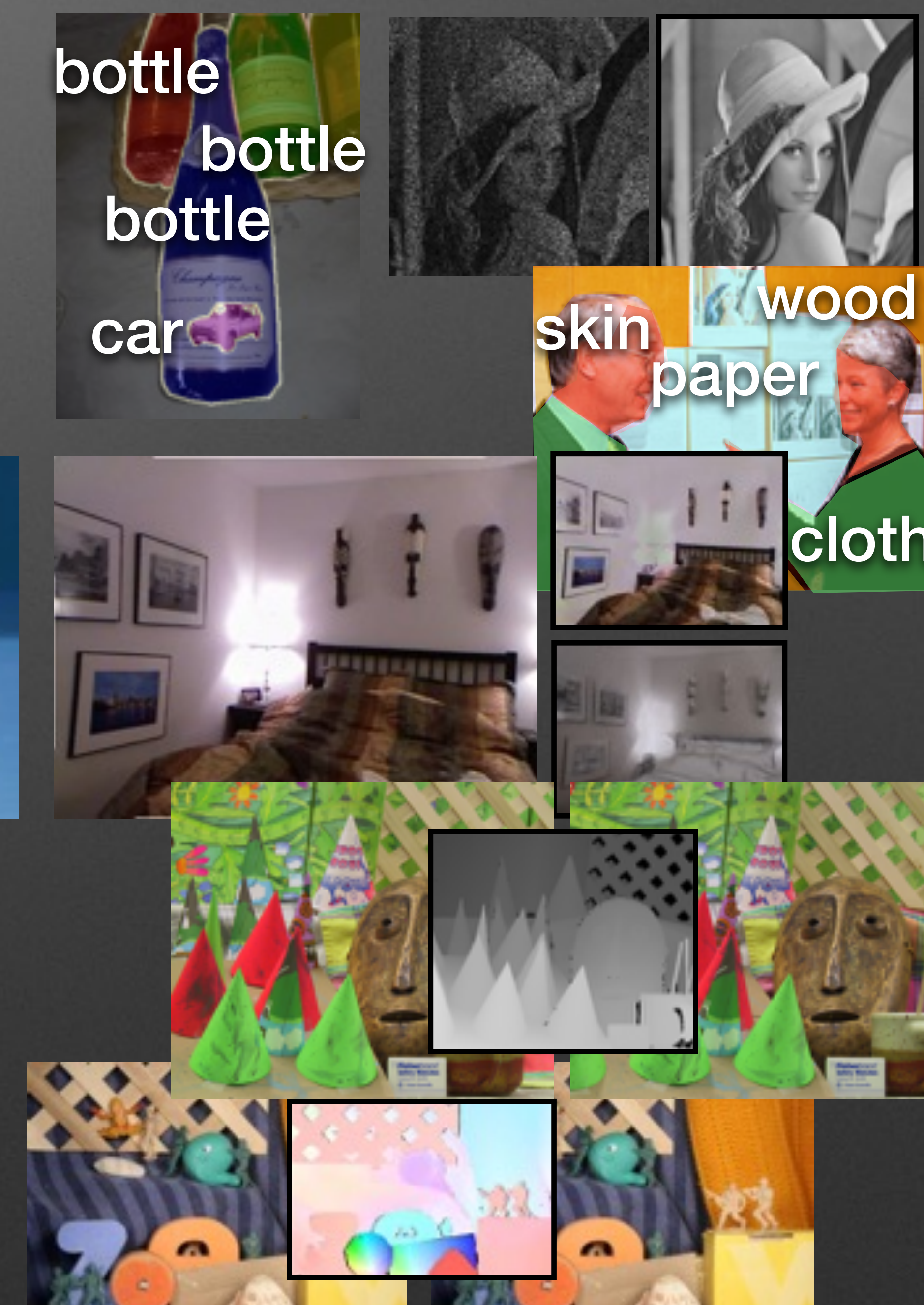
per image



sparse

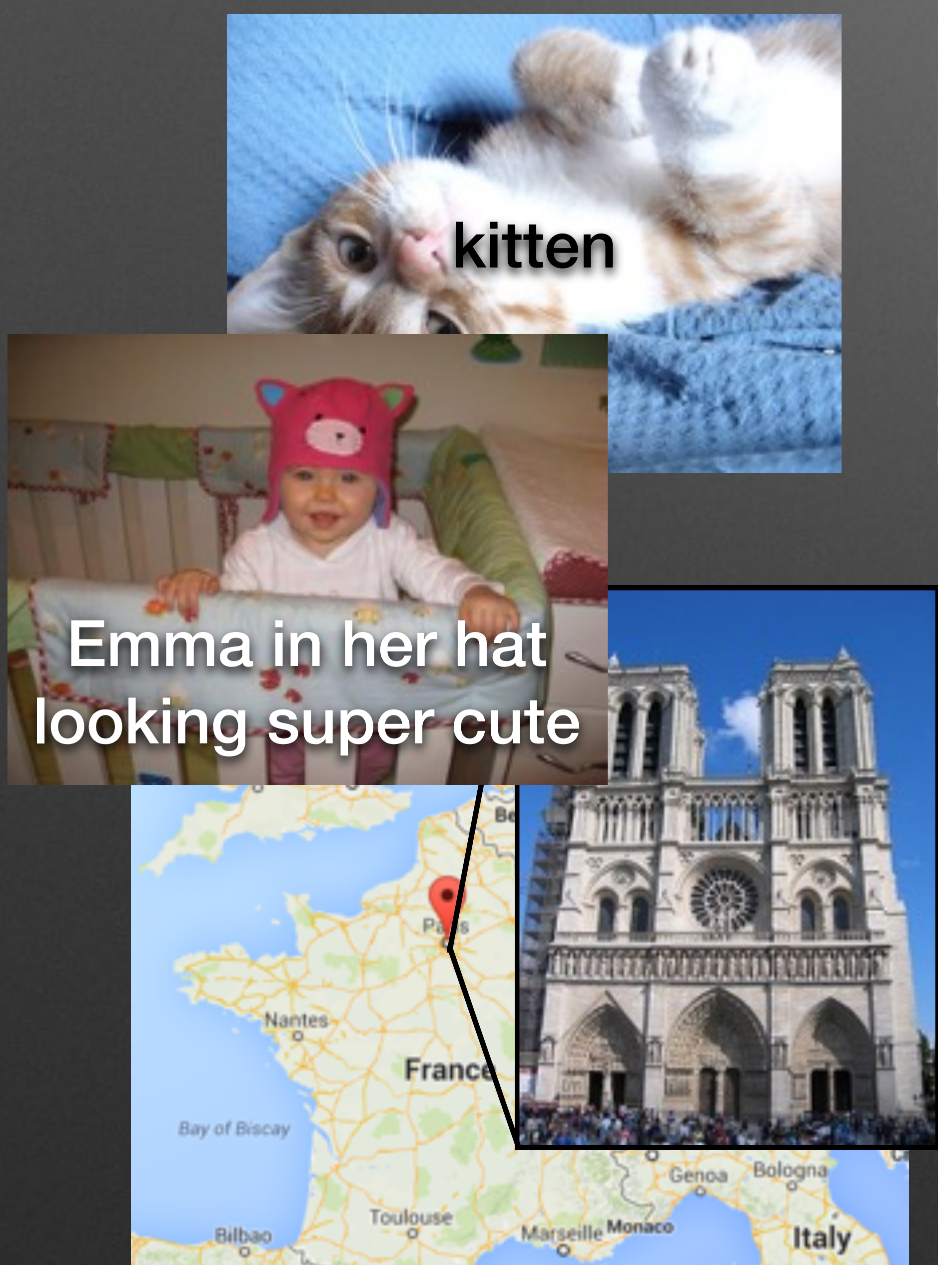


dense

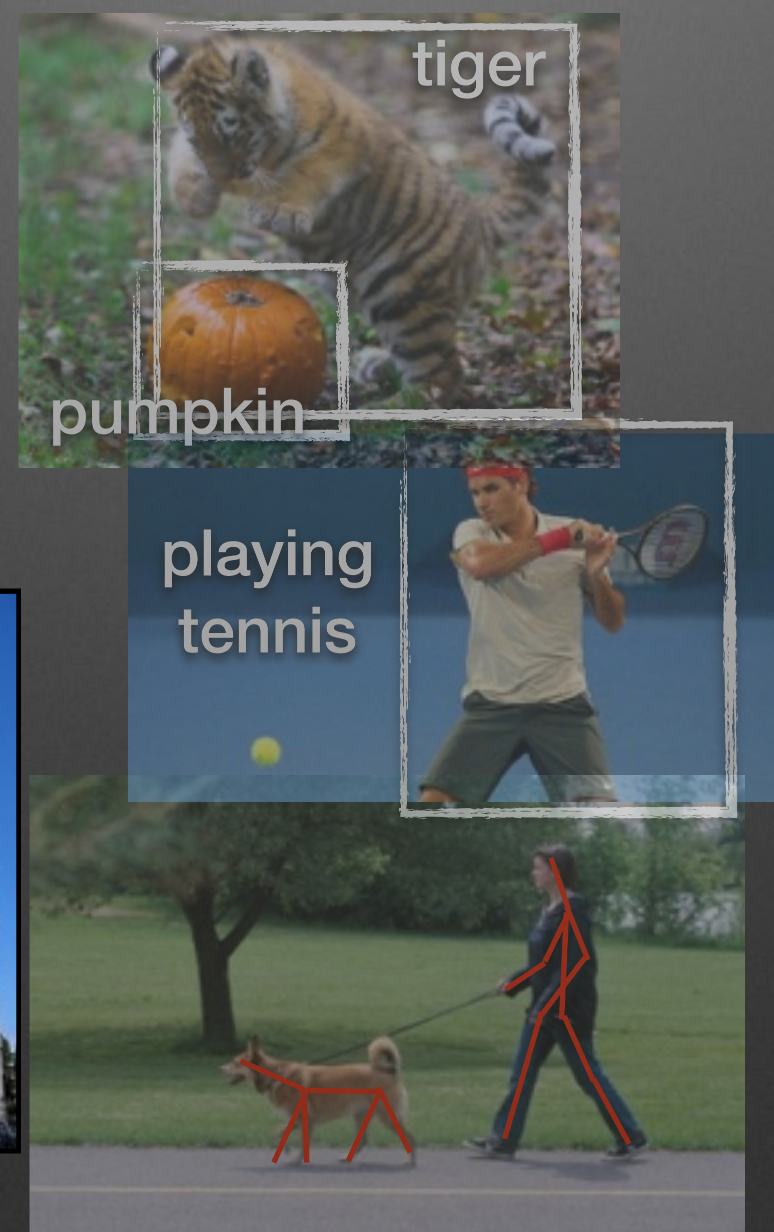


Labeling problems

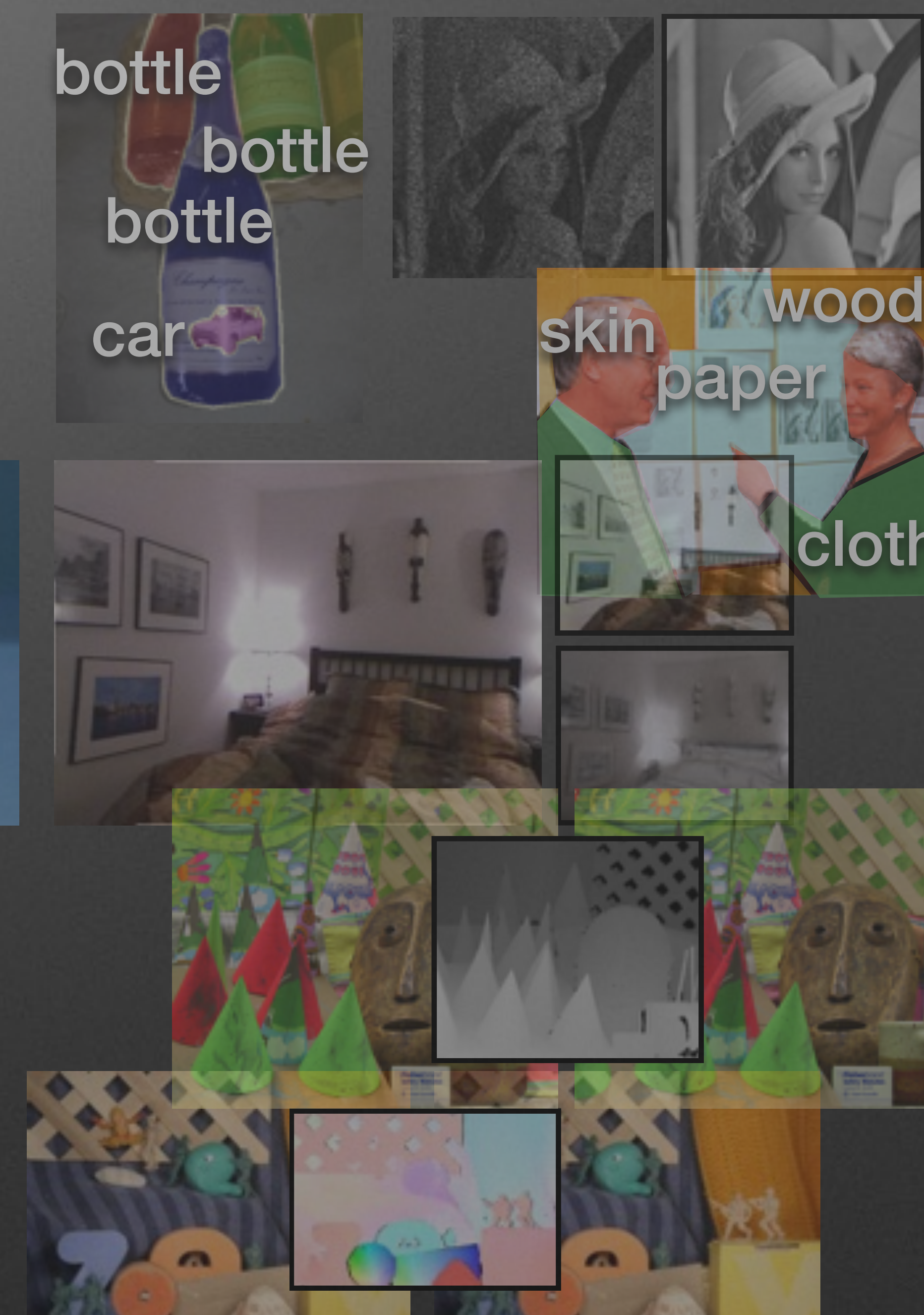
per image



sparse

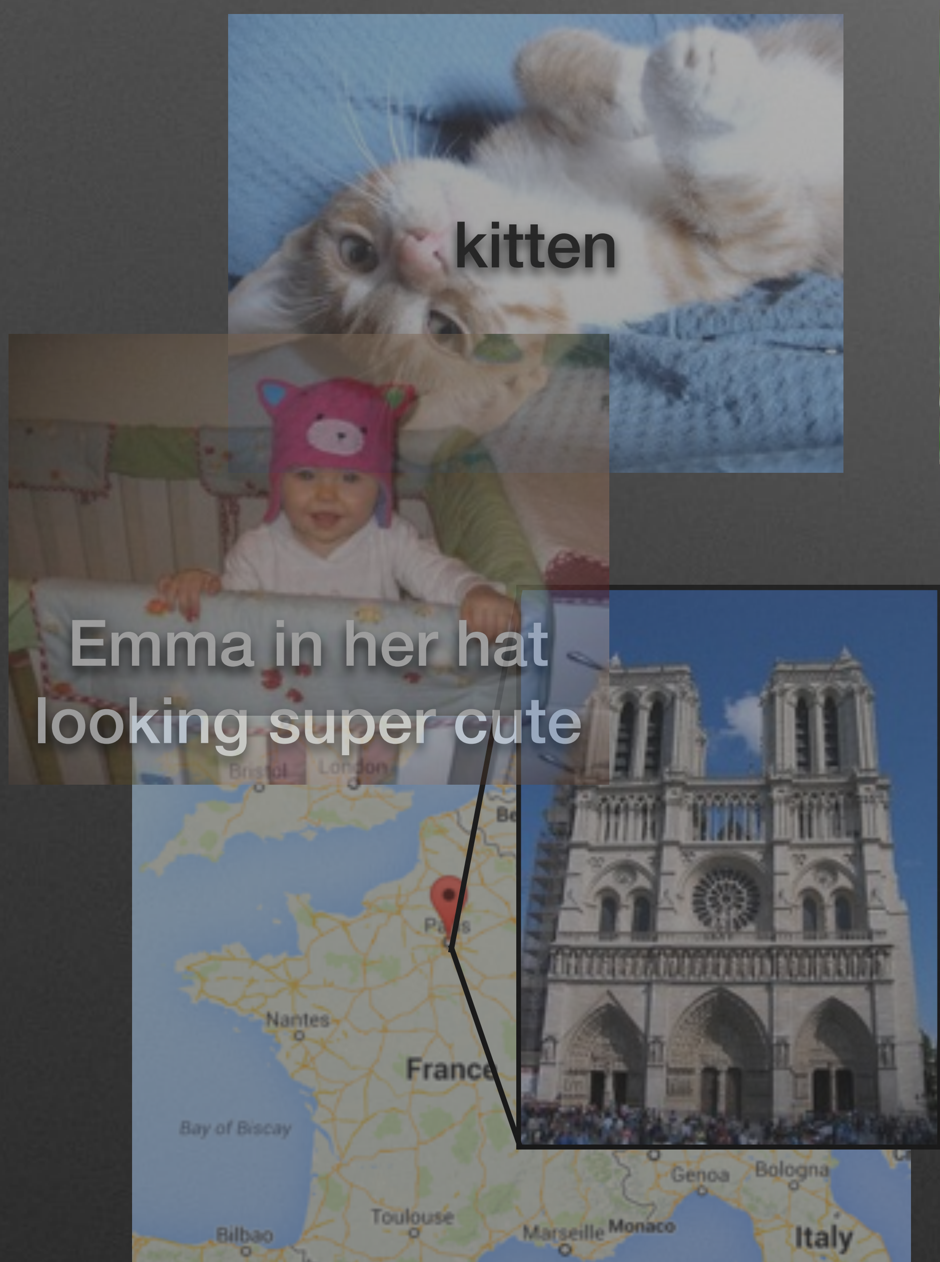


dense

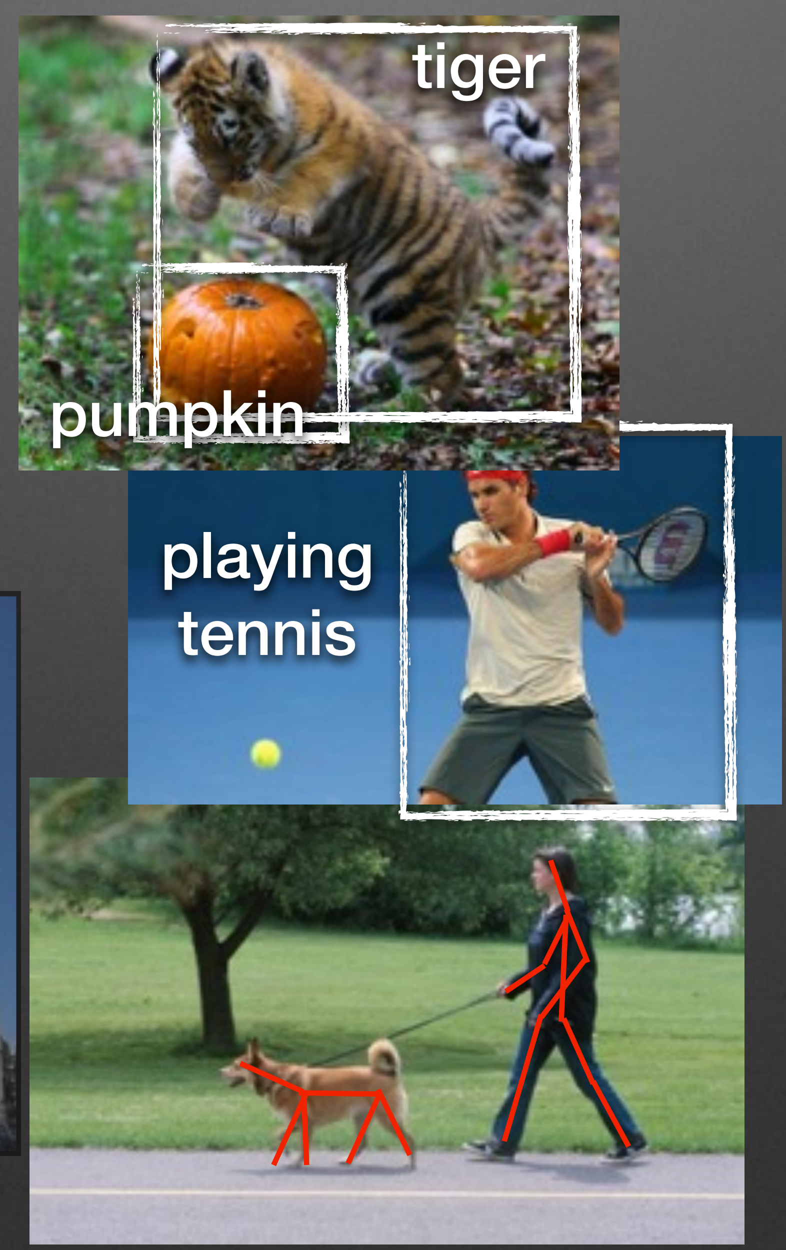


Labeling problems

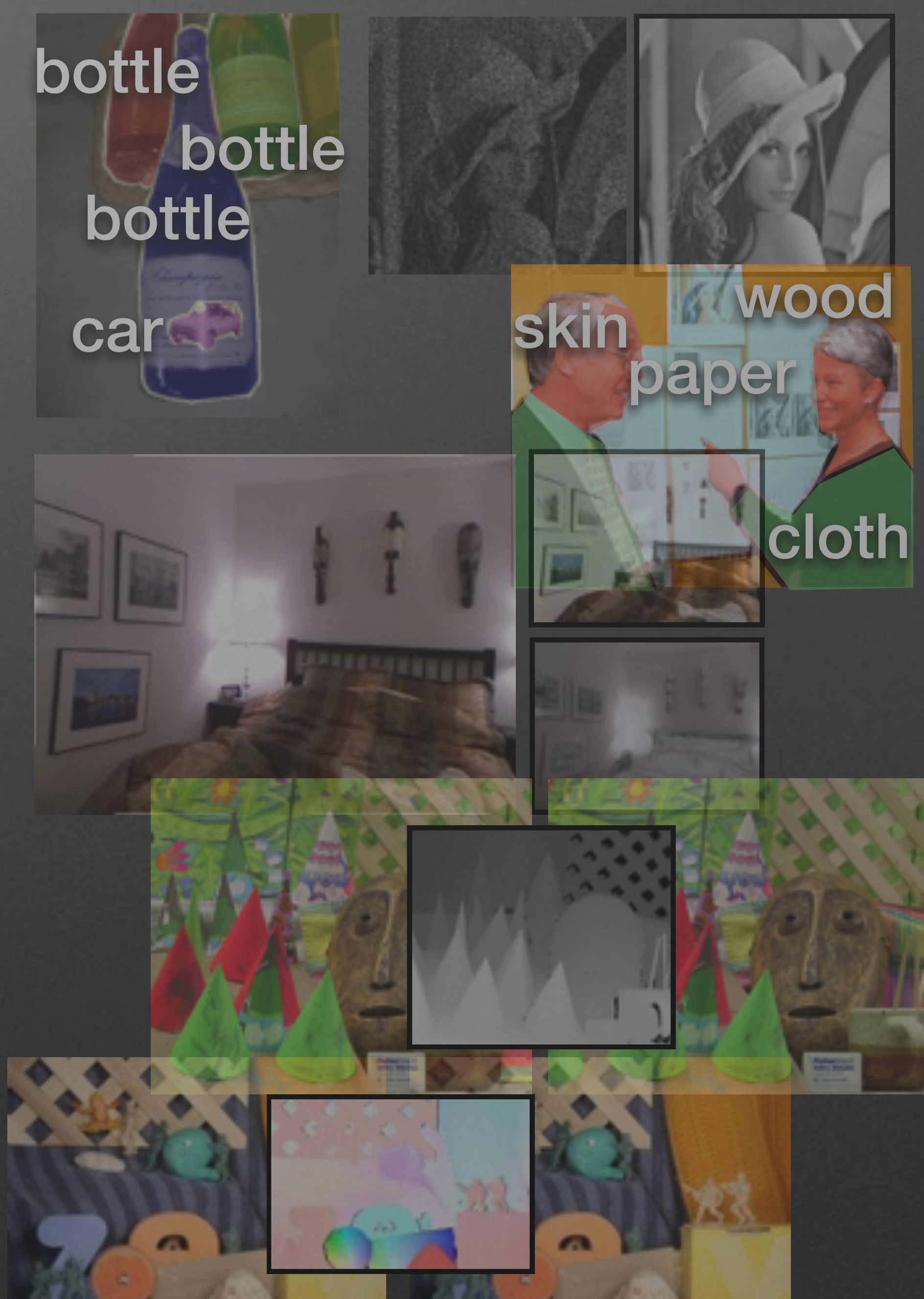
per image



sparse

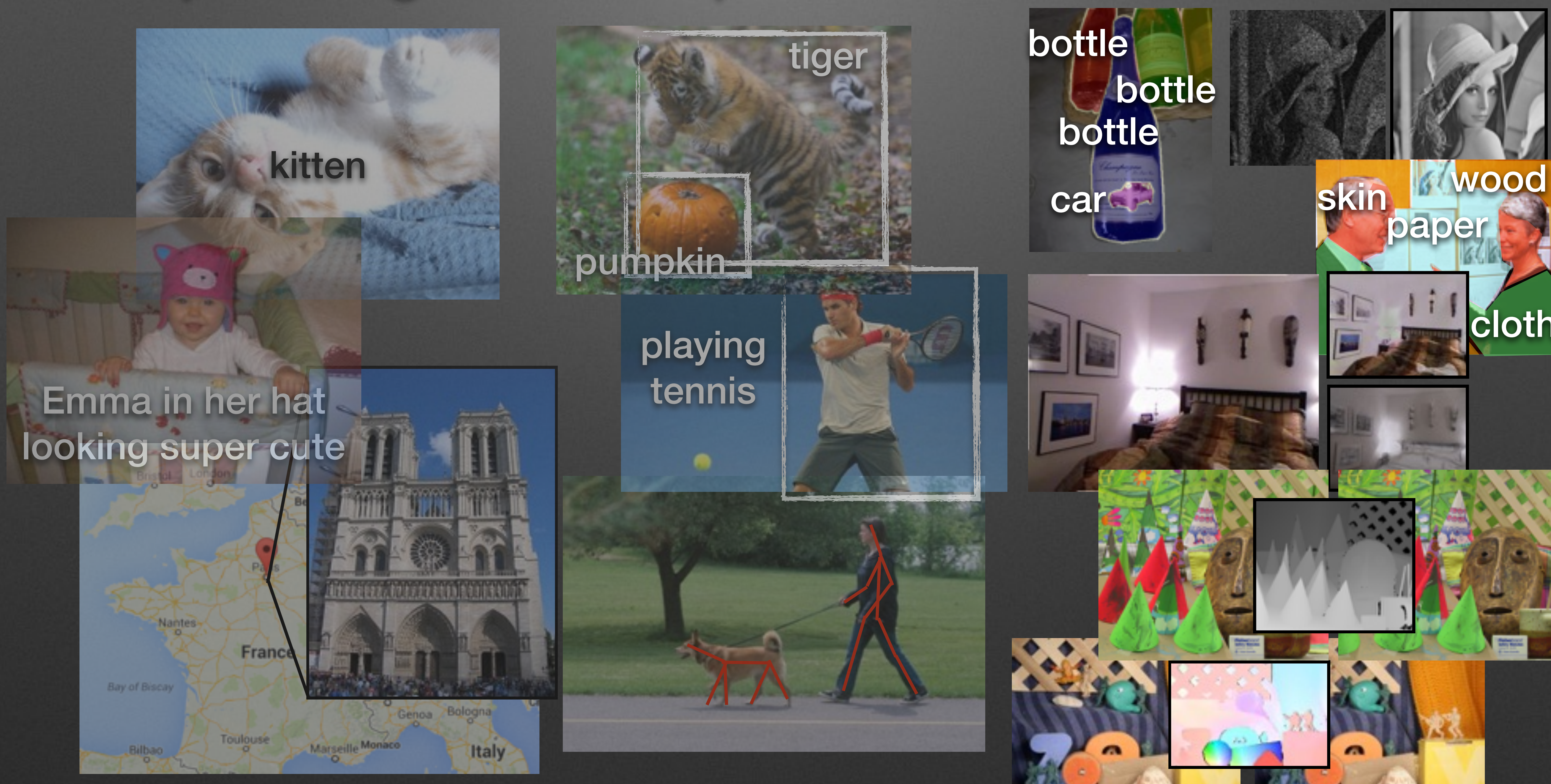


dense



Labeling problems

per image sparse dense

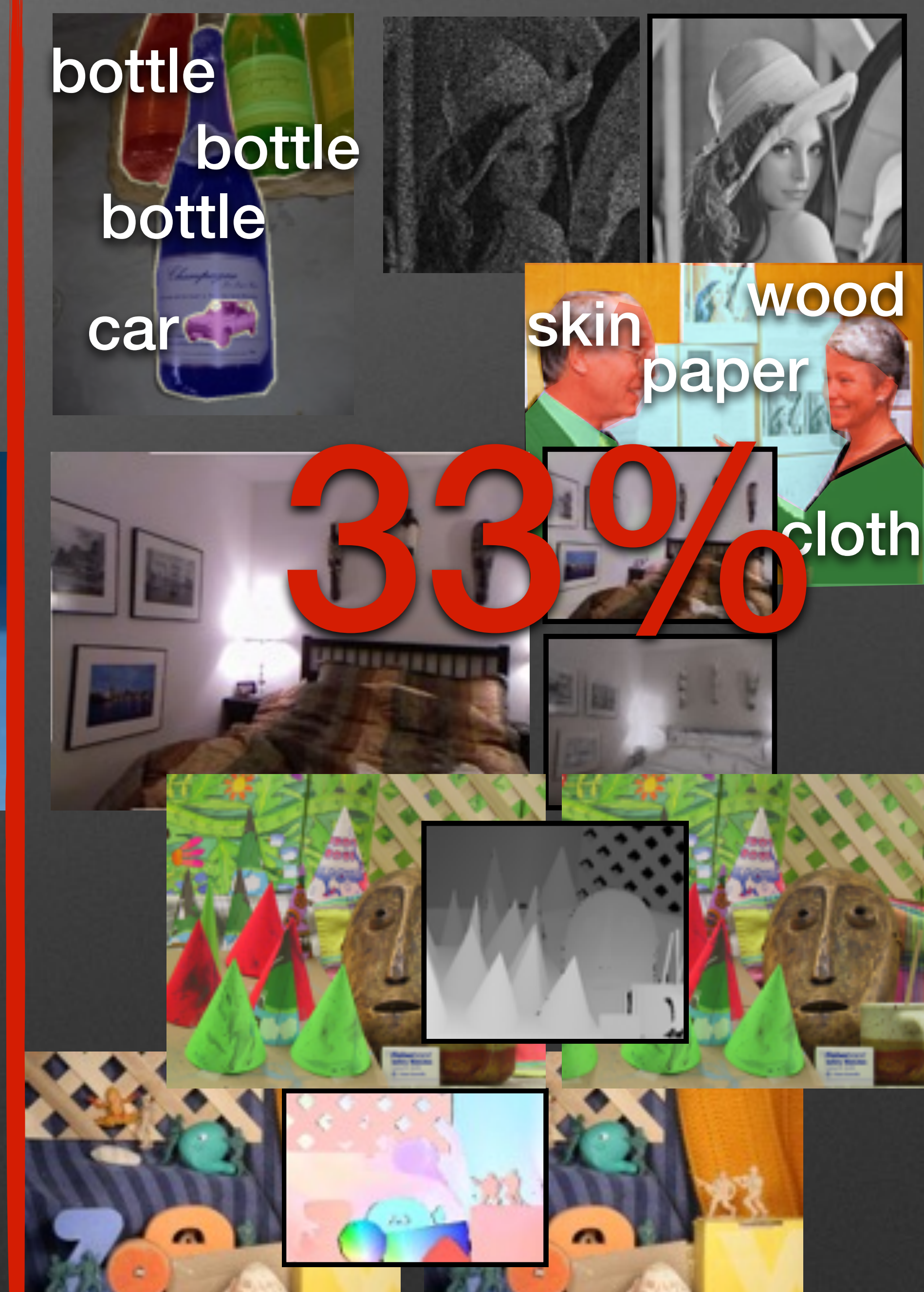


Labeling problems

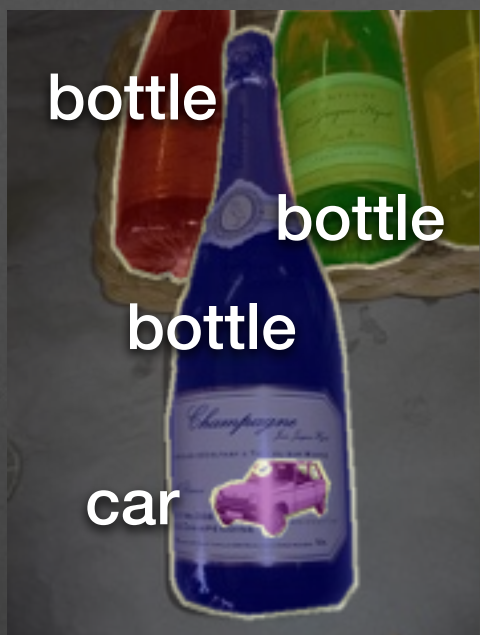
per image

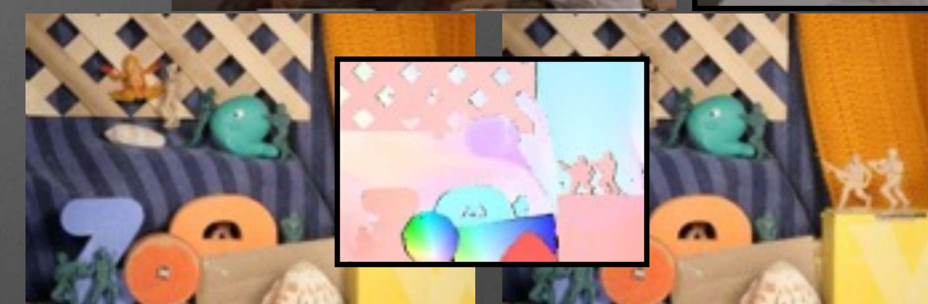
sparse

dense



Dense labeling problems

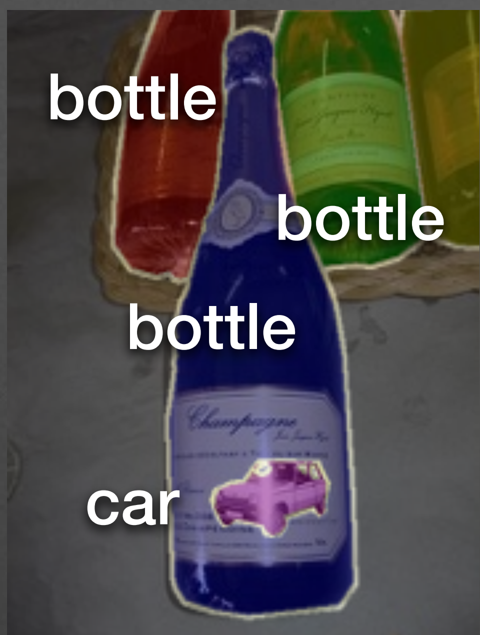


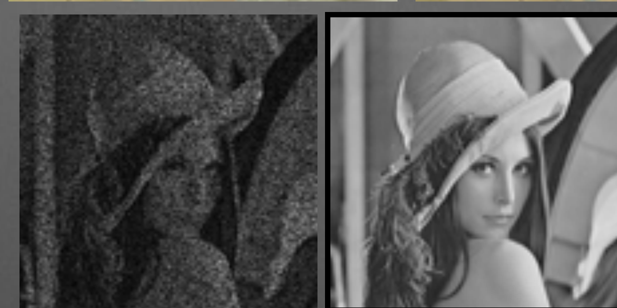


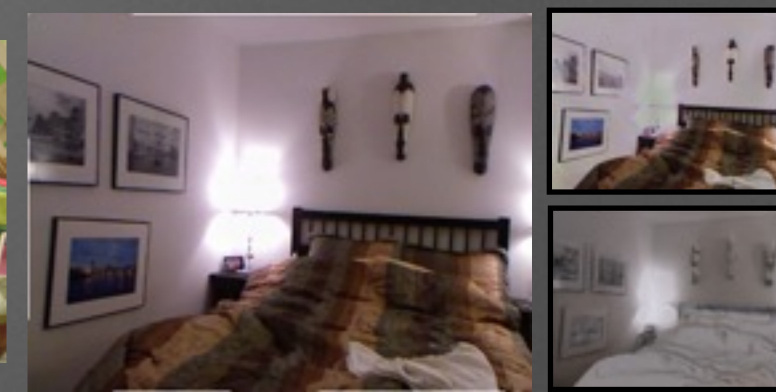
Dense labeling problems

- pixel-wise labeling

Dense labeling problems







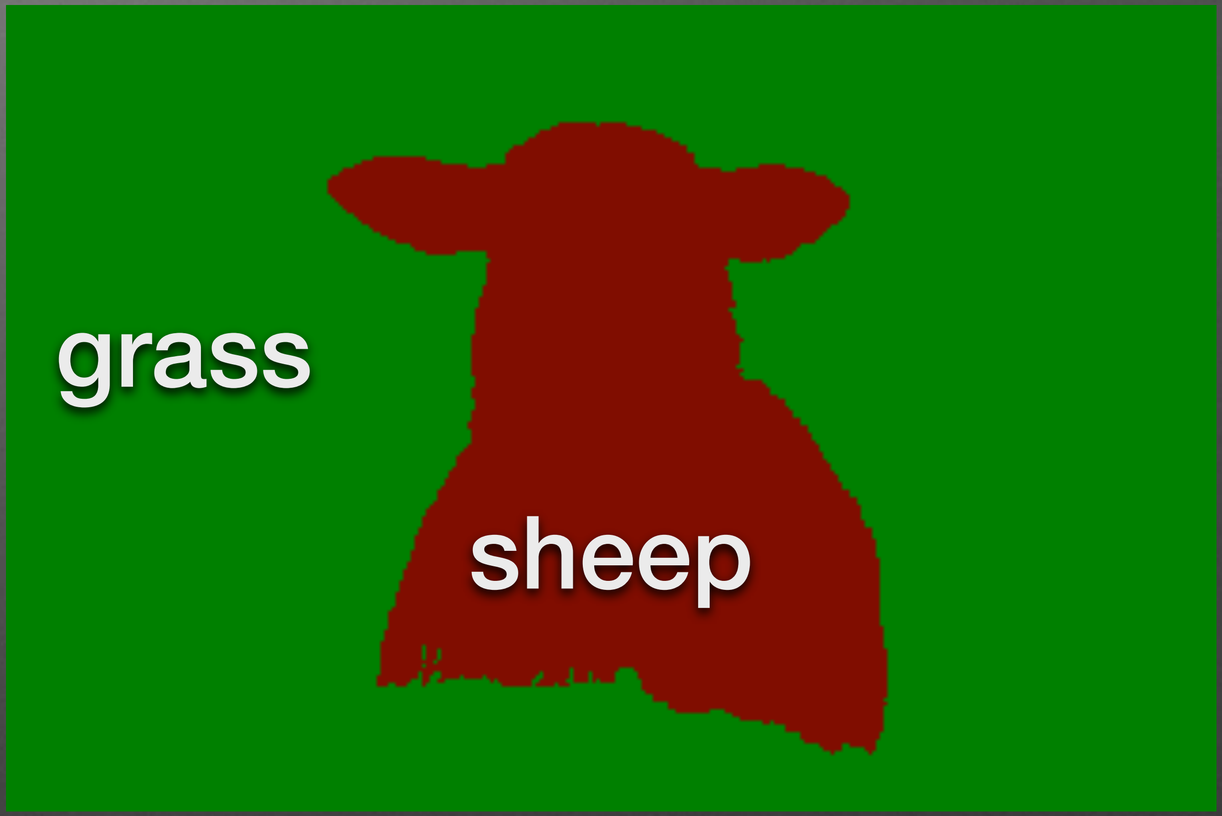
- pixel-wise labeling
- spatial coherence

Pixelswise vs Instance

Pixelswise vs Instance

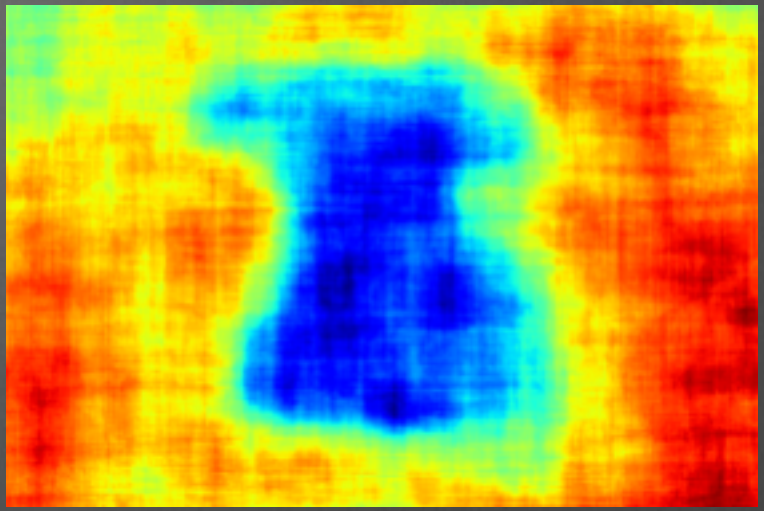
Multi-class image segmentation

Multi-class image segmentation

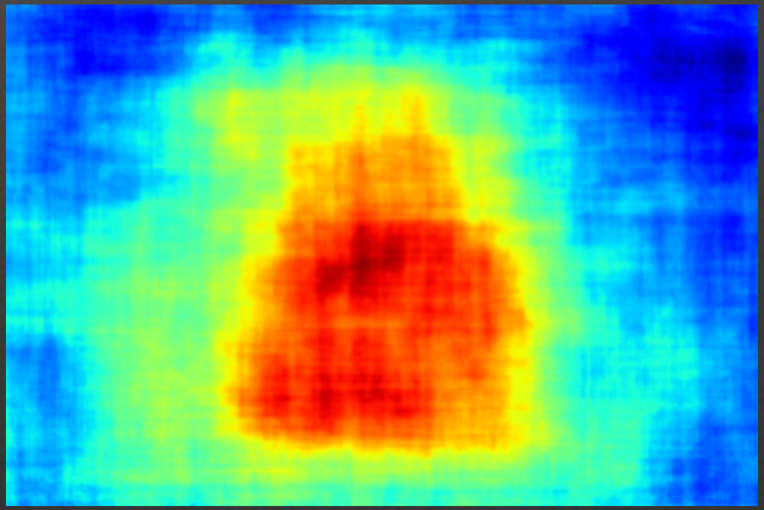


Classification

ψ(grass)



ψ(sheep)

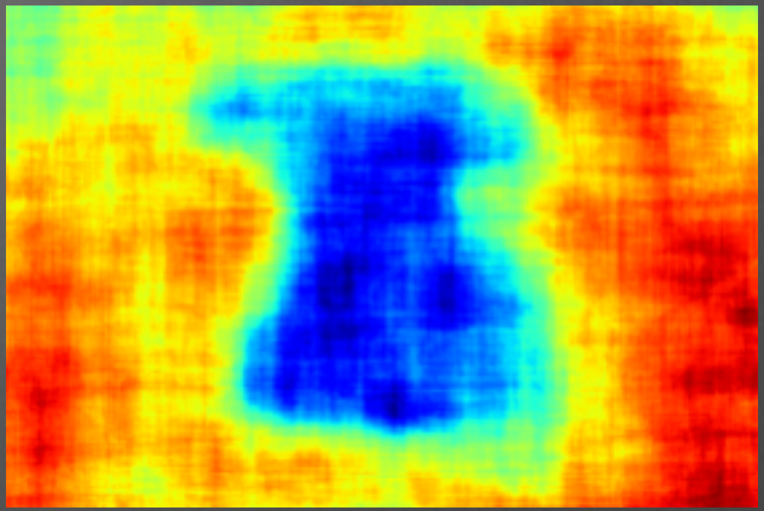


[1] TextonBoost for Image Understanding: Multi-Class Object Recognition and Segmentation by Jointly Modeling Texture, Layout, and Context, Shotton et.al. 2009

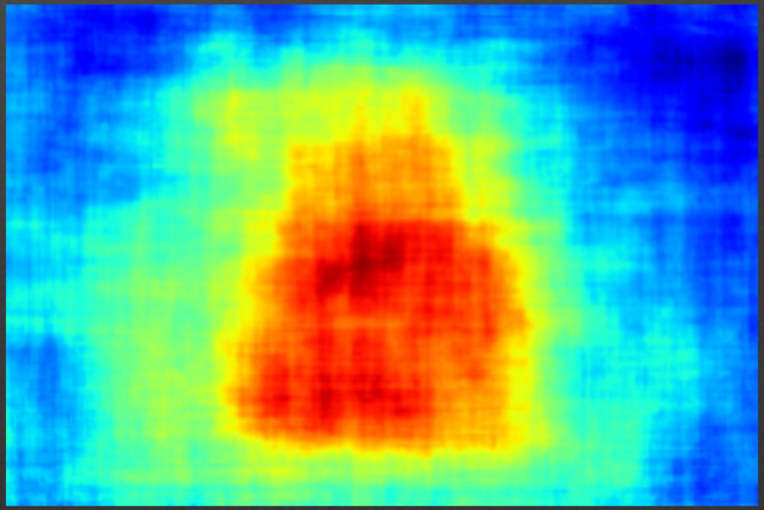
Classification

- Train classifier $\psi(l)$
    - for each class l
    - TextonBoost [1]

ψ(grass)



ψ(sheep)



[1] TextonBoost for Image Understanding: Multi-Class Object Recognition and Segmentation by Jointly Modeling Texture, Layout, and Context, Shotton et.al. 2009

Classification

- Train classifier *ψ(l)*
  - for each class l
  - TextonBoost [1]
- Pixels independent
  - noisy classification

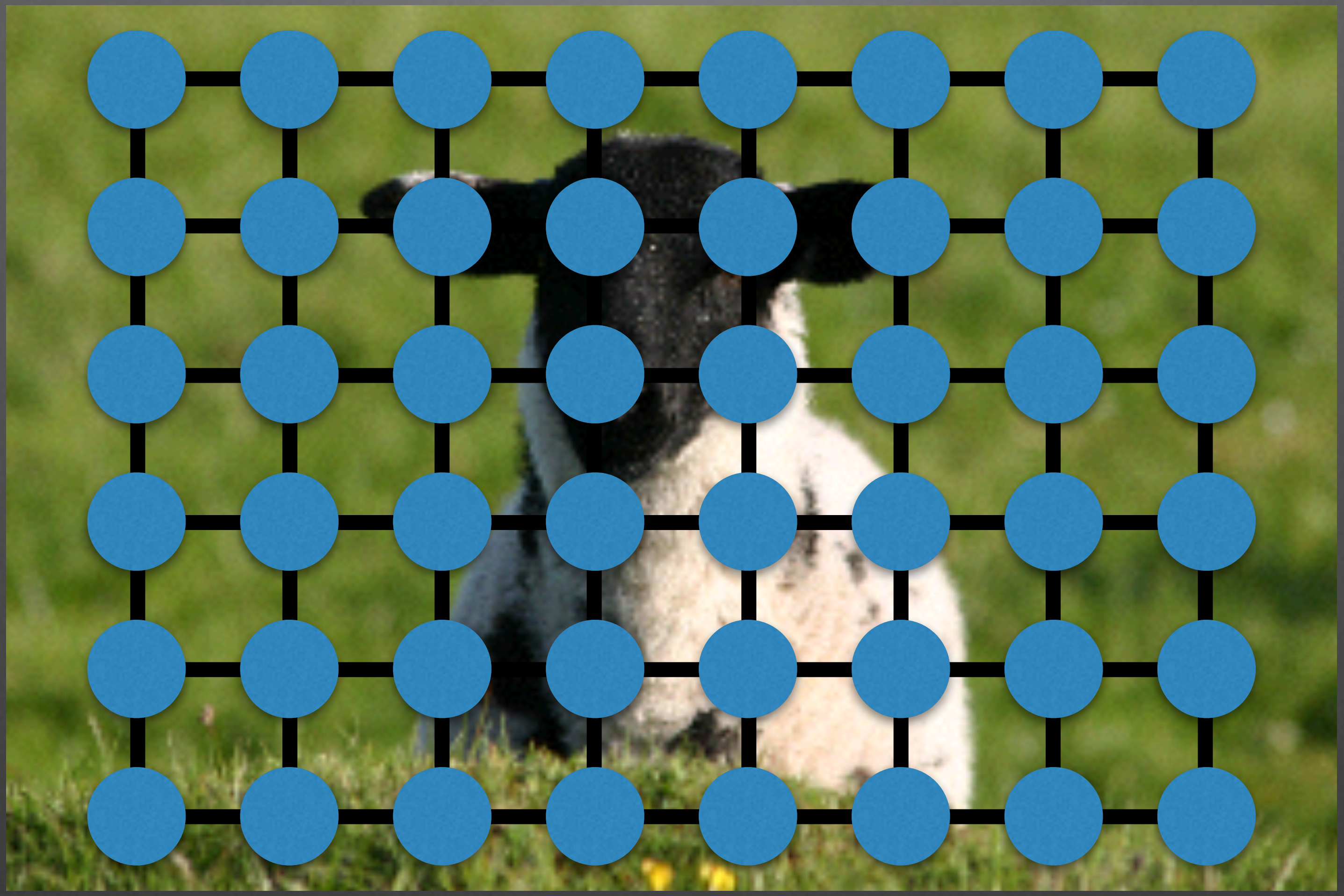
[1] TextonBoost for Image Understanding: Multi-Class Object Recognition and Segmentation by Jointly Modeling Texture, Layout, and Context, Shotton et.al. 2009

Classification

- Train classifier *ψ(l)*
  - for each class l
  - TextonBoost [1]
- Pixels independent
  - noisy classification
- Large regional context
  - inaccurate around boundaries

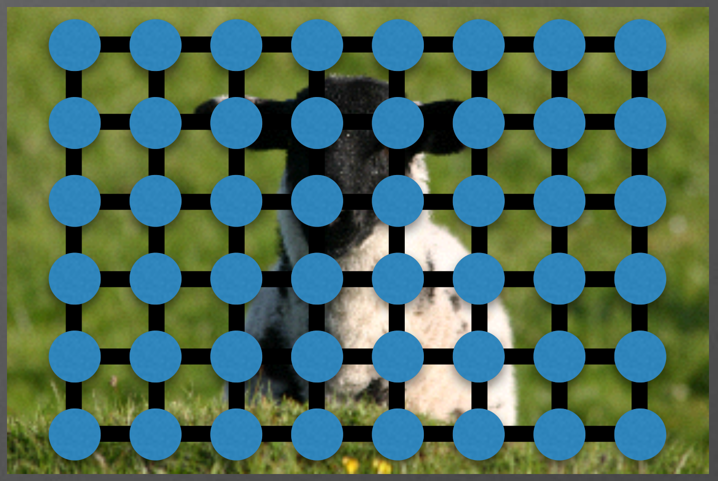
[1] TextonBoost for Image Understanding: Multi-Class Object Recognition and Segmentation by Jointly Modeling Texture, Layout, and Context, Shotton et.al. 2009

Random Field Models



Random Field Models

$$E(X) = \sum_{i} \psi_i(X_i) + \sum_{i,j \in N} \psi_{ij}(X_i, X_j)$$



unary term

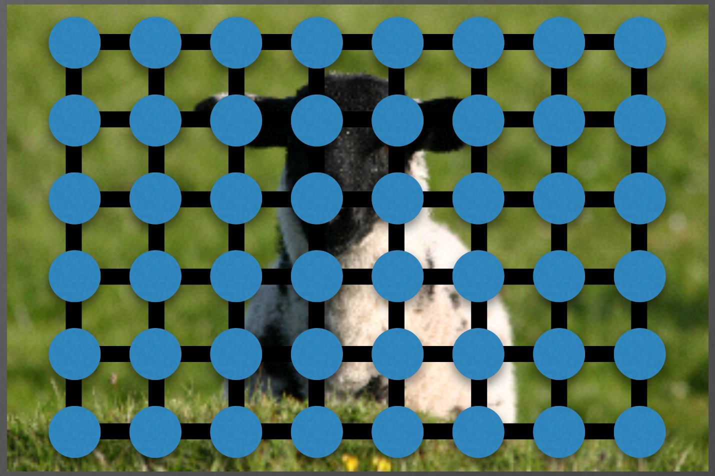
pairwise term

Random Field Models

$$E(X) = \sum_i \psi_i(X_i) + \sum_{i,j \in N} \psi_{ij}(X_i, X_j)$$

unary term

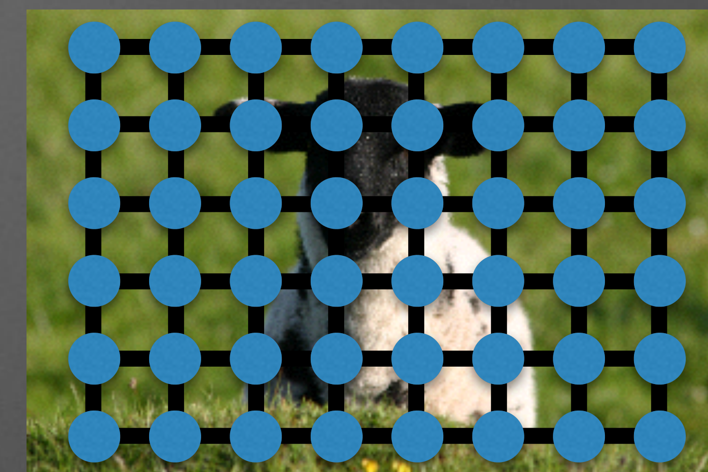
pairwise term



- Probabilistic interpretation $P(X) = \frac{1}{Z} \exp(-E(X))$

Random Field Models

$$E(X) = \sum_i \psi_i(X_i) + \sum_{i,j \in N} \psi_{ij}(X_i, X_j)$$



unary term

pairwise term

- Probabilistic interpretation $P(X) = \frac{1}{Z}\exp(-E(X))$
- MAP inference
  - most likely labeling
  - lowest energy

Random Field Models

$$E(X) = \sum_i \psi_i(X_i) + \sum_{i,j \in N} \psi_{ij}(X_i, X_j)$$

$\psi_{ij}(X_i, X_j) = 0$

unary term

Random Field Models

$$E(X) = \sum_i \psi_i(X_i) + \sum_{i,j \in N} \psi_{ij}(X_i, X_j)$$

$\psi_{ij}(X_i, X_j) = [X_i \neq X_j]$

conditional random field

Random Field Models

$$E(X) = \sum_i \psi_i(X_i) + \sum_{i,j \in N} \psi_{ij}(X_i, X_j)$$

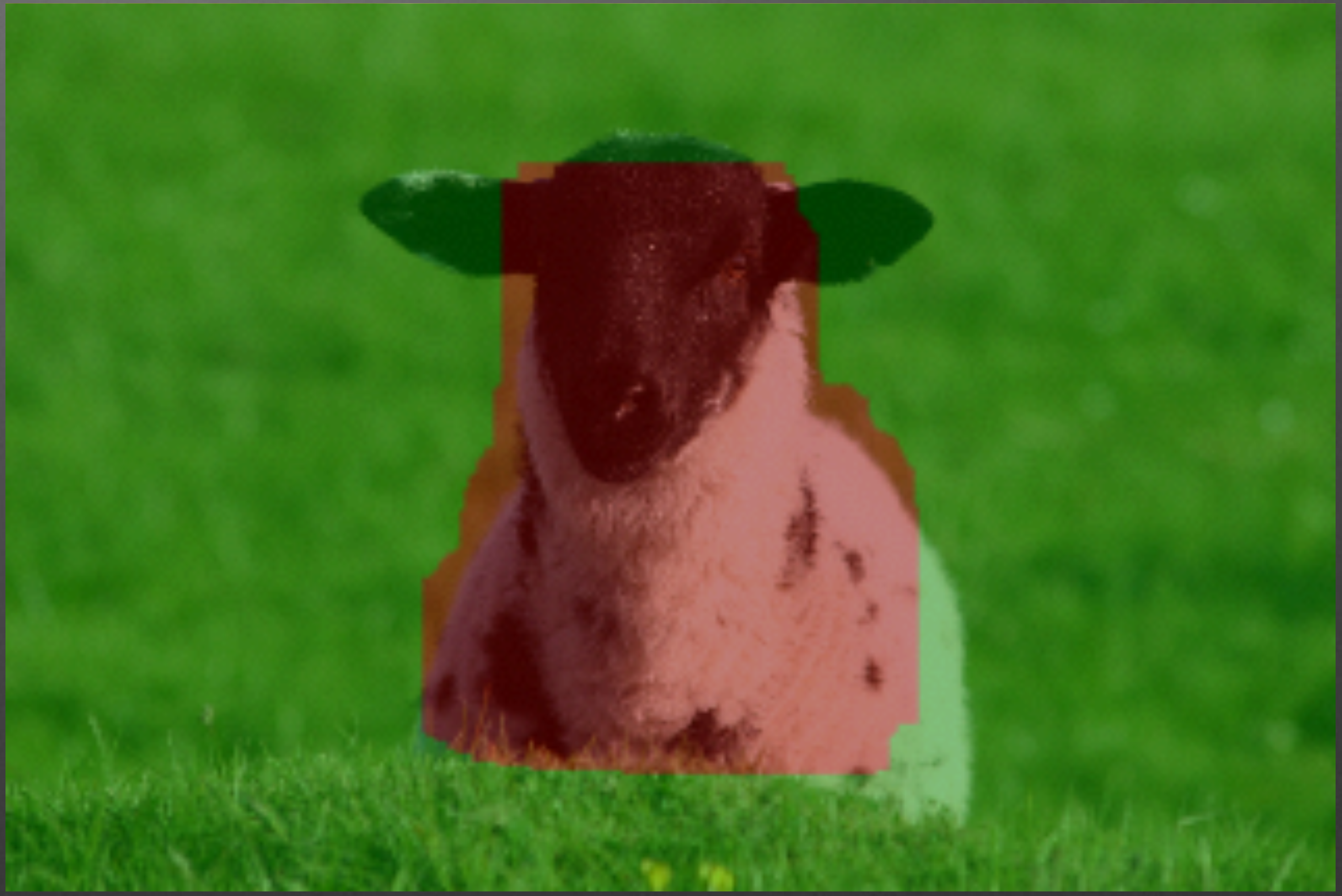
$\psi_{ij}(X_i,X_j) = 100[X_i \neq X_j]$

conditional random field

Random Field Models

$$E(X) = \sum_i \psi_i(X_i) + \sum_{i,j \in N} \psi_{ij}(X_i, X_j)$$

$\psi_{ij}(X_i,X_j) = 100[X_i \neq X_j]$



conditional random field

Random Field Models

$$E(X) = \sum_i \psi_i(X_i) + \sum_{i,j \in N} \psi_{ij}(X_i, X_j)$$

$\psi_{ij}(X_i,X_j) = w_{ij}[X_i \neq X_j]$

$w_{ij}=\exp(-\alpha(c_i-c_j)^2)$

conditional random field

Random Field Models

$$E(X) = \sum_i \psi_i(X_i) + \sum_{i,j \in N} \psi_{ij}(X_i, X_j)$$

$\psi_{ij}(X_i,X_j) = w_{ij}[X_i \neq X_j]$

$w_{ij}=\exp(-\alpha(c_i-c_j)^2)$

weight horizontal

Random Field Models

$$E(X) = \sum_i \psi_i(X_i) + \sum_{i,j \in N} \psi_{ij}(X_i, X_j)$$

$\psi_{ij}(X_i,X_j) = w_{ij}[X_i \neq X_j]$

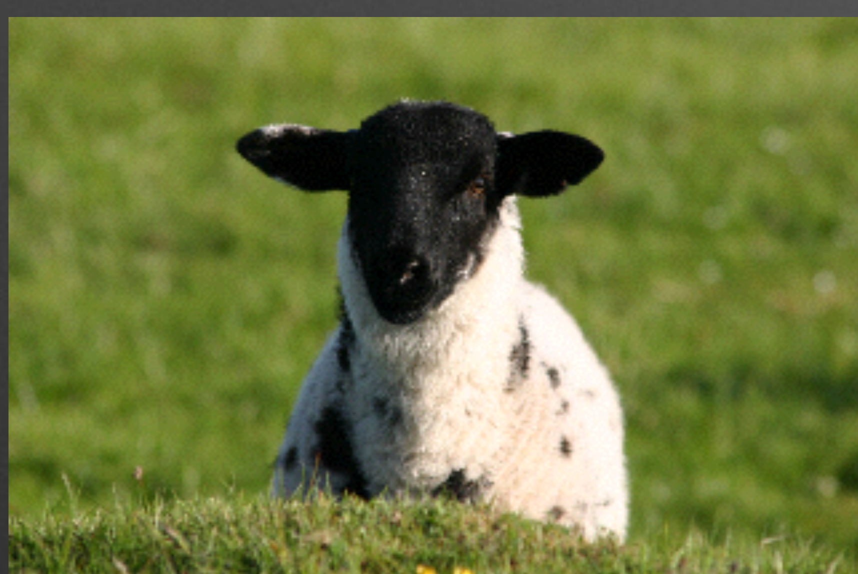
$w_{ij} = \exp(-\alpha(c_i - c_j)^2)$

weight vertical

Random Field Models

$$E(X) = \sum_i \psi_i(X_i) + \sum_{i,j \in N} \psi_{ij}(X_i, X_j)$$

$\psi_{ij}(X_i,X_j) = 100w_{ij}[X_i \neq X_j]$



conditional random field
color sensitive

Random Field Models

Random Field Models

Pros:

Random Field Models

Pros:

- Probabilistic interpretation

Random Field Models

Pros:

- Probabilistic interpretation
- Parameter learning

Random Field Models

Pros:

- Probabilistic interpretation
- Parameter learning
- Combine with other models

Random Field Models

Pros:

- Probabilistic interpretation
- Parameter learning
- Combine with other models

Cons:

Random Field Models

Pros:

- Probabilistic interpretation
- Parameter learning
- Combine with other models

Cons:

- Shrinking bias

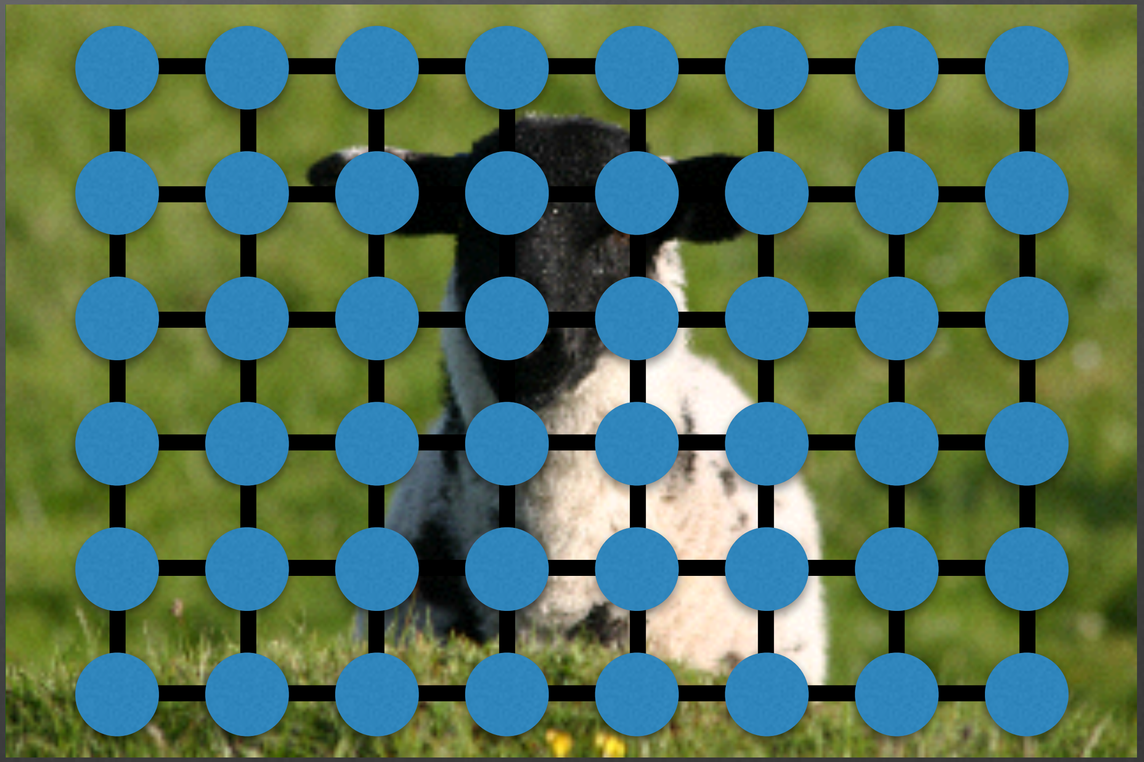
Random Field Models

Pros:

- Probabilistic interpretation
- Parameter learning
- Combine with other models

Cons:

- Shrinking bias
- Models only local interactions



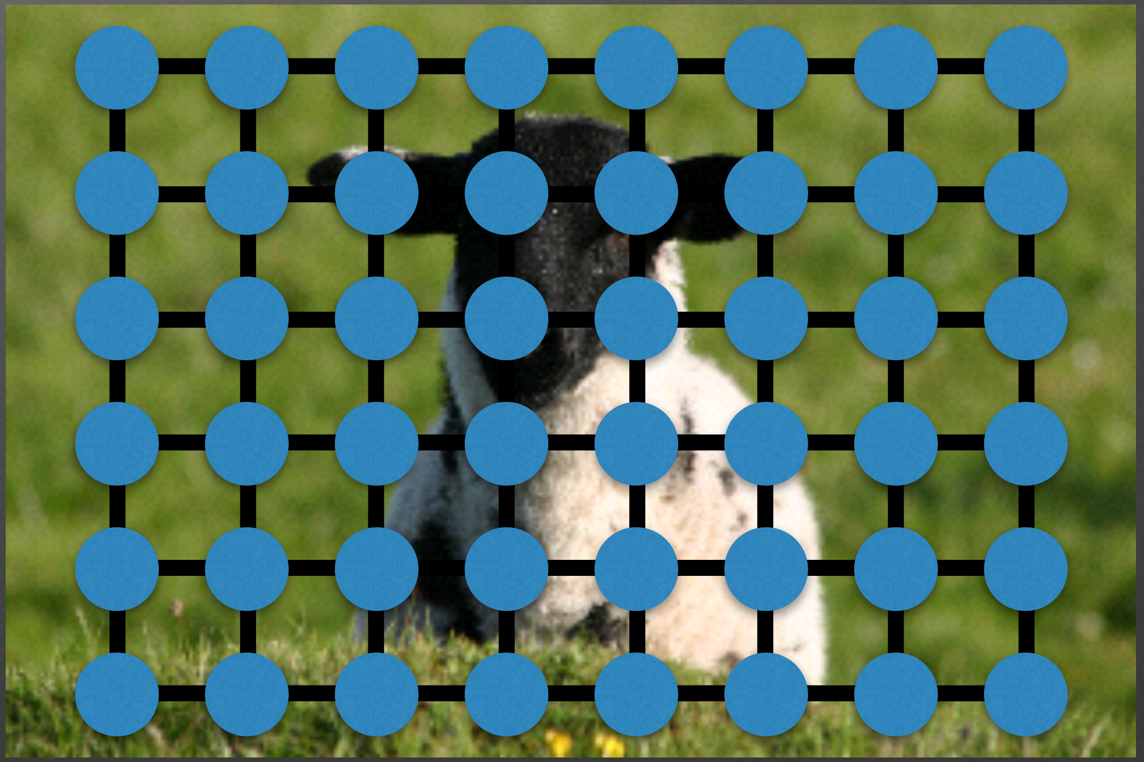
Random Field Models

Pros:

- Probabilistic interpretation
- Parameter learning
- Combine with other models

Cons:

- Shrinking bias
- Models only local interactions
  - hard for information to propagate

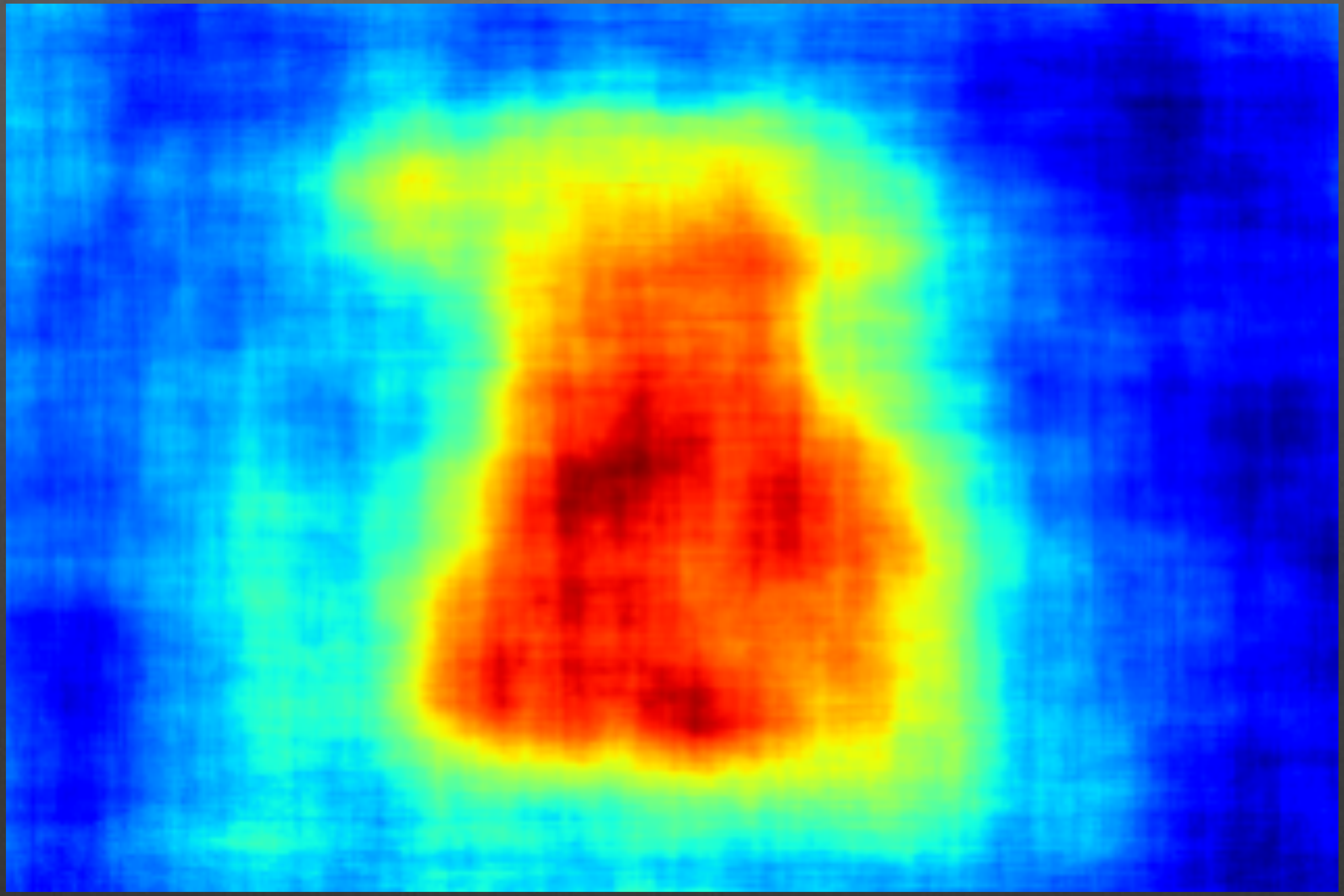


Filtering

classifier labeling

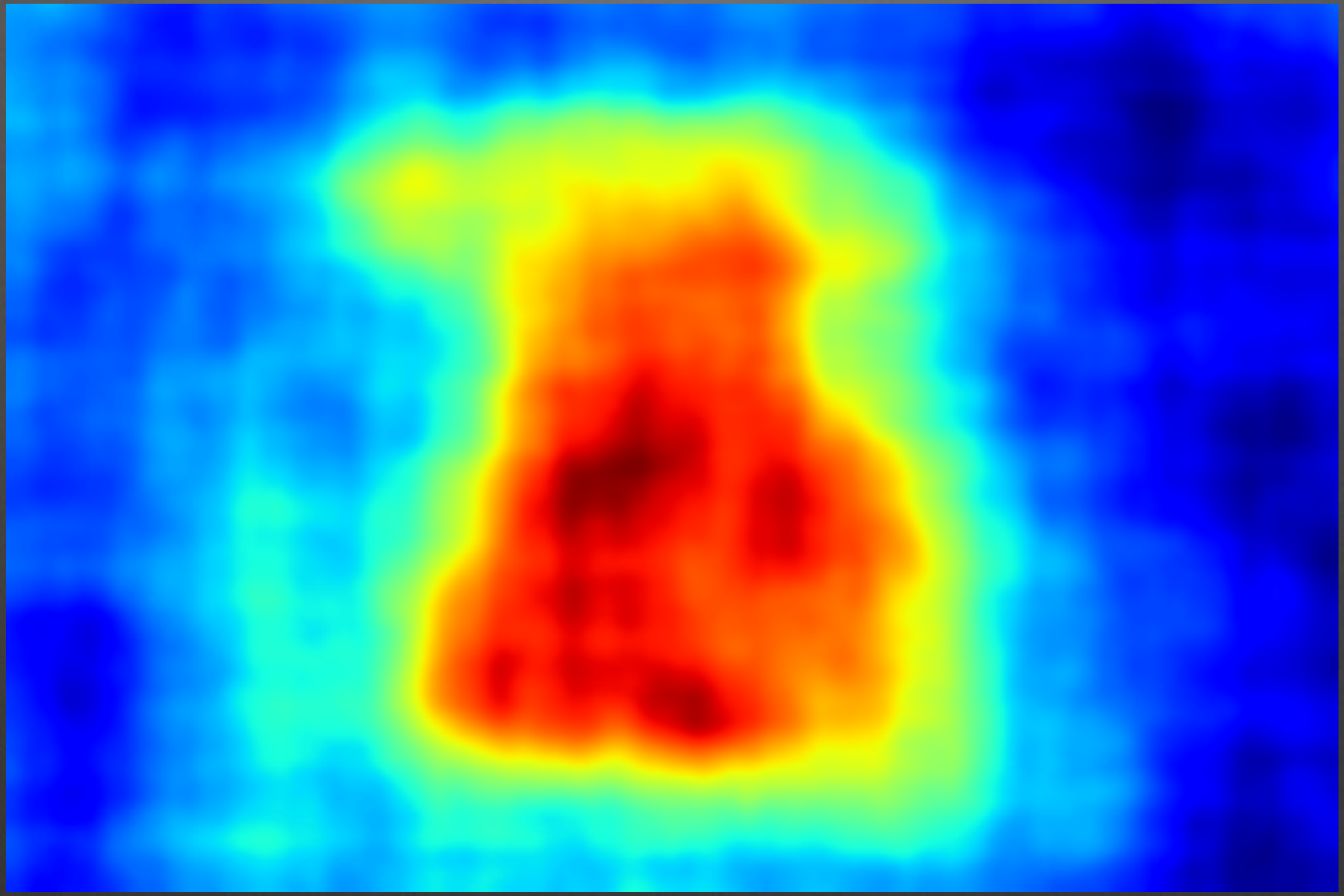
Filtering

classifier log likelihood



Filtering

blurred log likelihood
Gaussian σ_s=2px



Filtering

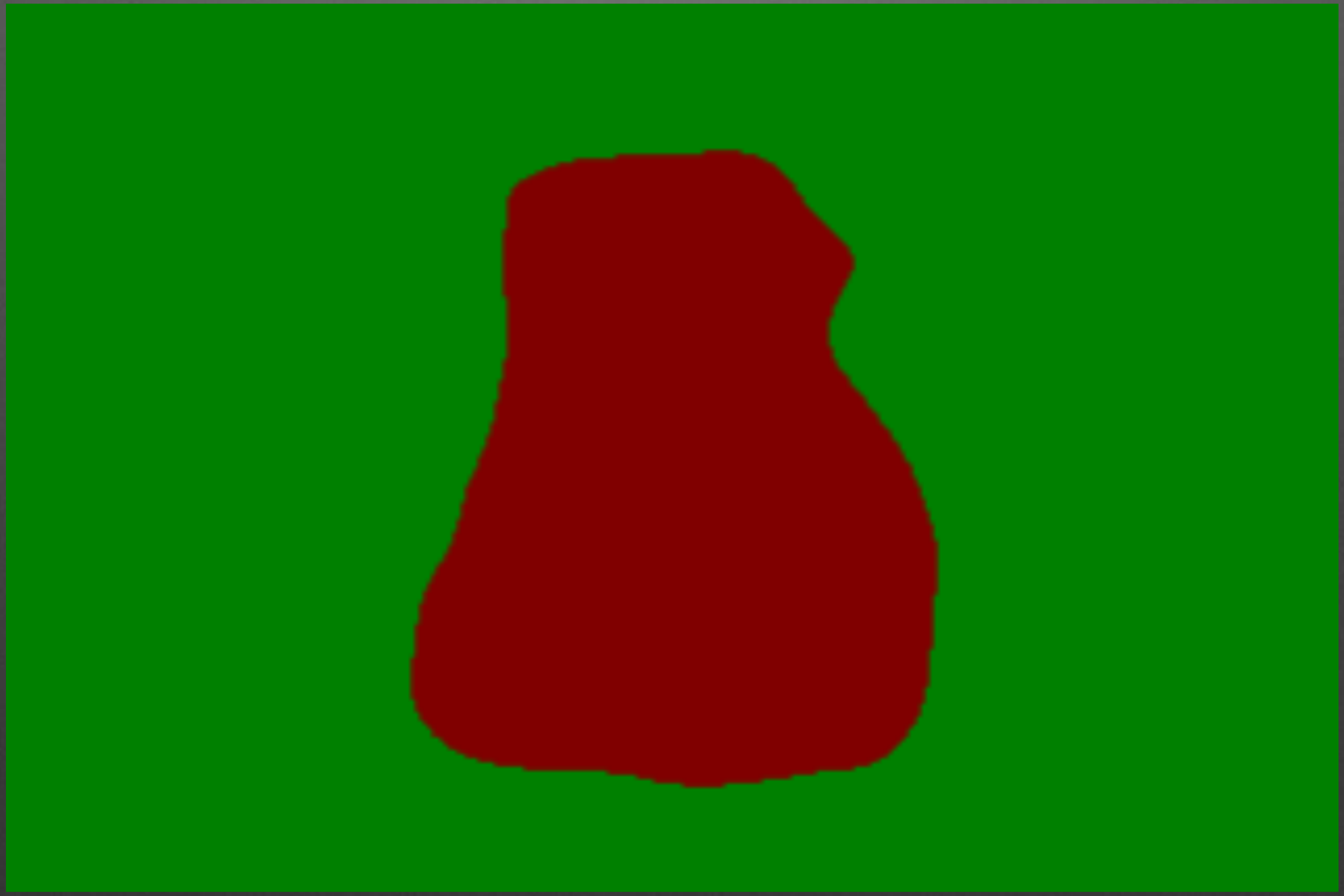
blurred labeling

Gaussian σ_s=2px

Filtering

blurred labeling

Gaussian σ_s=6px



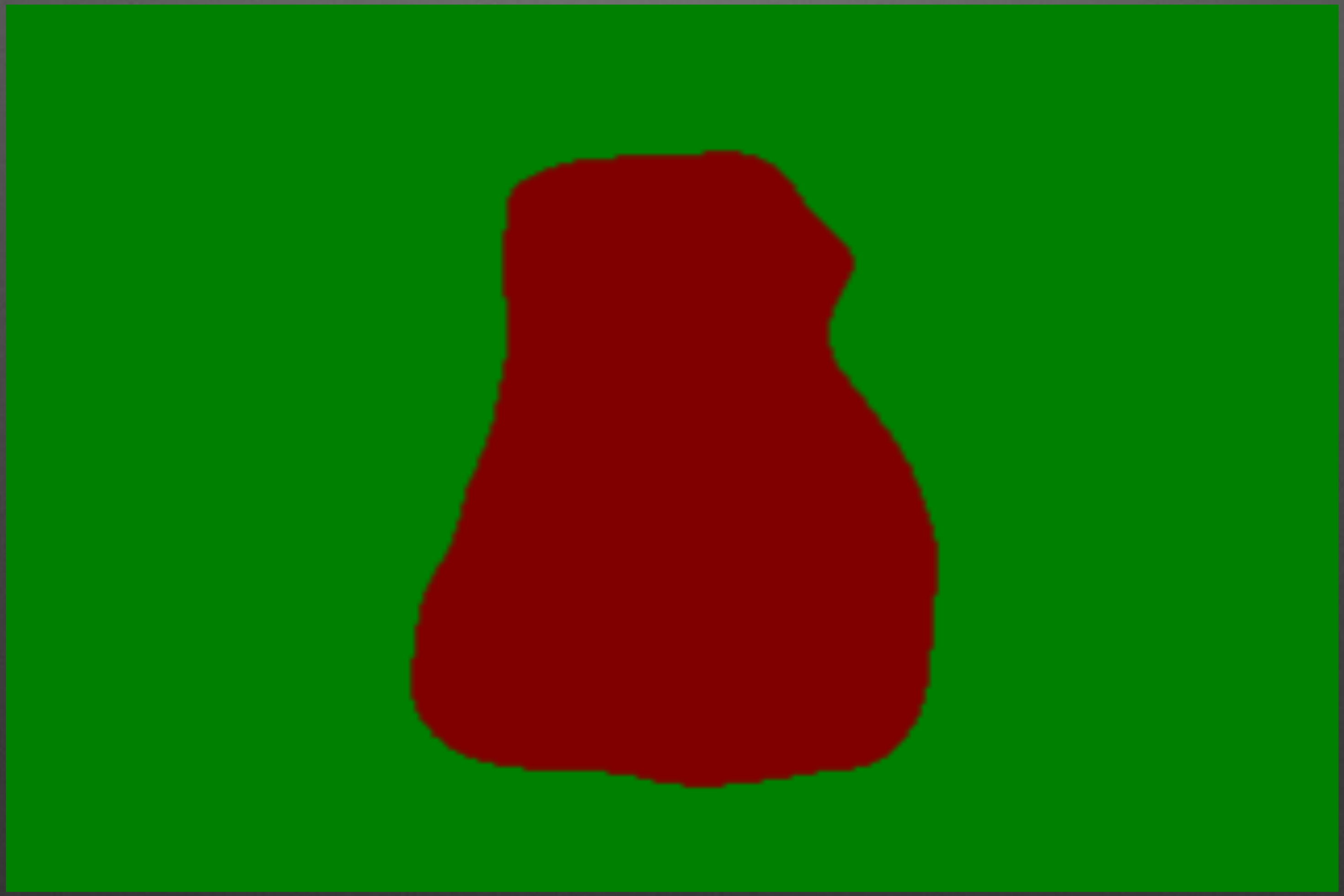
Filtering

Conditional Random Field (CRF)

Filtering

blurred labeling

Gaussian σ_s=6px



Filtering

blurred labeling

Bilateral σ_s=60px σ_c=15

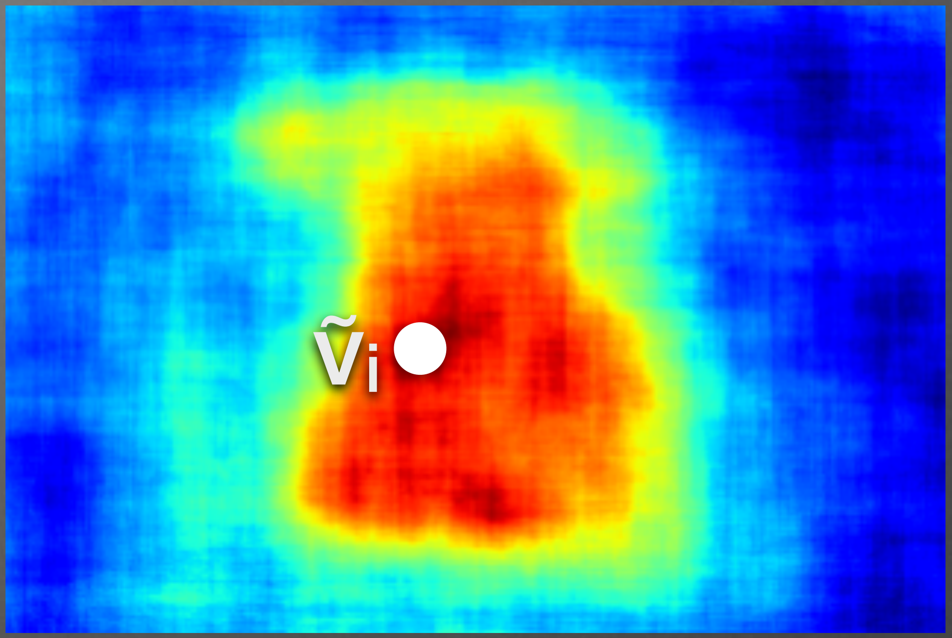
Filtering

Conditional Random Field (CRF)
color sensitive

Filtering

$$\tilde{v}_i = \sum_j w_{ij} v_j$$

$w_{ij} = \exp\left(-(s_i - s_j)^2/\sigma_s\right) \exp\left(-(c_i - c_j)^2/\sigma_c\right)$

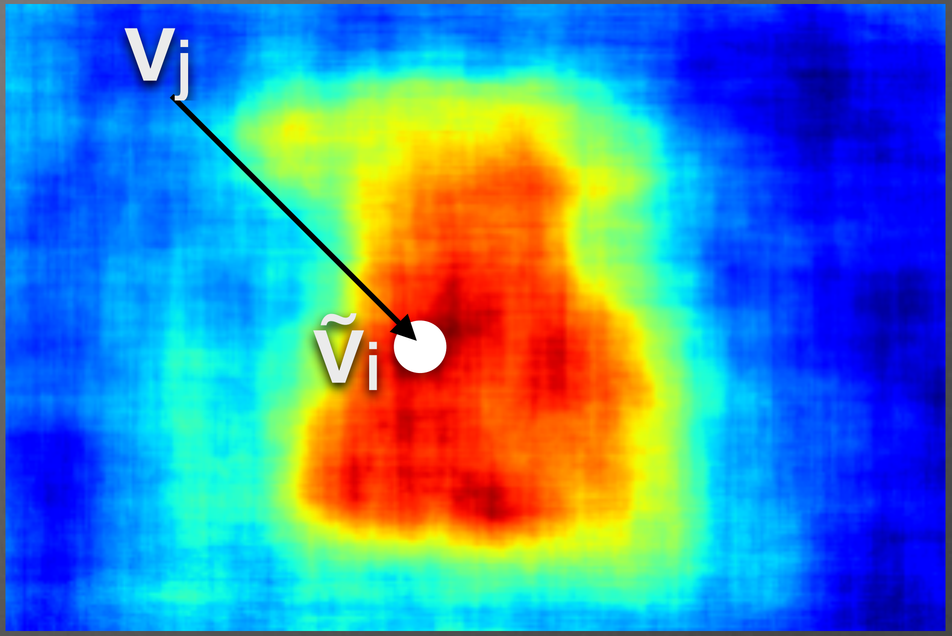


[2] Fast High-Dimensional Filtering Using the Permutohedral Lattice, Adams et.al. 2010

Filtering

$$\tilde{v}_i = \sum_j w_{ij} v_j$$

$w_{ij} = \exp(-(s_i-s_j)^2/\sigma_s)\ \exp(-(c_i-c_j)^2/\sigma_c)$

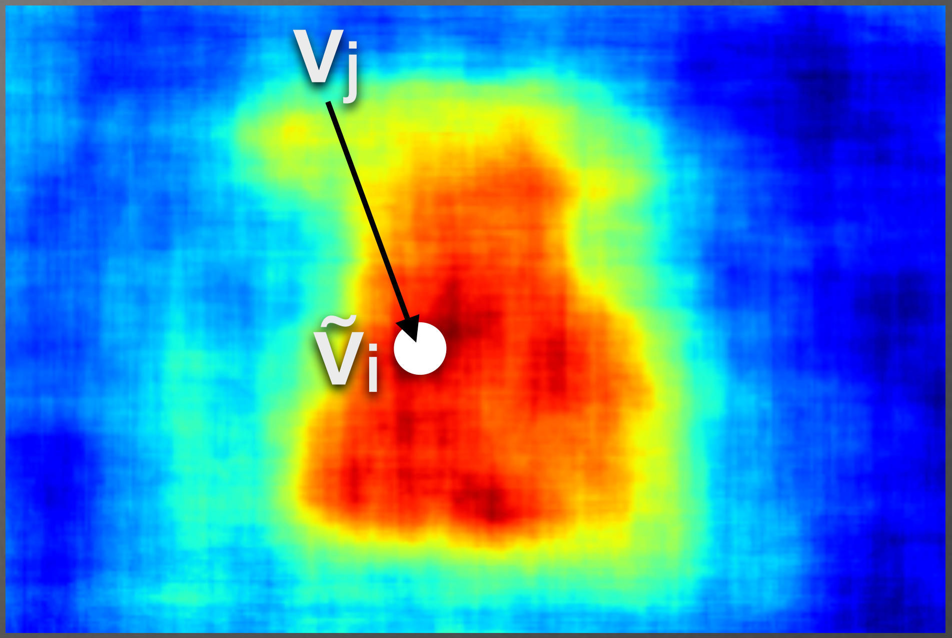


[2] Fast High-Dimensional Filtering Using the Permutohedral Lattice, Adams et.al. 2010

Filtering

$$\tilde{v}_i = \sum_j w_{ij} v_j$$

$w_{ij} = \exp(-(s_i - s_j)^2/\sigma_s)\ \exp(-(c_i - c_j)^2/\sigma_c)$

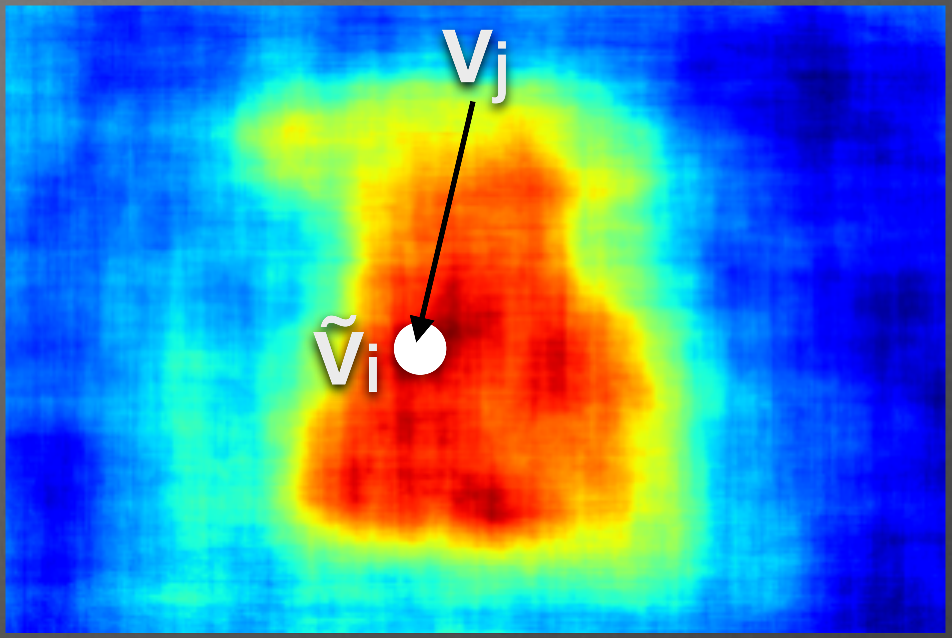


[2] Fast High-Dimensional Filtering Using the Permutohedral Lattice, Adams et.al. 2010

Filtering

$$\tilde{v}_i = \sum_j w_{ij} v_j$$

$w_{ij} = \exp(-(s_i-s_j)^2/\sigma_s)\ \exp(-(c_i-c_j)^2/\sigma_c)$

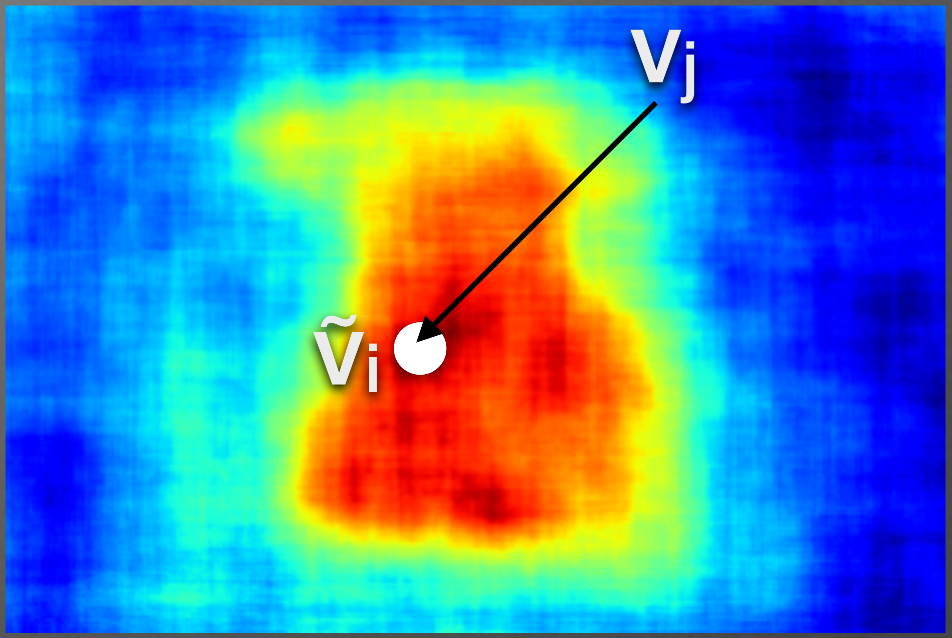


[2] Fast High-Dimensional Filtering Using the Permutohedral Lattice, Adams et.al. 2010

Filtering

$$\tilde{v}_i = \sum_j w_{ij} v_j$$

$w_{ij} = \exp(-(s_i - s_j)^2/\sigma_s)\ \exp(-(c_i - c_j)^2/\sigma_c)$

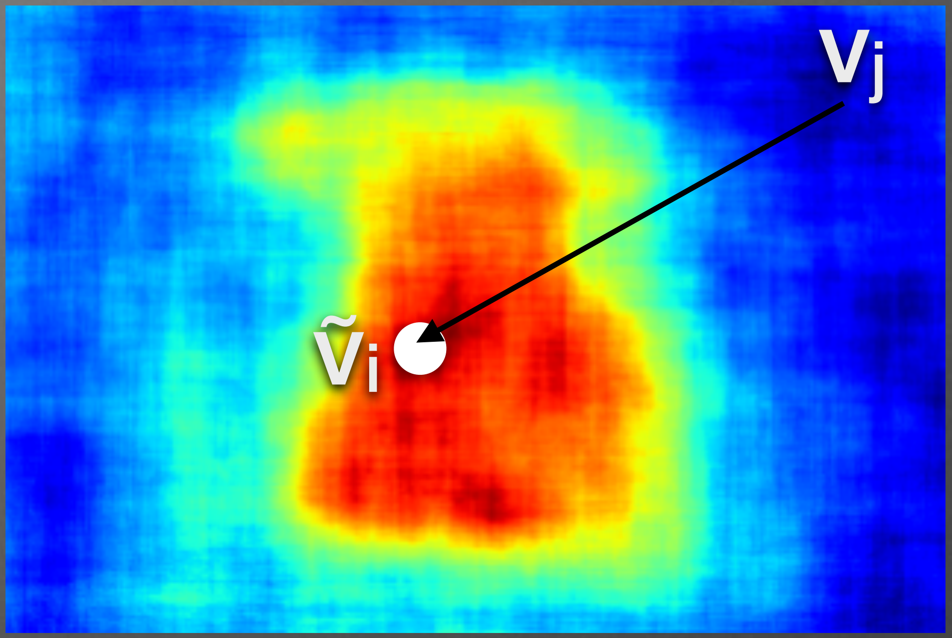


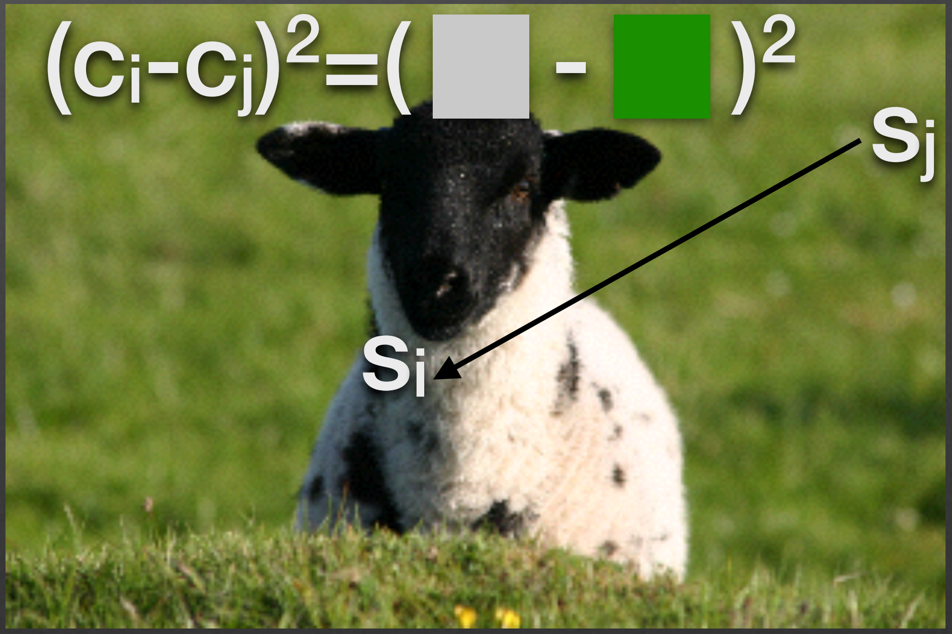
[2] Fast High-Dimensional Filtering Using the Permutohedral Lattice, Adams et.al. 2010

Filtering

$$\tilde{v}_i = \sum_j w_{ij} v_j$$

$w_{ij} = \exp(-(s_i - s_j)^2/\sigma_s)\ \exp(-(c_i - c_j)^2/\sigma_c)$



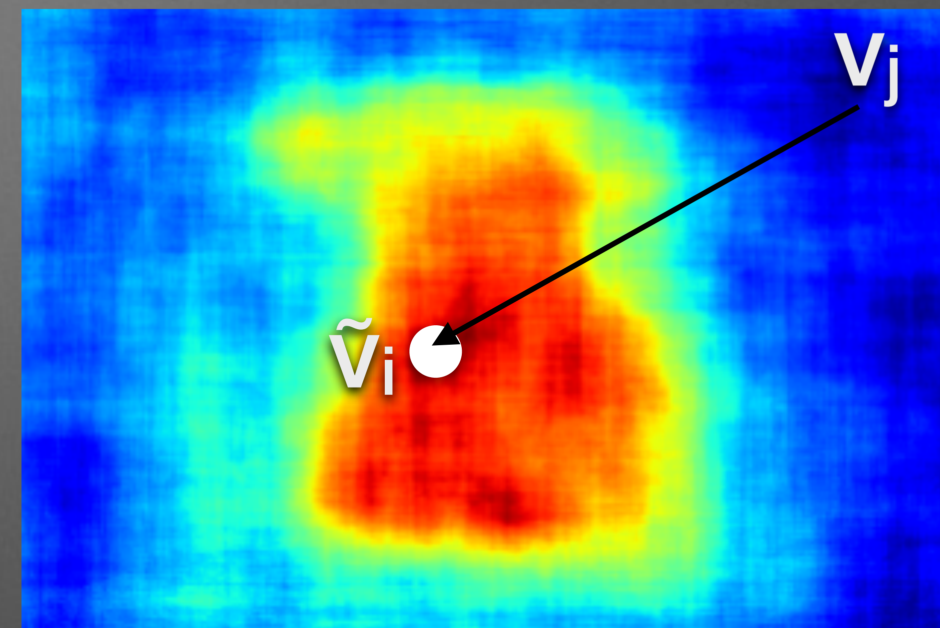


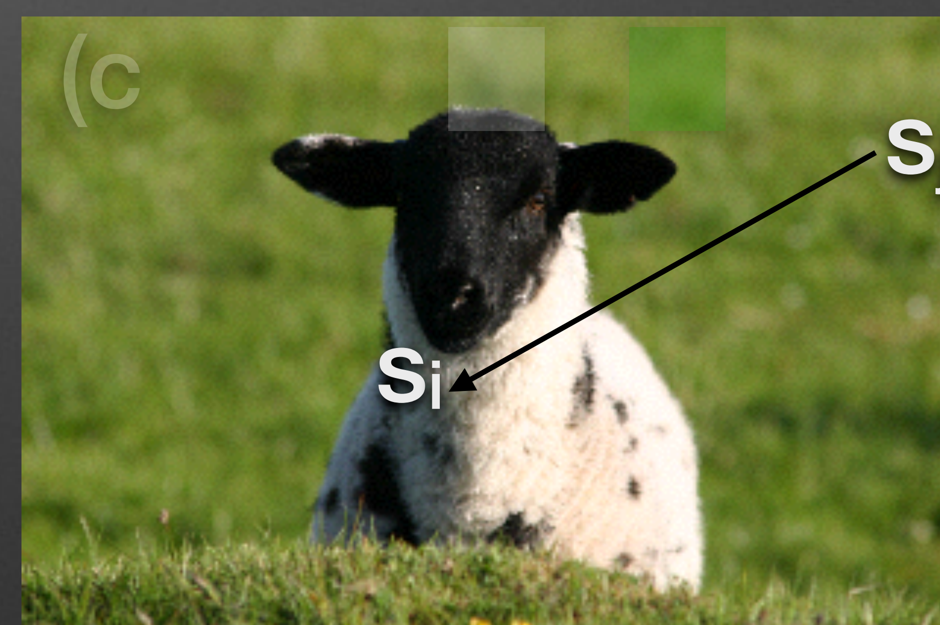
[2] Fast High-Dimensional Filtering Using the Permutohedral Lattice, Adams et.al. 2010

Filtering

$$\tilde{v}_i = \sum_j w_{ij} v_j$$

w $\exp(-(s_i-s_j)^2/\sigma_s)$ exp



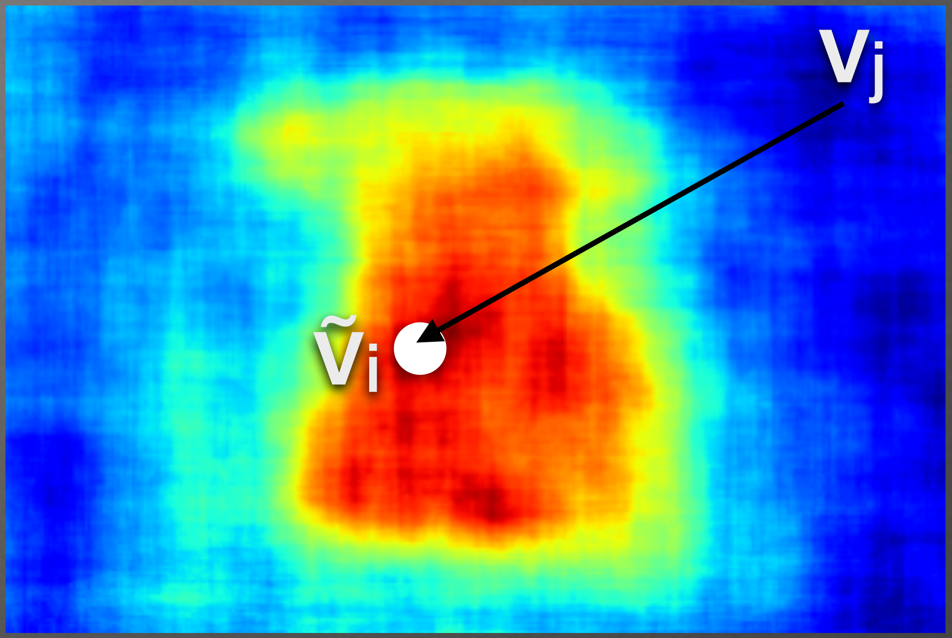


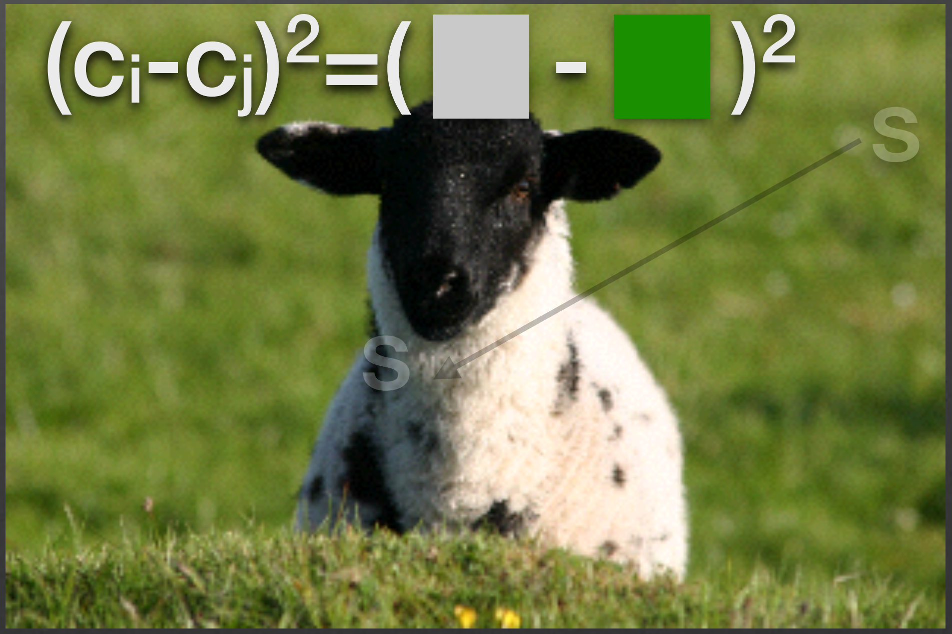
[2] Fast High-Dimensional Filtering Using the Permutohedral Lattice, Adams et.al. 2010

Filtering

$$\tilde{v}_i = \sum_j w_{ij} v_j$$

w exp $\exp(-(c_i - c_j)^2/\sigma_c)$





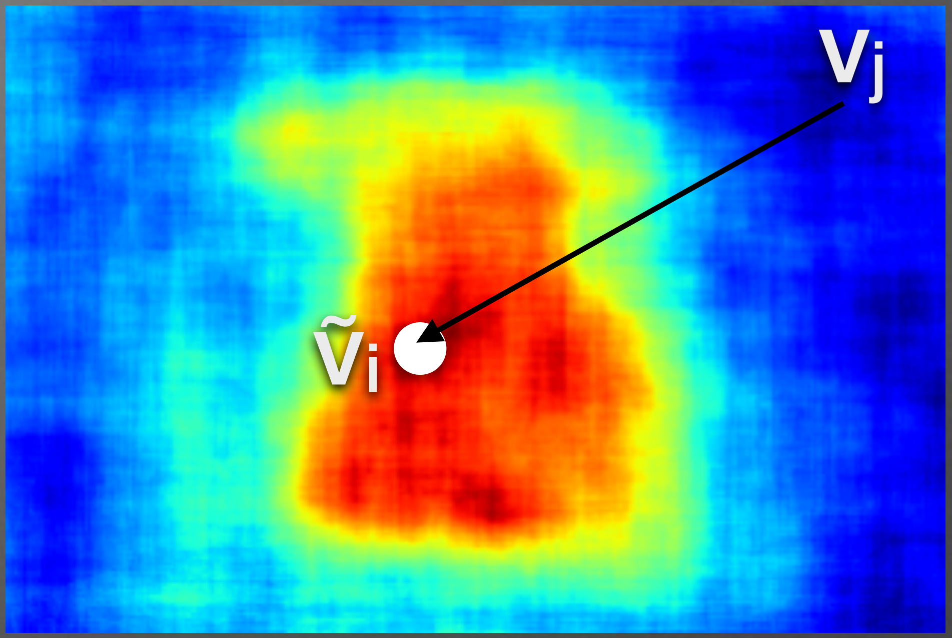
[2] Fast High-Dimensional Filtering Using the Permutohedral Lattice, Adams et.al. 2010

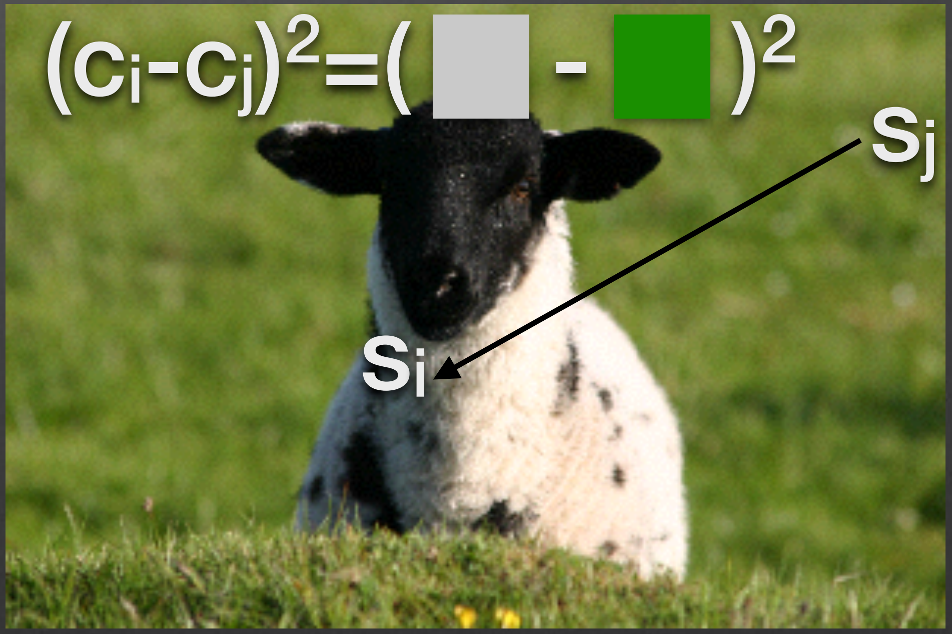
Filtering

$$\tilde{v}_i = \sum_j w_{ij} v_j$$

$w_{ij} = \exp(-(s_i - s_j)^2/\sigma_s)\ \exp(-(c_i - c_j)^2/\sigma_c)$

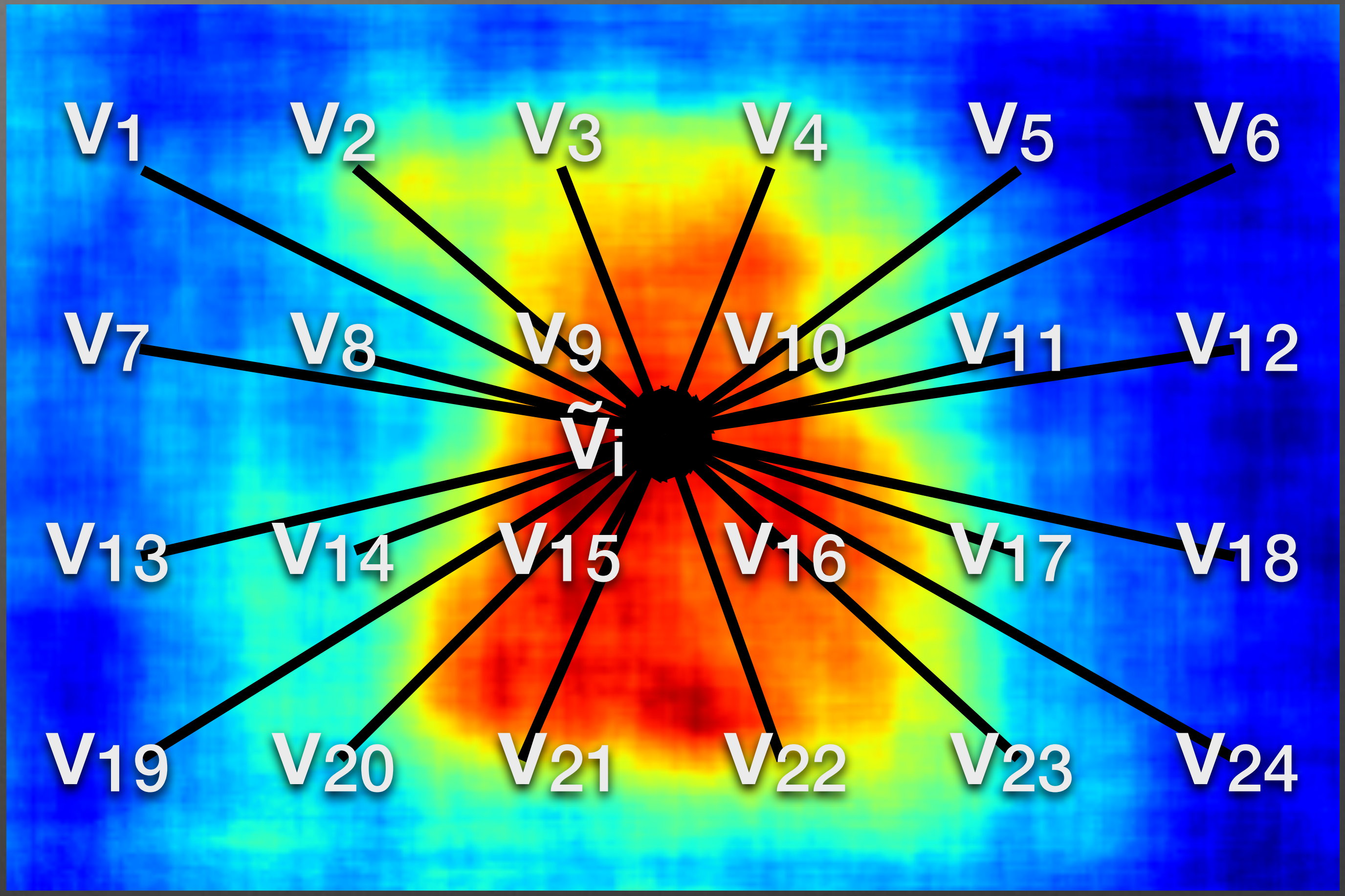
- Efficient convolution
  - Permutohedral lattice [2]
  - compute all $\tilde{v}_i$ in linear time
  - 50-100ms / image





[2] Fast High-Dimensional Filtering Using the Permutohedral Lattice, Adams et.al. 2010

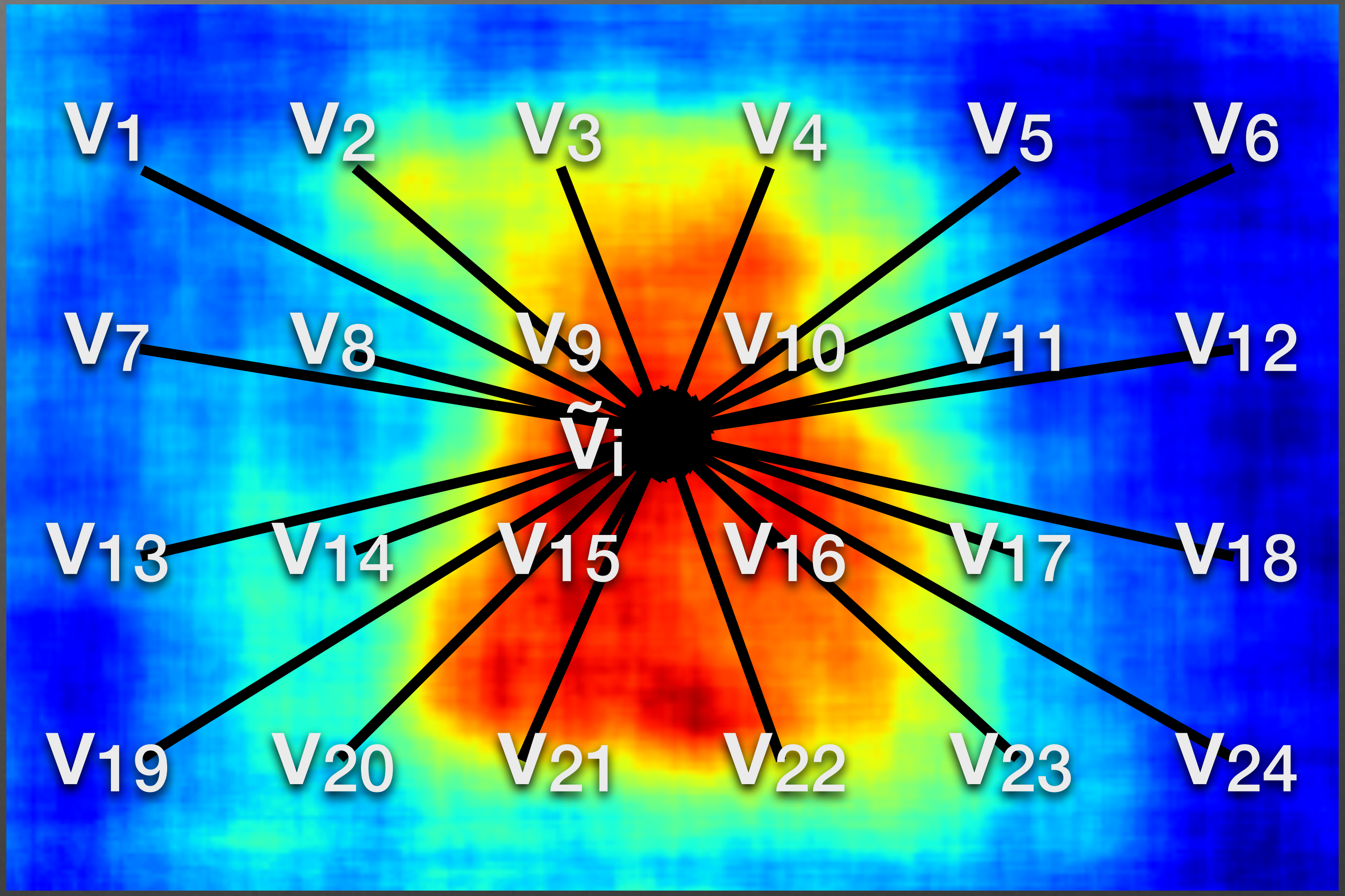
Filtering



v1
v2
v3
v4
v5
v6
v7
v8
v9
v10
v11
v12
ṽi
v13
v14
v15
v16
v17
v18
v19
v20
v21
v22
v23
v24

Filtering

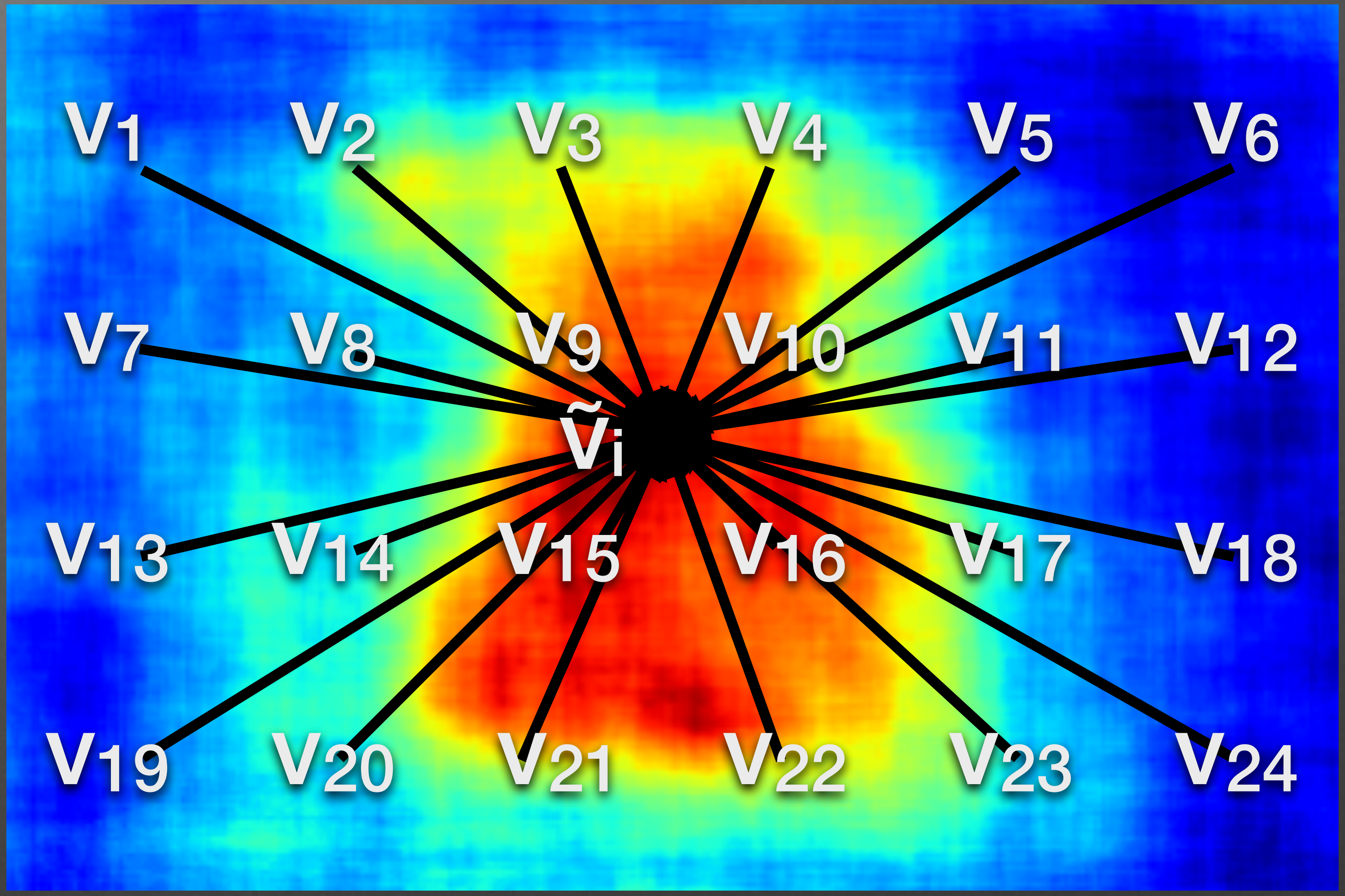
Pros:



Filtering

Pros:

- Propagates information over large distances
    - up to 1/3 of image

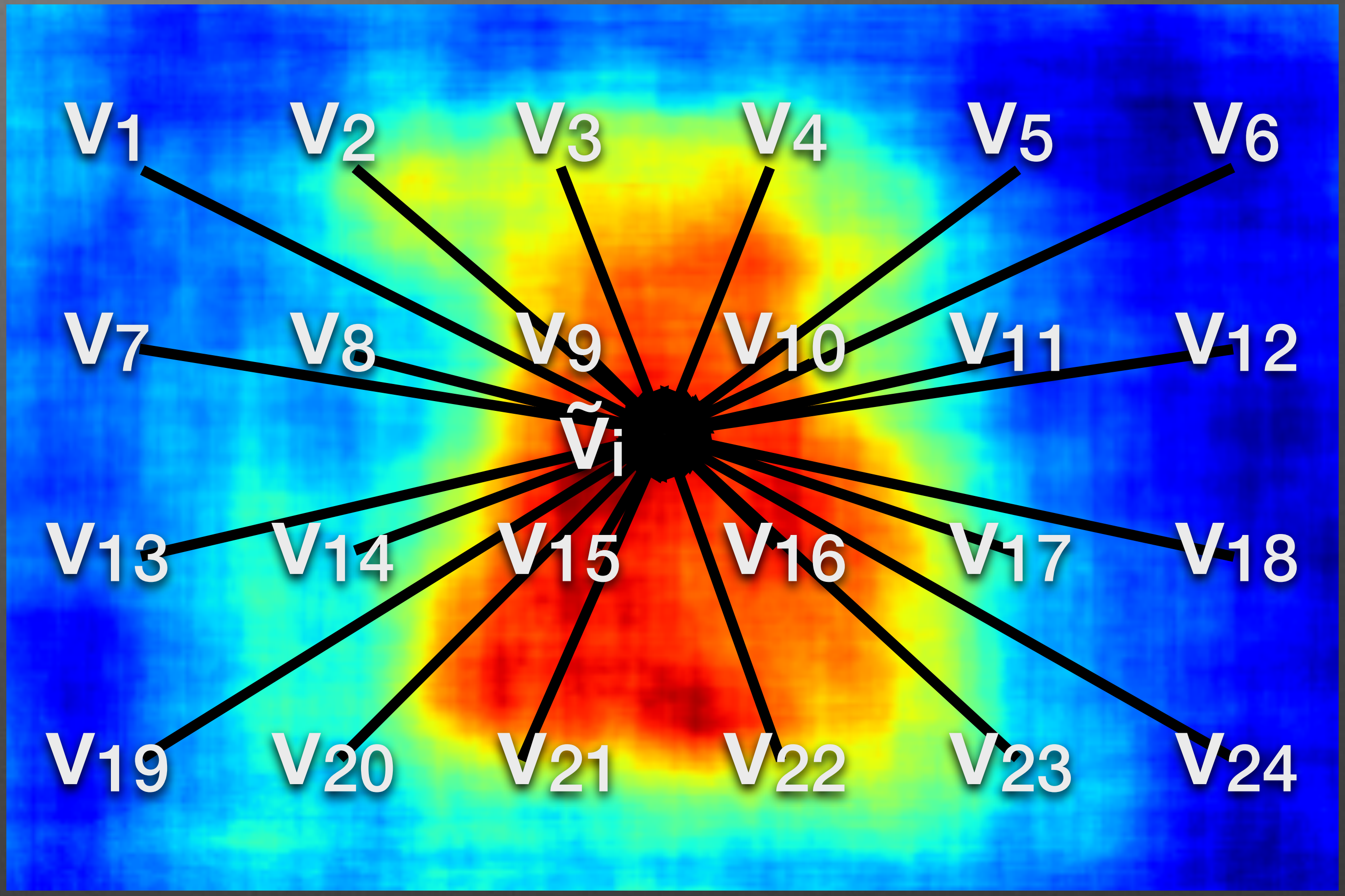


Filtering

Pros:

- Propagates information over large distances
  - up to 1/3 of image

Cons:



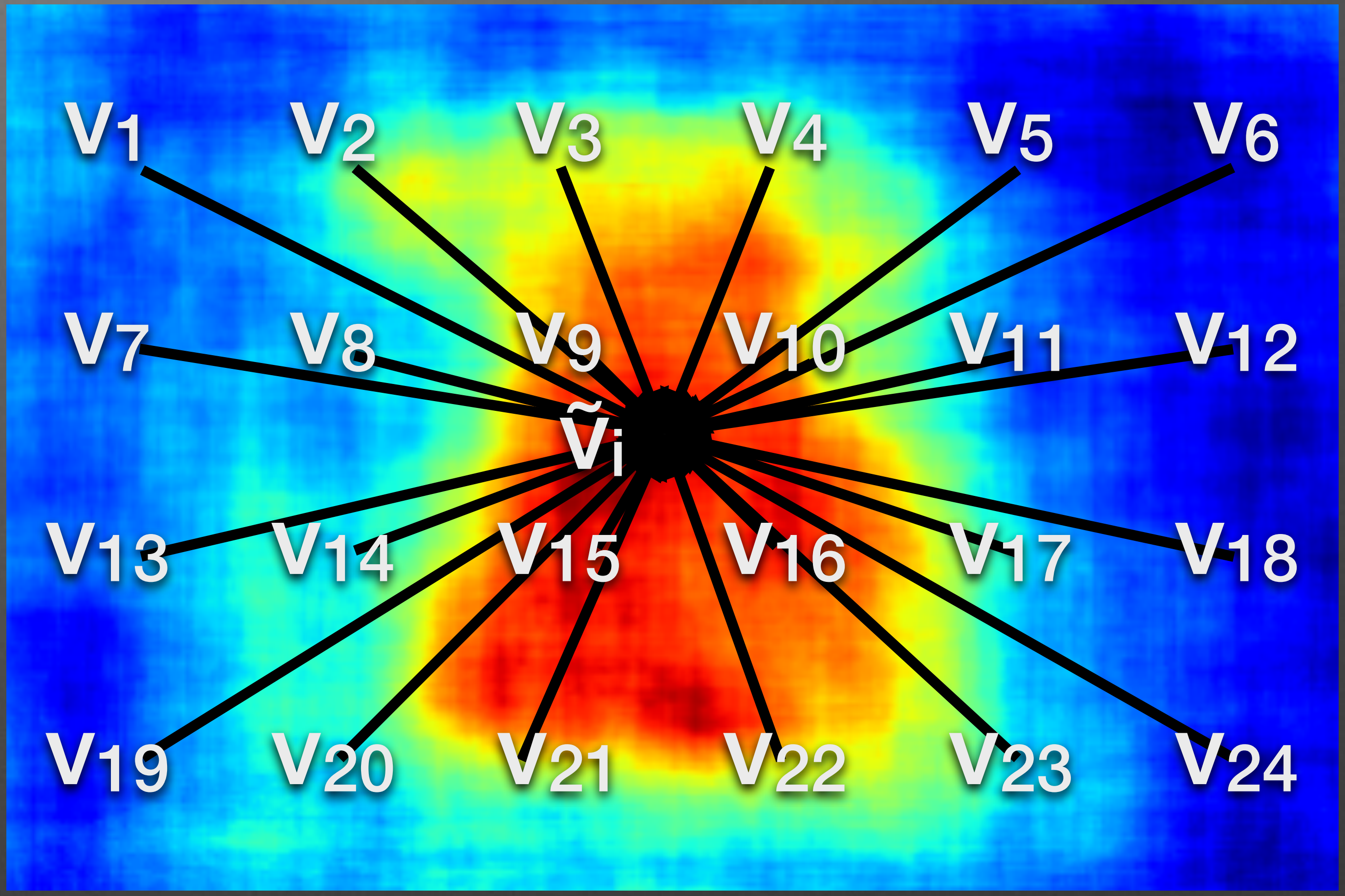
Filtering

Pros:

- Propagates information over large distances
  - up to 1/3 of image

Cons:

- No probabilistic interpretation



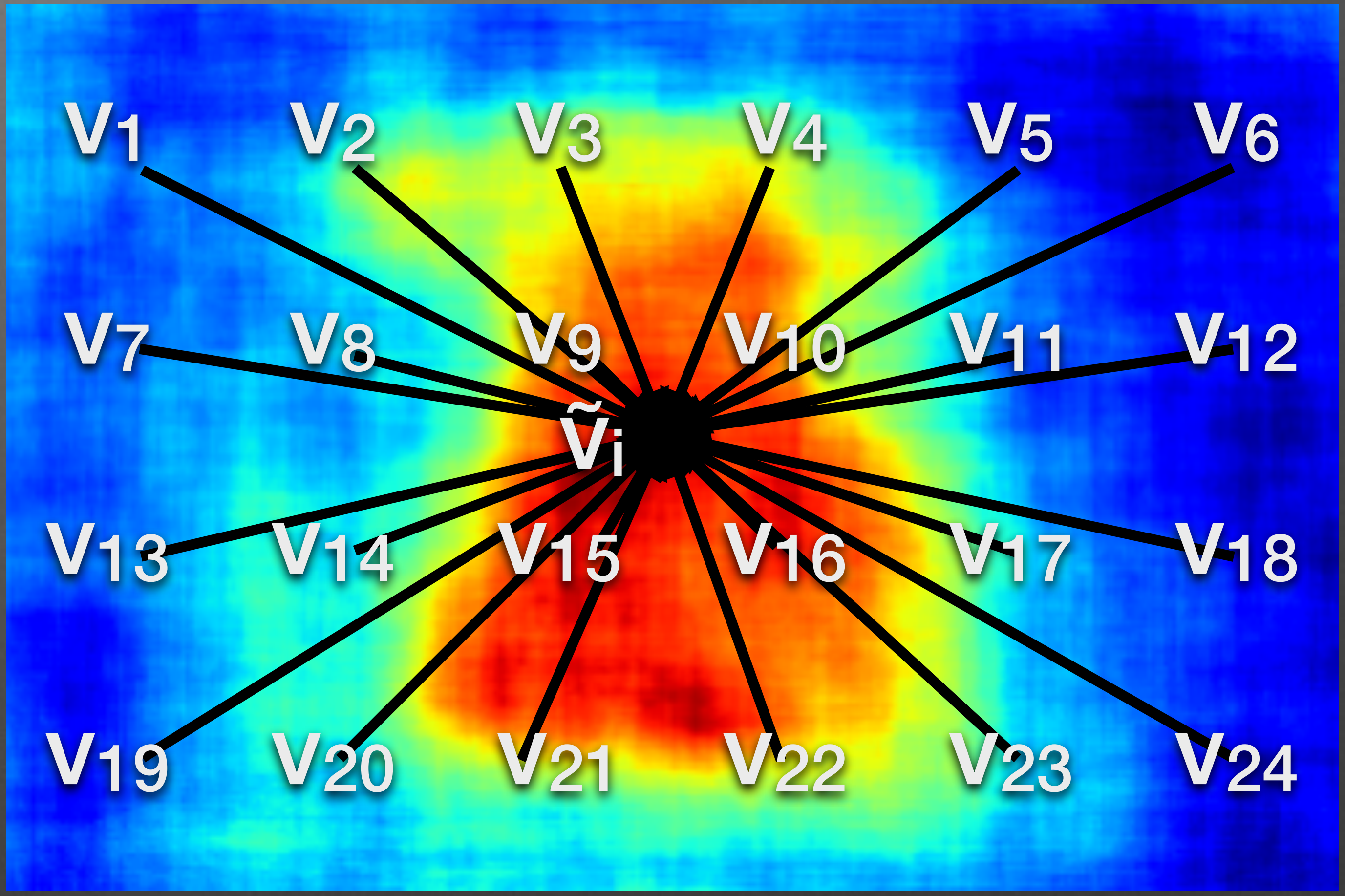
Filtering

Pros:

- Propagates information over large distances
  - up to 1/3 of image

Cons:

- No probabilistic interpretation
- No joint inference



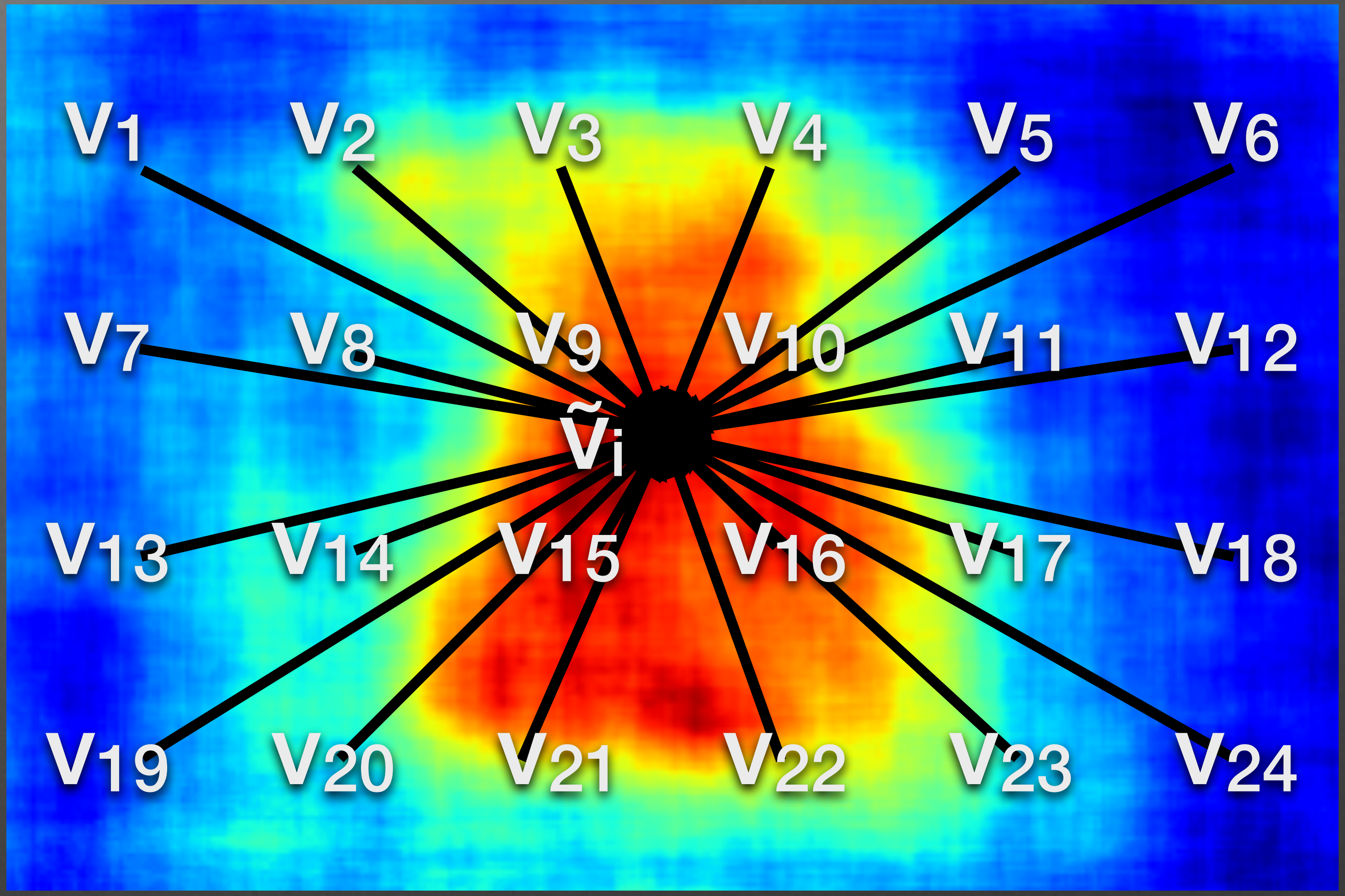
Filtering

Pros:

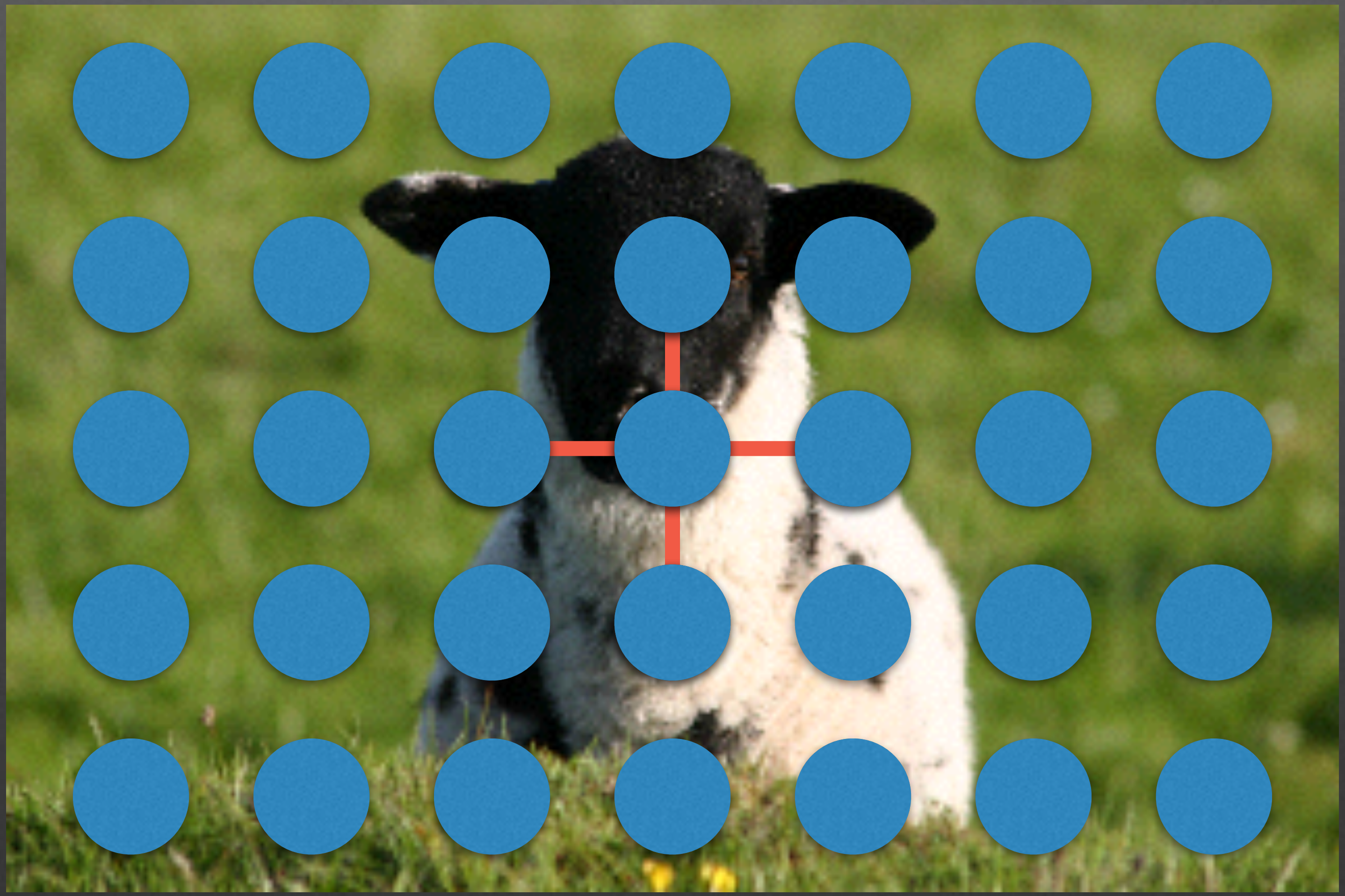
- Propagates information over large distances
    - up to 1/3 of image

Cons:

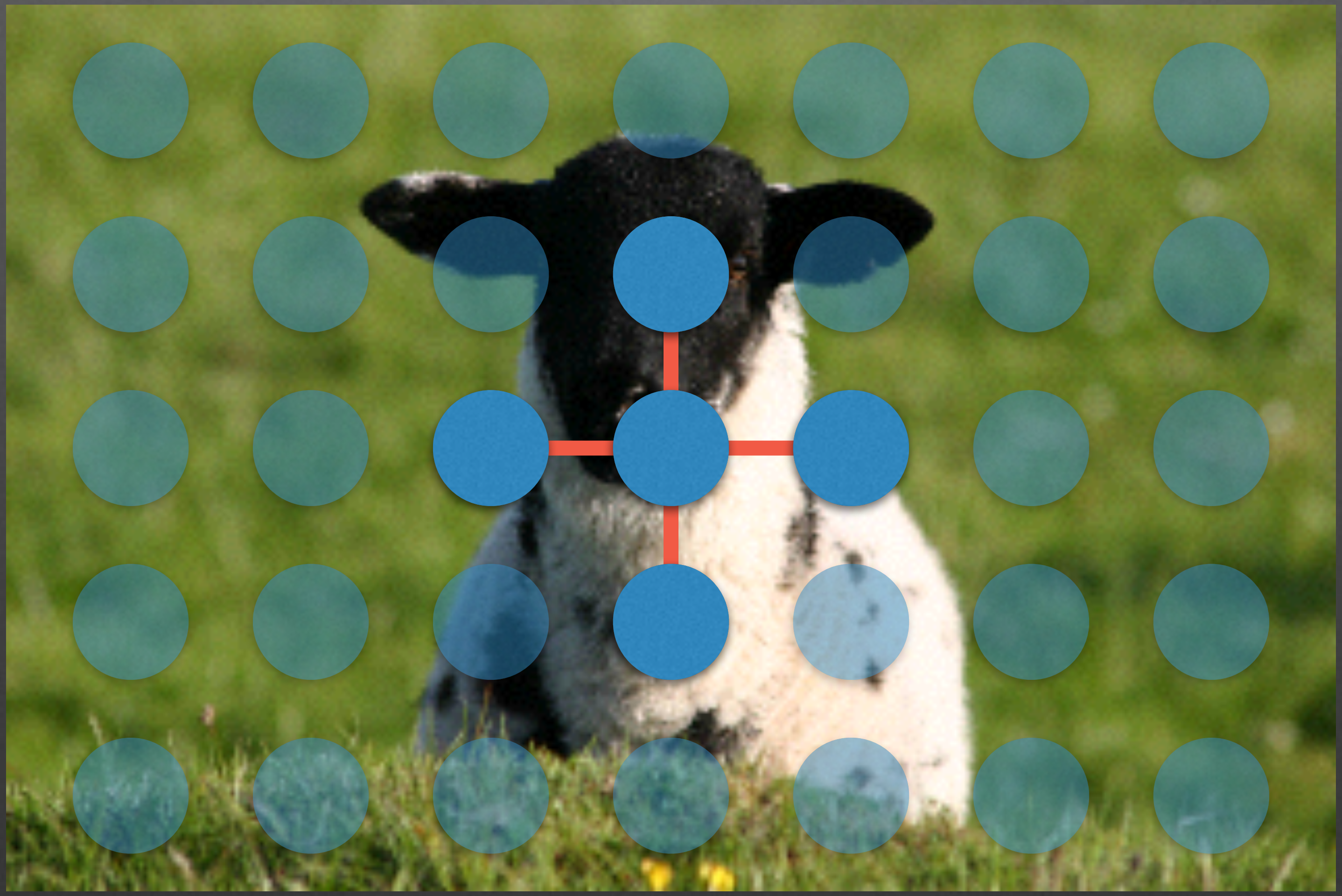
- No probabilistic interpretation
- No joint inference
- No learning



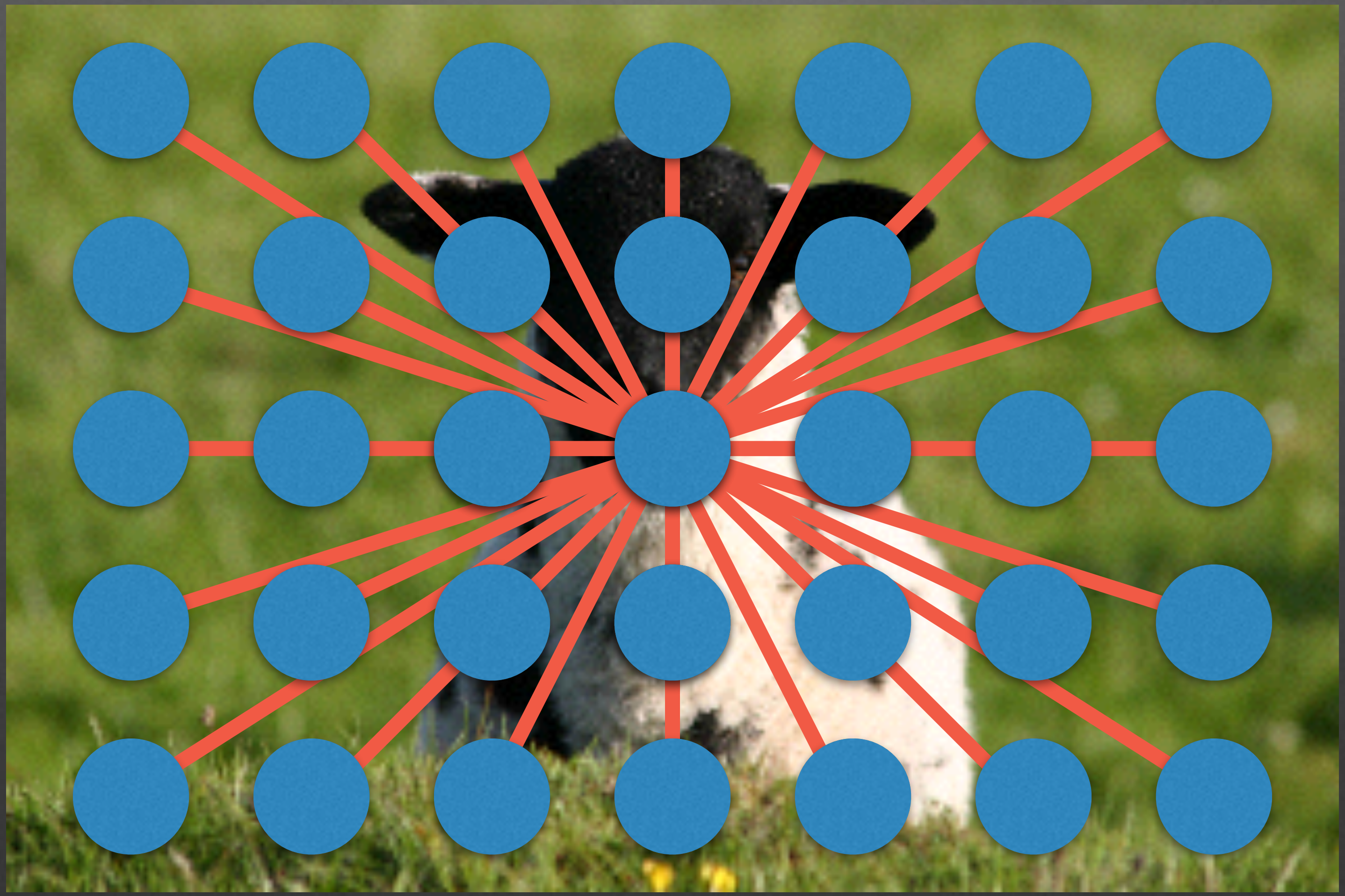
Dense Random Fields



Dense Random Fields



Dense Random Fields

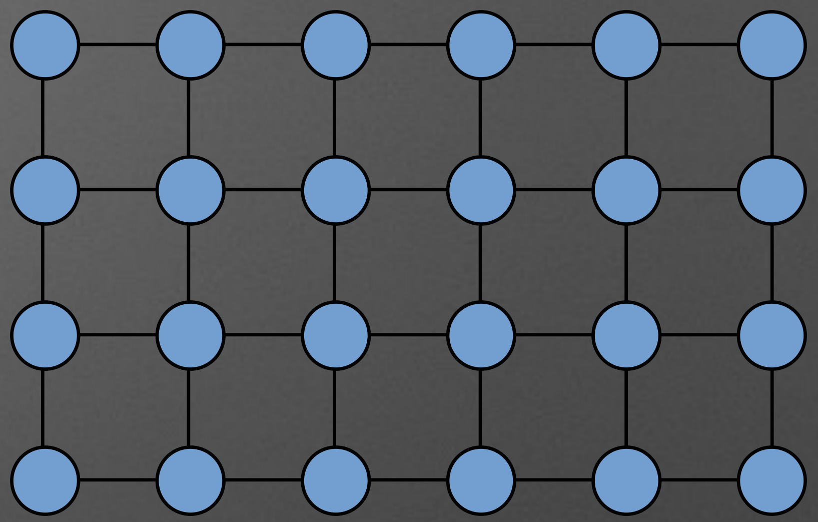


Dense Random Fields

$$E(X) = \sum_i \psi_i(X_i) + \sum_{i,j \in N} \psi_{ij}(X_i, X_j)$$

unary term

pairwise term

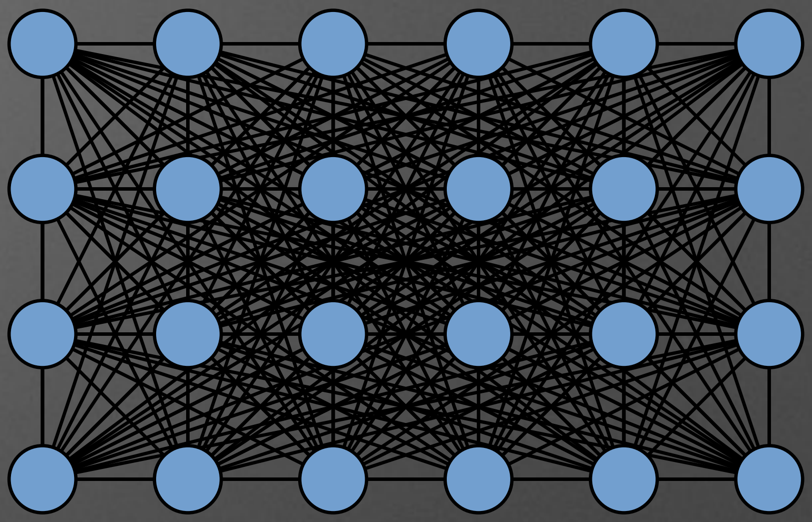


Dense Random Fields

$$E(X) = \sum_i \psi_i(X_i) + \sum_{i,j \in \cancel{N}} \psi_{ij}(X_i, X_j)$$

unary term

pairwise term



- Every node is connected to every other node
  - Connections weighted differently

Dense Random Fields

Dense Random Fields

Dense Random Fields

Dense Random Fields

Pros:

Dense Random Fields

Pros:

- Long range interactions

Dense Random Fields

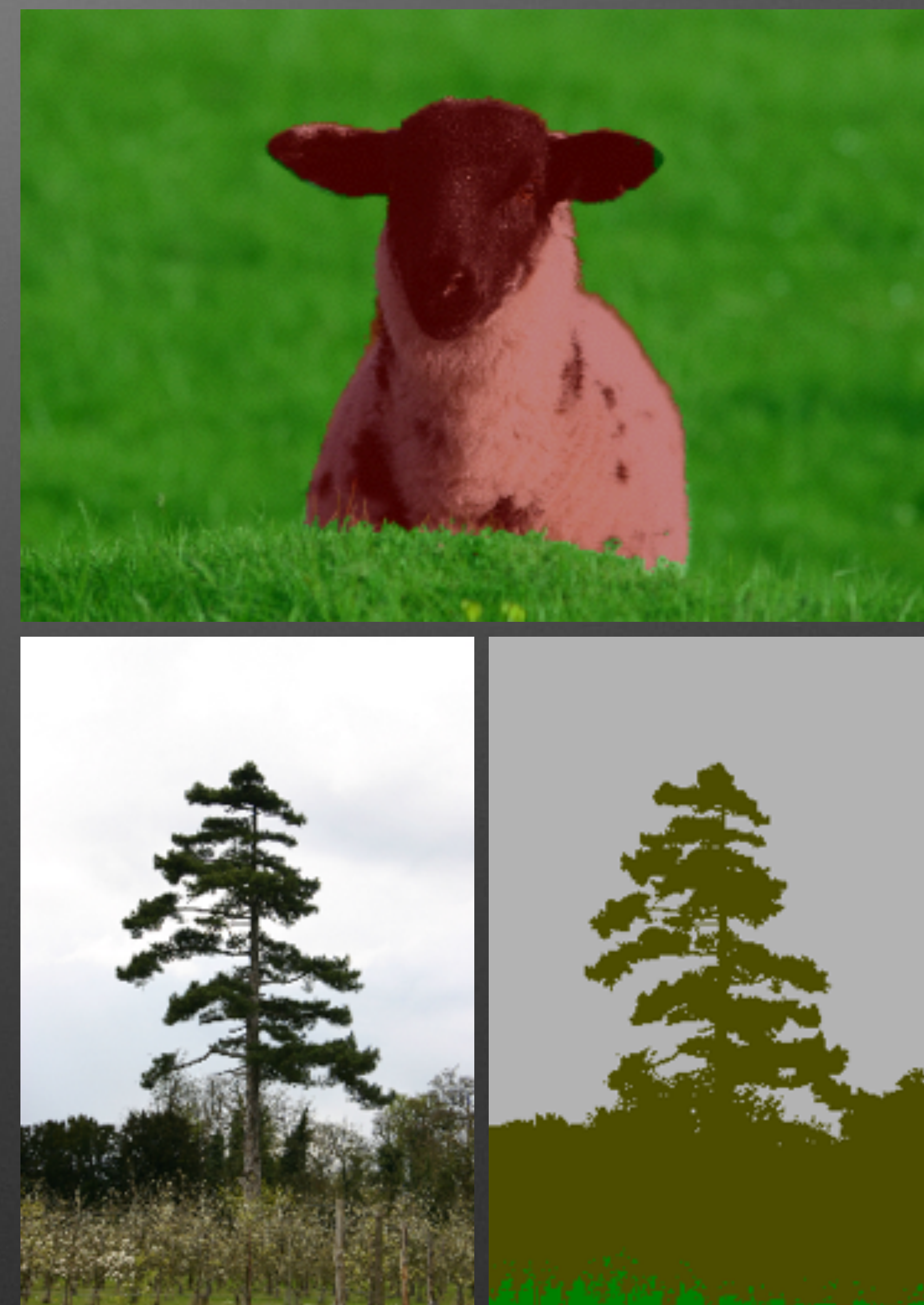
Pros:

- Long range interactions
- No shrinking bias

Dense Random Fields

Pros:

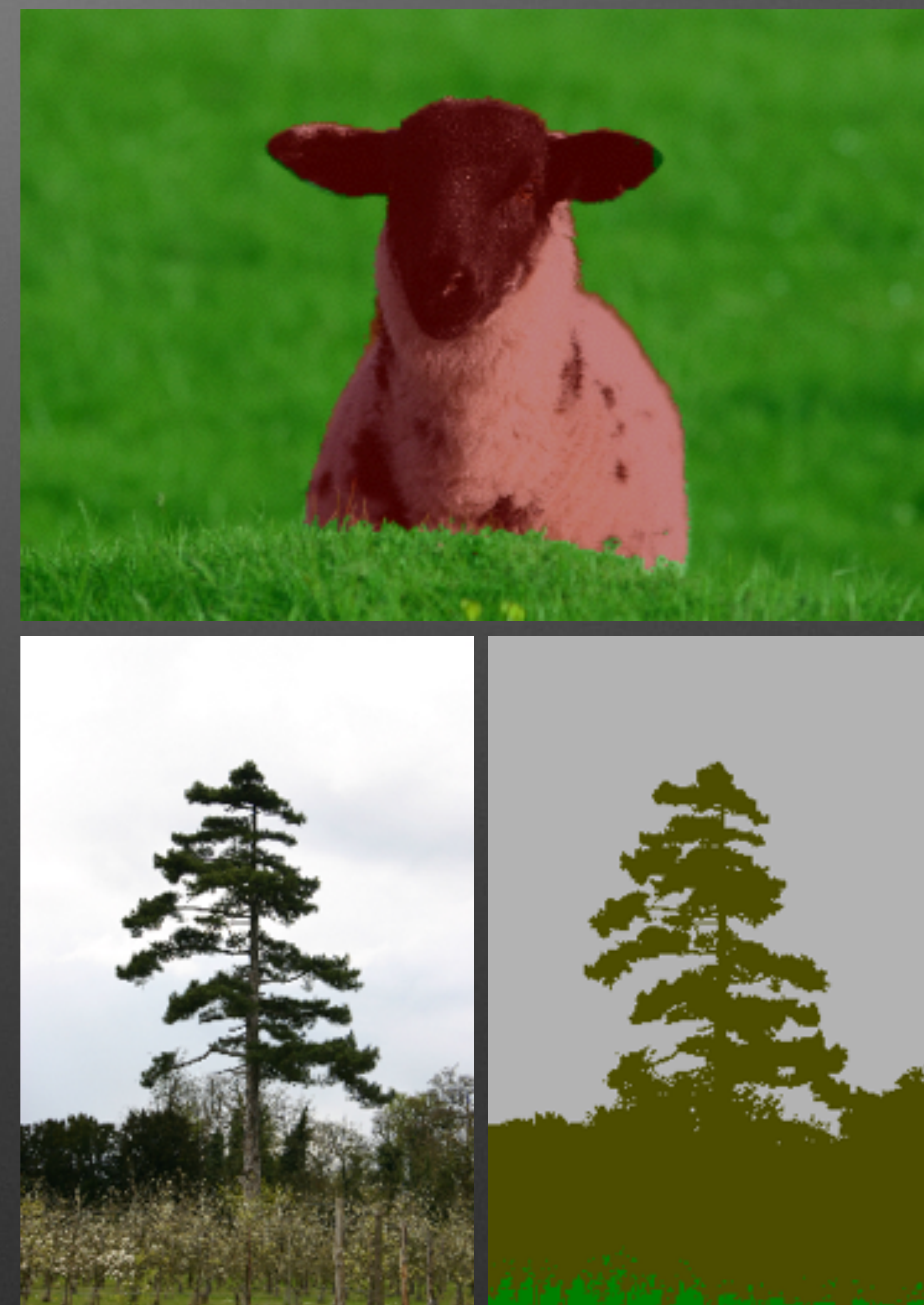
- Long range interactions
- No shrinking bias
- Probabilistic interpretation



Dense Random Fields

Pros:

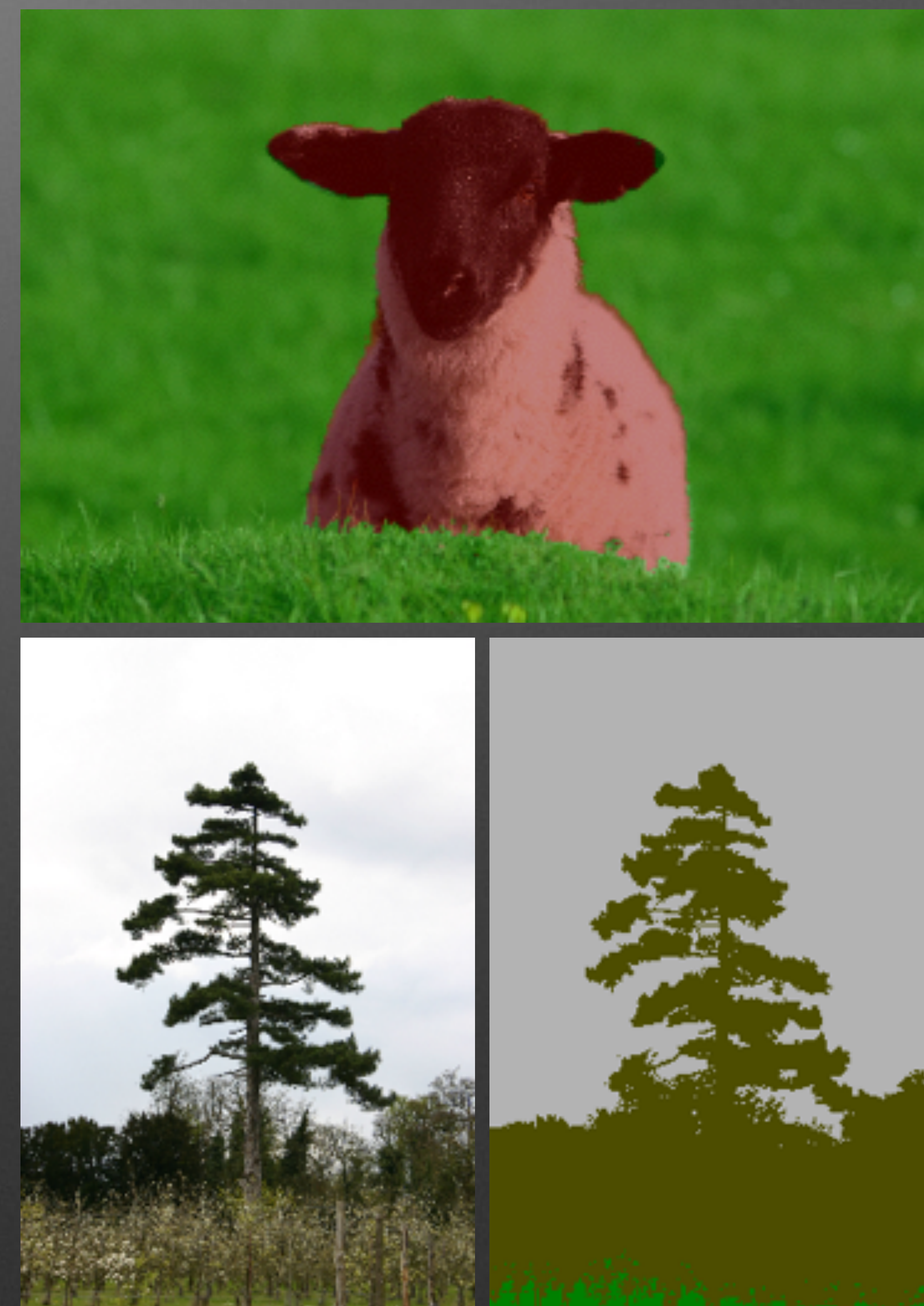
- Long range interactions
- No shrinking bias
- Probabilistic interpretation
- Parameter learning



Dense Random Fields

Pros:

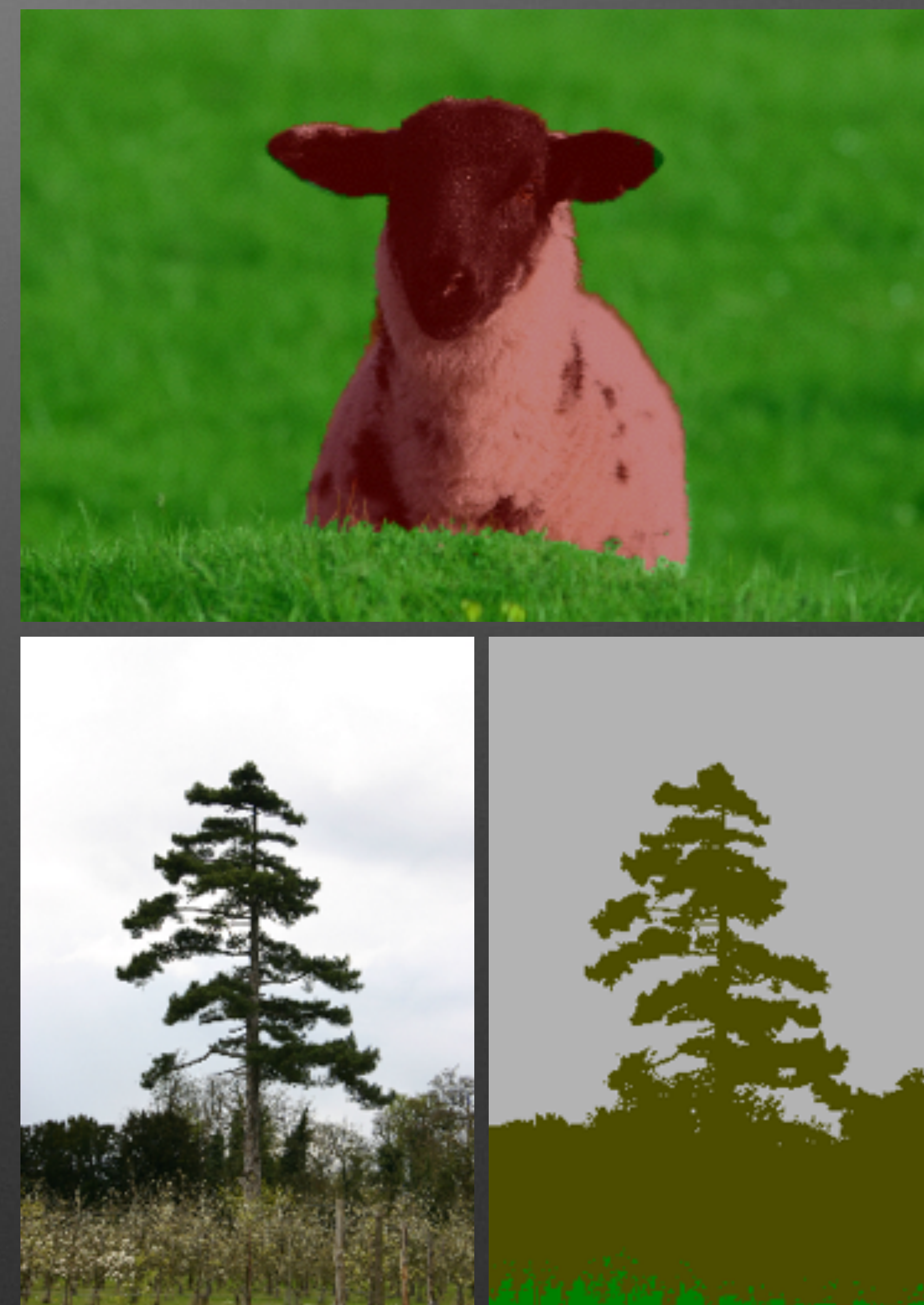
- Long range interactions
- No shrinking bias
- Probabilistic interpretation
- Parameter learning
- Combine with other models



Dense Random Fields

Dense Random Fields

Cons:



Dense Random Fields

Cons:

- Very large model
  - 50’000 - 100’000 variables
  - billions pairwise terms

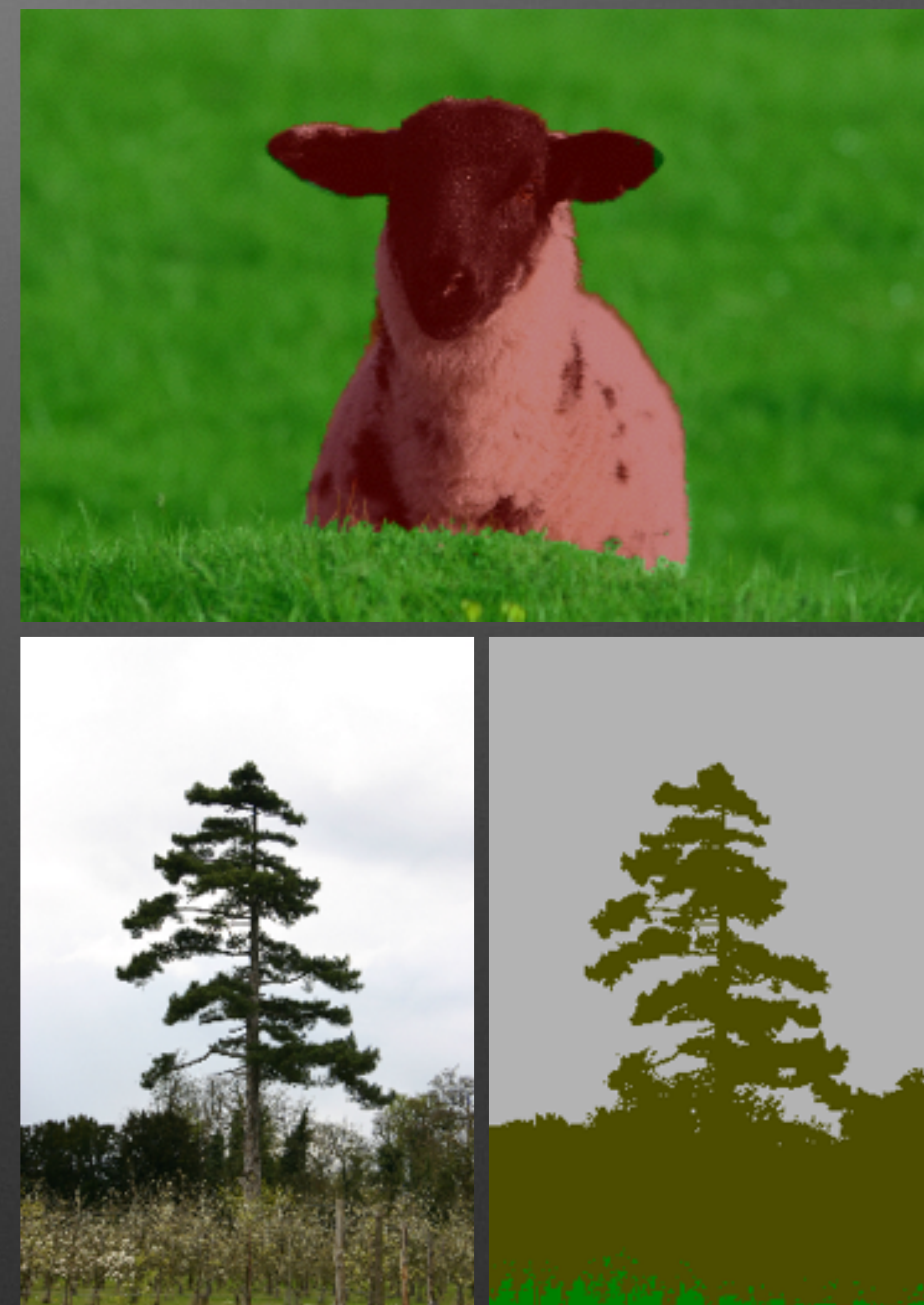
Dense Random Fields

Cons:

- Very large model
  - 50’000 - 100’000 variables
  - billions pairwise terms
- Traditional inference very slow
  - MCMC “converges” in 36h
  - GraphCuts and alpha-exp.: no convergence in 3 days

Dense Random Fields

- Efficient inference
  - 0.2s / image
- Pairwise term
  - linear combination of Gaussians

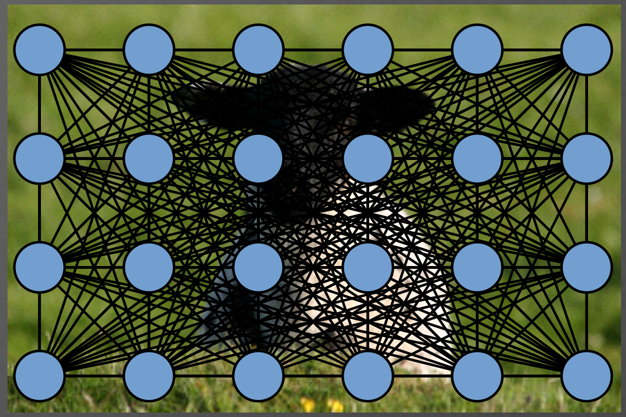


Dense Random Fields

$$E(X) = \sum_i \psi_i(X_i) + \sum_{i>j} \psi_{ij}(X_i, X_j)$$

Dense Random Fields

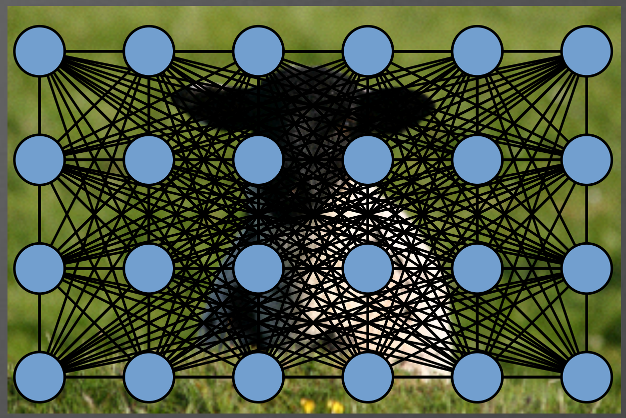
$$E(X) = \sum_i \psi_i(X_i) + \sum_{i>j} \psi_{ij}(X_i, X_j)$$



$$\psi_{ij}(X_i, X_j) = \sum_m k^{(m)}(f_i, f_j)\, \mu^{(m)}(X_i, X_j)$$

Dense Random Fields

$$E(X) = \sum_i \psi_i(X_i) + \sum_{i>j} \psi_{ij}(X_i, X_j)$$

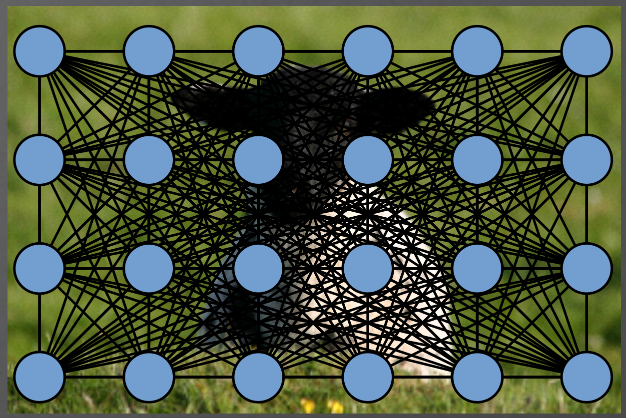


$$\psi_{ij}(X_i, X_j) = \sum_m k^{(m)}(f_i, f_j)\, \mu^{(m)}(X_i, X_j)$$

Gaussian kernel $k^{(m)}$

Dense Random Fields

$$E(X) = \sum_i \psi_i(X_i) + \sum_{i>j} \psi_{ij}(X_i, X_j)$$



$$\psi_{ij}(X_i, X_j) = \sum_m k^{(m)}(f_i, f_j)\, \mu^{(m)}(X_i, X_j)$$

Gaussian kernel $k^{(m)}$

$\otimes$

Label compatibility $\mu^{(m)}$

μ	GRASS	SHEEP	WATER	...
GRASS	0	1	1	...
SHEEP	1	0	10	...
WATER	1	10	0	...
...	...	...	...	0

Dense Random Fields

$$\psi_{ij}(X_i, X_j) = \mu_1(X_i, X_j) \exp\left(-\frac{|s_i - s_j|^2}{2\sigma_\alpha^2} - \frac{|c_i - c_j|^2}{2\sigma_\beta^2}\right) + \mu_2(X_i, X_j) \exp\left(-\frac{|s_i - s_j|^2}{2\sigma_\gamma^2}\right)$$

Dense Random Fields

$$\psi_{ij}(X_i, X_j) = \mu_1(X_i, X_j) \exp\left(-\frac{|s_i - s_j|^2}{2\sigma_\alpha^2} - \frac{|c_i - c_j|^2}{2\sigma_\beta^2}\right) +$$

$$\mu_2(X_i, X_j) \exp\left(-\frac{|s_i - s_j|^2}{2\sigma_\gamma^2}\right)$$

- Label compatibility

Dense Random Fields

$$\psi_{ij}(X_i, X_j) = \mu_1(X_i, X_j) \exp\left(-\frac{|s_i - s_j|^2}{2\sigma_\alpha^2} - \frac{|c_i - c_j|^2}{2\sigma_\beta^2}\right) +$$

$$\mu_2(X_i, X_j) \exp\left(-\frac{|s_i - s_j|^2}{2\sigma_\gamma^2}\right)$$

- Label compatibility
  - Potts model: μ(Xi,Xj) = [Xi≠Xj]

μ	GRASS	SHEEP	WATER	...
GRASS	0	1	1	1
SHEEP	1	0	1	1
WATER	1	1	0	1
...	1	1	1	0

Dense Random Fields

$$\psi_{ij}(X_i, X_j) = \mu_1(X_i, X_j) \exp\left(-\frac{|s_i - s_j|^2}{2\sigma_\alpha^2} - \frac{|c_i - c_j|^2}{2\sigma_\beta^2}\right) +$$

$$\mu_2(X_i, X_j) \exp\left(-\frac{|s_i - s_j|^2}{2\sigma_\gamma^2}\right)$$

- Label compatibility
  - Potts model: μ(Xi,Xj) = [Xi≠Xj]
  - Learned from data

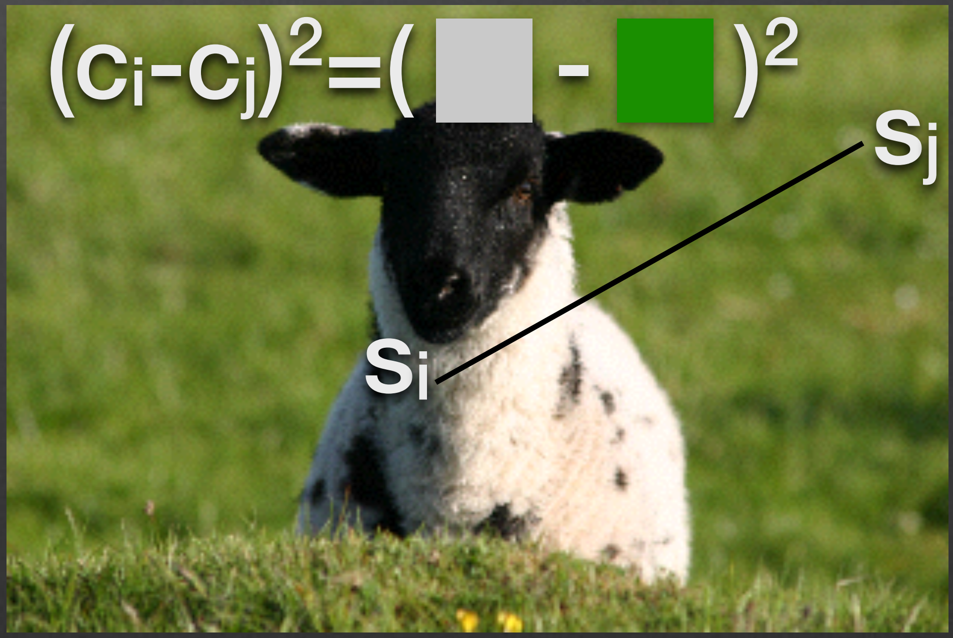
μ	GRASS	SHEEP	WATER	...
GRASS	0	?	?	?
SHEEP	?	0	?	?
WATER	?	?	0	?
...	?	?	?	0

Dense Random Fields

$$\psi_{ij}(X_i, X_j) = \mu_1(X_i, X_j) \exp\left(-\frac{|s_i - s_j|^2}{2\sigma_\alpha^2} - \frac{|c_i - c_j|^2}{2\sigma_\beta^2}\right) +$$

$$\mu_2(X_i, X_j) \exp\left(-\frac{|s_i - s_j|^2}{2\sigma_\gamma^2}\right)$$

- Label compatibility
  - Potts model: μ(Xi,Xj) = [Xi≠Xj]
  - Learned from data
- Appearance kernel

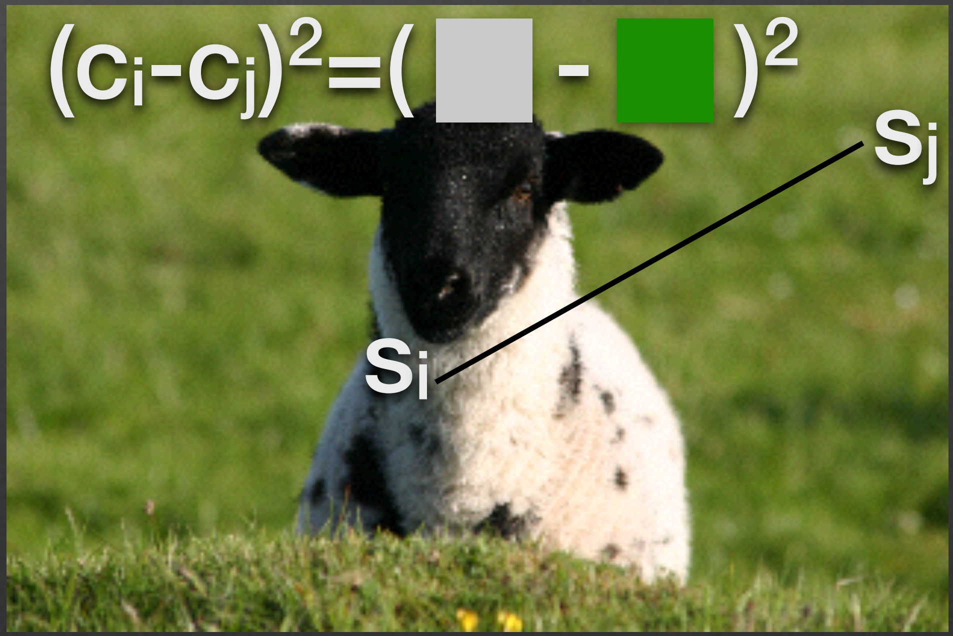


Dense Random Fields

$$\psi_{ij}(X_i, X_j) = \mu_1(X_i, X_j) \exp\left(-\frac{|s_i - s_j|^2}{2\sigma_\alpha^2} - \frac{|c_i - c_j|^2}{2\sigma_\beta^2}\right) +$$

$$\mu_2(X_i, X_j) \exp\left(-\frac{|s_i - s_j|^2}{2\sigma_\gamma^2}\right)$$

- Label compatibility
  - Potts model: μ(Xi,Xj) = [Xi≠Xj]
  - Learned from data
- Appearance kernel
  - Color—sensitive

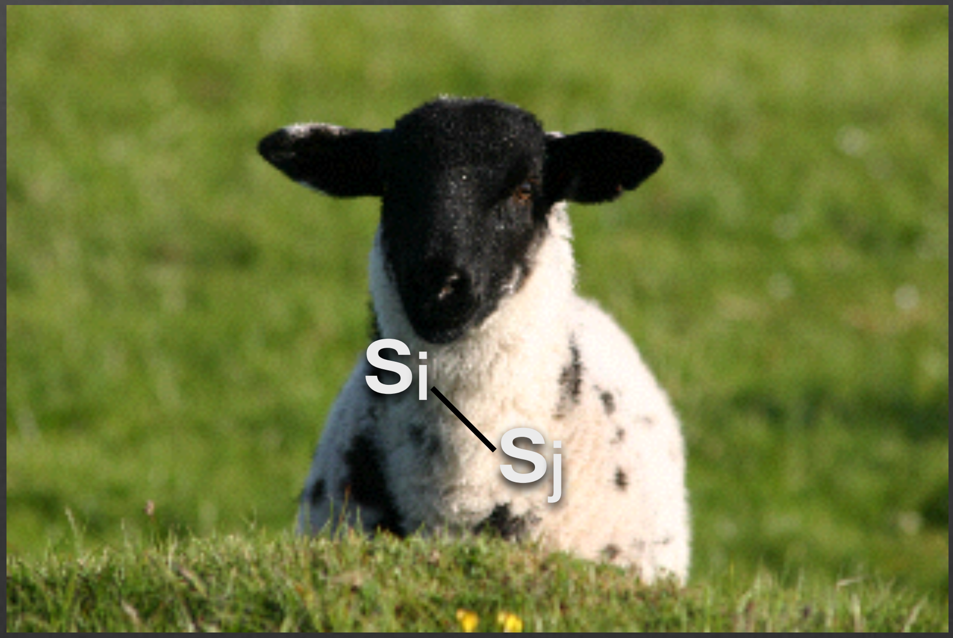


Dense Random Fields

$$\psi_{ij}(X_i, X_j) = \mu_1(X_i, X_j) \exp\left(-\frac{|s_i - s_j|^2}{2\sigma_\alpha^2} - \frac{|c_i - c_j|^2}{2\sigma_\beta^2}\right) +$$

$$\mu_2(X_i, X_j) \exp\left(-\frac{|s_i - s_j|^2}{2\sigma_\gamma^2}\right)$$

- Label compatibility
  - Potts model: μ(Xi,Xj) = [Xi≠Xj]
  - Learned from data
- Appearance kernel
  - Color—sensitive
- Local smoothness

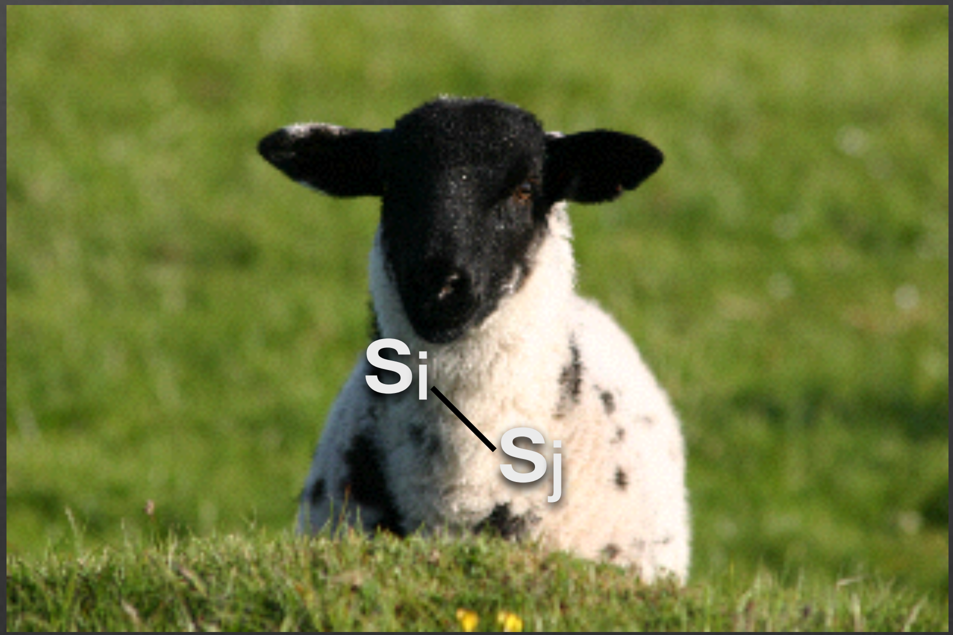


Dense Random Fields

$$\psi_{ij}(X_i, X_j) = \mu_1(X_i, X_j) \exp\left(-\frac{|s_i - s_j|^2}{2\sigma_\alpha^2} - \frac{|c_i - c_j|^2}{2\sigma_\beta^2}\right) +$$

$$\mu_2(X_i, X_j) \exp\left(-\frac{|s_i - s_j|^2}{2\sigma_\gamma^2}\right)$$

- Label compatibility
  - Potts model: μ(Xi,Xj) = [Xi≠Xj]
  - Learned from data
- Appearance kernel
  - Color—sensitive
- Local smoothness
  - Discourages single pixel noise



Efficient inference

$$E(X) = \sum_i \psi_i(X_i) + \sum_{i>j} \psi_{ij}(X_i, X_j)$$

Efficient inference

$$E(X) = \sum_i \psi_i(X_i) + \sum_{i>j} \psi_{ij}(X_i, X_j)$$

Find most likely assignment (MAP)

$$\hat{x} = \arg\max_X P(X)$$ where $$P(X) = \frac{1}{Z}\exp(-E(X))$$

Efficient inference

$$E(X) = \sum_i \psi_i(X_i) + \sum_{i>j} \psi_{ij}(X_i, X_j)$$

Find most likely as… (MAP)

$$\hat{x} = \arg\max_X P(\ldots$$

$$P(X) = \frac{1}{Z} \exp(-E(X))$$

NP-Hard

Efficient inference

$$E(X) = \sum_i \psi_i(X_i) + \sum_{i>j} \psi_{ij}(X_i, X_j)$$

Find most likely as… (MAP)

NP-Hard

$$\hat{x} = \arg\max_X P(\ldots$$

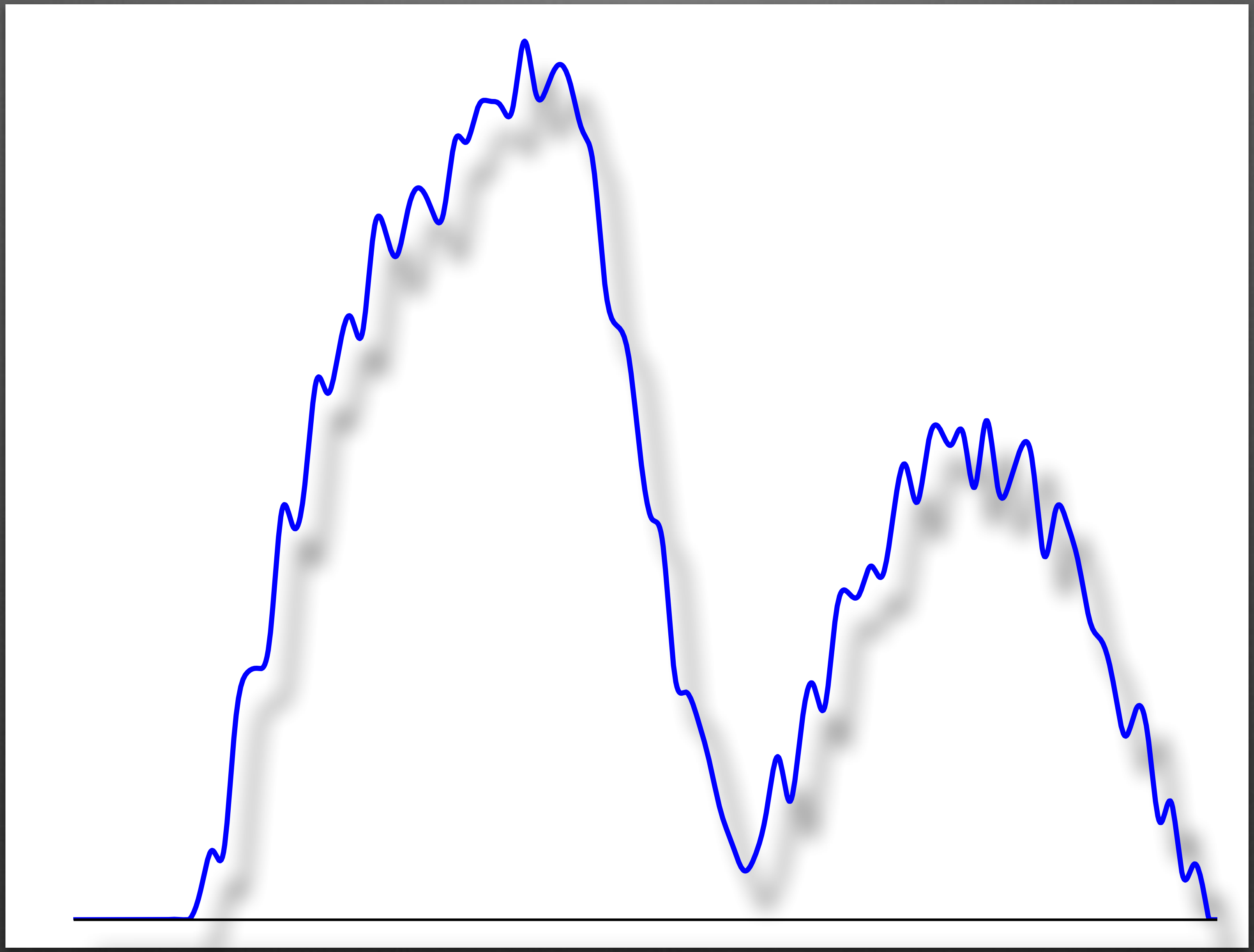
$$P(X) = \frac{1}{Z}\exp(-E(X))$$

Mean Field approximation

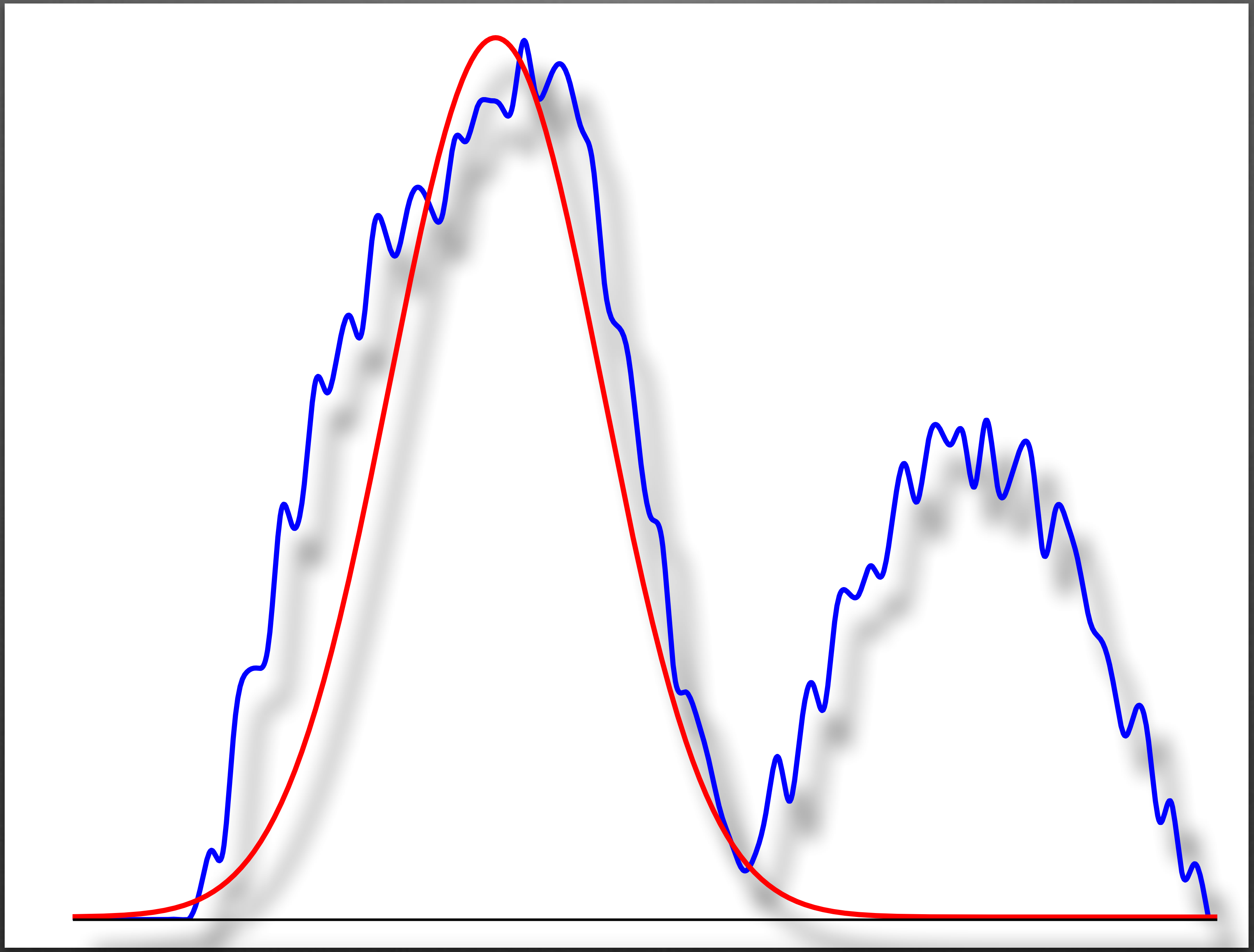
Find $Q(X)=\prod_i Q_i(X_i)$ close to P(X) in terms of KL-divergence D(Q||P)

$$\hat{x} \approx \arg\max_X Q(X)$$

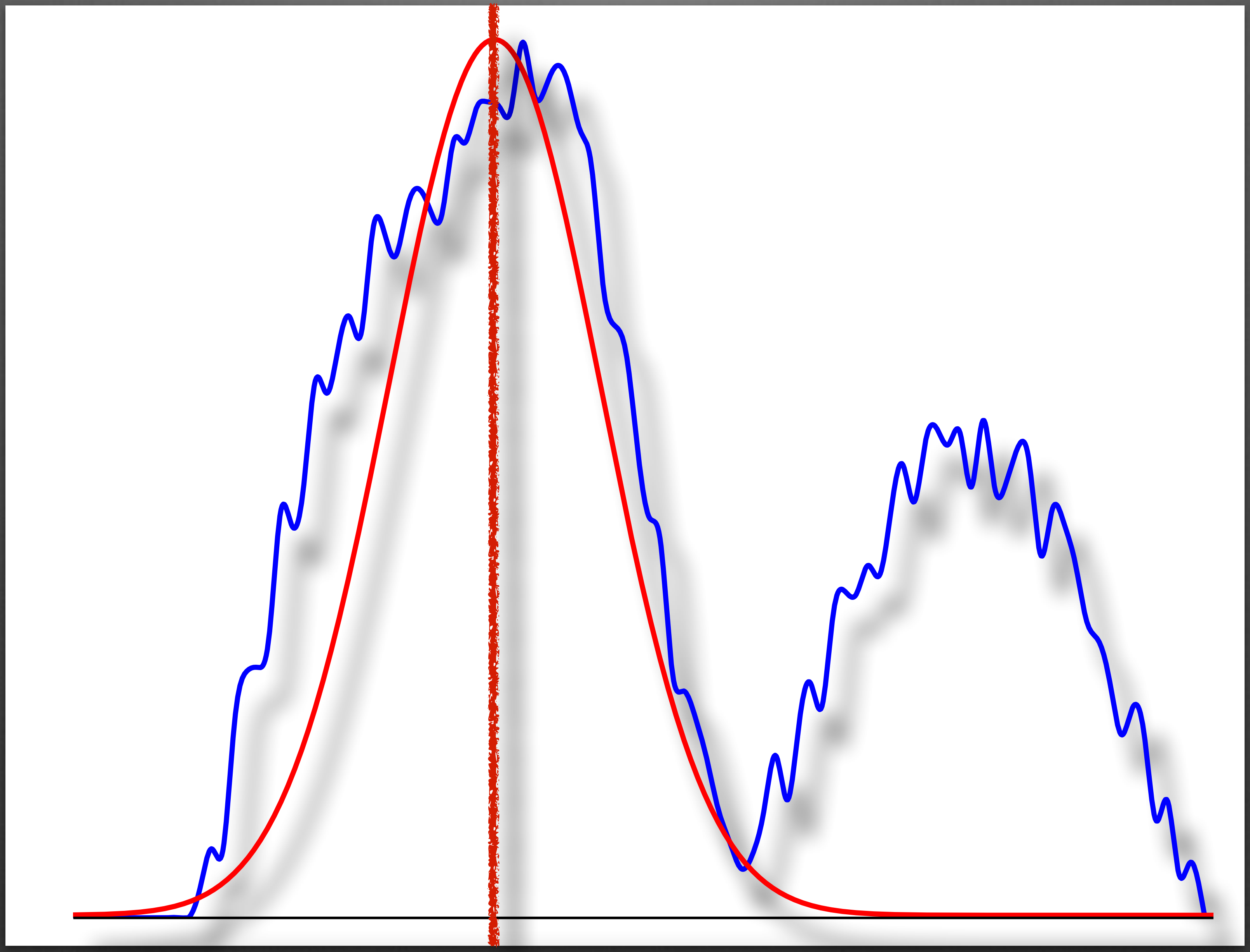
Mean-Field approximation



Mean-Field approximation



Mean-Field approximation



Efficient inference

Mean Field algorithm

Efficient inference

Mean Field algorithm

Initialize $Q_i(l) = \frac{1}{Z_i} \exp(-\psi_i(l))$

Efficient inference

Mean Field algorithm

Initialize $Q_i(l) = \frac{1}{Z_i}\exp(-\psi_i(l))$

Until convergence:

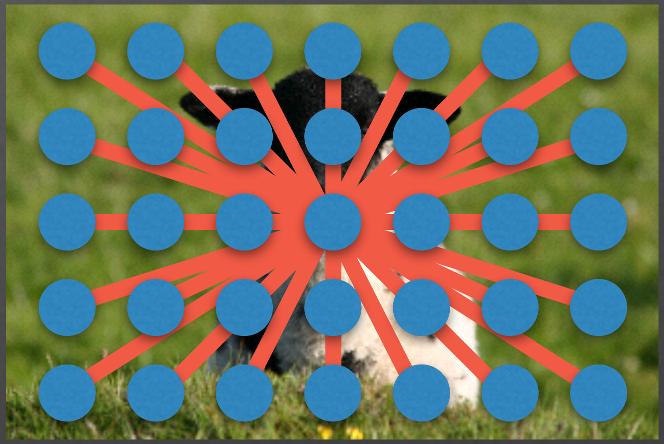
Efficient inference

Mean Field algorithm

Initialize $Q_i(l) = \frac{1}{Z_i}\exp(-\psi_i(l))$

Until convergence:

- Message passing: $\tilde{Q}_i^{(m)}(l) = \sum_j k^{(m)}(f_i, f_j) Q_j(l)$



Efficient inference

Mean Field algorithm

Initialize $Q_i(l) = \frac{1}{Z_i} \exp(-\psi_i(l))$

Until convergence:

- Message passing: $\tilde{Q}_i^{(m)}(l) = \sum_j k^{(m)}(f_i, f_j) Q_j(l)$
- Compatibility transform: $\hat{Q}_i^{(m)}(l') = \sum_l \mu^{(m)}(l', l) \tilde{Q}_i^{(m)}(l)$

μ	GRASS	SHEEP	WATER	...
GRASS	0	1	1	...
SHEEP	1	0	10	...
WATER	1	10	0	...
...	...	...	...	0

Efficient inference

Mean Field algorithm

Initialize $Q_i(l) = \frac{1}{Z_i} \exp(-\psi_i(l))$

Until convergence:

- Message passing: $\tilde{Q}_i^{(m)}(l) = \sum_j k^{(m)}(f_i, f_j) Q_j(l)$
- Compatibility transform: $\hat{Q}_i^{(m)}(l') = \sum_l \mu^{(m)}(l', l) \tilde{Q}_i^{(m)}(l)$
- Local update: $Q_i(l) = \exp(-\psi_i(l) - \sum_m \hat{Q}_i^{(m)}(l))$

Efficient inference

Mean Field algorithm

Initialize $Q_i(l) = \frac{1}{Z_i} \exp(-\psi_i(l))$

Until convergence:

- Message passing: $\tilde{Q}_i^{(m)}(l) = \sum_j k^{(m)}(f_i, f_j) Q_j(l)$
- Compatibility transform: $\hat{Q}_i^{(m)}(l') = \sum_l \mu^{(m)}(l', l) \tilde{Q}_i^{(m)}(l)$
- Local update: $Q_i(l) = \exp(-\psi_i(l) - \sum_m \hat{Q}_i^{(m)}(l))$
- Normalize Q_i

Efficient inference

Mean Field algorithm

Initialize $Q_i(l) = \frac{1}{Z_i} \exp(-\psi_i(l))$ O(N)

Until convergence:

- Message passing: $\tilde{Q}_i^{(m)}(l) = \sum_j k^{(m)}(f_i, f_j) Q_j(l)$
- Compatibility transform: $\hat{Q}_i^{(m)}(l') = \sum_l \mu^{(m)}(l', l) \tilde{Q}_i^{(m)}(l)$
- Local update: $Q_i(l) = \exp(-\psi_i(l) - \sum_m \hat{Q}_i^{(m)}(l))$
- Normalize Q_i

Efficient inference

Mean Field algorithm

Initialize $Q_i(l) = \frac{1}{Z_i} \exp(-\psi_i(l))$ O(N)

Until convergence:

- Message passing: $\tilde{Q}_i^{(m)}(l) = \sum_j k^{(m)}(f_i, f_j) Q_j(l)$

O(N) Compatibility transform: $\hat{Q}_i^{(m)}(l') = \sum_l \mu^{(m)}(l', l) \tilde{Q}_i^{(m)}(l)$

- Local update: $Q_i(l) = \exp(-\psi_i(l) - \sum_m \hat{Q}_i^{(m)}(l))$
- Normalize Q_i

Efficient inference

Mean Field algorithm

Initialize $Q_i(l) = \frac{1}{Z_i} \exp(-\psi_i(l))$ O(N)

Until convergence:

- Message passing: $\tilde{Q}_i^{(m)}(l) = \sum_j k^{(m)}(f_i, f_j) Q_j(l)$

O(N) Compatibility transform: $\hat{Q}_i^{(m)}(l') = \sum_l \mu^{(m)}(l', l) \tilde{Q}_i^{(m)}(l)$

O(N) Local update: $Q_i(l) = \exp(-\psi_i(l) - \sum_m \hat{Q}_i^{(m)}(l))$

- Normalize Q_i

Efficient inference

Mean Field algorithm

Initialize $Q_i(l) = \frac{1}{Z_i} \exp(-\psi_i(l))$ O(N)

Until convergence:

- Message passing: $\tilde{Q}_i^{(m)}(l) = \sum_j k^{(m)}(f_i, f_j) Q_j(l)$

O(N) Compatibility transform: $\hat{Q}_i^{(m)}(l') = \sum_l \mu^{(m)}(l', l) \tilde{Q}_i^{(m)}(l)$

O(N) Local update: $Q_i(l) = \exp(-\psi_i(l) - \sum_m \hat{Q}_i^{(m)}(l))$

O(N) Normalize Q_i

Efficient inference

Mean Field algorithm

Initialize $Q_i(l) = \frac{1}{Z_i}\exp(-\psi_i(l))$ O(N)

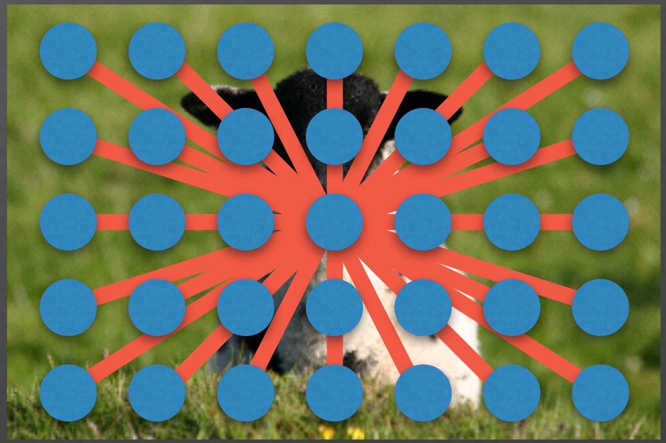
Until convergence:

O(N²) Message passing: $\tilde{Q}_i^{(m)}(l) = \sum_j k^{(m)}(f_i, f_j) Q_j(l)$

O(N) Compatibility transform: $\hat{Q}_i^{(m)}(l') = \sum_l \mu^{(m)}(l', l)\tilde{Q}_i^{(m)}(l)$

O(N) Local update: $Q_i(l) = \exp(-\psi_i(l) - \sum_m \hat{Q}_i^{(m)}(l))$

O(N) Normalize Q_i



Efficient message passing

- Update all variables simultaneously

$$\tilde{Q}_i^{(m)}(l) = \sum_j k^{(m)}(f_i, f_j) Q_j(l)$$

- Gaussian Convolution
    - Efficient approximation

Efficient inference

Mean Field algorithm

Initialize $Q_i(l) = \frac{1}{Z_i} \exp(-\psi_i(l))$ O(N)

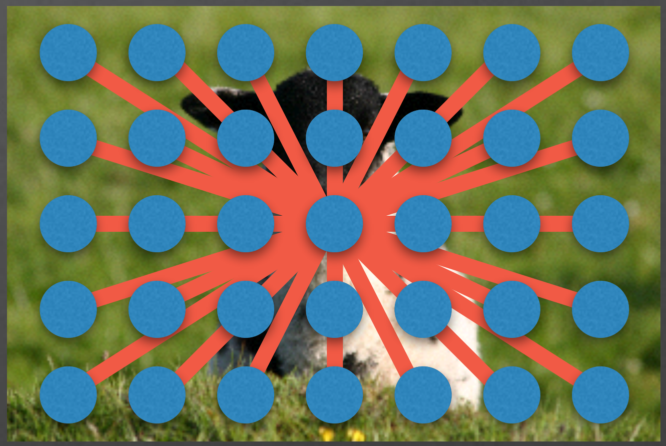
Until convergence:

O(N²) Message passing: $\tilde{Q}_i^{(m)}(l) = \sum_j k^{(m)}(f_i, f_j) Q_j(l)$

O(N) Compatibility transform: $\hat{Q}_i^{(m)}(l') = \sum_l \mu^{(m)}(l', l) \tilde{Q}_i^{(m)}(l)$

O(N) Local update: $Q_i(l) = \exp(-\psi_i(l) - \sum_m \hat{Q}_i^{(m)}(l))$

O(N) Normalize Q_i



Efficient inference

Mean Field algorithm

Initialize $Q_i(l) = \frac{1}{Z_i}\exp(-\psi_i(l))$ O(N)

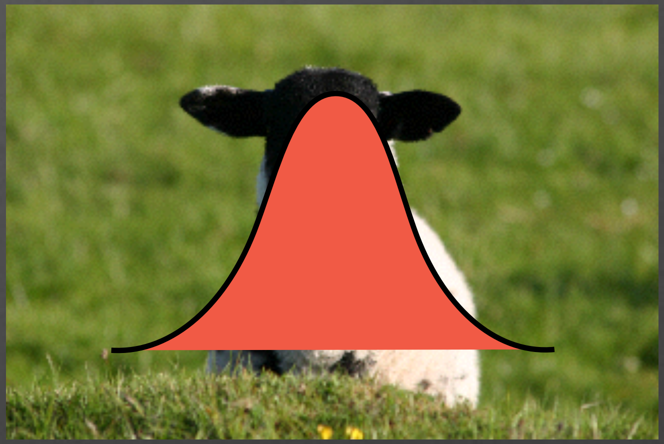
Until convergence:

O(N) Message passing: High-dimensional filter

O(N) Compatibility transform: $\hat{Q}_i^{(m)}(l') = \sum_l \mu^{(m)}(l', l)\tilde{Q}_i^{(m)}(l)$

O(N) Local update: $Q_i(l) = \exp(-\psi_i(l) - \sum_m \hat{Q}_i^{(m)}(l))$

O(N) Normalize Q_i



Efficient inference

Mean Field algorithm

Initialize $Q_i(l) = \frac{1}{Z_i}\exp(-\psi_i(l))$ O(N)

Until convergence:

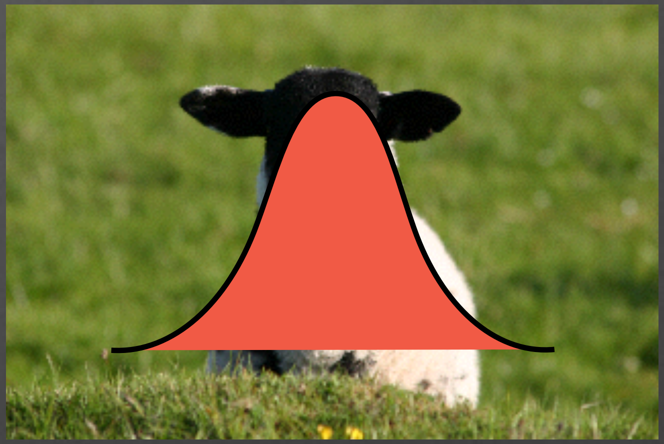
O(N) Message passi… …sional filter

O(N) Comp… $Q_i^{(m)}(l') = \sum_l \mu^{(m)}(l',l)\tilde{Q}_i^{(m)}(l)$

O(N) … $(l) = \exp(-\psi_i(l) - \sum_m \hat{Q}_i^{(m)}(l))$

O(N) …ize Q_i

linear in number of variables
independent of number of pairwise terms



Parallel Mean-Field

Parallel Mean-Field

- Not guaranteed to converge for general models

Parallel Mean-Field

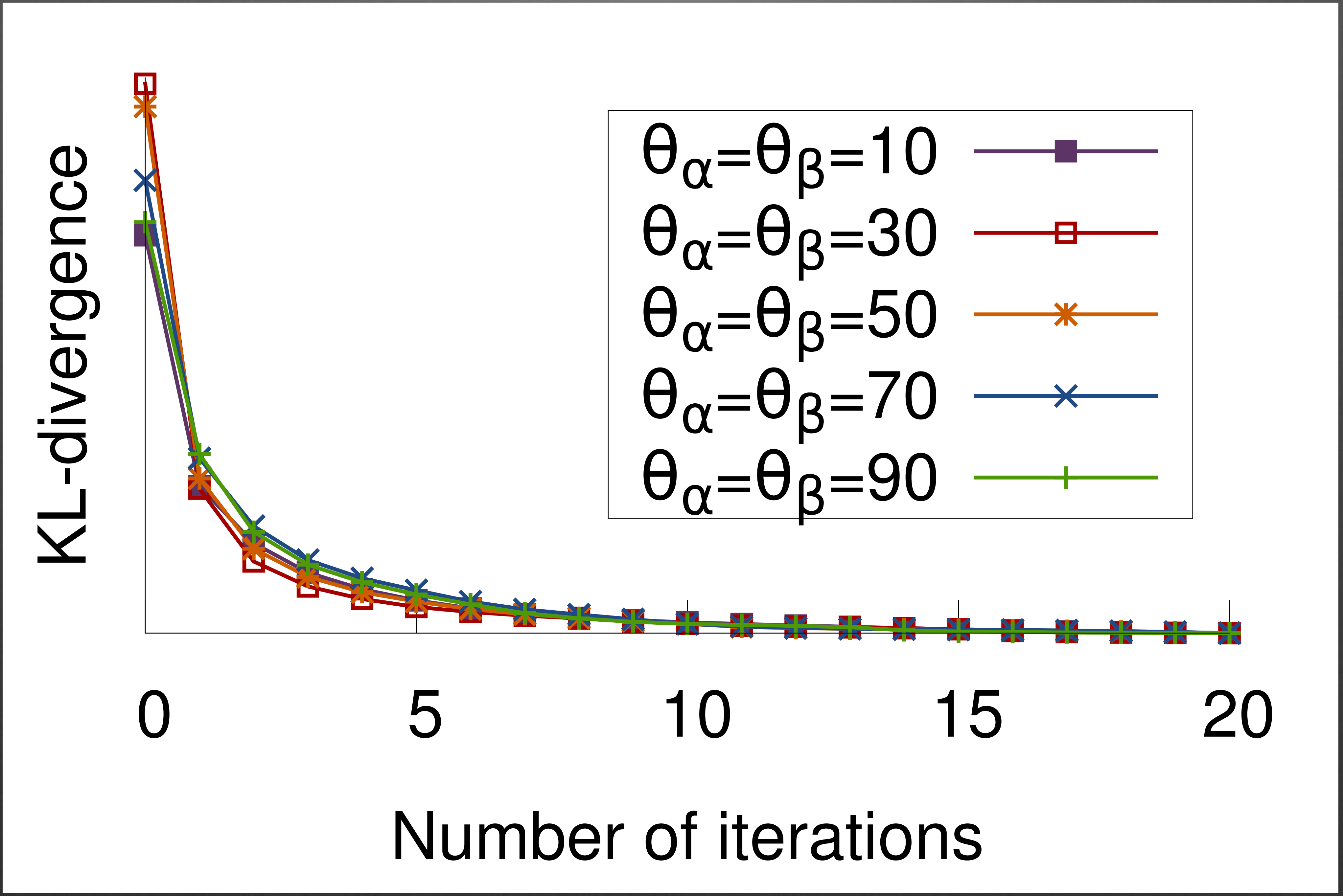
- Not guaranteed to converge for general models
- Guaranteed to converge for fully-connected models with negative definite label compatibility
    - Potts models
    - L1 norms
    - ...

Parallel Mean-Field

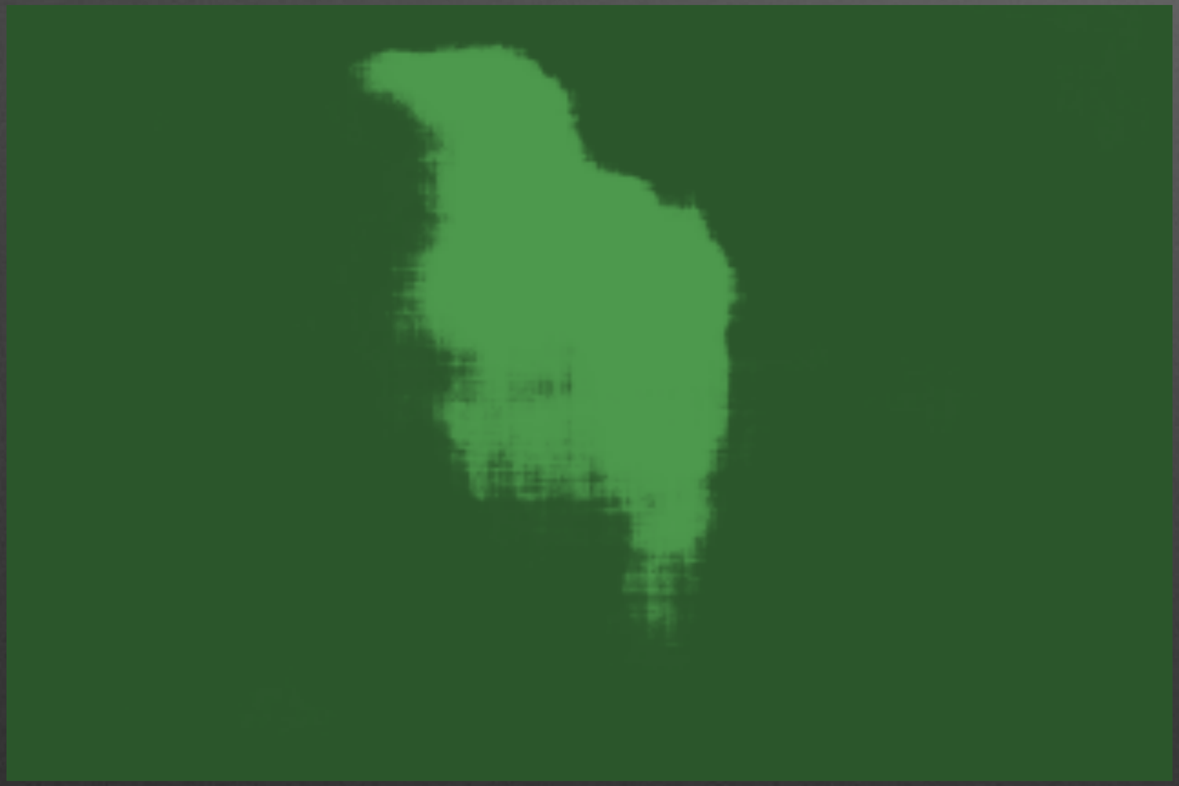
- Not guaranteed to converge for general models
- Guaranteed to converge for fully-connected models with negative definite label compatibility
    - Potts models
    - L1 norms
    - ...
- Proof see Thesis or [3]
    - Reduction of Parallel Mean-Field to CCCP

[3] Parameter Learning and Convergent Inference for Dense Random Fields, Krähenbühl and Koltun, ICML 2013

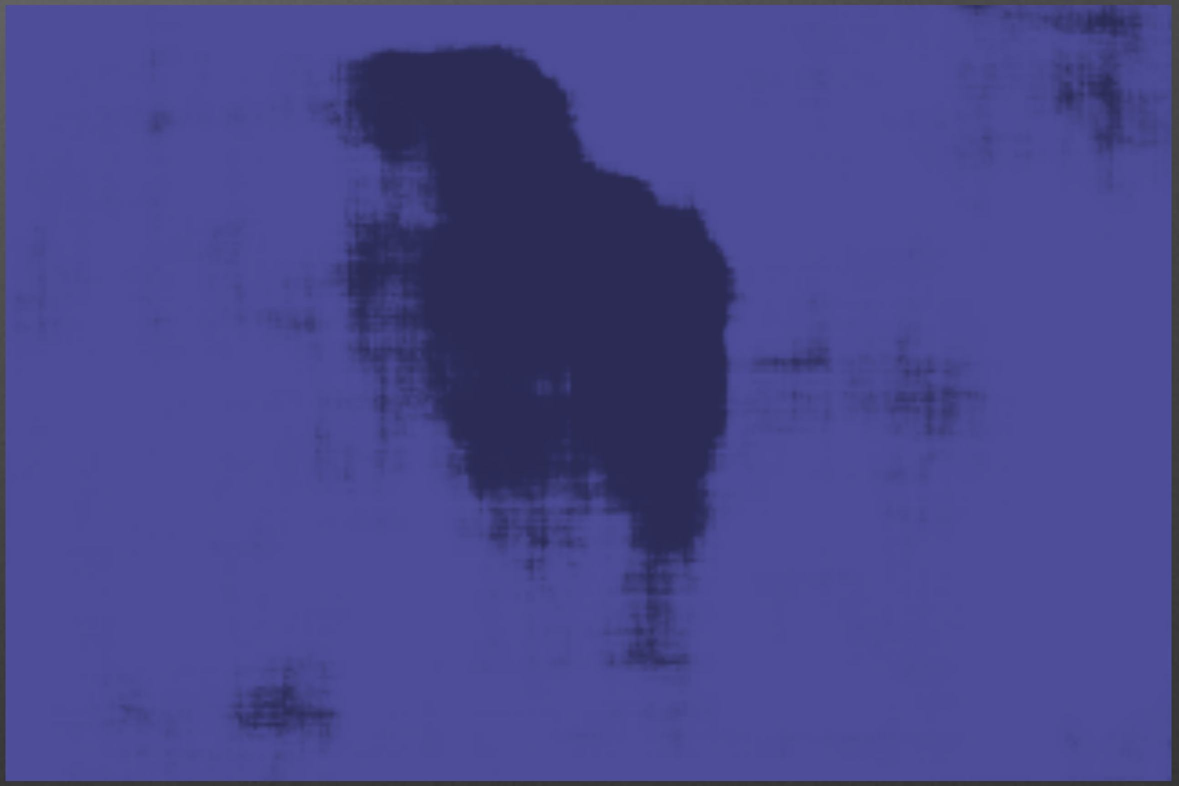
How fast will it converge



0 iterations

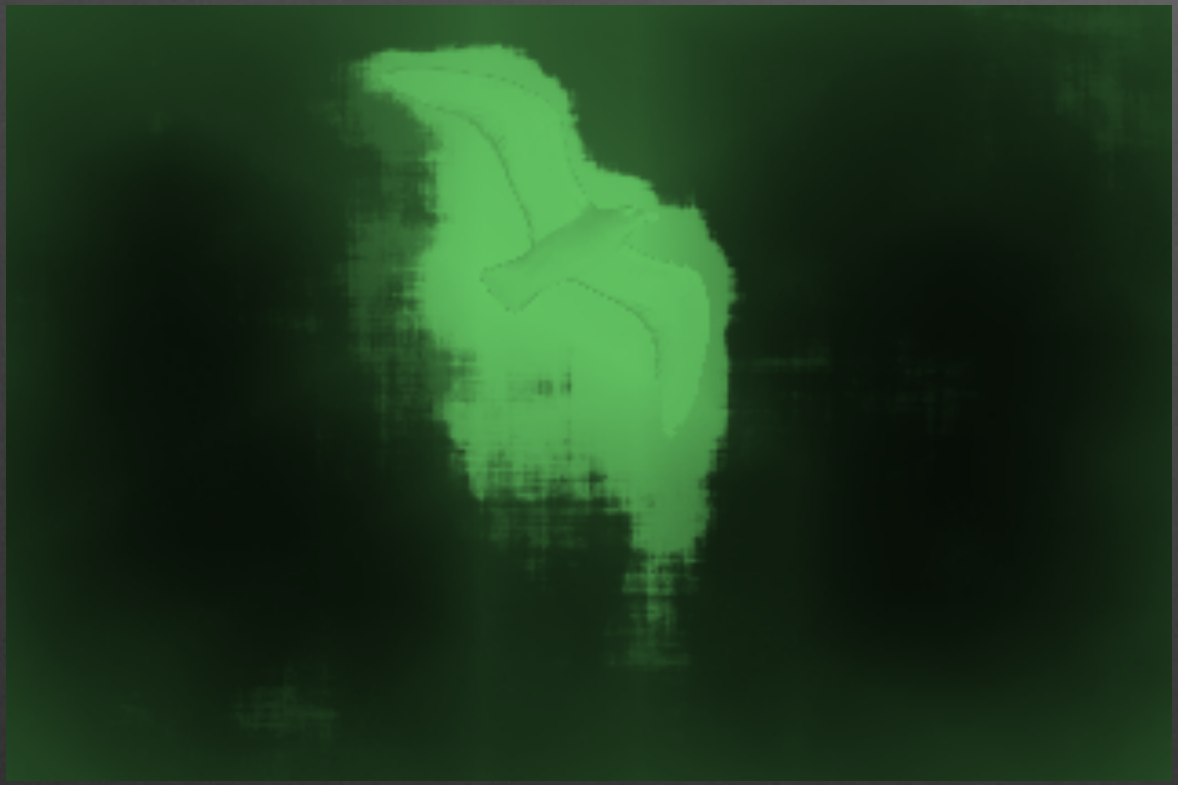


Q(bird)



Q(sky)

1 iterations



Q(bird)

Q(sky)

2 iterations

Q(bird)

Q(sky)

10 iterations

Q(bird)

Q(sky)

Results - MSRC

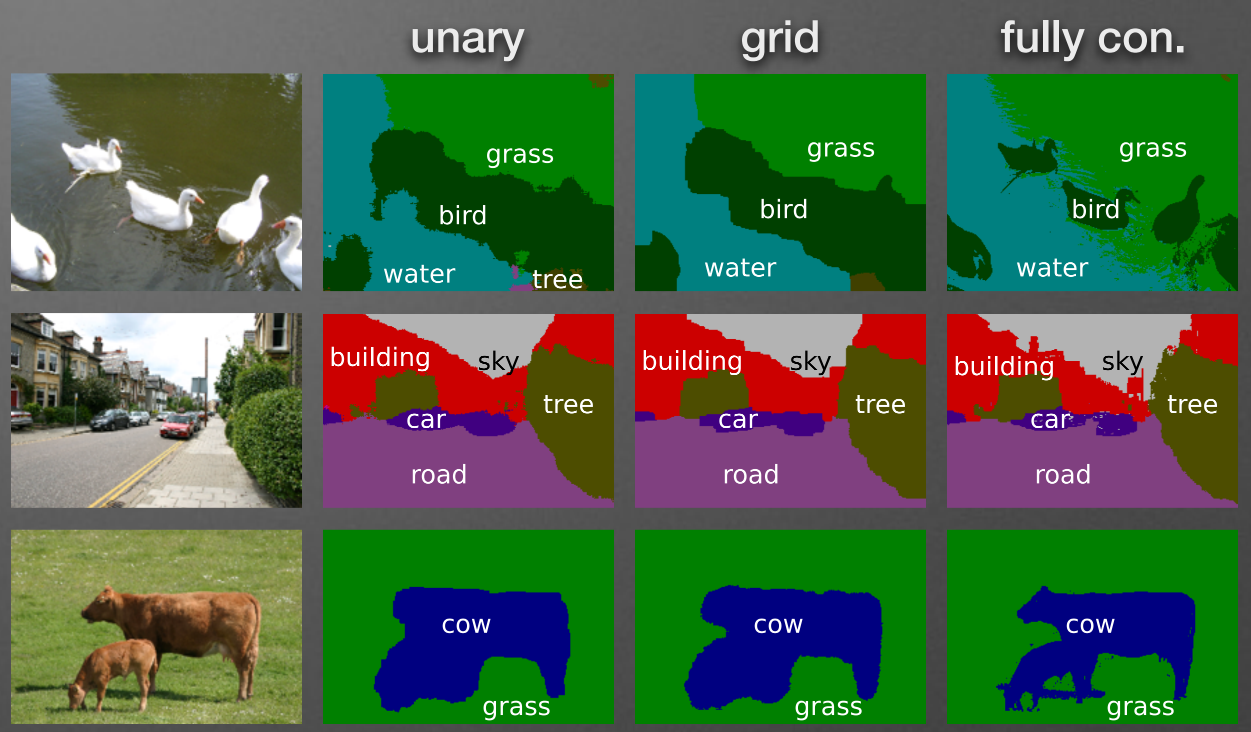
unary grid fully con.

Results - MSRC

MSRC dataset

- 591 images
- 21 classes

	TIME	GLOBAL	AVERAGE
UNARY	-	84.0	76.6

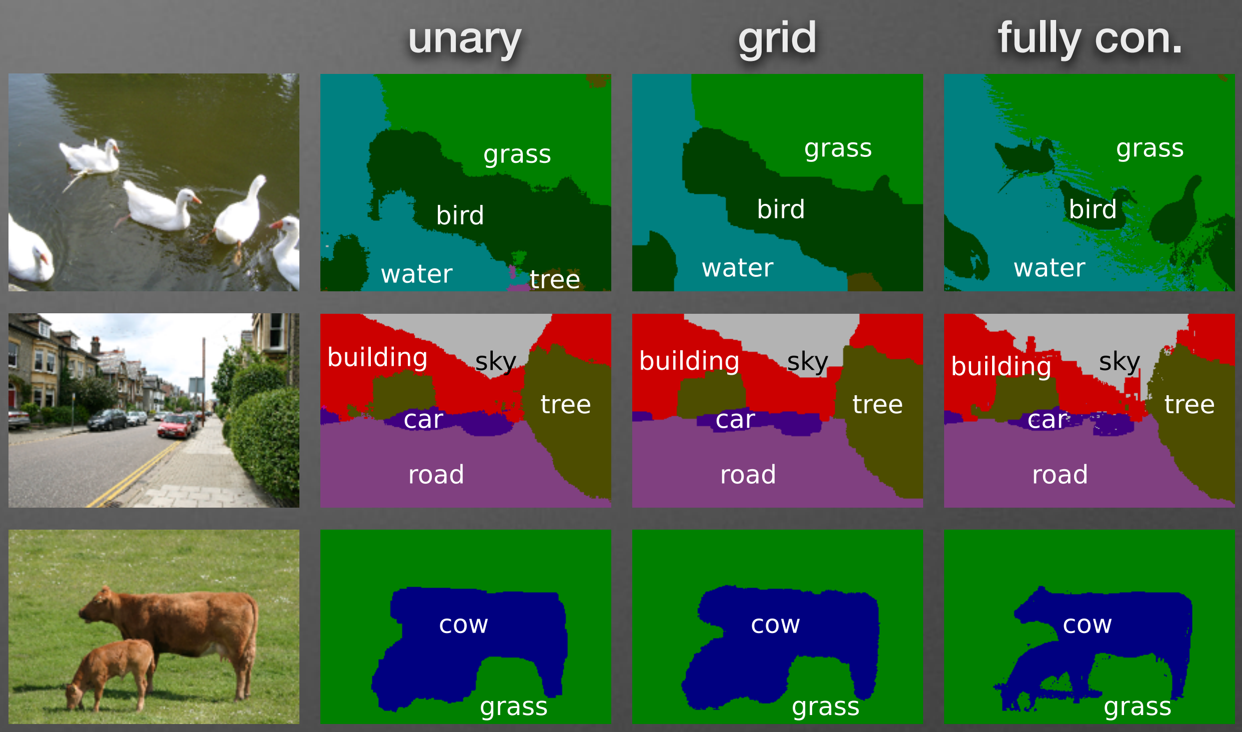


Results - MSRC

MSRC dataset

- 591 images
- 21 classes

	TIME	GLOBAL	AVERAGE
UNARY	-	84.0	76.6
GRID CRF	1s	84.6	77.2

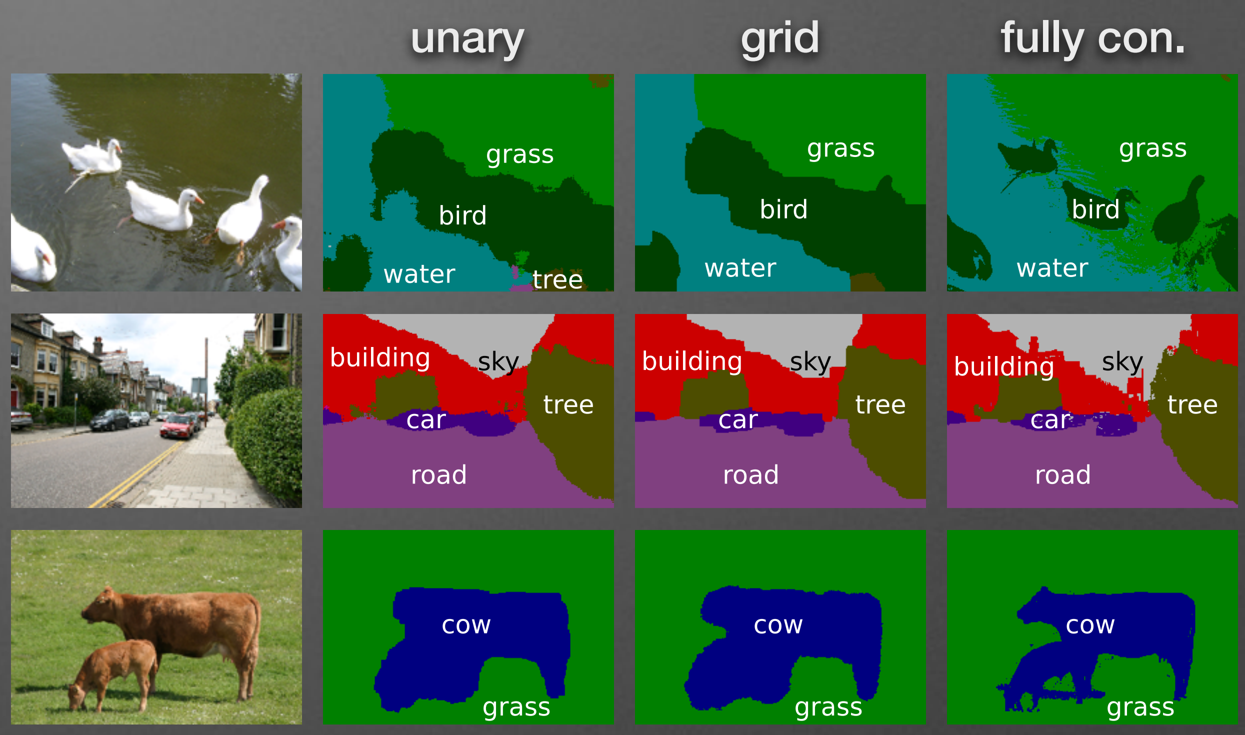


Results - MSRC

MSRC dataset

- 591 images
- 21 classes

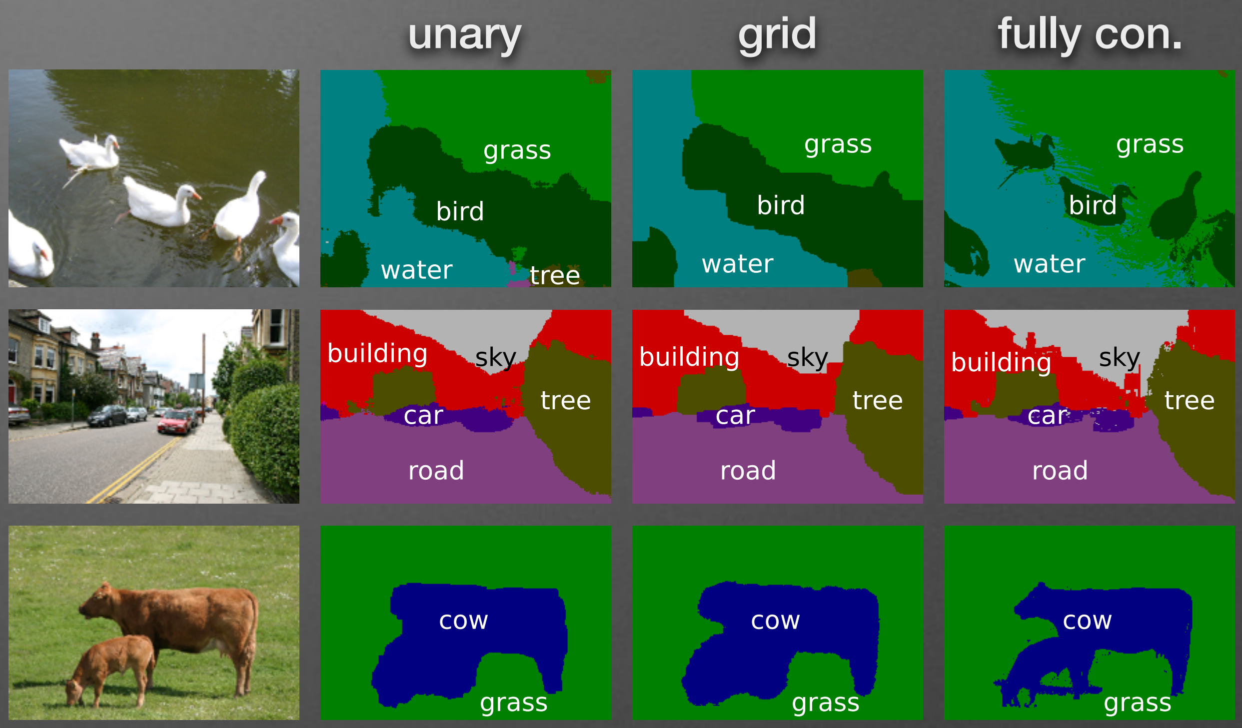
	TIME	GLOBAL	AVERAGE
UNARY	-	84.0	76.6
GRID CRF	1s	84.6	77.2
FC CRF	0.2s	**86.0**	**78.3**



Results - MSRC

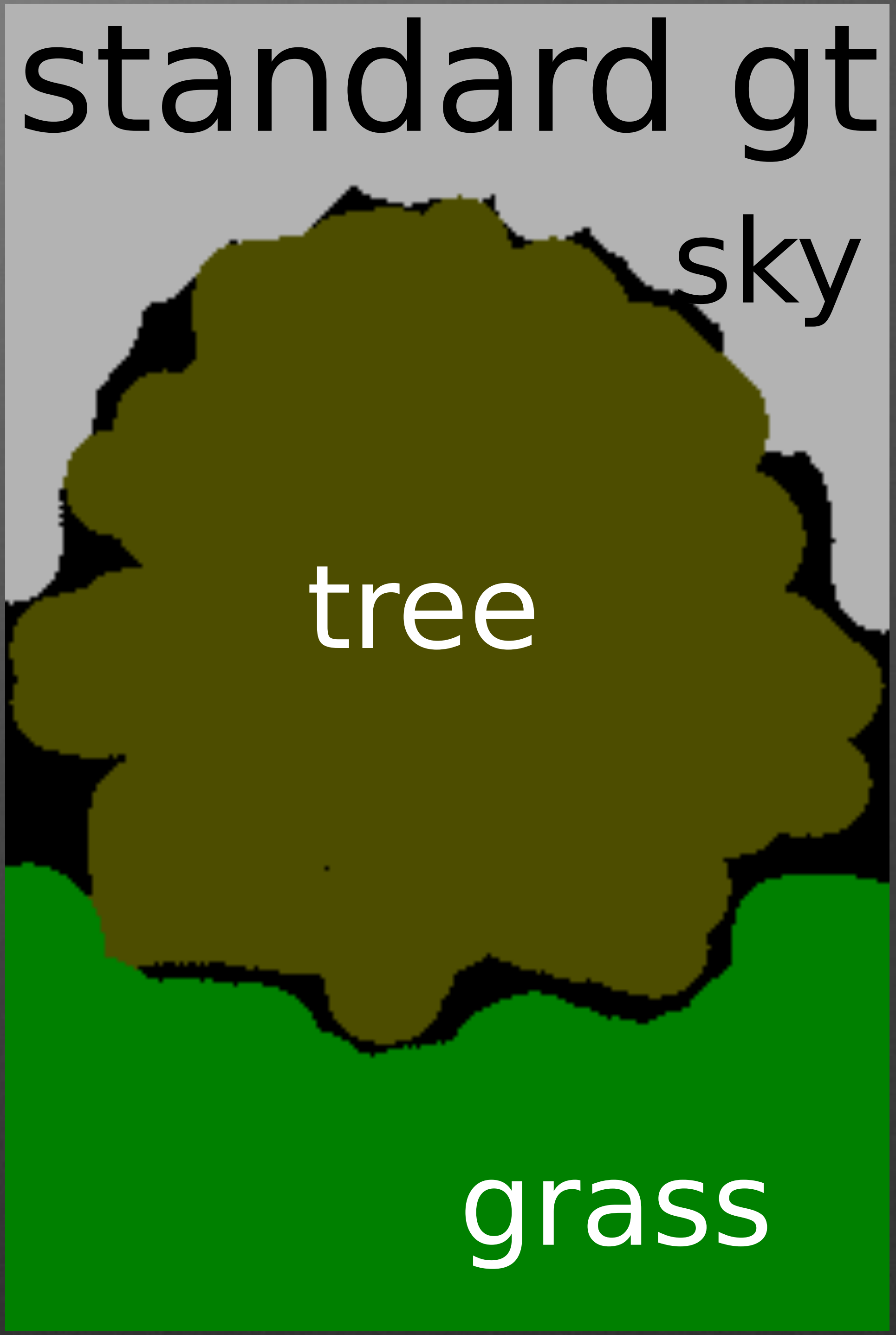
MSRC dataset

- 591 images
- 21 classes



	TIME	GLOBAL	AVERAGE
UNARY	-	84.0	76.6
GRID CRF	1s	84.6	77.2
FC CRF	0.2s	**86.0**	**78.3**
FILTER	**0.05s**	85.0	77.5

Results - MSRC



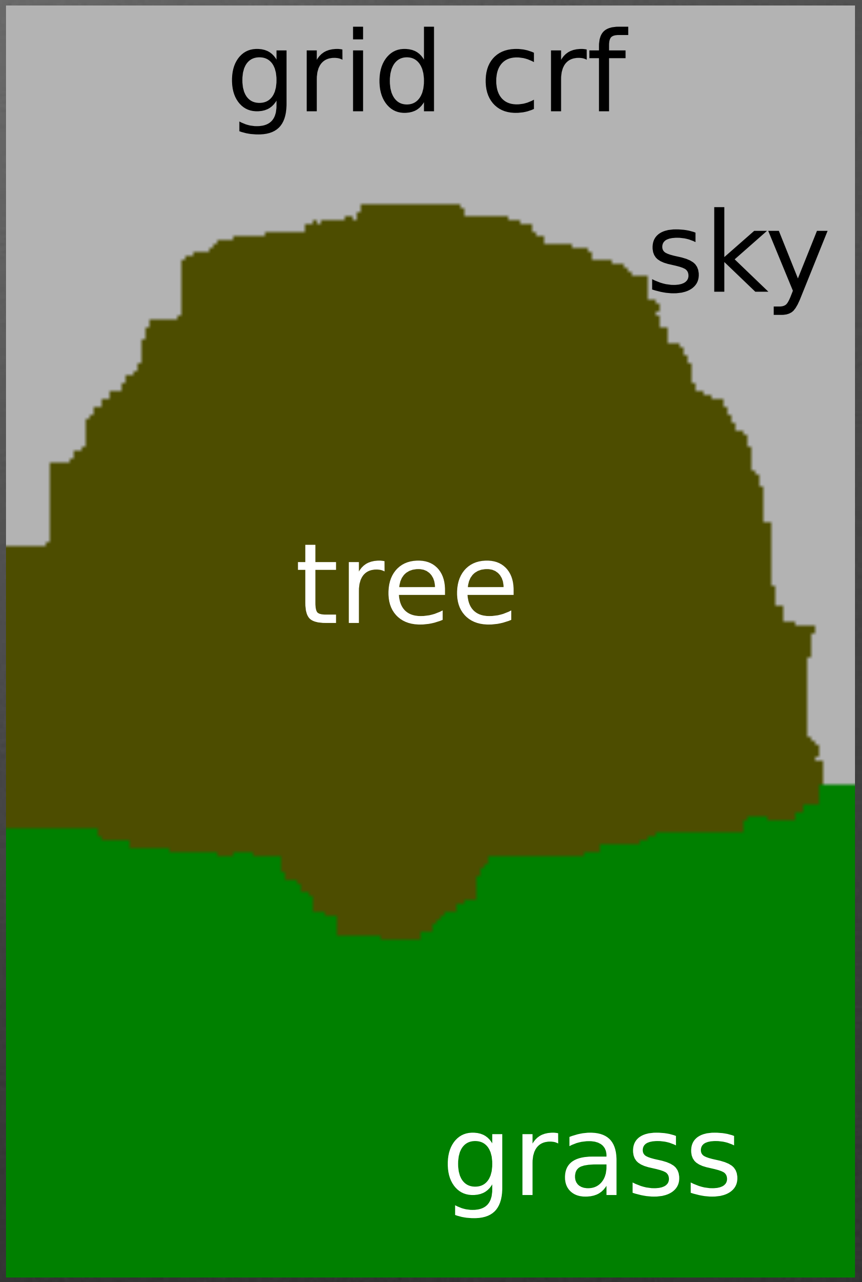
Results - MSRC

Results - MSRC

MSRC Accurate annotations

- 94 images
- hand annotated (30 min each)
- unary train on standard anno.
- 5-fold cross validation

	GLOBAL	AVERAGE
UNARY	83.2±1.5	80.6±2.3
GRID CRF	84.8±1.5	82.4±1.8
FC CRF	**88.2±0.7**	**84.7±0.7**

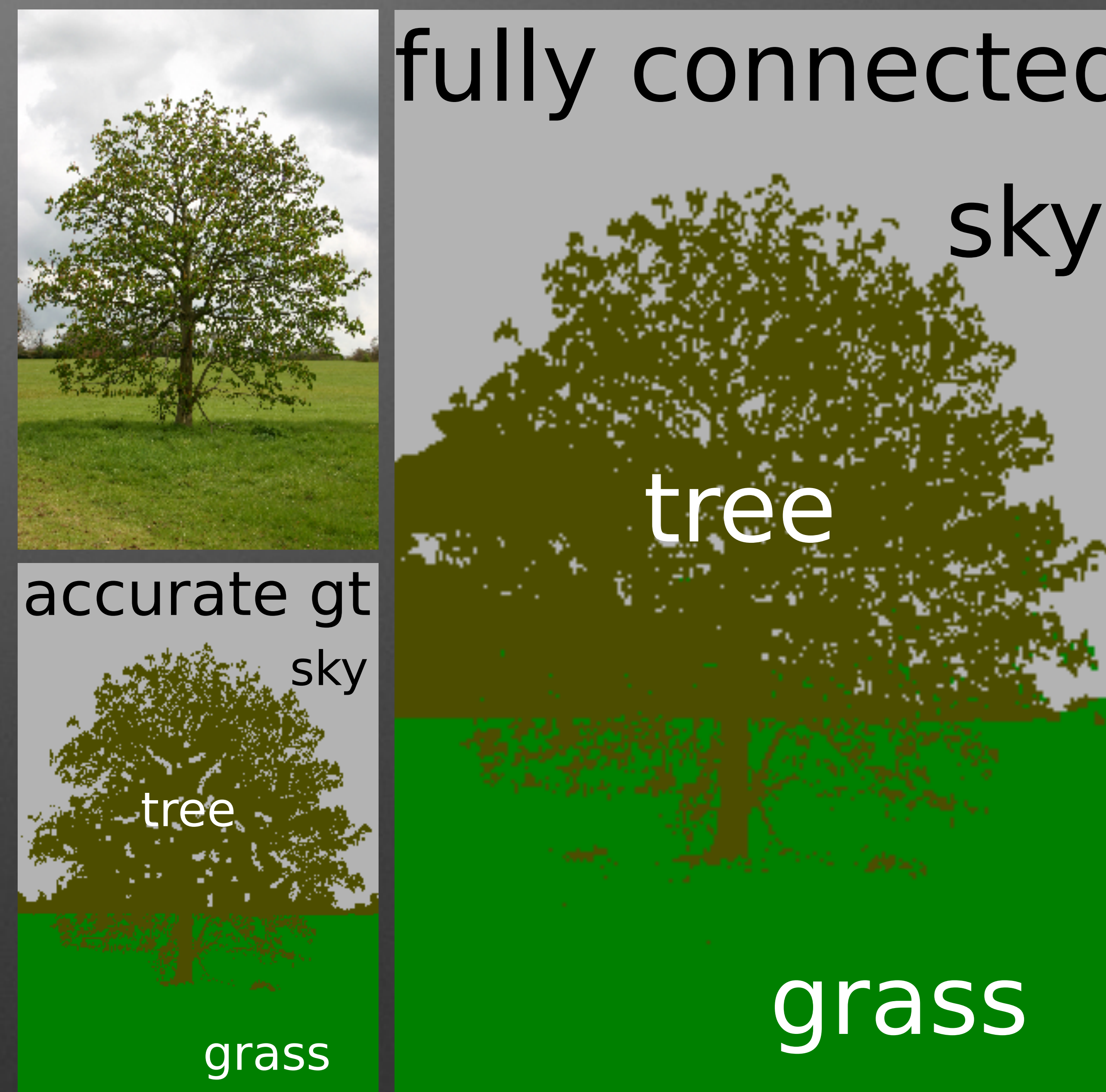


Results - MSRC

MSRC Accurate annotations

- 94 images
- hand annotated (30 min each)
- unary train on standard anno.
- 5-fold cross validation

	GLOBAL	AVERAGE
UNARY	83.2±1.5	80.6±2.3
GRID CRF	84.8±1.5	82.4±1.8
FC CRF	**88.2±0.7**	**84.7±0.7**



Results - VOC 2010

PASCAL VOC 2010

- 1928 images
- 20 classes + background

	TIME	IOU ACCURACY
UNARY	-	27.6
GRID CRF	2.5s	28.3
FC CRF	**0.5s**	**29.1**

Average Precision (AP %)

	mean	aero plane	bicycle	bird	boat	bottle	bus	car	cat	chair	cow	dining table	dog	horse	motor bike	person	potted plant	sheep	sofa	train	tv/ monitor	submission date
Oxford_TVG_CRF_RNN_COCO [?]	**73.4**	87.7	52.1	**88.5**	**66.9**	68.6	87.5	82.8	84.3	30.5	76.1	**63.5**	77.5	**79.9**	85.4	81.1	57.6	80.4	51.7	76.6	**68.7**	10-Apr-2015
DeepLab-CRF-COCO-LargeFOV [?]	72.7	**89.1**	38.3	88.1	63.3	69.7	87.1	**83.1**	**85.0**	29.3	**76.5**	56.5	**79.8**	77.9	**85.8**	82.4	57.4	**84.3**	54.9	**80.5**	64.1	18-Mar-2015
DeepLab-MSc-CRF-LargeFOV [?]	71.6	84.4	**54.5**	81.5	63.6	65.9	85.1	79.1	83.4	30.7	74.1	59.8	79.0	76.1	83.2	80.8	59.7	82.2	50.4	73.1	63.7	02-Apr-2015
MSRA_BoxSup [?]	71.0	86.4	35.5	79.7	65.2	65.2	84.3	78.5	83.7	30.5	76.2	62.6	79.3	76.1	82.1	81.3	57.0	78.2	**55.0**	72.5	68.1	10-Feb-2015
Context_Deep_CNN_CRF [?]	70.7	87.5	37.7	75.8	57.4	**72.3**	**88.4**	82.6	80.0	**33.4**	71.5	55.0	79.3	78.4	81.3	**82.7**	56.1	79.8	48.6	77.1	66.3	14-Apr-2015
DeepLab-CRF-COCO-Strong [?]	70.4	85.3	36.2	84.8	61.2	67.5	84.6	81.4	81.0	30.8	73.8	53.8	77.5	76.5	82.3	81.6	56.3	78.9	52.3	76.6	63.3	11-Feb-2015
Oxford_TVG_CRF_RNN [?]	70.4	85.5	36.7	77.2	62.9	66.7	85.9	78.1	82.5	30.1	74.8	59.2	77.3	75.0	82.8	79.7	**59.8**	78.3	50.0	76.9	65.7	26-Mar-2015
DeepLab-CRF-LargeFOV [?]	70.3	83.5	36.6	82.5	62.3	66.5	85.4	78.5	83.7	30.4	72.9	60.4	78.5	75.5	82.1	79.7	58.2	82.0	48.8	73.7	63.3	28-Mar-2015
DeepLab-CRF-MSc [?]	67.1	80.4	36.8	77.4	55.2	66.4	81.5	77.5	78.9	27.1	68.2	52.7	74.3	69.6	79.4	79.0	56.9	78.8	45.2	72.7	59.3	30-Dec-2014
DeepLab-CRF [?]	66.4	78.4	33.1	78.2	55.6	65.3	81.3	75.5	78.6	25.3	69.2	52.7	75.2	69.0	79.1	77.6	54.7	78.3	45.1	73.3	56.2	23-Dec-2014
CRF_RNN [?]	65.2	80.9	34.0	72.9	52.6	62.5	79.8	76.3	79.9	23.6	67.7	51.8	74.8	69.9	76.9	76.9	49.0	74.7	42.7	72.1	59.6	10-Feb-2015
TTI_zoomout_16 [?]	64.4	81.9	35.1	78.2	57.4	56.5	80.5	74.0	79.8	22.4	69.6	53.7	74.0	76.0	76.6	68.8	44.3	70.2	40.2	68.9	55.3	24-Nov-2014
Hypercolumn [?]	62.6	68.7	33.5	69.8	51.3	70.2	81.1	71.9	74.9	23.9	60.6	46.9	72.1	68.3	74.5	72.9	52.6	64.4	45.4	64.9	57.4	09-Apr-2015
FCN-8s [?]	62.2	76.8	34.2	68.9	49.4	60.3	75.3	74.7	77.6	21.4	62.5	46.8	71.8	63.9	76.5	73.9	45.2	72.4	37.4	70.9	55.1	12-Nov-2014
MSRA_CFM [?]	61.8	75.7	26.7	69.5	48.8	65.6	81.0	69.2	73.3	30.0	68.7	51.5	69.1	68.1	71.7	67.5	50.4	66.5	44.4	58.9	53.5	17-Dec-2014
TTI_zoomout [?]	58.4	70.3	31.9	68.3	46.4	52.1	75.3	68.4	75.3	19.2	58.4	49.9	69.6	63.0	70.1	67.6	41.5	64.0	34.9	64.2	47.3	17-Nov-2014
SDS [?]	51.6	63.3	25.7	63.0	39.8	59.2	70.9	61.4	54.9	16.8	45.0	48.2	50.5	51.0	57.7	63.3	31.8	58.7	31.2	55.7	48.5	21-Jul-2014
NUS_UDS [?]	50.0	67.0	24.5	47.2	45.0	47.9	65.3	60.6	58.5	15.5	50.8	37.4	45.8	59.9	62.0	52.7	40.8	48.2	36.8	53.1	45.6	29-Oct-2014
TTIC-divmbest-rerank [?]	48.1	62.7	25.6	46.9	43.0	54.8	58.4	58.6	55.6	14.6	47.5	31.2	44.7	51.0	60.9	53.5	36.6	50.9	30.1	50.2	46.8	15-Nov-2012
BONN_O2PCPMC_FGT_SEGM [?]	47.8	64.0	27.3	54.1	39.2	48.7	56.6	57.7	52.5	14.2	54.8	29.6	42.2	58.0	54.8	50.2	36.6	58.6	31.6	48.4	38.6	08-Aug-2013
BONN_O2PCPMC_FGT_SEGM [?]	47.5	63.4	27.3	56.1	37.7	47.2	57.9	59.3	55.0	11.5	50.8	30.5	45.0	58.4	57.4	48.6	34.6	53.3	32.4	47.6	39.2	23-Sep-2012
BONNGC_O2P_CPMC_CSI [?]	46.8	63.6	26.8	45.6	41.7	47.1	54.3	58.6	55.1	14.5	49.0	30.9	46.1	52.6	58.2	53.4	32.0	44.5	34.6	45.3	43.1	23-Sep-2012
BONN_CMBR_O2P_CPMC_LIN [?]	46.7	63.9	23.8	44.6	40.3	45.5	59.6	58.7	57.1	11.7	45.9	34.9	43.0	54.9	58.0	51.5	34.6	44.1	29.9	50.5	44.5	23-Sep-2012

Questions?

Questions?


fully connect
all the things


fully connected
cat
background