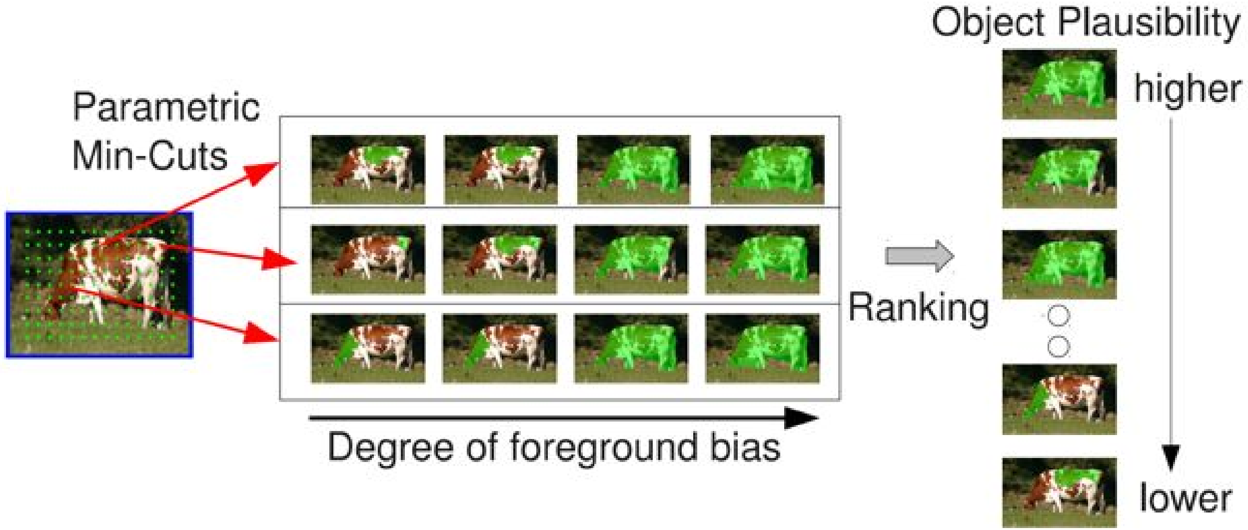
Constrained Parametric Min-Cuts for Automatic Object Segmentation

Joao Carreira and Cristian Sminchisescu
Computer Vision and Machine Learning Group,
Institute for Numerical Simulation,
Faculty of Mathematics and Natural Sciences, University of Bonn

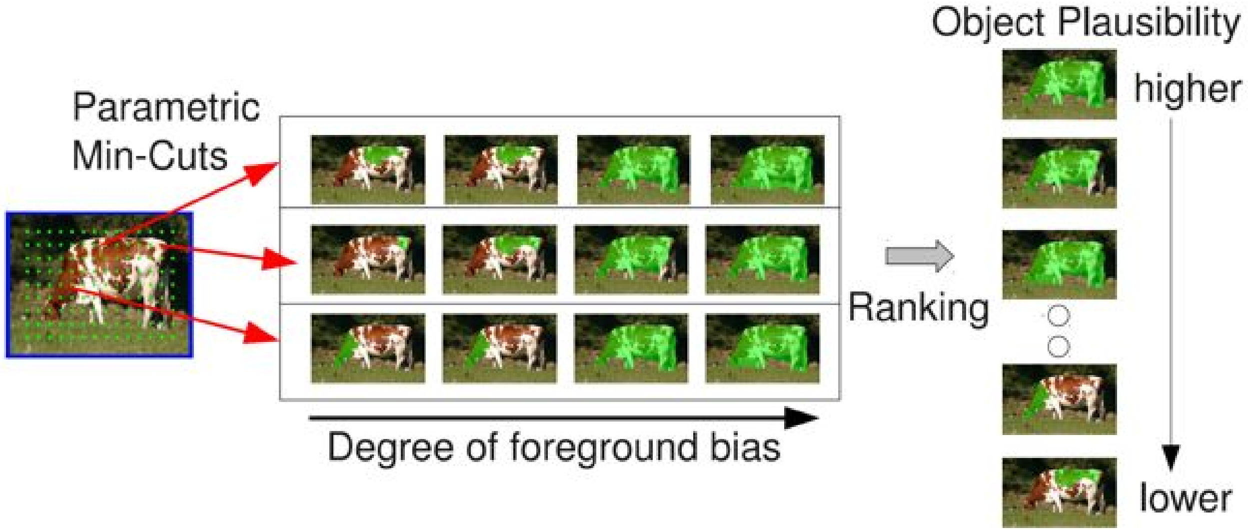
Outline

- Introduction
- Implementation
- Evaluation
- Conclusion



Outline

- **Introduction**
- Implementation
- Evaluation
- Conclusion



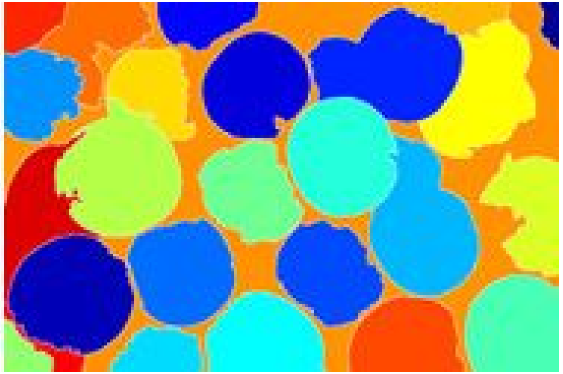
Introduction

- Problem
  - Segmentation - partition image into sets of pixels (superpixels)
  - S: I -> R, where I is an image, R = { {(0, 0), (0, 1)}, {(1, 0)}, {(1, 1)} }

$$\bigcup_{i=1}^{n} R_i = I$$

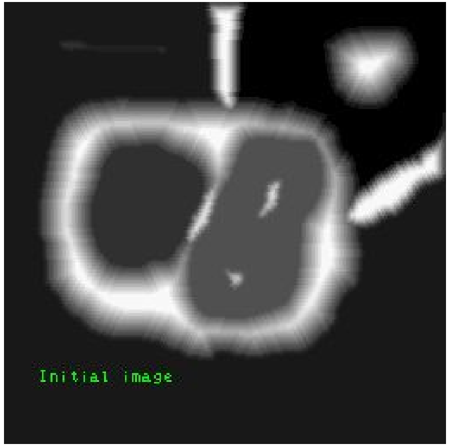
$$R_i \text{ is connected}$$

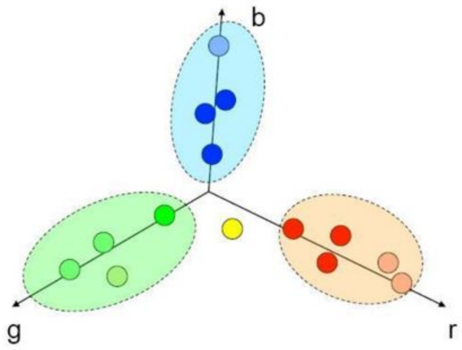
$$R_i \cap R_j = \emptyset \quad \forall_{i,j} \quad i \neq j$$

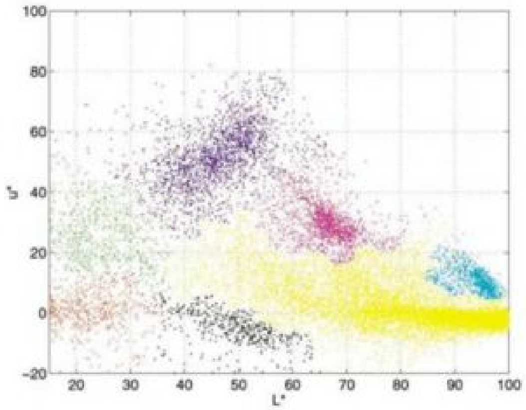


Introduction

- Techniques
  - K-means (rgb cluster)
  - Mean-shift (rgb cluster)
  - Watershed (gradient)

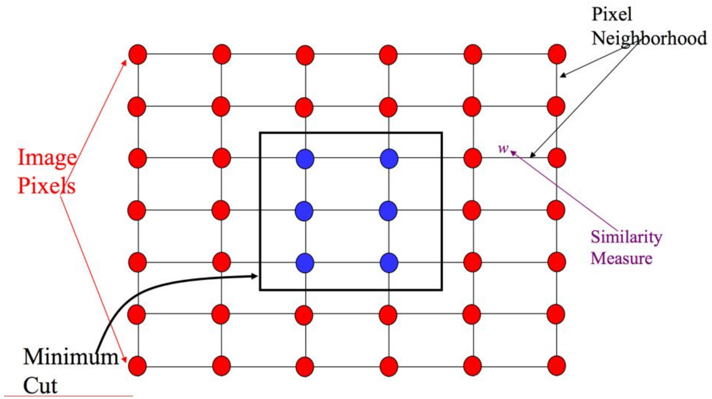




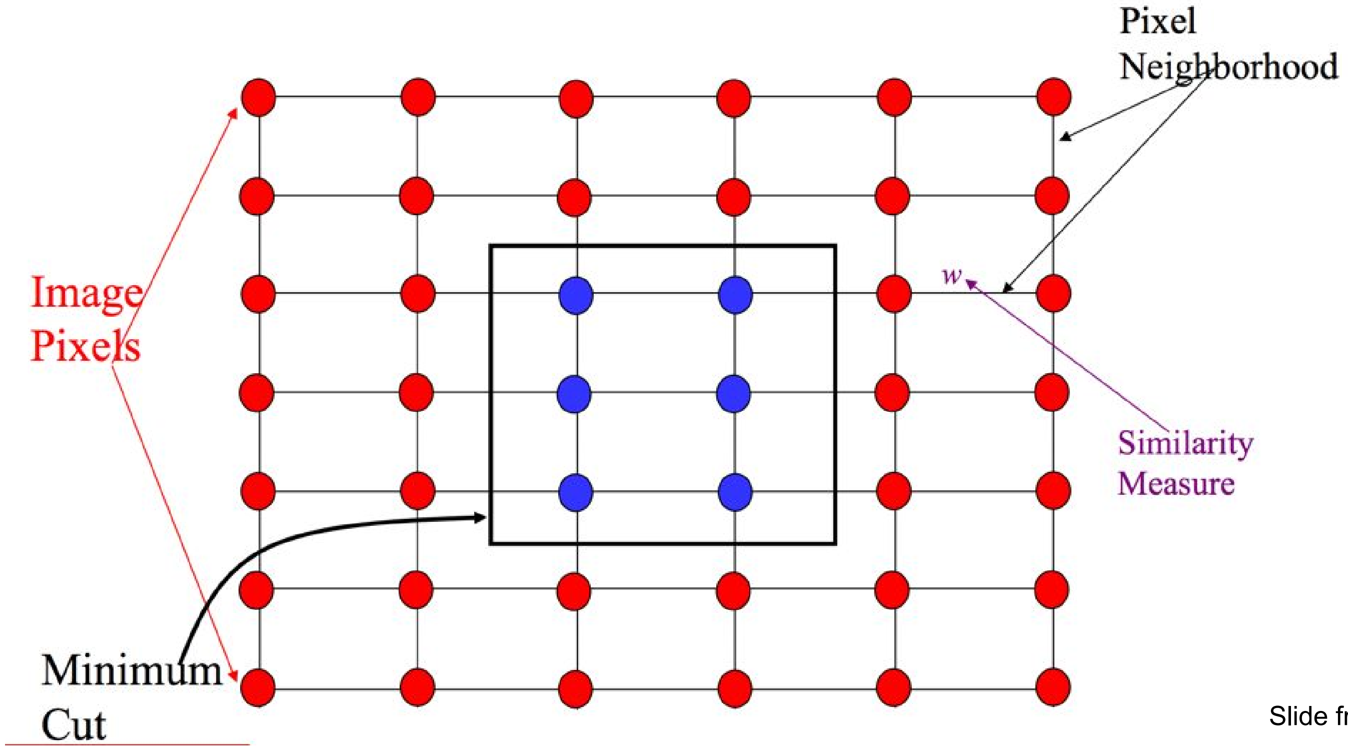


Introduction

- Intuition
  - I -> G (V, E)
    - Nodes are vertices
    - Edges are weighted by affinity/similarity
  - Find the min cut



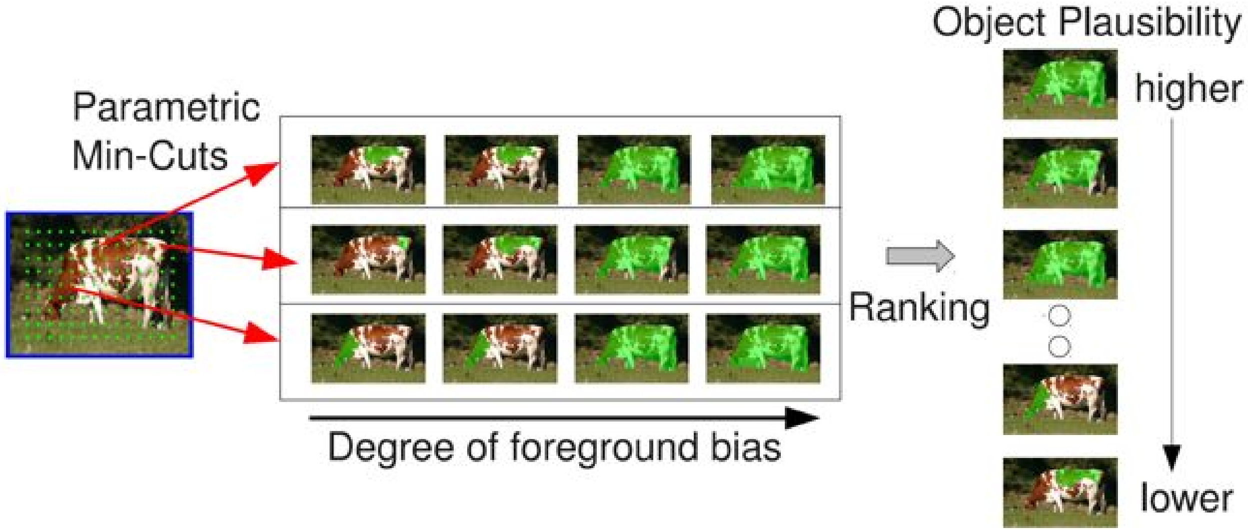
Introduction



Slide from Cornell University

Outline

- Introduction
- **Implementation**
- Evaluation
- Conclusion

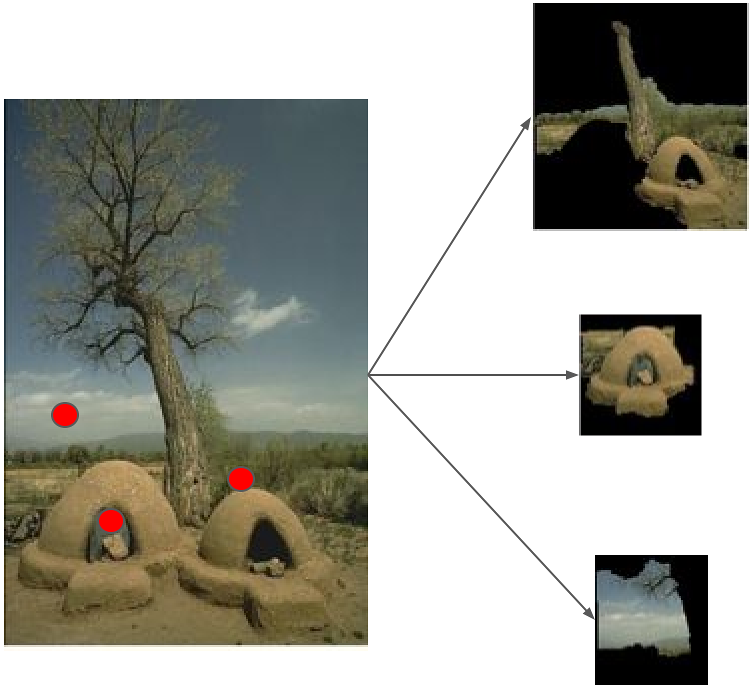


Implementation

- Framework
  - Multiple Hypothesis Generation
    - Solve large number of flows
  - Filter results
    - Remove trivial results
  - Pick a candidate segmentation
    - Regressor to rank based on overlap

Implementation

- Framework
  - **Multiple Hypothesis Generation**
    - Solve large number of flows
  - Filter results
    - Remove trivial results
  - Pick a candidate segmentation
    - Regressor to rank based on overlap

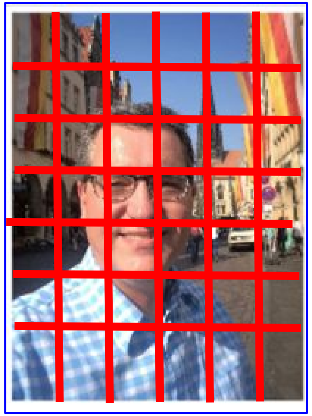


Implementation

- Foreground Seeds
  - Regular grid geometry
  - Centroids of normalized cuts
  - Centroids of algorithm by P. F. Felzenszwalb and D. P. Huttenlocher (FH)
- Background Seeds
  - Image boundary
  - Vertical
  - Horizontal
  - All but bottom

Seed placement	MSRC score	Weizmann score
Grid	0.85 ± 0.1	0.93 ± 0.06
NCuts	0.86 ± 0.09	0.93 ± 0.07
FH	0.87 ± 0.08	0.93 ± 0.07

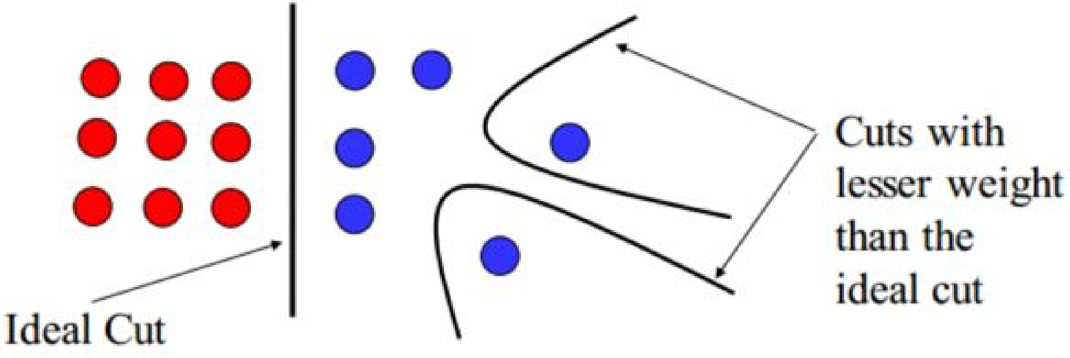
Background Seeds

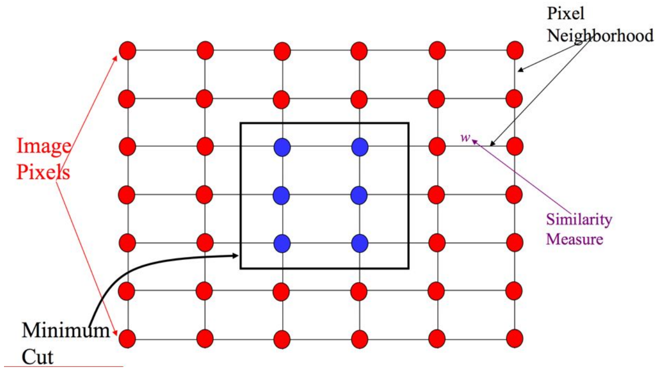


Foreground Seeds

Implementation

- Vanilla min cut





How do we get by this?

Implementation

- Objective Function
  - Source s, sink t represent foreground, background
  - Given seeds V_f, V_b, minimize energy over pixel labels
  - Solve for multiple values of lambda (log scale)

$$E^{\lambda}(X) = \sum_{u \in \mathcal{V}} D_{\lambda}(x_u) + \sum_{(u,v) \in \mathcal{E}} V_{uv}(x_u, x_v)$$

$$D_{\lambda}(x_u) = \begin{cases} 0 & \text{if } x_u = 1, u \notin \mathcal{V}_b \\ \infty & \text{if } x_u = 1, u \in \mathcal{V}_b \\ \infty & \text{if } x_u = 0, u \in \mathcal{V}_f \\ f(x_u) + \lambda & \text{if } x_u = 0, u \notin \mathcal{V}_f \end{cases} \qquad V_{uv}(x_u, x_v) = \begin{cases} 0 & \text{if } x_u = x_v \\ g(u,v) & \text{if } x_u \neq x_v \end{cases}$$

Implementation

- Objective Function
  - Bias Function
    - 0
    - $\ln p_f(x_u) - \ln p_b(x_u)$
  - Similarity Function
    - $g(u,v) = \exp\left[-\frac{\max(gPb(u), gPb(v))}{\sigma^2}\right]$

$$E^\lambda(X) = \sum_{u \in \mathcal{V}} D_\lambda(x_u) + \sum_{(u,v) \in \mathcal{E}} V_{uv}(x_u, x_v)$$

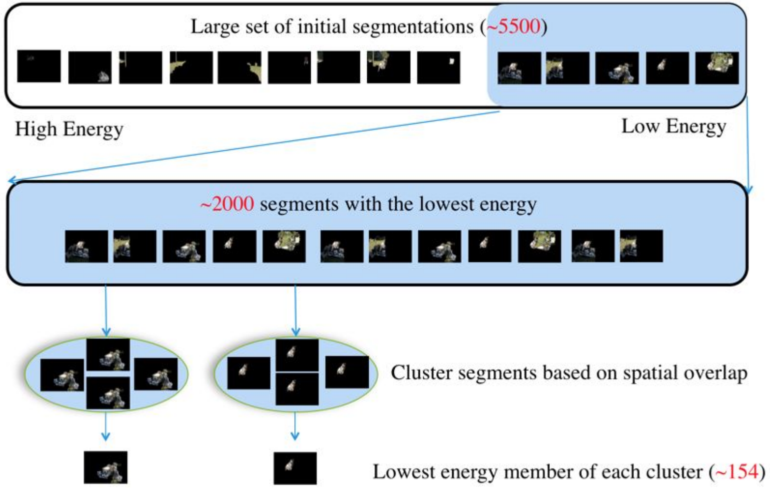
$$D_\lambda(x_u) = \begin{cases} 0 & \text{if } x_u = 1, u \notin \mathcal{V}_b \\ \infty & \text{if } x_u = 1, u \in \mathcal{V}_b \\ \infty & \text{if } x_u = 0, u \in \mathcal{V}_f \\ f(x_u) + \lambda & \text{if } x_u = 0, u \notin \mathcal{V}_f \end{cases} \qquad V_{uv}(x_u, x_v) = \begin{cases} 0 & \text{if } x_u = x_v \\ g(u,v) & \text{if } x_u \neq x_v \end{cases}$$

Implementation

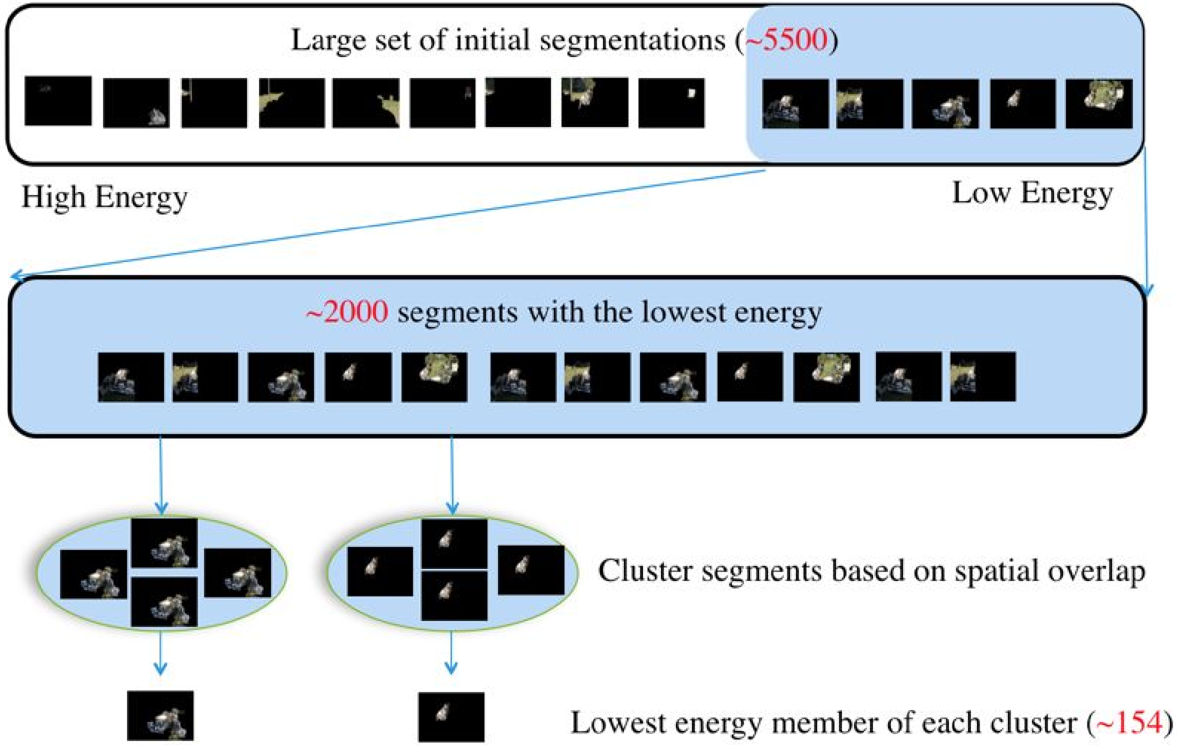
- Framework
    - Multiple Hypothesis Generation
        - Solve large number of flows
    - **Filter results**
        - Remove trivial results
    - Pick a candidate segmentation
        - Regressor to rank based on overlap

Implementation

- Fast Rejection Step
  - Large homogenous regions
  - Filter out small segments
  - Sort by ratio cut energy
  - Group overlapping regions



Implementation

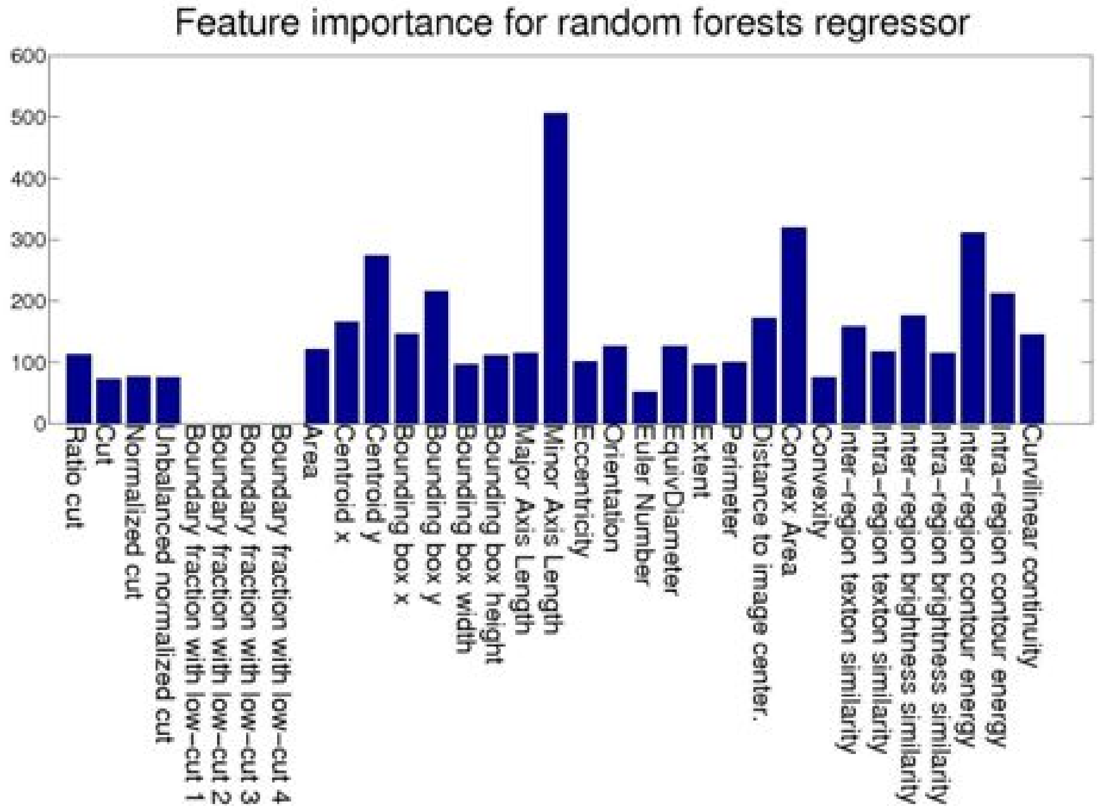


Implementation

- Framework
  - Multiple Hypothesis Generation
    - Solve large number of flows
  - Filter results
    - Remove trivial results
  - **Pick a candidate segmentation**
    - Regressor to rank based on overlap

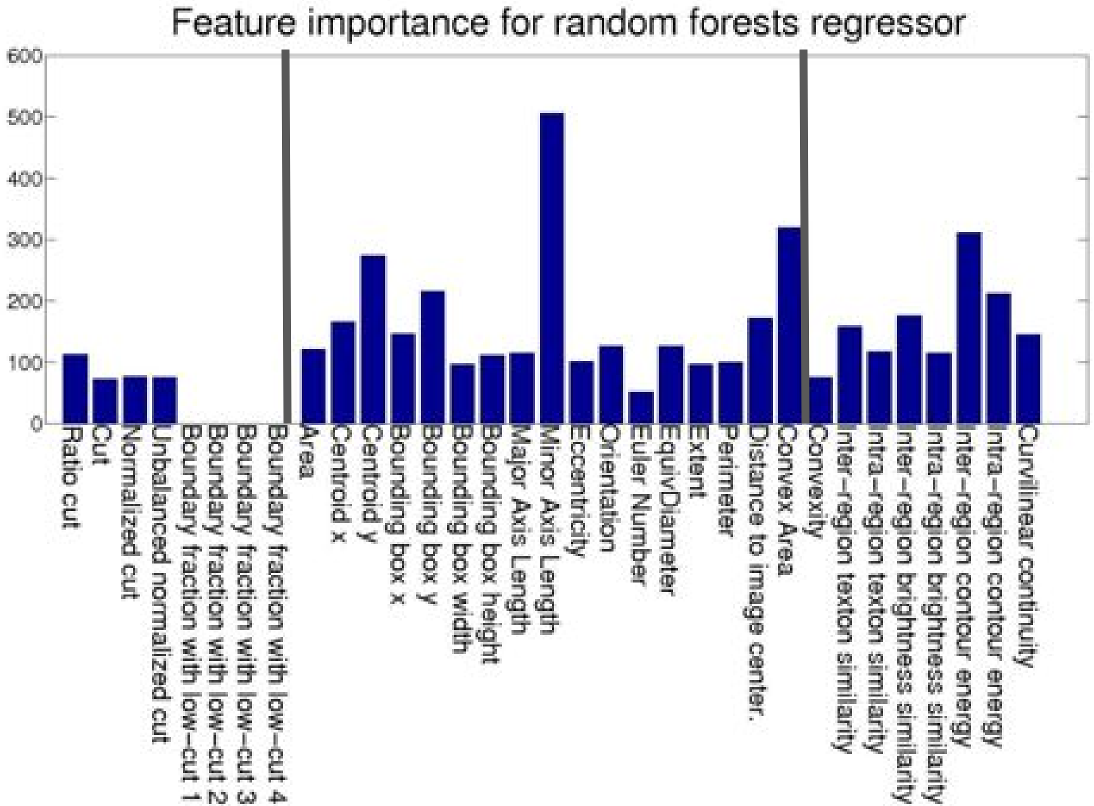
Implementation

- Features
  - Graph partition (8 features)
    - Cut
    - Normalized cut
  - Region (18 features)
    - Area
    - Perimeter
  - Gestalt (8 features)
    - Texton similarity
    - Contour energy



Implementation

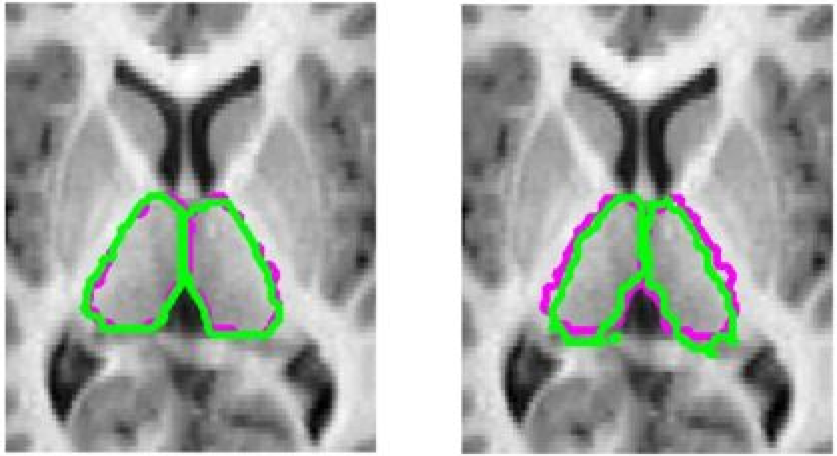
- What really matters?
  - Graph partition
    - Not that important
  - Region
    - Minor axis length
  - Gestalt
    - Significant weight
    - Contour energy

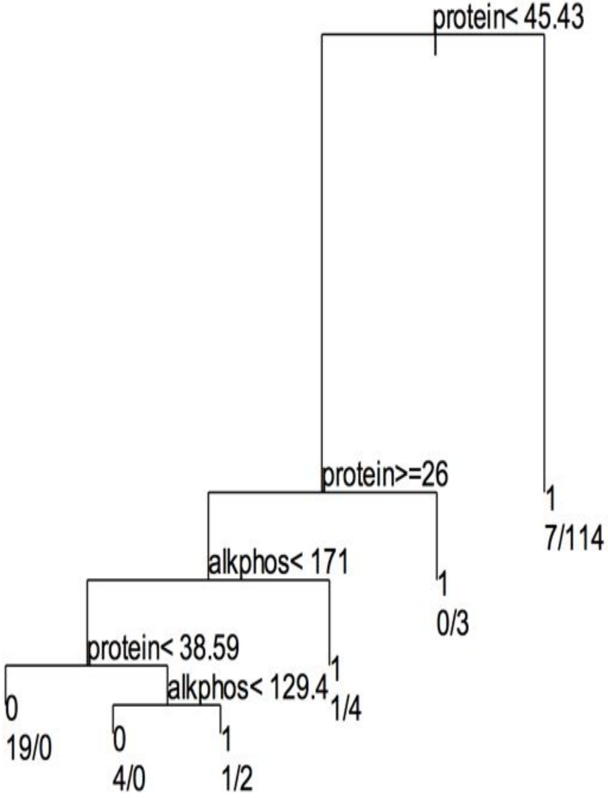


Implementation

- Random Forest Regressor
  - Regress on largest overlap
  - How object-like a segment is
  - Penalizes under and over segmentations
  - Scale invariant

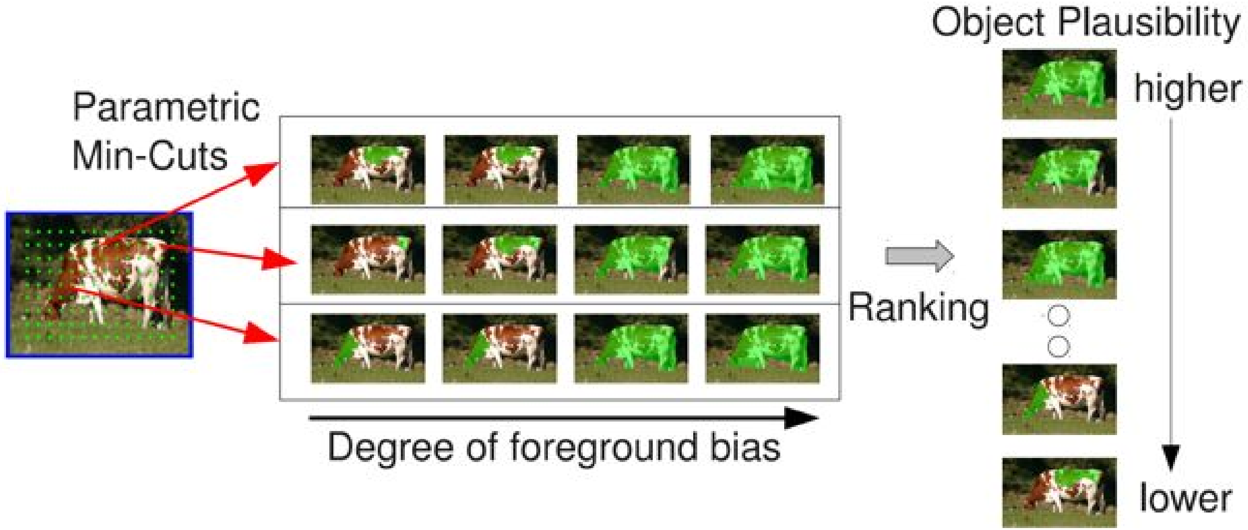
$$O(S,G) = \frac{|S \cap G|}{|S \cup G|}$$





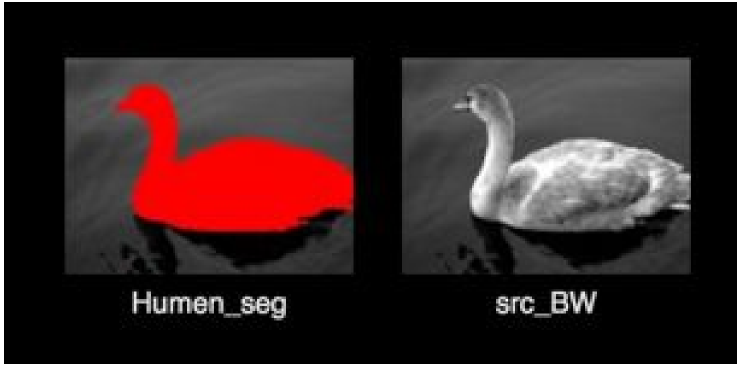
Outline

- Introduction
- Implementation
- **Evaluation**
- Conclusion



Evaluation

- Datasets
  - Weizmann
    - Single prominent foreground
  - MSRC
    - 23 Classes (grass, water)
  - VOC2009
    - Challenging Flickr data
- Measures
  - F-Measure
    - $F = \frac{2RP}{P+R}$
  - Segmentation covering
    - $C(S', S) = \frac{1}{N} \sum_{R \in S} |R| * \max_{R' \in S'} O(R, R')$



Weizmann

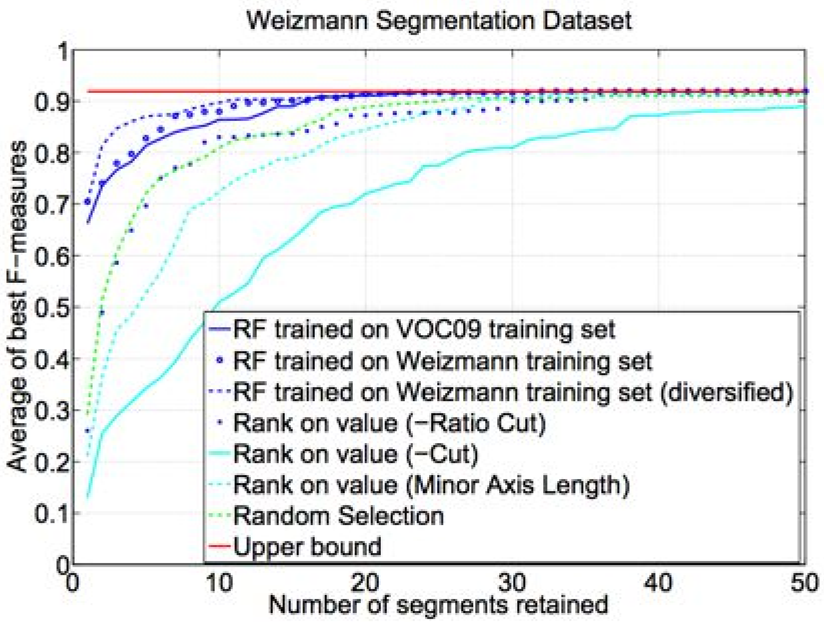
VOC2009

Evaluation

- Weizmann
  - 100 images
  - Grey scale
  - F-Measure
- Other works
  - Bagon - user clicks inside region
  - Alpert - automatic

$$F = \frac{2RP}{P+R}$$

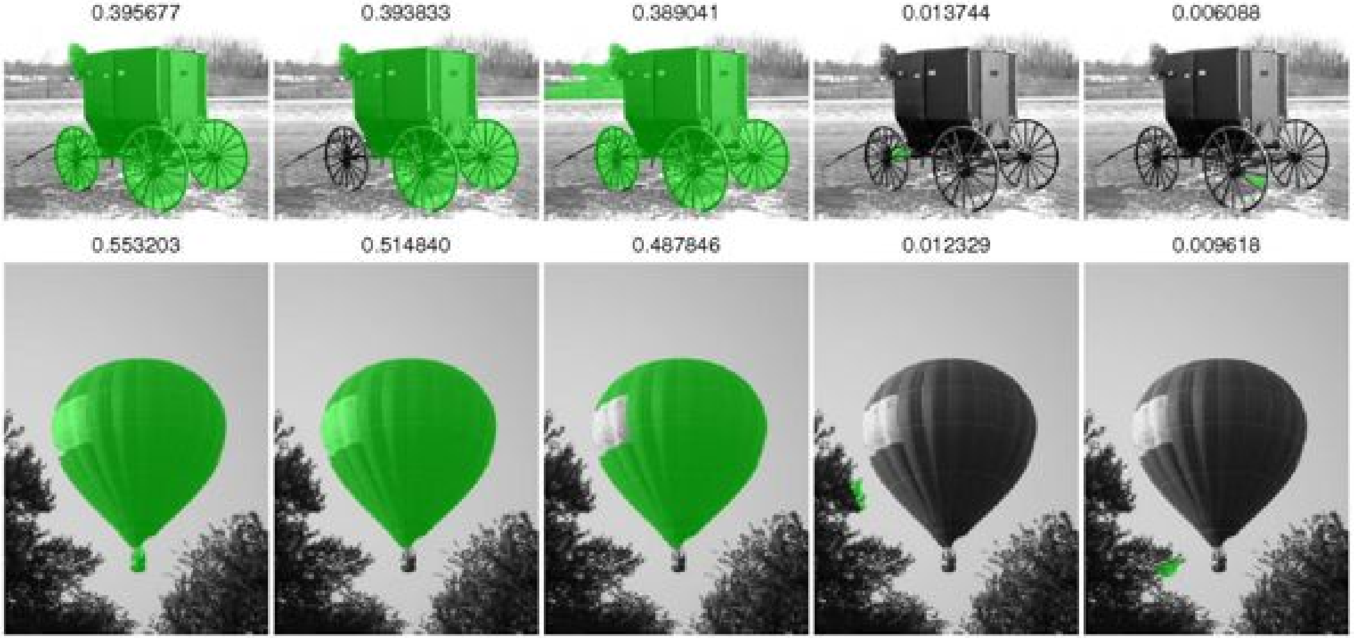
Weizmann	F-measure
CPMC	0.93 ± 0.009
Bagon *et al.*	0.87 ± 0.010
Alpert *et al.*	0.86 ± 0.012



Evaluation

- Weizmann

Values are very similar...

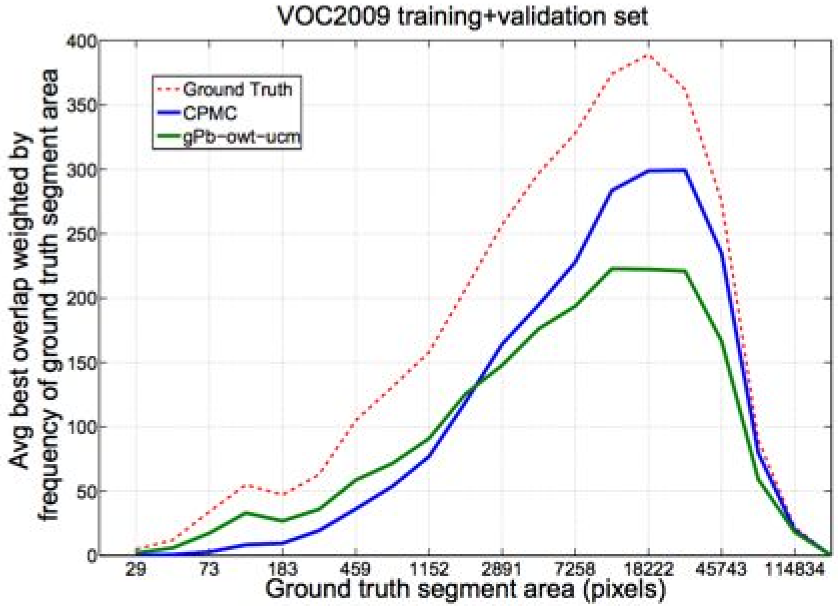


Evaluation

- VOC2009
  - Flickr data
  - 20 different object classes
  - Multiple ground truths
  - Segmentation covering measure

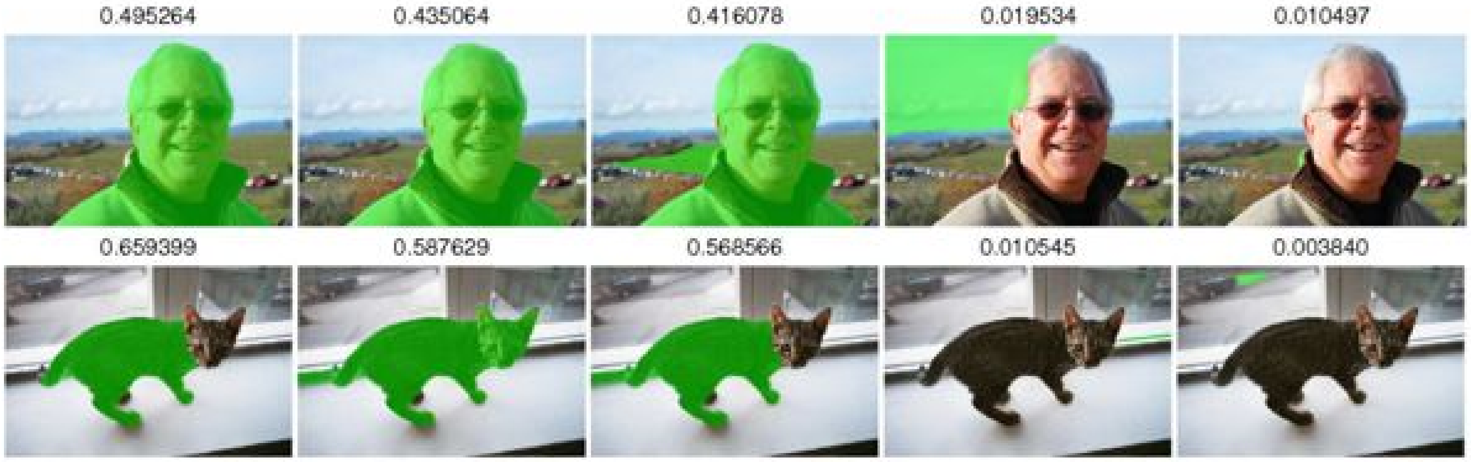
$$C(S', S) = \frac{1}{N} \sum_{R \in S} |R| * \max_{R' \in S'} O(R, R')$$

VOC2009	Covering	N Segments
CPMC	0.78 ± 0.18	154
gPb-owt-ucm	0.61 ± 0.20	1286



Evaluation

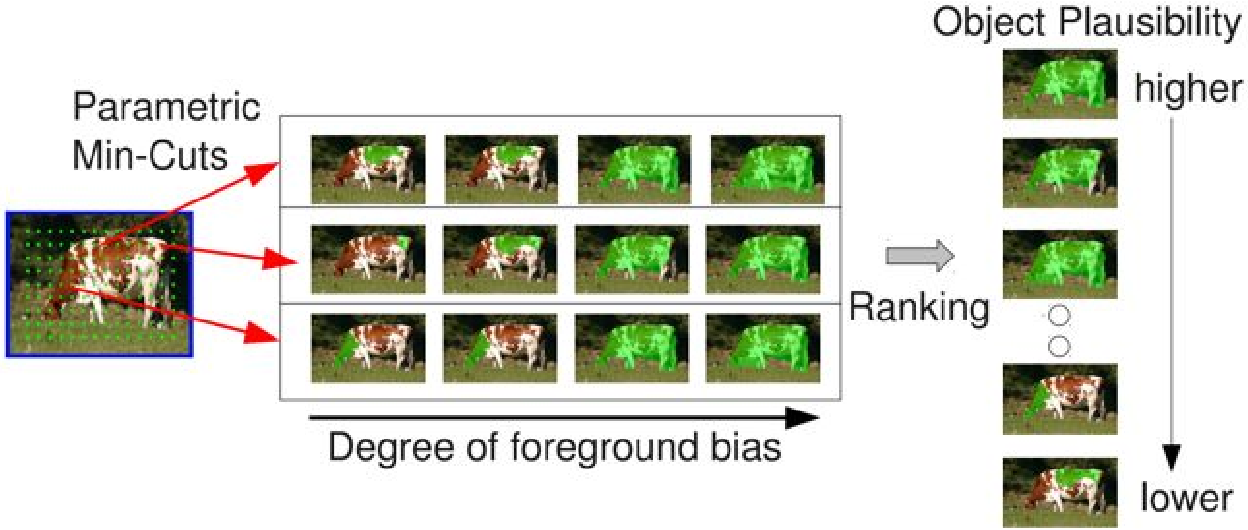
- VOC2009



Ranked a correct segmentation much lower!

Outline

- Introduction
- Implementation
- Evaluation
- **Conclusion**



Conclusion

- Feature selection
  - Can we learn what features to use?
- Regression model
  - Improvements?
- Properties
  - Uses only low to mid level image features