On Combining CRF and CNN

Lena Gorelick and Olga Veksler

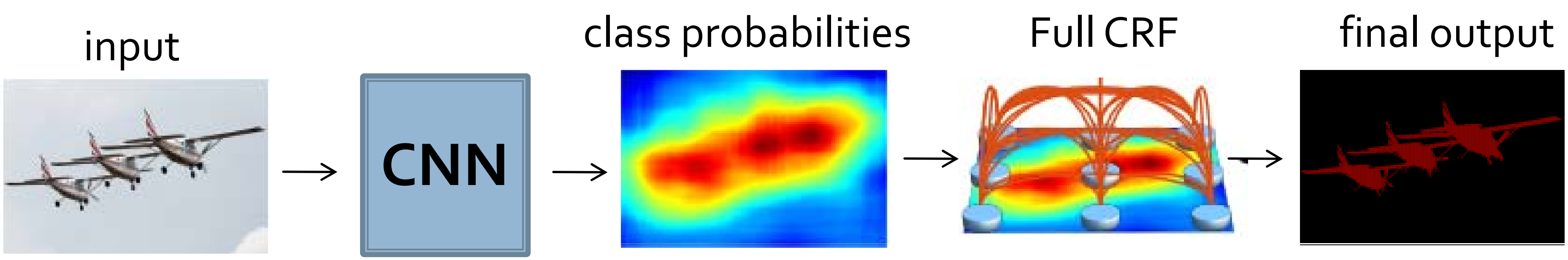
Contents

1. CNN+ [CRF as post processing]
   - Full CRFs with Quantized Edges
     - understand properties of full CRFs
     - efficient graph-cut optimization algorithm
2. [CNN+CRF] in end-end trainable system
   - CNN architecture to simulate CRF

Part 1: Full CRF with Quantized Edges

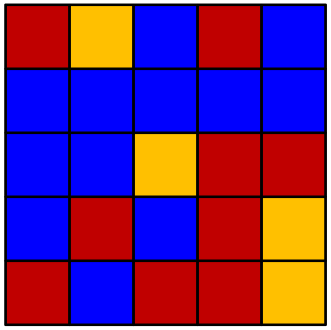
CNN + CRF Post-processing

- CNN does not directly model spatial regularity
- Combine with CRF
  - Chen et.al. ICLR'2015

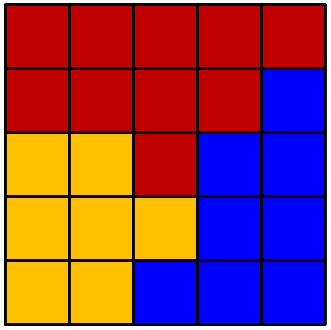


- Fully connected CRF with Gaussain weighted edges is often used, optimized with mean-field
  - regularization properties of full-CRF?
  - does mean field work well?

CRF Energy with Potts Potentials



high energy



low energy

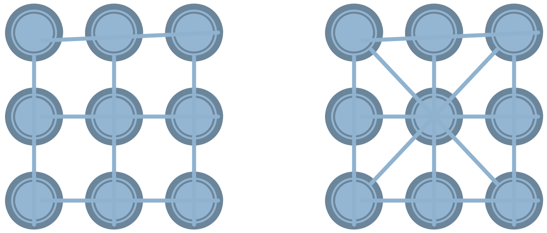
- Find labeling $\boldsymbol{x}$ minimizing energy

$$E(x) = \sum_{p} D_p(x_p) + \sum_{(p,q)\in N} w_{pq}[x_p \neq x_q]$$

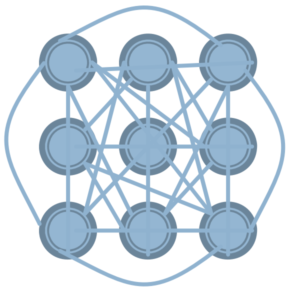
- Optimization
  - Solved exactly in binary case with a graph cut
  - NP hard in multi-label case
  - expansion algorithm approximation (factor of 2) [Boykov et.al.'TPAMI01]

Sparse vs. Fully Connected CRF

- Sparsely connected CRFs
  - 4, 8, or small neighbourhood connected
  - TRWS [Kolmogorov 'TPAMI2006] or expansion algorithms work well
  - length regularization [Boykov&Kolmogorov'ICCV2003]



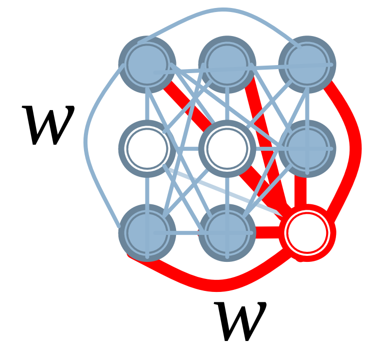
- Fully connected CRFs [Krahenbul&Koltun'NIPS2011]
  - all pixels are neighbors, **n** pixels, $\mathbf{O(n^2)}$ edges
  - naïve application of expansion algorithm, TRWS, etc. is not efficient
  - regularization properties?

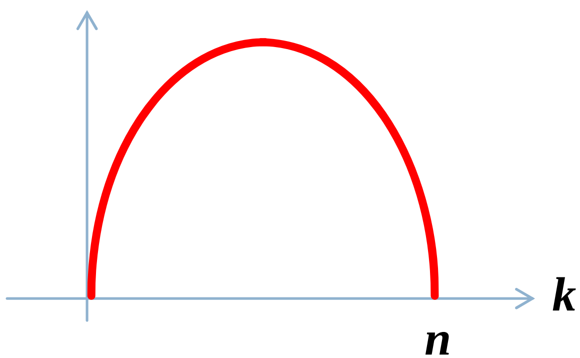


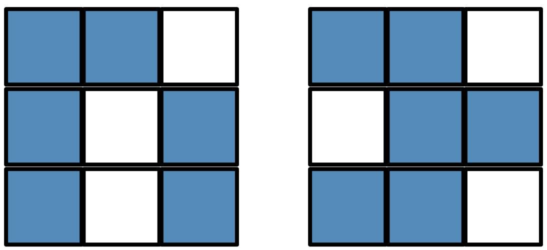
Binary Full CRF with Uniform Weights

- Labels in {0,1}
- All edges have weight w
- Cardinality regularization
  - n pixels in the image, k pixels assigned to label 1
  - pairwise energy is

$$w \cdot (n - k) \cdot k$$



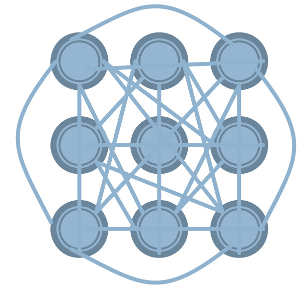




same pairwise cost

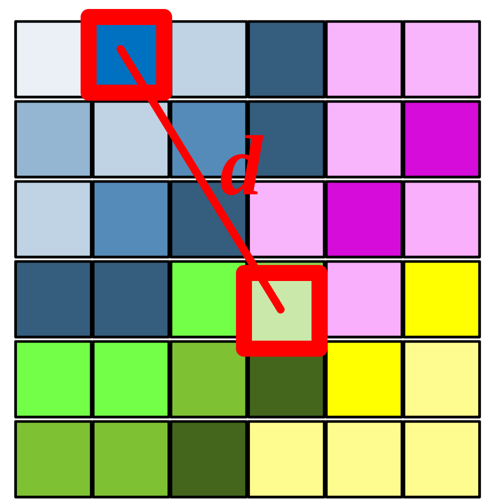
- Efficient optimization
  - for each k
    - find the k pixels with lowest cost for label 1
    - compute total energy
  - chose k corresponding to the smallest energy

Fully Connected CRFs



- Fully connected CRFs [Krahenbul&Koltun'NIPS2011]
  - Gaussian edge weights

$$w_{pq} = \mathbf{exp}\left(-\left|\blacksquare - \square\right|\right)\mathbf{exp}\left(-d\right)$$

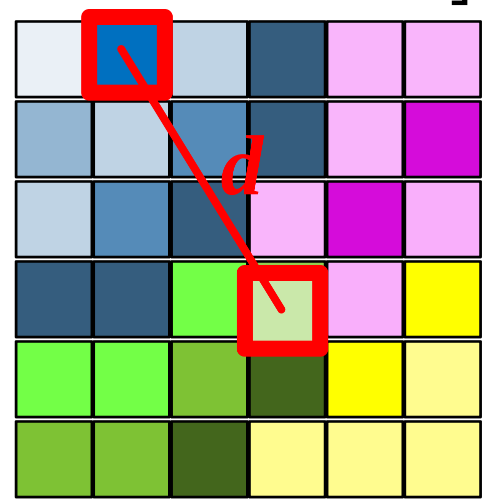


  - efficient mean field inference
    - approximate bilateral filter [Paris&Durand, IJCV'2009]
  - mean field does not work well [Weiss'2001]

Quantized Edge Fully Connected CRFs

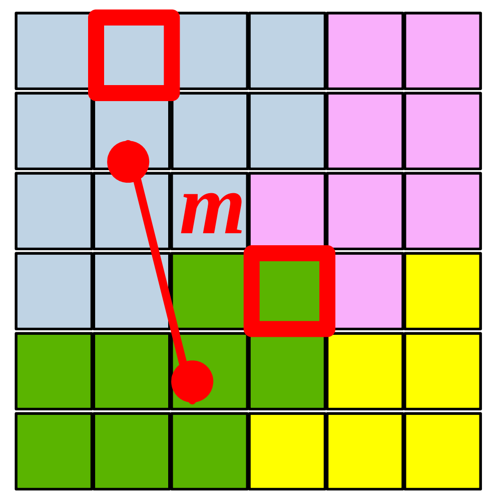
- Gaussian Edge Weights [Krahenbul&Koltun'NIPS2011]

$$w_{pq} = \mathbf{exp}\left(-\left|\ \square - \square\ \right|\right)\ \mathbf{exp}\left(-\,d\,\right)$$



- Quantized edge weights

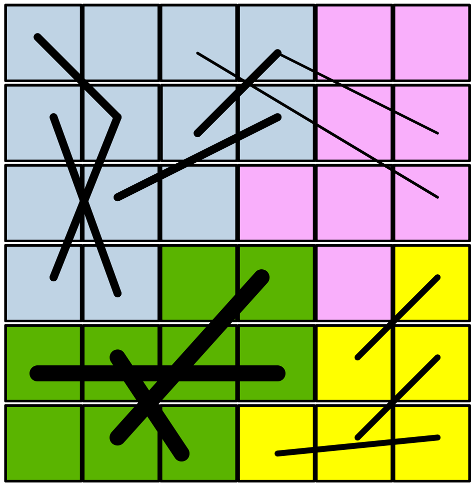
$$w_{pq} = \mathbf{exp}\left(-\left|\ \square - \square\ \right|\right)\ \mathbf{exp}\left(-\,m\,\right)$$



superpixels

Quantized Edge Fully Connected CRFs

- Edge weights depend on superpixel membership
  - do not have to be Gaussian weighted



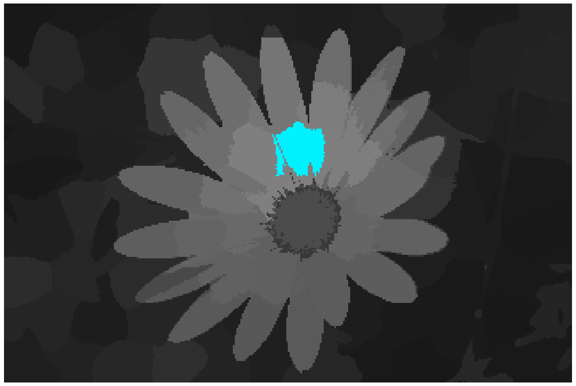
superpixels

Quantized Edge Fully Connected CRFs

input image

superpixels

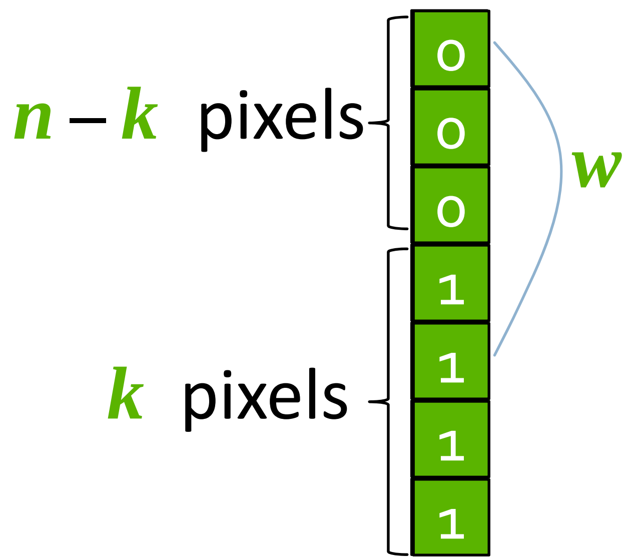
interior weights

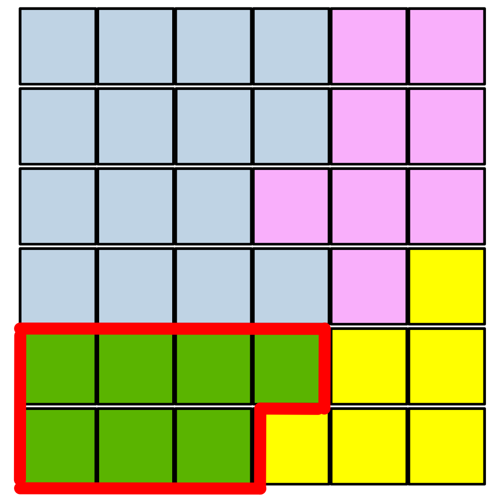


exterior weights

Optimization for 2 labels: Internal Cost

- Consider one superpixel of size n
  - internal edges weight w

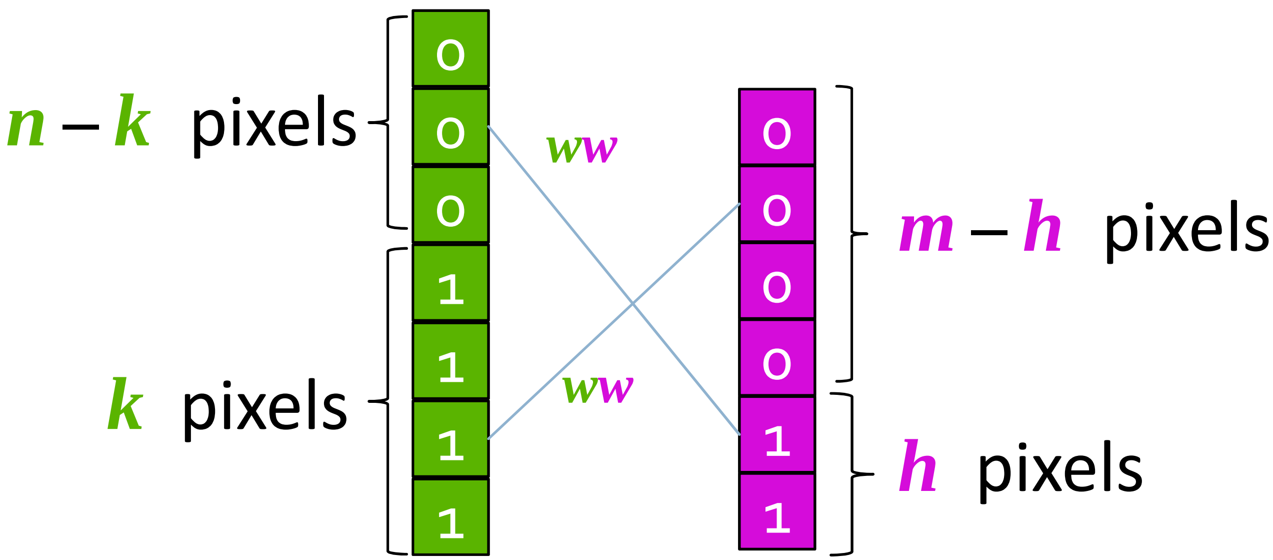


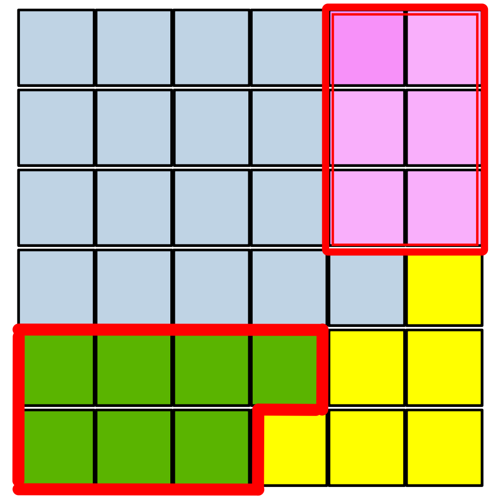


- Internal pairwise cost is $w \cdot k \cdot (n - k)$
  - depends only on k

Optimization for 2 labels: External Cost

- Consider two superpixels of sizes n, m
  - external edge weight ww



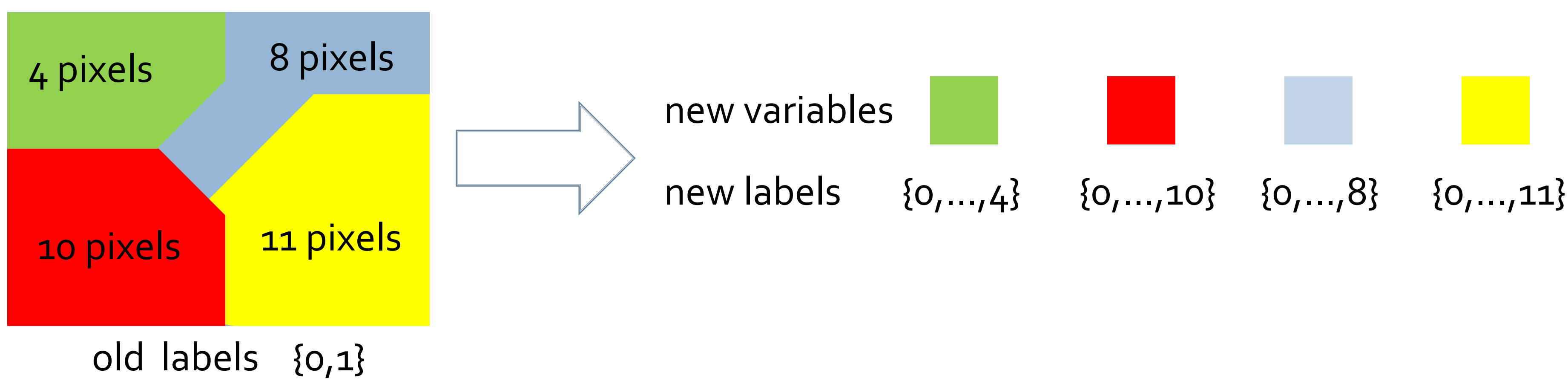


- External pairwise cost is $ww \cdot [k \cdot (m - h) + (n - k) \cdot h]$
  - cost depends on k, h
- **Suppose know that k pixels in a superpixel assigned to label 1**
  - **must be pixels that have the smallest cost for label 1**

Optimization for 2 labels: Overview

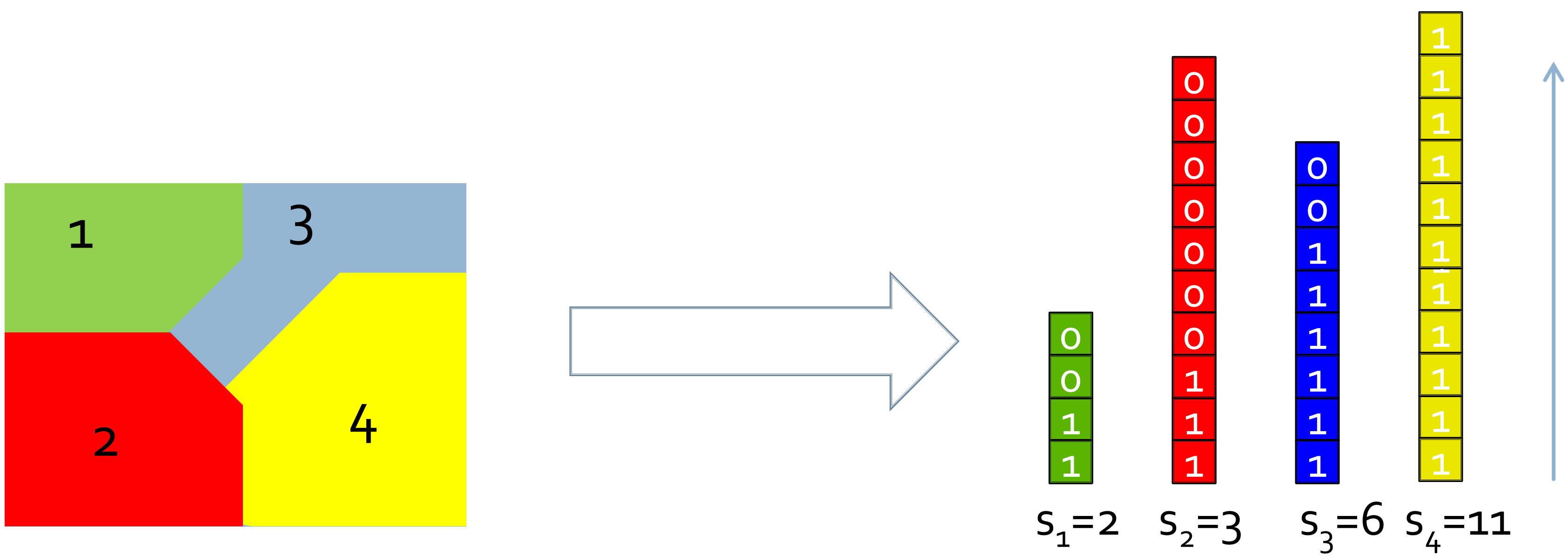
- Binary energy on pixels ➡multi-label energy on superpixels

  [Felzenszwalb&Veksler, CVPR’2010]
  - new variables are superpixels
  - new *cardinality* labels are 0,1,…,superpixelSize
  - assume unary cost for label 0 is 0



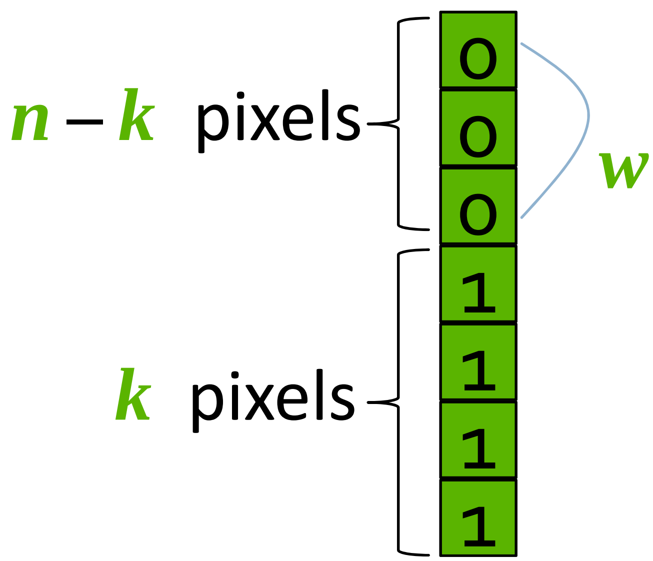
Optimization for 2 labels: Conversion

- Sort pixels in each superpixel by increasing cost of label 1
- New variables are the superpixels
- New labels are 0,1,…,superpixelSize
- Meaning of label k for superpixel
  - k smallest cost pixels of the superpixel are assigned to label 1 in original problem



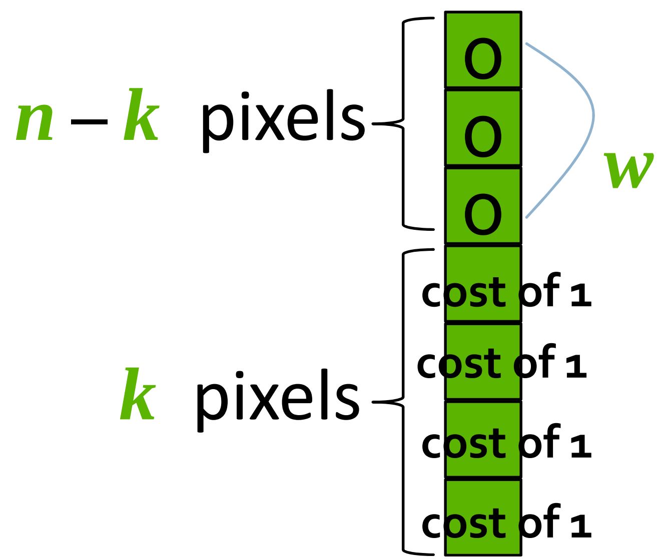
Unary Cost for Transformed Problem

- Unary cost for green superpixel to have label k
  - account for unary terms of the original binary problem



Unary Cost for Transformed Problem

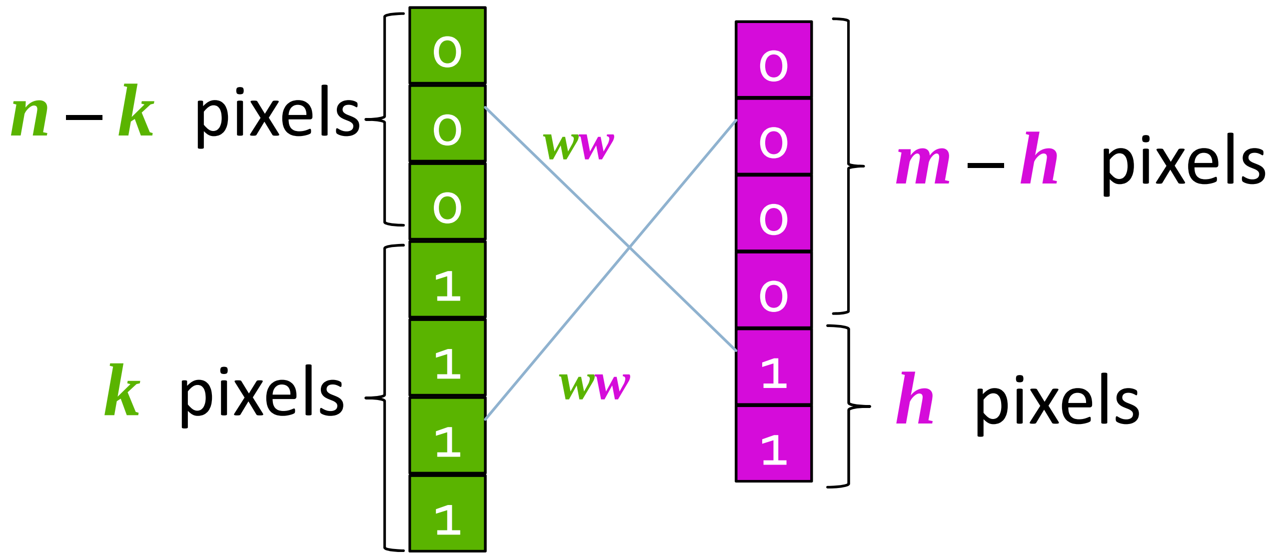
- Unary cost for label k
  - unary terms of the original binary problem
  - internal pairwise terms of original binary problem



$+ \, w \cdot k \cdot (n - k)$

Pairwise Cost for Transformed Problem

- Pairwise cost for labels k, h
  - external pairwise terms of original problem



$$ww \cdot [\,k\cdot(m-h) + (n-k)\cdot h\,]$$

Optimization for Transformed Problem

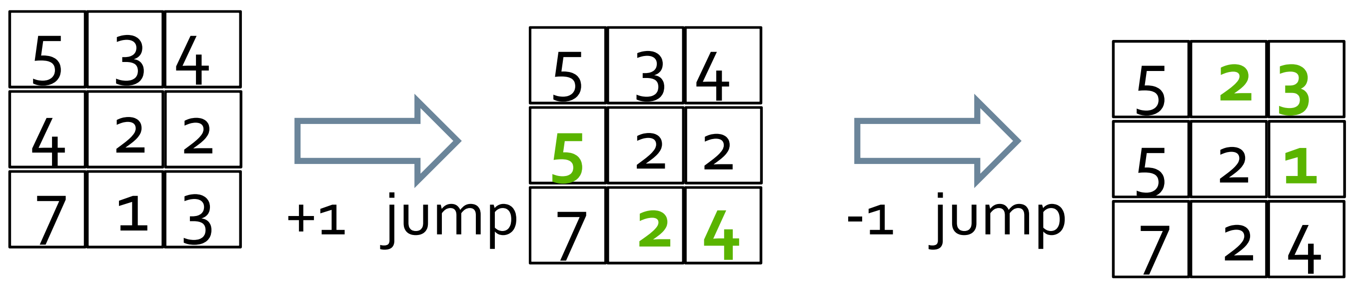
- Pairwise cost for labels k, h

$$ww \cdot [k\cdot(m - h) + (n - k)\cdot h]$$

- Rewrite $\quad$ $unary\ terms + (h - k)^2$
- Optimize exactly with [Ishikawa'TPAMI04]
  - number of edges is quadratic in the number of labels
  - memory inefficient, time complexity almost as bad as the original binary problem
- Or with [Ajanthan'CVPR2016]
  - memory efficient,
  - time complexity almost as bad as the original binary problem

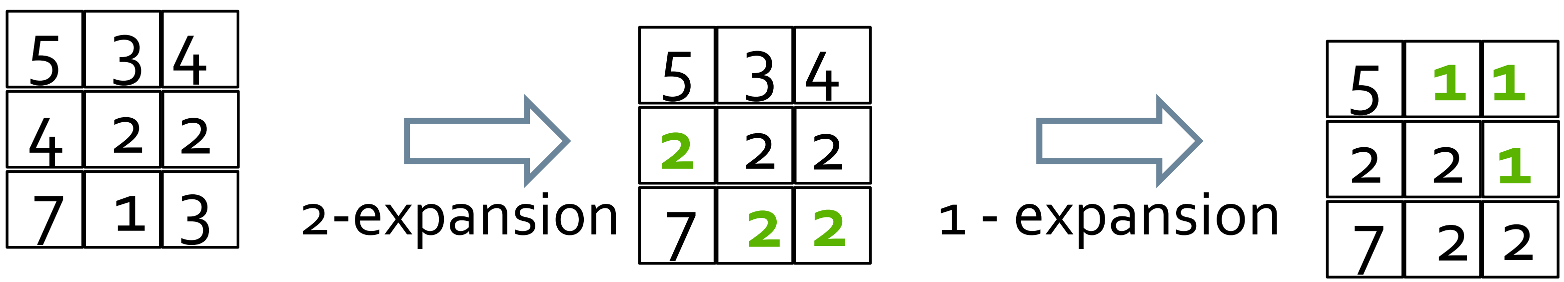
Optimization: Jump Moves

- Pairwise cost is quadratic $(h - k)^2$



- Jump moves [Veksler'99, Kolmogorov &Shioura'09]
  - each move is optimization of binary energy
  - efficient: number of edges is linear in the number of pixels
  - give exact minimum efficiently if unary terms are also convex
- Our unary terms are not convex
  - jump moves do not work well in practice

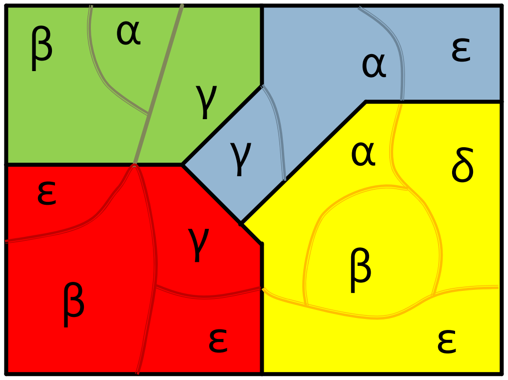
Optmization: *E*xpansion Moves



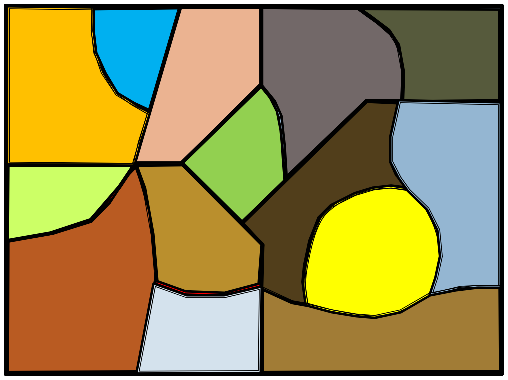
- Expansion moves [Boykov et.al., PAMI'2001]
  - each expansion move is optimization of binary energy
  - efficient: number of edges is linear in the number of pixels
  - Not submodular for quadratic potential
    - but does find the optimum in the overwhelming majority of cases

Multi-Label Quantized Full-CRF

- Apply expansion algorithm
  - each expansion step is optimization of binary energy
  - already know how to optimize 2-label Edge Quantized Full CRF
  - problem
    - meaning of label 0 is not fixed for expansion algorithm
  - solution
    - construct new superpixels according to the current labeling



old superpixels



new superpixels

Final Algorithm, Multi-Label Case

for each $\alpha \in \mathrm{L}$

perform α-expansion

1. compute new superpixels
2. transform binary expansion energy from pixel domain to multi-label energy in superpixel domain
3. **for** each $\beta \in \mathrm{L}^{\text{transformed}}$

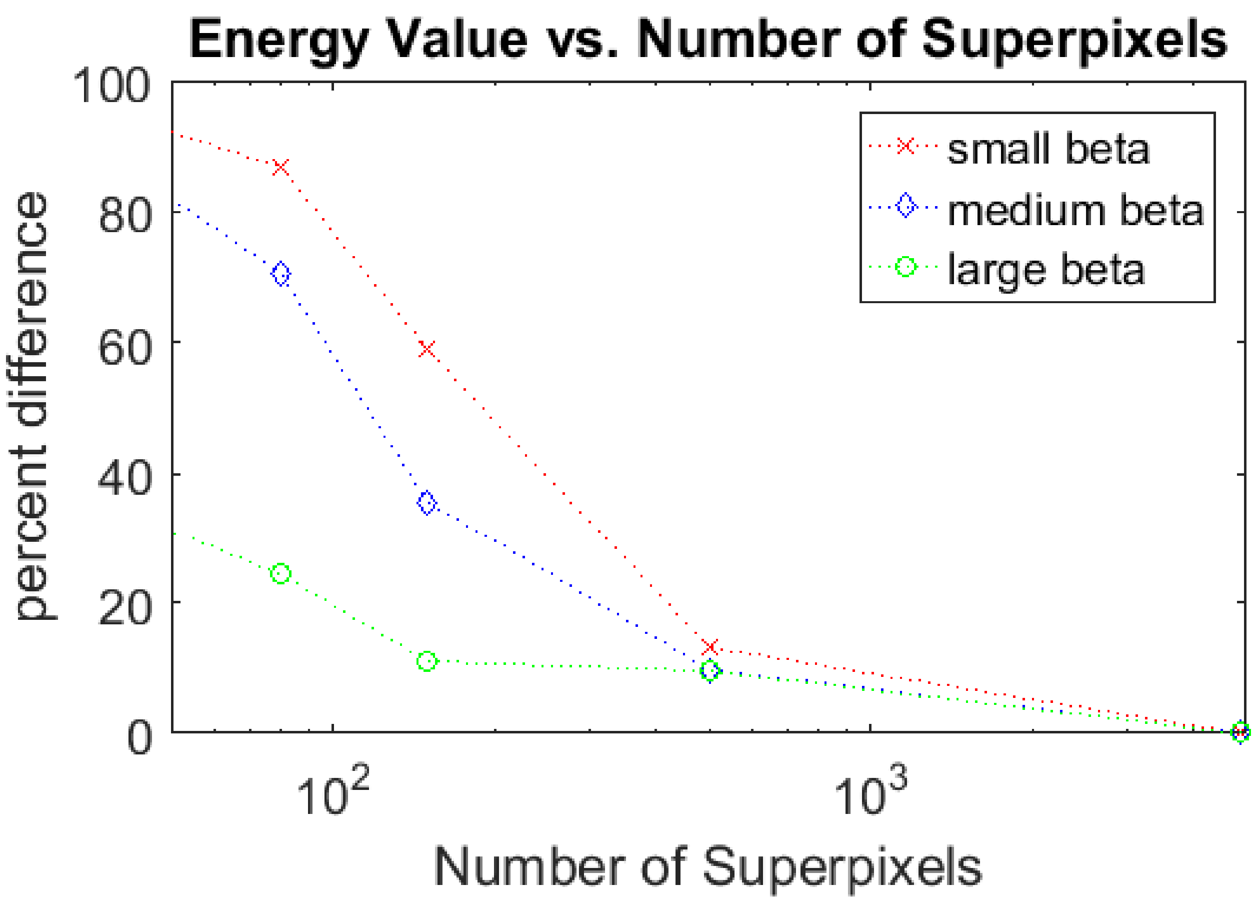
   perform β -expansion

   until convergence

until convergence

Connection to Gaussian Full-CRF

- Quantized edge CRF gets close to Gaussian edge CRF
  - as number of superpixels increases
  - as beta increases

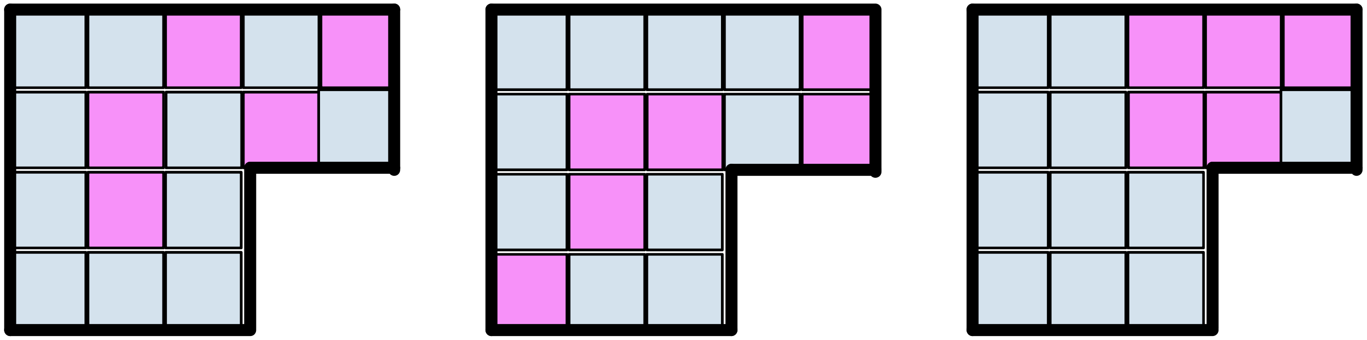


Connection to Gaussian Full-CRF

- Regularization properties of full Gaussian CRF not well understood
  - "all pixels connected", "preserves fine structure"

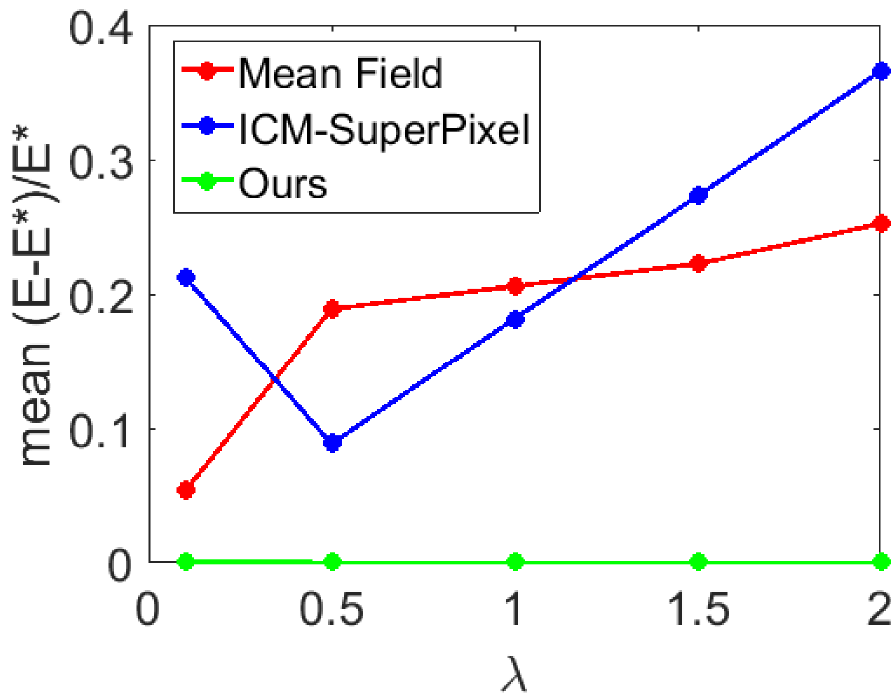
ground truth Gaussian Full-CRF results

- Quantized Edge CRF model helps to understand Gaussian CRF
  - If k pixels in a superpixel split from the rest, shape of the split does not matter
  - equal cost labelings



Optimization Results: Full-CRF, 2 labels

- validation fold of Pascal 2012 dataset
- reduced to 70x70 pixels
- 2 most likely labels
- global optimum with a graph cut
- our method is exact in 89% of cases

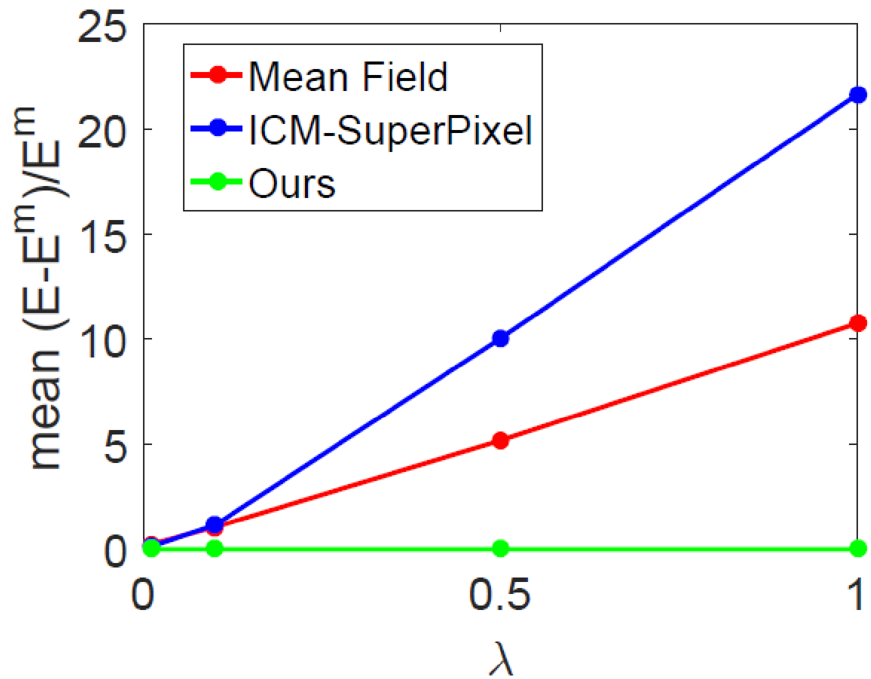


- running time in seconds

Mean Field	Superpixel ICM	Ours	Exact
0.012	0.014	0.31	7.1

Optimization Results: Full-CRF, multilabel

- validation fold of Pascal 2012 dataset
- 21 labels
- our method is always better than mean-field, ICM



- running time in seconds

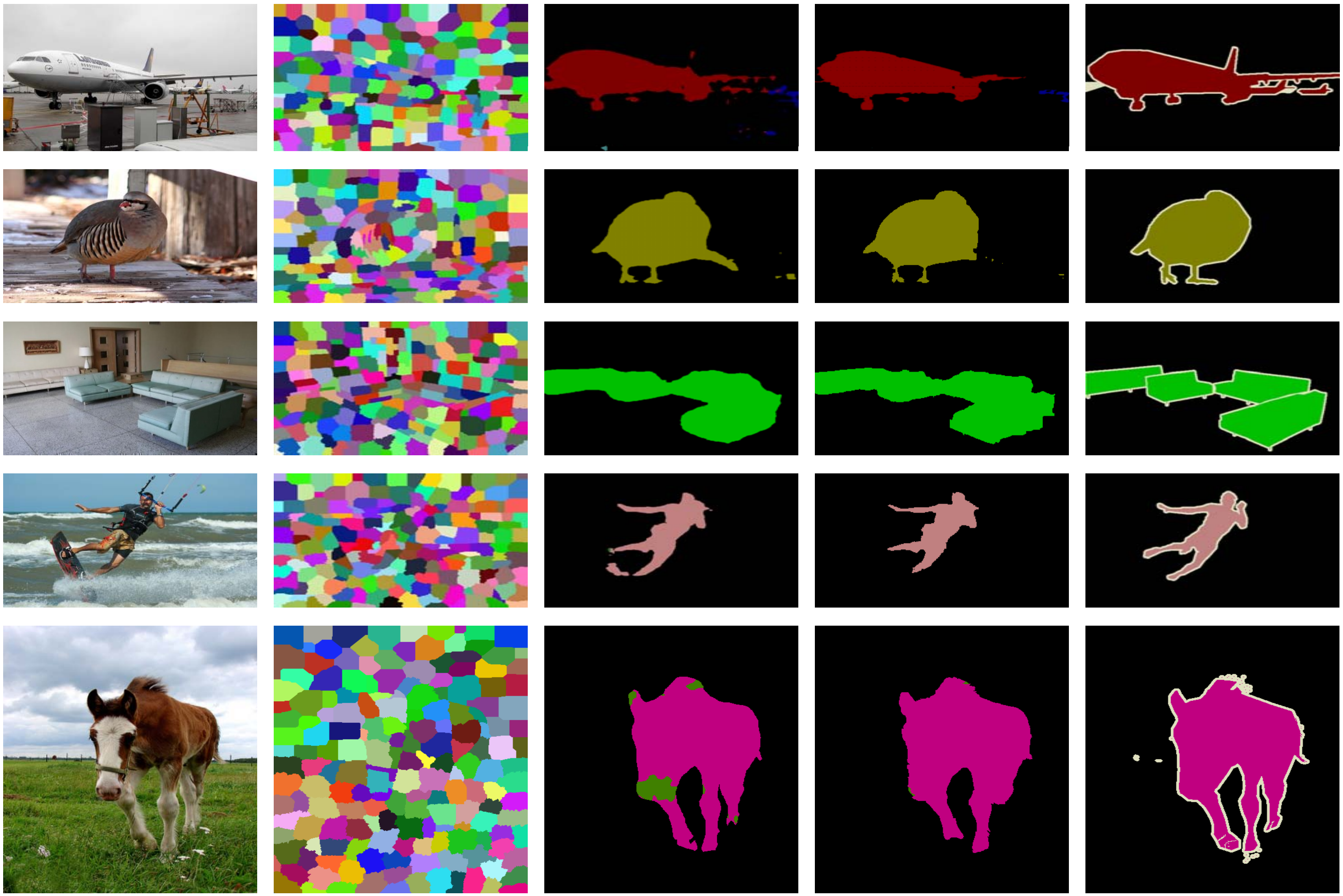
Mean Field	Superpixel ICM	Our Method
2.16	0.11	15.73

Full CRFs :Semantic Segmentaiton

- Test fold of Pascal 2012 dataset
- 21 labels
- Overall IOU
  - Unary 67.143
  - Superpixels 65.89
  - Mean Field 67.3
  - Ours 67.75

object class	Superpixels	Unary	Ours
Overall	65.8899	67.143	67.7484
background	91.996607	92.505	92.6236
aeroplane	81.7341	83.5563	83.7498
bicycle	41.1970	51.1836	51.2267
bird	81.2498	81.8296	83.2405
boat	58.5404	60.2947	60.1668
bottle	58.4436	59.62620	59.6262
bus	79.8713	80.30270	81.0952
car	73.8574	75.22980	76.0474
cat	78.1484	78.23960	79.4247
chair	26.9773	27.49680	27.2861
cow	65.7162	66.69770	67.5622
diningtable	55.9211	56.62960	56.6296
dog	68.5041	69.3166	69.9815
horse	66.6537	66.9853	67.7631
motorbike	80.2764	81.5684	82.4261
person	77.1641	77.9252	78.5284
pottedplant	49.1919	49.65990	50.5761
sheep	69.5786	71.6253	71.9729
sofa	42.1142	42.3743	43.0364
train	70.8761	73.0517	73.4008
tvmonitor	65.6757	65.5707	66.3532

Full CRFs :Semantic Segmentaiton



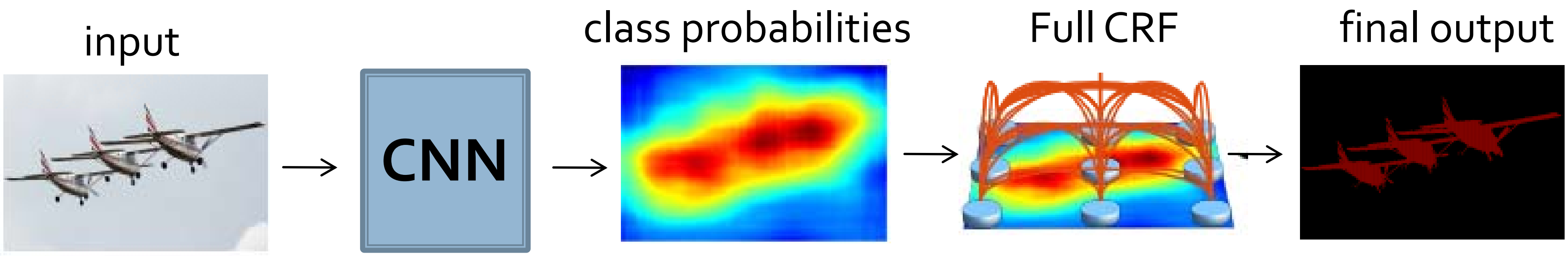
(a) Input image (b) superpixels (c) unary terms (d) our result (e) ground truth

Summary of Part 1

- Quantized Edge Full CRF model
  - Approximation to Gaussian Edge CRF
  - Helps to understand properties of Gaussian Edge CRF
- Efficient optimization of Quantized Edge full CRF with graph cuts
  - Transform the original problem to a smaller domain
  - Optimization quality significantly better than mean field inference

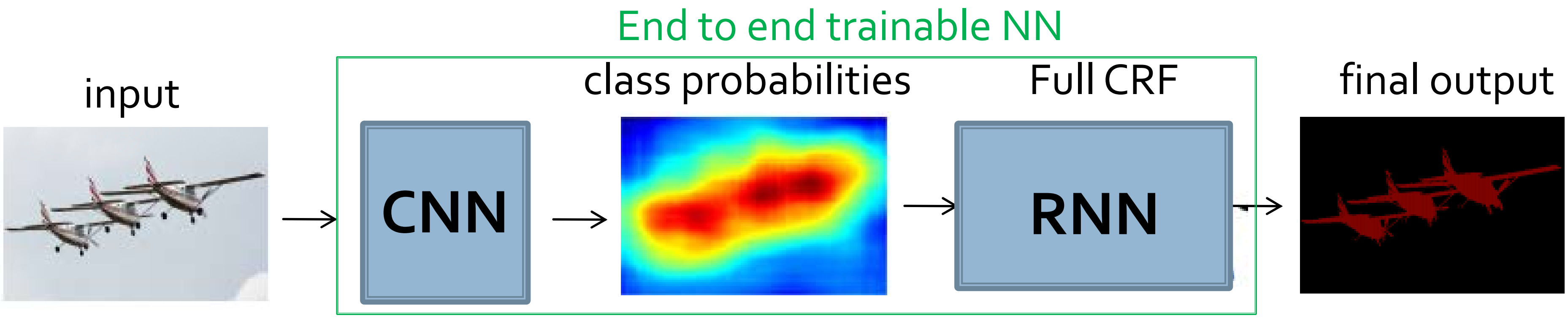
Part 2: CNN for Simulating CRF

CNN+CRF in End-to-End System



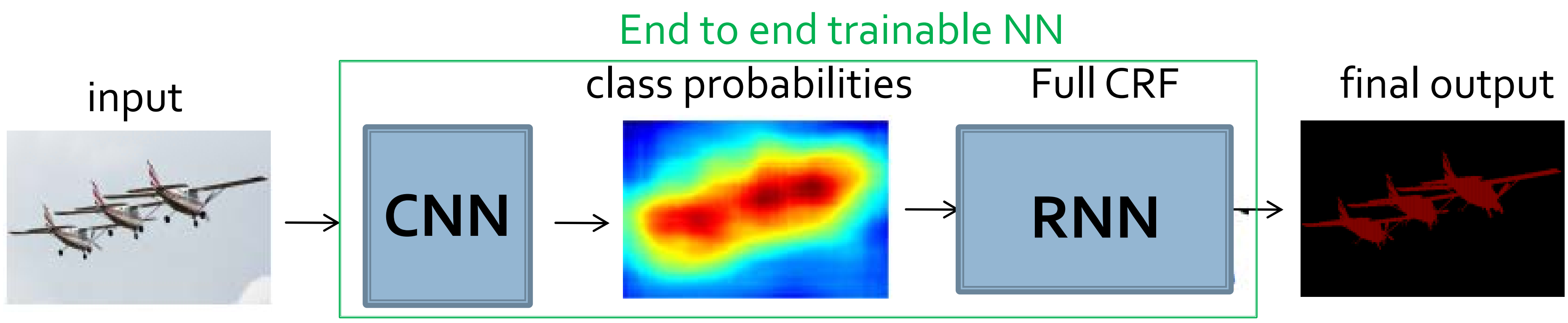
- End-to-end trainable system
  - [Zheng et.al., ICCV'2015], etc.
- Implement mean field inference as RNN

CNN+CRF in End-to-End System



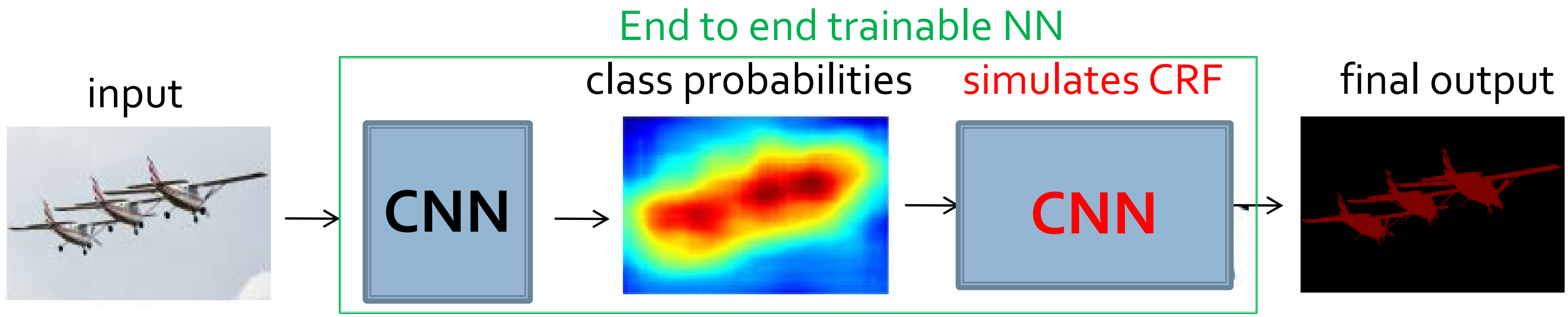
- End-to-end trainable system
  - [Zheng et.al., ICCV'2015], etc.
- Implement mean field inference as RNN
- Advantages
  - end-to-end training
- Disadvantages
  - architecture specific to concrete CRF
  - and mean field annealing

CRF simulator



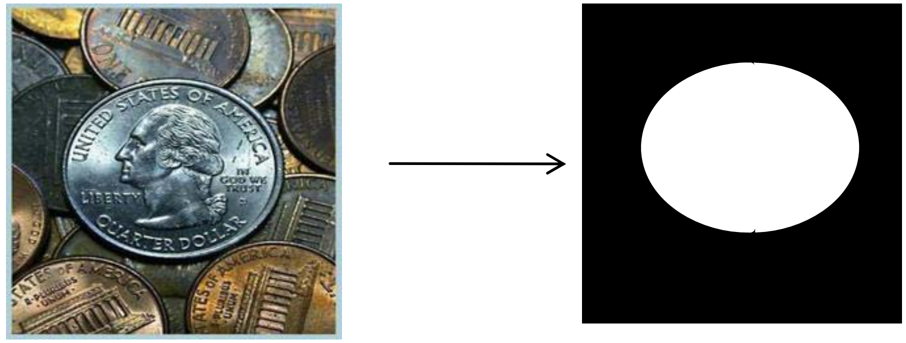
- Replace with CRF simulator

CRF simulator



- Replace with CRF simulator
  - use standard CNN architecture
  - train separately on large dataset
    - access to good CRF optimizer to create training dataset
    - training dataset of unlimited size
  - can simulate any desired CRF

CRF simulator: Salient Object Segmentation



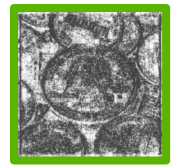
- CRF is binary
- Efficient and exact optimization with a graph cut

Dataset for CRF Simulator

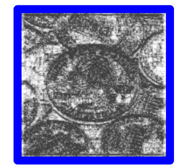
- Binary CRF energy

$$E(x) = \sum_{p} D_p(x_p) + \sum_{(p,q)\in N} w_{pq}[x_p \neq x_q]$$

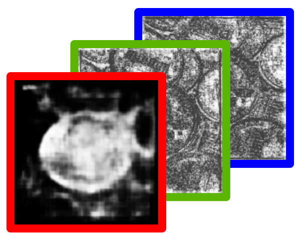
vertical w_{pq}



horizontal w_{pq}



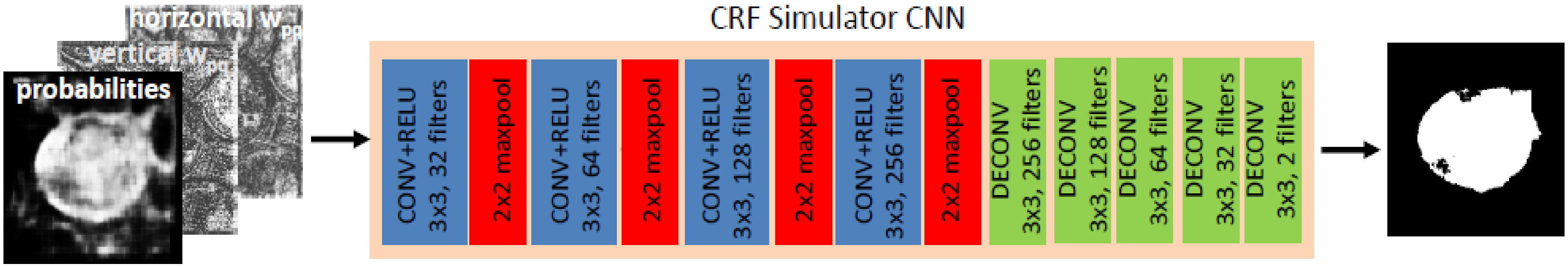
- Training Example



- Optimal solution gives ground truth

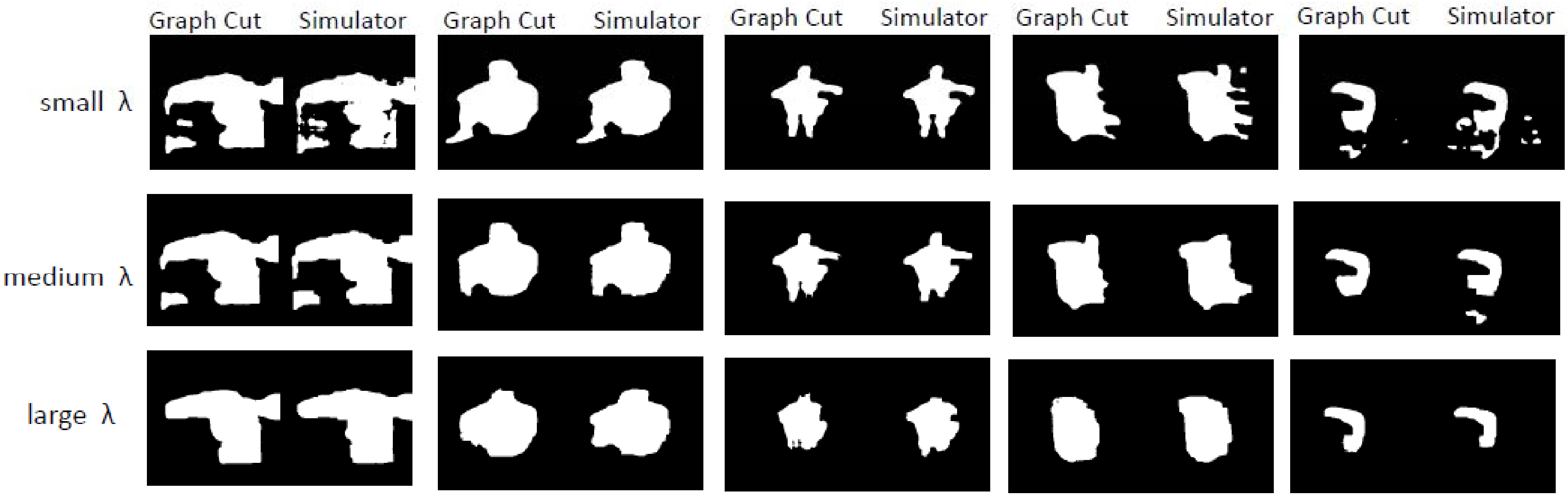
- Used over 100,000 examples for training
  - data terms from saliency and other problems

Architecture for CRF Simulator



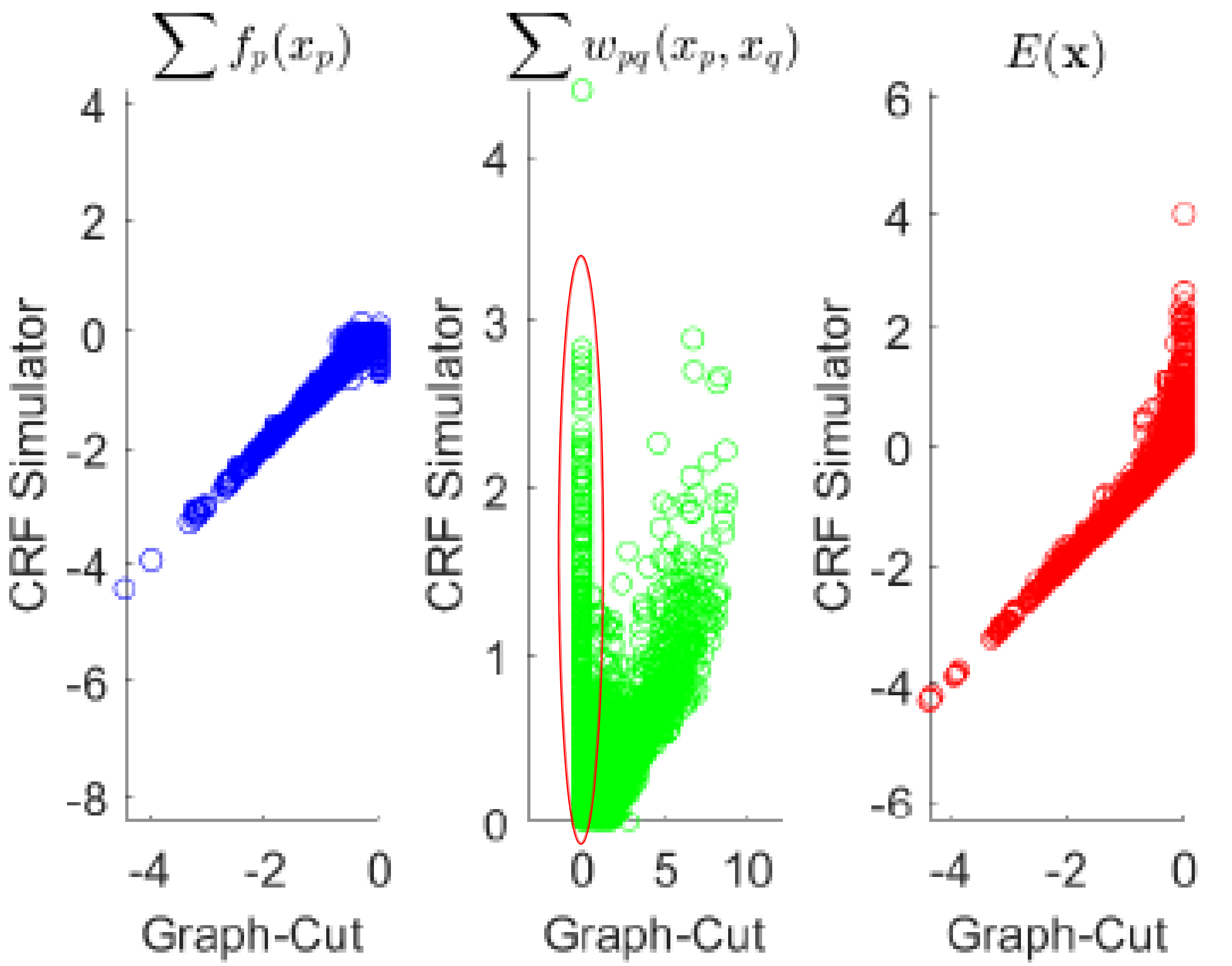
- Standard Encoder/Decoder architecture
- Pre-trained features do not help

CRF Simulator vs Graph Cut

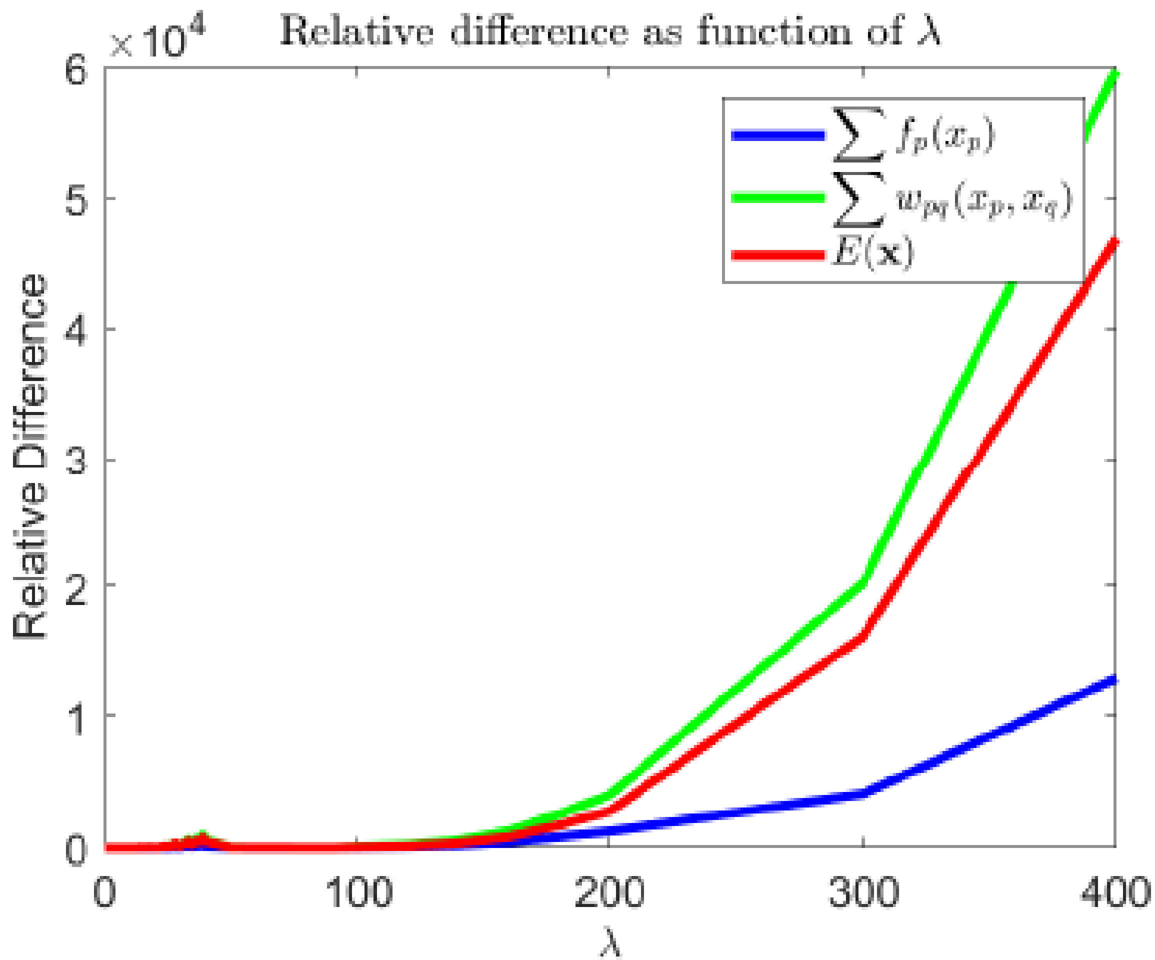


- Captures the ‘spirit’ of regularization
- F-measure is 90.44%

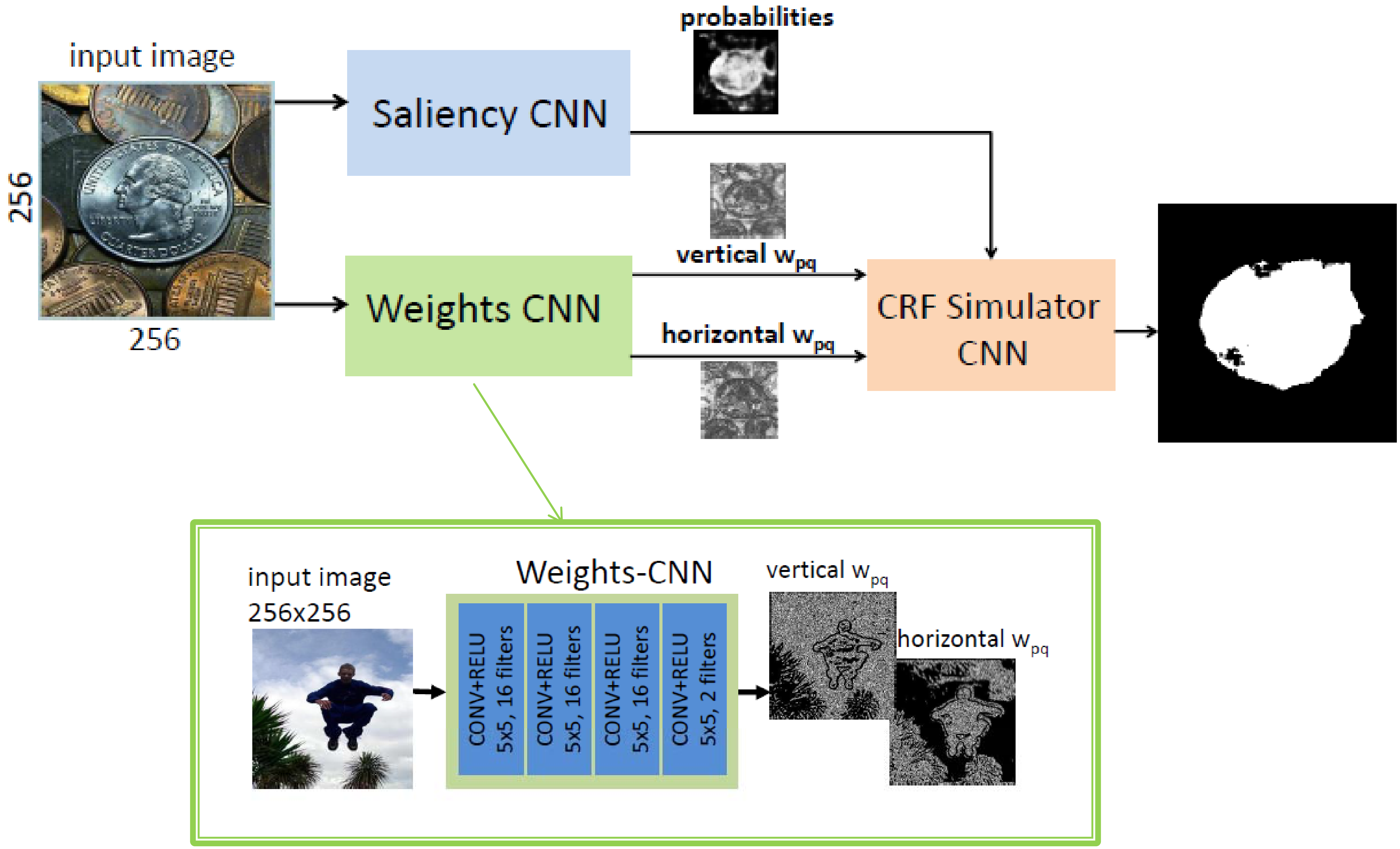
CRF Simulator vs Graph Cut: Energy Values



CRF Simulator vs Graph Cut: Energy Values

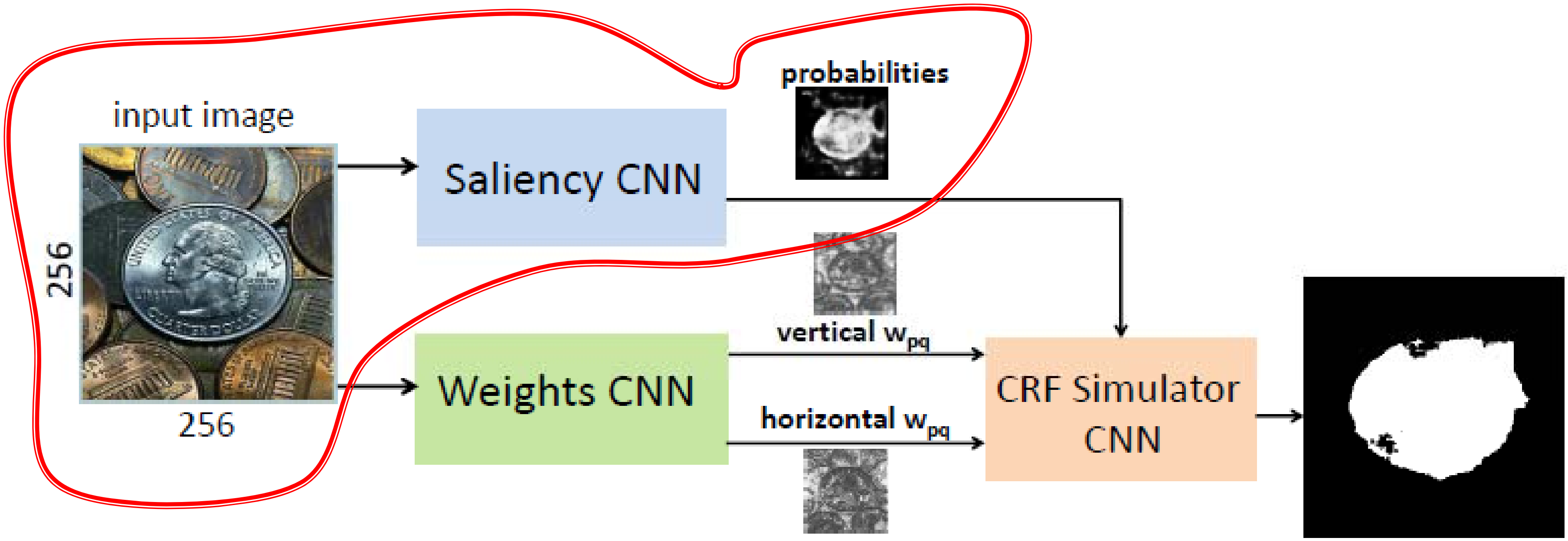


CNN with CRF Simulator: Complete System



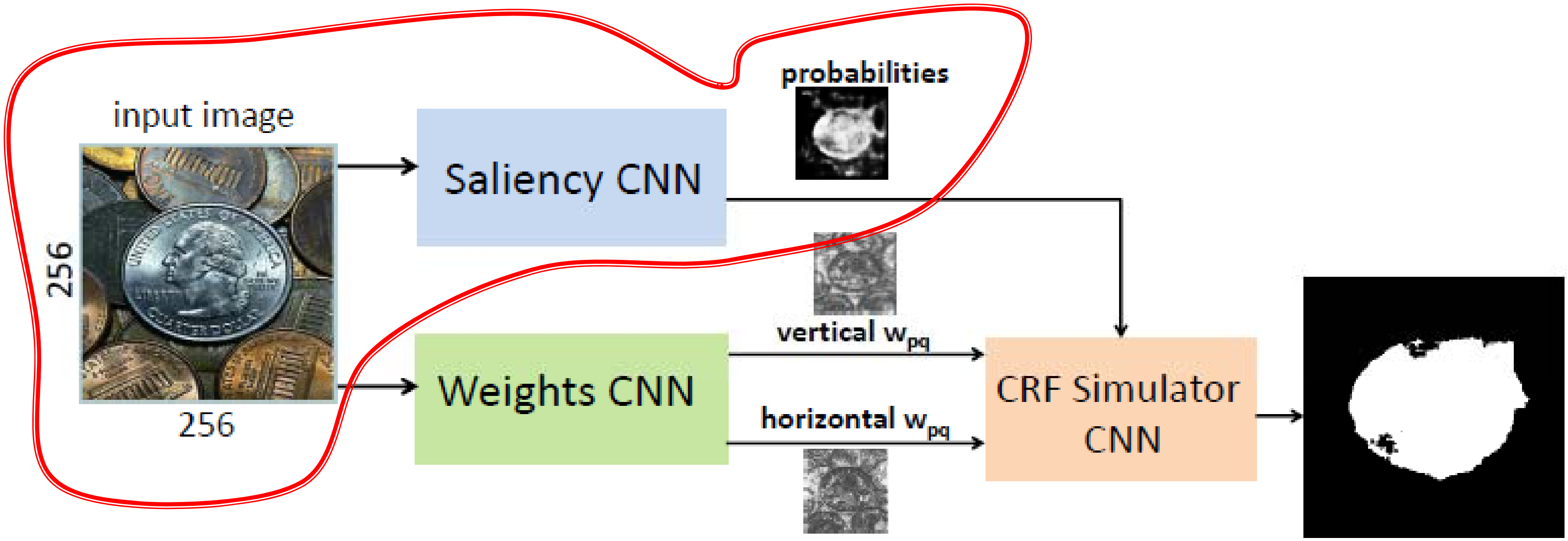
Complete System Results

Saliency CNN	+CRF optimizer	Complete TF	Complete TT	Complete FT	Complete FF	Complete Random
88.22						



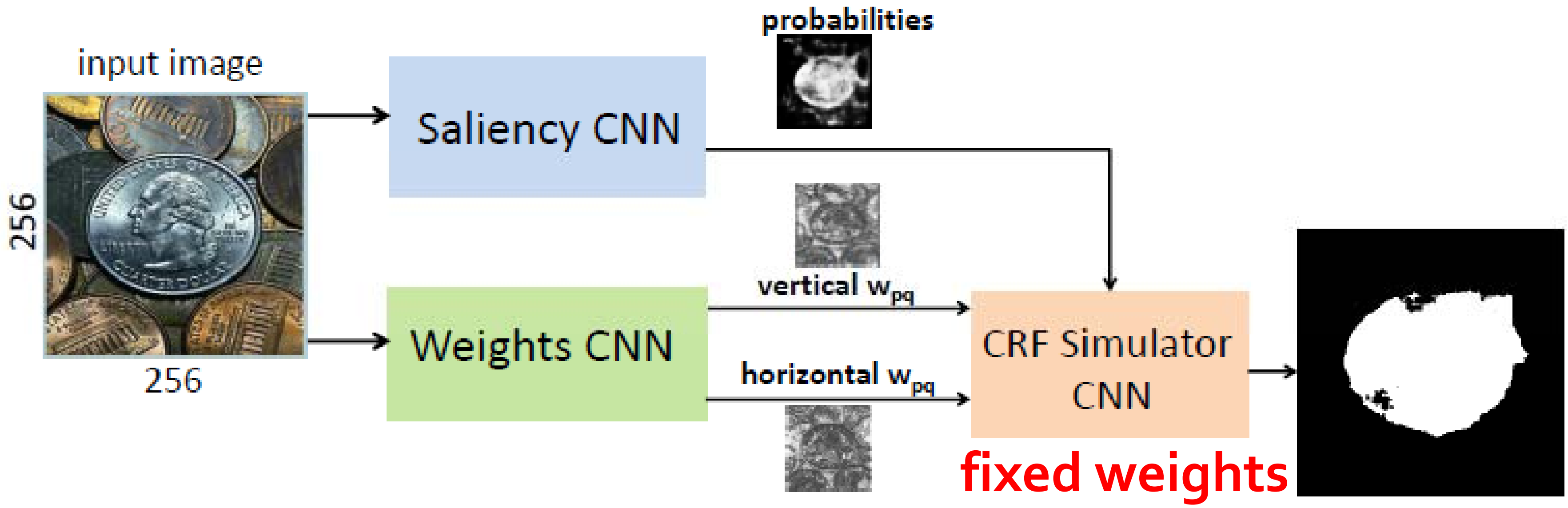
Complete System Results

Saliency CNN	+CRF optimizer	Complete TF	Complete TT	Complete FT	Complete FF	Complete Random
88.22	88.81					



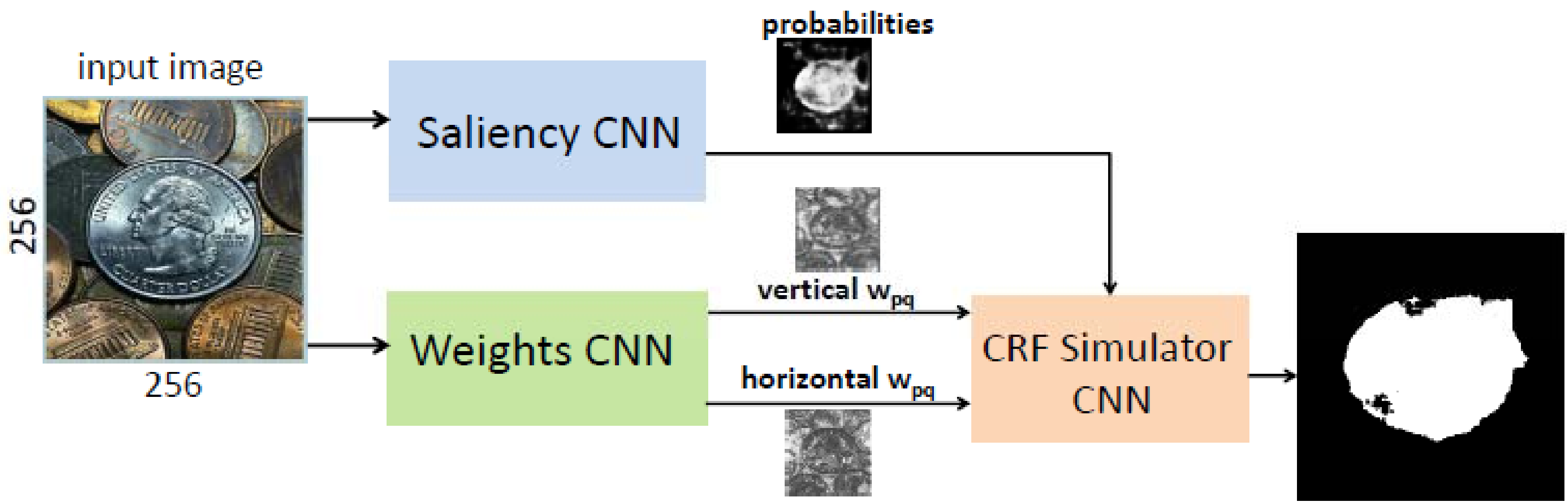
Complete System Results

Saliency CNN	+CRF optimizer	Complete TF	Complete TT	Complete FT	Complete FF	Complete Random
88.22	88.81	89.3				



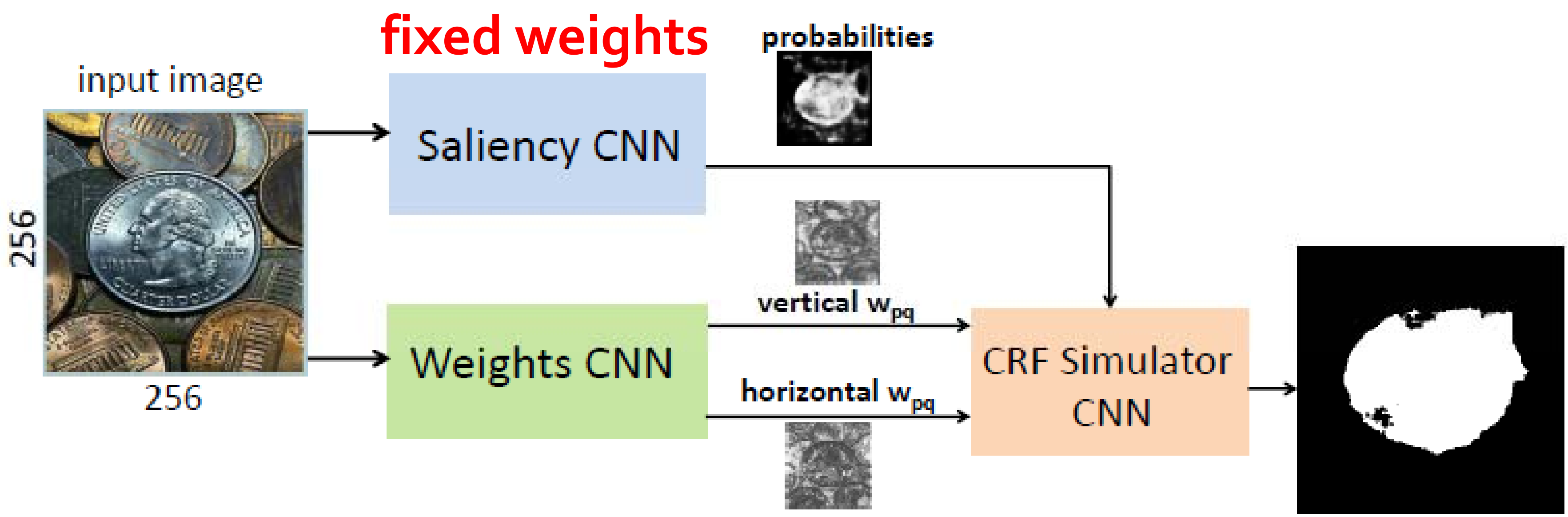
Complete System Results

Saliency CNN	+CRF optimizer	Complete TF	Complete TT	Complete FT	Complete FF	Complete Random
88.22	88.81	89.3	**89.4**			



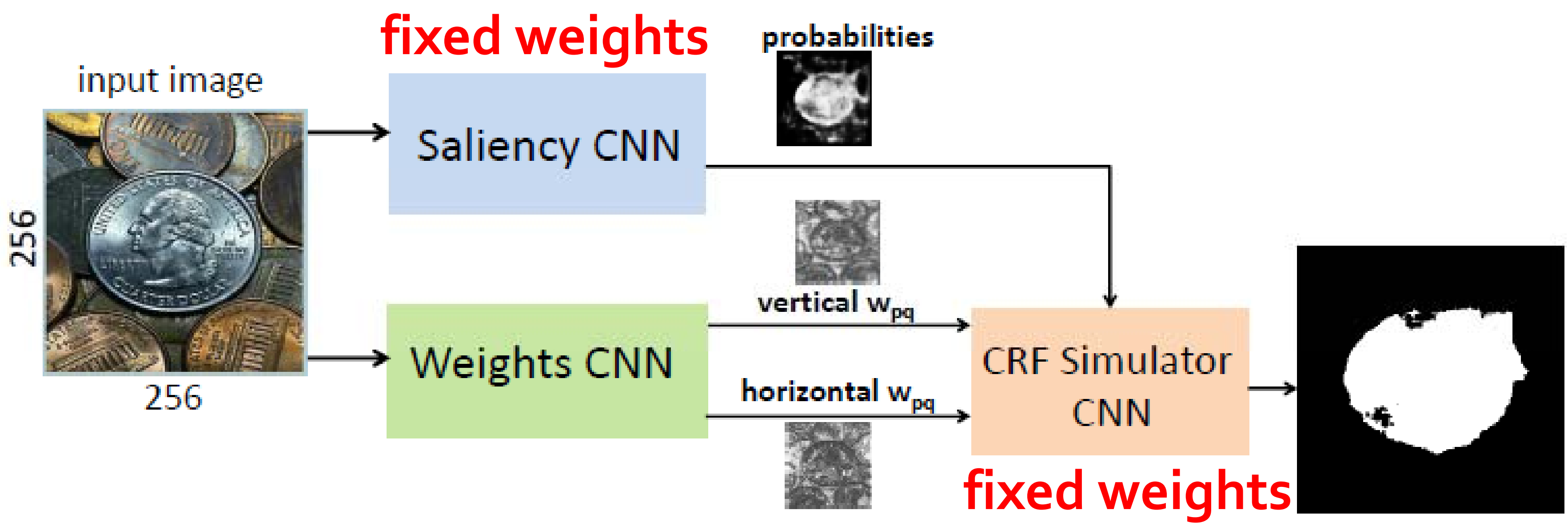
Complete System Results

Saliency CNN	+CRF optimizer	Complete TF	Complete TT	Complete FT	Complete FF	Complete Random
88.22	88.81	89.3	**89.4**	89.05		



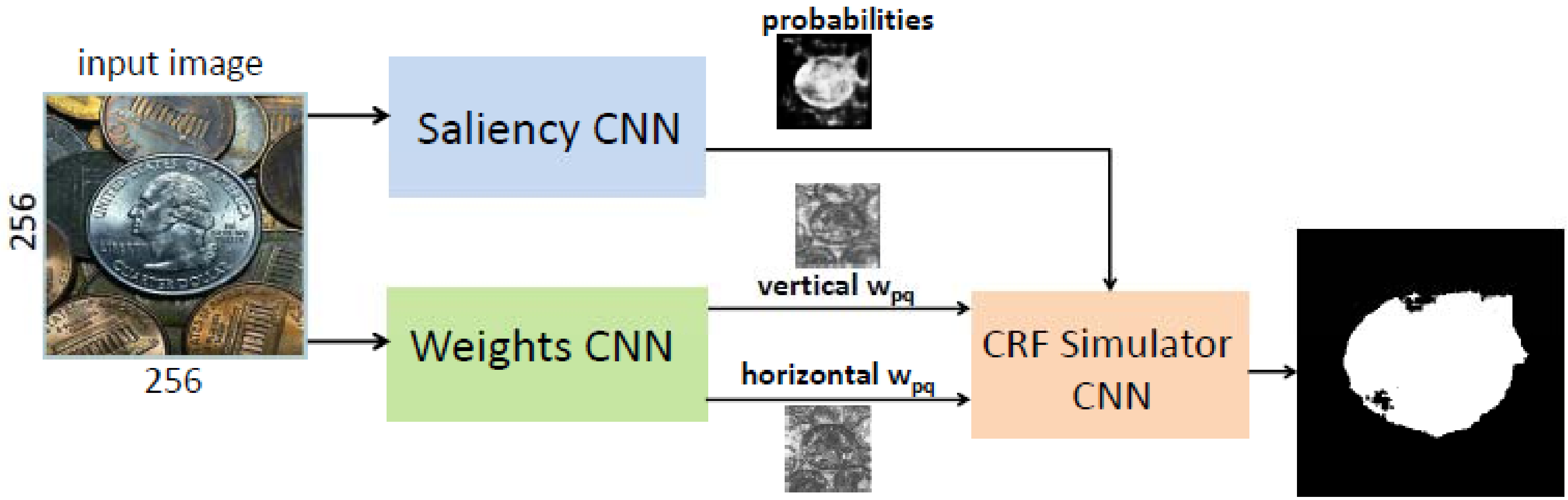
Complete System Results

Saliency CNN	+CRF optimizer	Complete TF	Complete TT	Complete FT	Complete FF	Complete Random
88.22	88.81	89.3	**89.4**	89.05	88.93	



Complete System Results

Saliency CNN	+CRF optimizer	Complete TF	Complete TT	Complete FT	Complete FF	Complete Random
88.22	88.81	89.3	**89.4**	89.05	88.93	88.35



Summary of Part 2

- Can simulate CRF regularization with CNN
- Easy to incorporate into any CNN system
  - standard architecture
- Easy to handle any energy
  - provided efficient optimizer is available
    - collect dataset for new energy
    - dataset size is unlimited
    - train simulator