Semi-supervised Learning for Large Scale Image Cosegmentation

Zhengxiang Wang and Rujie Liu

Fujitsu R&D Center Co. Ltd, Beijing, China

International Conference on Computer Vision Sydney 2013

FUJITSU shaping tomorrow with you

Overview

We propose:

- A semi-supervised learning problem for large scale image cosegmentation, which makes use of:
  - the limited **training** segmentation groundtruth
  - the common object shared between **unsegmented** images
- An effective energy minimization method for semi-supervised cosegmentation, where the energy function consists of:
  - the **inter-image** distance
  - the **intra-image** distance
  - the **balance** term
- An efficient algorithm to solve the energy function by decomposition and iterative updating each sub-problem

Energy Function

Goal: predict the superpixel labels (y_i) by solving an energy minimization problem

- The **inter-image** distance measures the Euclidean distance between the foreground histograms ($H \cdot y$) of two images:

$$E_{inter} = \sum_{i=1}^{N_u}\sum_{j=1}^{N_s} \| H_i \cdot y_i - H_j^{tr} \cdot y_j^{tr} \|^2 + \sum_{i=1}^{N_u}\sum_{j=i+1}^{N_u} \| H_i \cdot y_i - H_j \cdot y_j \|^2$$

- The **intra-image** distance tries to assign the same label to visually similar adjacent superpixels inside an unsegmented image (adjacent superpixels assigned with different labels are penalized):

$$E_{intra} = \sum_{i=1}^{N_u}\sum_{j=1,k=1}^{s_i} W_i(j,k) \cdot \delta(j,k)$$

where

$$\delta(j,k) = \begin{cases} 1, & if \ \ y_i(j) \neq y_i(k) \\ 0, & if \ \ y_i(j) = y_i(k) \end{cases} = |y_i(j) - y_i(k)|$$

$$W_i(j,k) = \frac{\alpha(j,k)}{\sum_{l \in N(j)} \alpha(j,l)} \cdot \exp(-\frac{\| h_i(j) - h_i(k) \|^2}{\theta})$$

- The **balance term** prevents all superpixels belonging to the same label, measured by the proportion of foreground and background superpixels:

$$E_{bal} = \sum_{i=1}^{N_u}(P_i^f \log P_i^f + P_i^b \log P_i^b)$$

where

$$P_i^f = \frac{\sum_{j=1}^{N_u} y_i(j)}{s_i} = \frac{y_i^T \cdot e_i}{s_i}$$

The **whole energy function** is the sum of these three terms:

$$E = E_{inter} + \lambda_1 \cdot E_{intra} + \lambda_2 \cdot E_{bal}$$

Binary QP Problem

The energy minimization problem is solved by converting to a **binary quadratic programming (QP) problem**. Each term is converted as:

- Inter-image distance:

$$E_{inter} = \sum_{i=1}^{N_u} y_i^T \cdot M_{ii}^{inter} \cdot y_i + \sum_{i=1}^{N_u}\sum_{j=i+1}^{N_u} y_i^T \cdot M_{ij}^{inter} \cdot y_j + \sum_{i=1}^{N_u} y_i^T \cdot V_i + C$$

- Intra-image distance:

$$E_{intra} = \sum_{i=1}^{N_u} y_i^T \cdot M_i^{intra} \cdot y_i$$

- Balance term:

$$E_{bal} = \sum_{i=1}^{N_u}(2\frac{y_i^T \cdot e_i \cdot e_i^T \cdot y_i}{s_i^2} - 2\frac{y_i^T \cdot e_i}{s_i} - \frac{1}{2})$$

The whole energy function is therefore:

$$E = \sum_{i=1}^{N_u} y_i^T \cdot (M_{ii}^{inter} + \lambda_1 M_i^{intra} + \lambda_2 \frac{e_i \cdot e_i^T}{s_i^2}) \cdot y_i + \sum_{i=1}^{N_u}\sum_{j=i+1}^{N_u} y_i^T \cdot M_{ij}^{inter} \cdot y_j + \sum_{i=1}^{N_u} y_i^T \cdot (V_i - \lambda_2 \frac{e_i}{s_i})$$

By concatenating all superpixel label vectors into a long vector Y, we can get a simple form as:

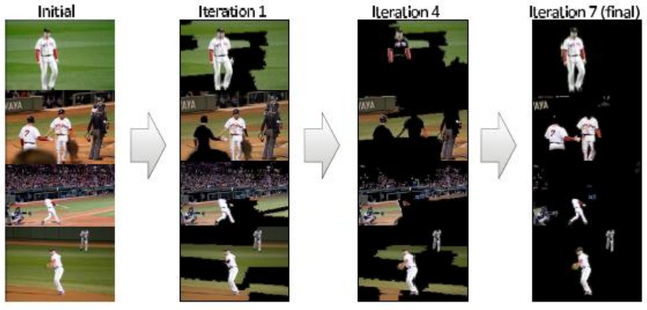
$$\min_Y E = Y^T \cdot M \cdot Y + Y^T \cdot V$$

Iterative Updating Algorithm

- Decompose the whole problem into sub-problems
- Update each image one by one alternatively in each iteration
  - When updating the current image, the superpixel labels of other images are fixed
  - The sub-problem for updating (y_i) is:

$$\min_{y_i} E_i = y_i^T \cdot M_i' \cdot y_i + y_i^T \cdot V_i' + C_i'$$

  - This is also a binary QP problem, but with **much less** binary variables
- The iteration repeats until convergence
- Example:



Experiment

- Unsupervised Cosegmentation

iCoseg

Accuracy

Classes	Ours	Joulin 2010 [13]	Kim 2011 (Best K) [17]	Kim 2011 (K=2) [17]	Joulin 2012 (Best K) [14]	Joulin 2012 (K=2) [14]
Baseball	0.592	0.179	0.621	0.123	0.617	0.197
Football	0.463	0.188	0.446	0.176	0.522	0.396
Panda	0.665	0.472	0.517	0.495	0.457	0.340
Goose	0.718	0.745	0.781	0.772	0.795	0.795
Airplane	0.477	0.577	0.054	0.049	0.500	0.146
Cheetah	0.476	0.358	0.614	0.496	0.668	0.636
Kite	0.539	0.651	0.107	0.093	0.532	0.208
Balloon	0.620	0.484	0.465	0.227	0.599	0.298
Statue	0.688	0.907	0.584	0.579	0.887	0.852
Kendo	0.781	0.802	0.716	0.716	0.871	0.709
Average	**0.602**	0.536	0.491	0.373	0.645	0.458

Running time

Classes	# images	Ours	Joulin 2010 [13]	Kim 2011 (K=2) [17]	Joulin 2012 (K=2) [14]
Baseball	25	6.8	963.8	38.6	998.4
Football	33	6.7	1449.4	47.6	1557.4
Panda	25	5.9	1449.6	42.3	941.9
Goose	31	5.9	1028.2	47.6	1050.1
Airplane	39	12.6	1763.8	31.6	1822.5
Cheetah	33	4.6	1533.9	31.5	1642.1
Kite	18	4.1	583.3	20.3	734.3
Balloon	24	2.6	941.2	23.6	829.4
Statue	41	6.0	1257.6	51.6	2018.3
Kendo	30	11.2	2501.8	47.6	1247.9
Average	29.9	**6.6**	1347.3	38.2	1284.2

Pascal VOC 2012

Accuracy

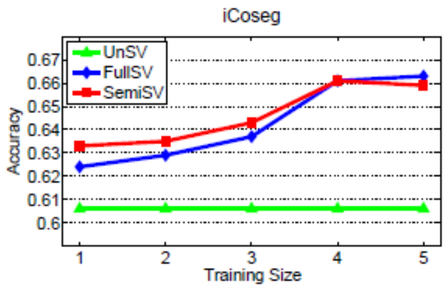
Classes	Ours	Kim 2011 (Best K) [17]	Kim 2011 (K=2) [17]
Aeroplane	0.335	0.166	0.142
Boat	0.231	0.100	0.098
Bus	0.392	0.342	0.335
Diningtable	0.255	0.228	0.228
Dog	0.248	0.145	0.131
Motorbike	0.280	0.222	0.222
Sheep	0.205	0.148	0.146
Train	0.332	0.220	0.200
Average	**0.285**	0.196	0.188

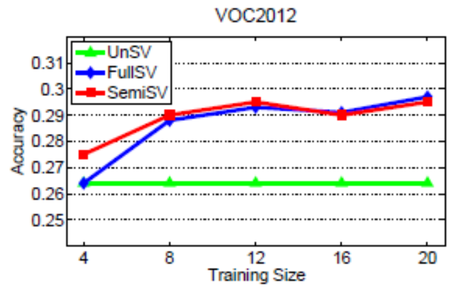
Running time

Classes	# images	Ours	Kim 2011 (K=2) [17]	Kim 2011 (K=9) [17]
Aeroplane	178	25.8	341.4	1807.3
Boat	150	13.3	348.9	1432.5
Bus	152	15.3	439.9	1631.6
Diningtable	157	11.7	467.6	2225.8
Dog	249	51.7	527.0	2165.1
Motorbike	157	19.4	432.6	1869.6
Sheep	120	34.3	249.0	1142.4
Train	167	15.7	480.3	1898.5
Average	166.3	**23.4**	410.8	1771.6

- Competitive cosegmentation accuracy to state-of-the-art
- Fast:
  - ~6s for cosegmenting ~30 images
  - ~23s for cosegmenting 100-200 images
  - ~5m for cosegmenting 1000 images
  - (Running time for superpixel extraction and histogram generation is not included)

- Semi-supervised Cosegmentation





- SemiSV outperforms both FullSV and UnSV in case of fewer training images
- With increased training images, FullSV catches up with SemiSV

Correspondence Email: **Zhengxiang Wang** wangzhengxiang@cn.fujitsu.com