Recurrent Pixel Embedding for Grouping

Shu Kong

CS, ICS, UCI

UCIrvine
University of California, Irvine

Outline

1. Problem Statement -- Pixel Grouping
2. Pixel-Pair Spherical Max-Margin Embedding
3. Recurrent Mean Shift Grouping
4. Experiment
5. Conclusion and Extension

Note: the slides were made before paper submission, please treat them as supplemental material and refer to the paper for updated content.

Pixel Labeling

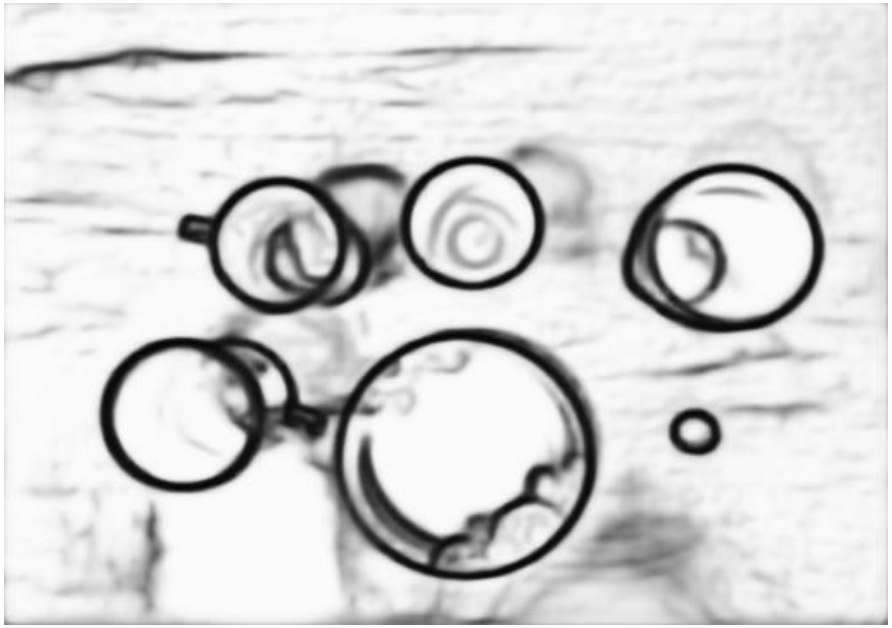
Tasks diving into pixels --

Pixel Labeling: Low-Level Vision

Tasks diving into pixels --

Low-level vision:

edge, boundary, contour



Pixel Labeling: Mid-Level Vision

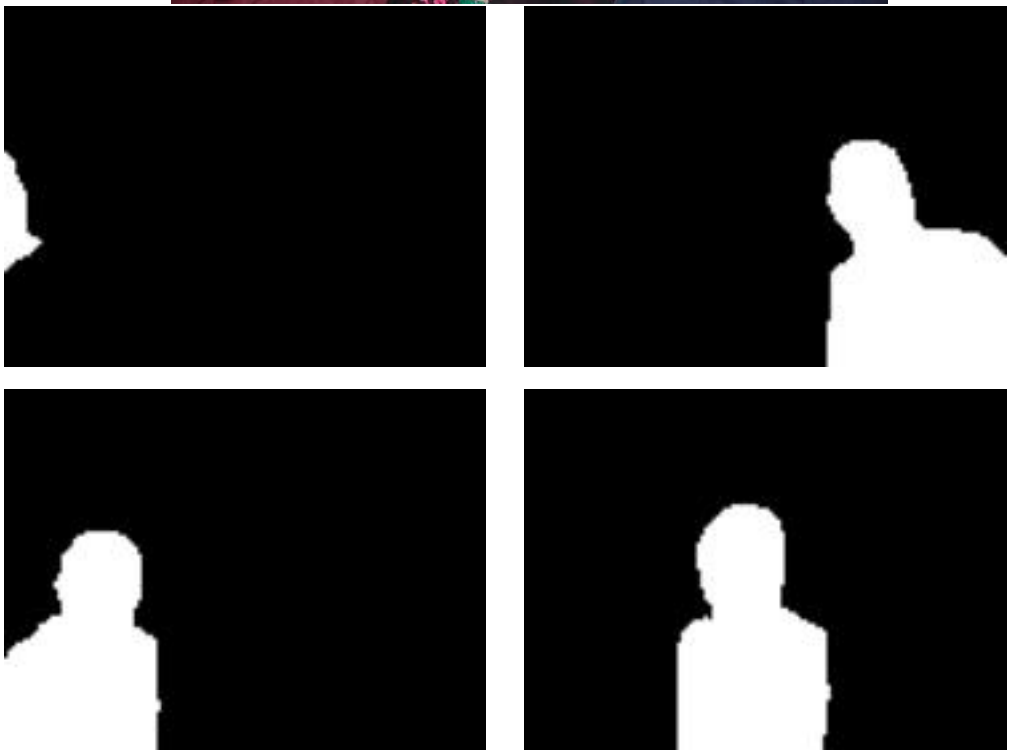
Tasks diving into pixels --

Low-level vision:

edge, boundary, contour

Mid-level vision:

object proposal



Pixel Labeling: High-Level Vision

Tasks diving into pixels --

Low-level vision:

edge, boundary, contour

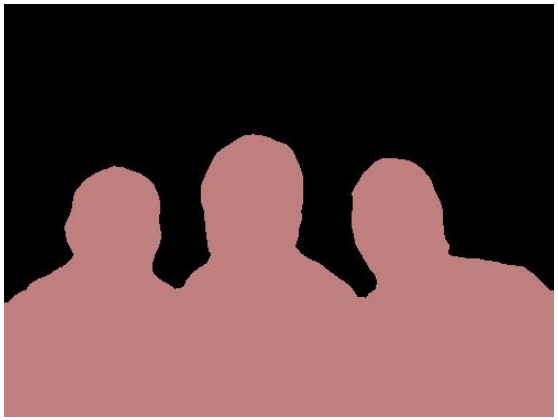
Mid-level vision:

object proposal

High-level vision:

semantic segmentation

instance-level semantic segmentation



Pixel Labeling: Learning

Tasks diving into pixels --

Low-level vision:	logistic loss
edge, boundary, contour	
Mid-level vision:	logistic loss for score
object proposal	regression for location
High-level vision:	
semantic segmentation	logistic loss for mask&score
instance-level semantic segmentation	cross-entropy for category

Pixel Labeling: New Framework

A new framework consisting of two novel modules --

Pixel Labeling: New Framework

This framework is agnostic to architecture, so ignore deep learning for now!

Pixel Labeling: New Framework

A new framework consisting of two novel modules --

1. pixel-pair spherical max-margin regression

2. recurrent mean shift grouping

Pixel Labeling: New Framework

A new framework consisting of two novel modules --

1. pixel-pair spherical max-margin regression

- learning an embedding space on the hyper-sphere such that
  - if meeting the pair-wise **criterion**, learn to push pixels to be close to each other,

    e.g. both are boundaries, from same instance;
  - if not, learn to pull them apart.

2. recurrent mean shift grouping

Pixel Labeling: New Framework

A new framework consisting of two novel modules --

1. pixel-pair spherical max-margin regression

- learning an embedding space on the hyper-sphere such that
  - if meeting the pair-wise **criterion**, learn to push pixels to be close to each other;

    e.g. both are boundaries, from same instance;
  - if not, learn to pull them apart.

2. recurrent mean shift grouping

- iteratively group the pixels into discrete clusters, such as **criteria:**

  boundary vs. non-boundary; object proposals; semantic segments

Pixel-Pair Spherical Max-Margin Regression

Pixel-Pair Spherical Max-Margin Regression

date back to Fisher Linear discriminant analysis (LDA)

Pixel-Pair Spherical Max-Margin Regression

date back to Fisher Linear discriminant analysis (LDA)

To utilize the label information in finding informative projection, maximizing the following objective

$$J(\mathbf{w}) = \frac{\mathbf{w}^T S_B \mathbf{w}}{\mathbf{w}^T S_W \mathbf{w}}$$

where

$$\begin{aligned} S_B &= \sum_c (\boldsymbol{\mu}_c - \bar{\mathbf{x}})(\boldsymbol{\mu}_c - \bar{\mathbf{x}})^T \\ S_W &= \sum_c \sum_{i \in c} (\mathbf{x}_i - \boldsymbol{\mu}_c)(\mathbf{x}_i - \boldsymbol{\mu}_c)^T \end{aligned}$$

Pixel-Pair Spherical Max-Margin Regression

What loss functions can we use at pixel-level?

Pixel-Pair Spherical Max-Margin Regression

What loss functions can we use at pixel-level?

Principle --

1. for positive pairs of pixels (meeting the criterion), minimizing the pair-wise discrepancy/distance;

2. for negative pairs, minimizing the similarity.

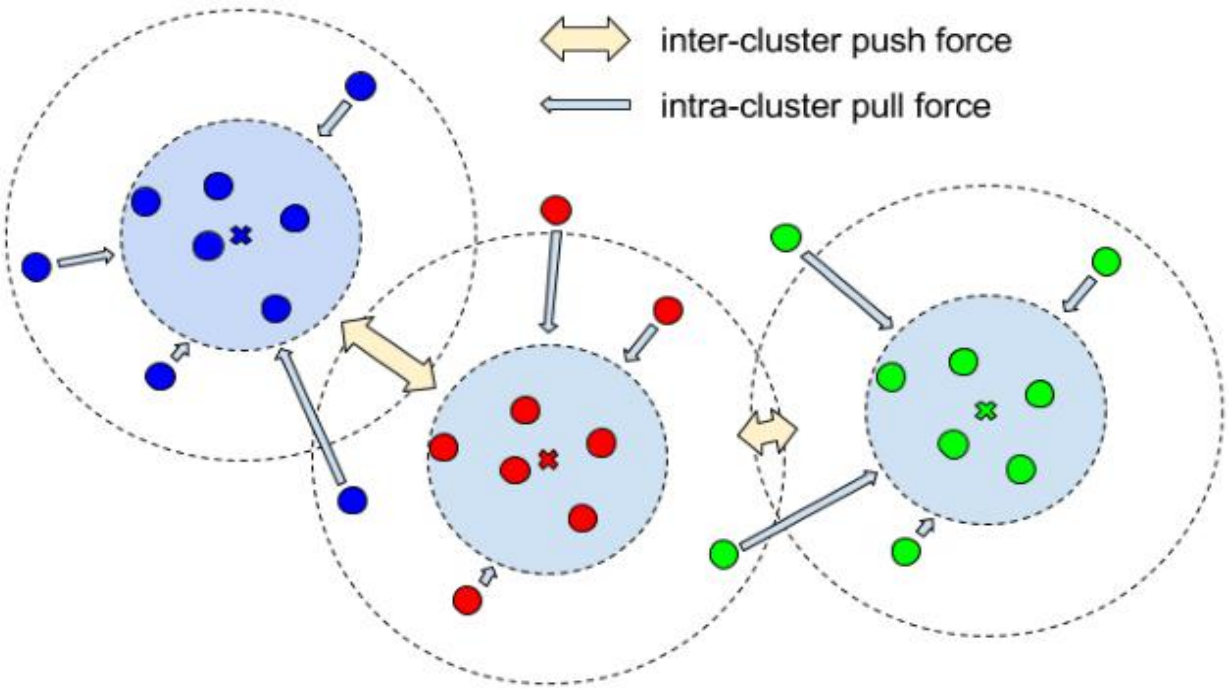
Pixel-Pair Spherical Max-Margin Regression

What loss functions can we use at pixel-level?

Principle --

1. for positive pairs of pixels (meeting the criterion), minimizing the pair-wise discrepancy/distance;

2. for negative pairs, minimizing the similarity.



Bert De Brabandere, Davy Neven, Luc Van GoolSemantic Instance Segmentation with a Discriminative Loss Function, arxiv, 2017

Pixel-Pair Spherical Max-Margin Regression

What loss functions can we use at pixel-level?

Principle --

1. for positive pairs of pixels (meeting the criterion), minimizing the pair-wise discrepancy/distance;

2. for negative pairs, minimizing the similarity.

for example:

Euclidean distance between pixel feature vectors for measuring distance.

Its inverse, or Gaussian transform, can measure the similarity.

.....

Bert De Brabandere, Davy Neven, Luc Van GoolSemantic Instance Segmentation with a Discriminative Loss Function, arxiv, 2017
Alejandro Newell, Jia Deng, Associative Embedding: End-to-End Learning for Joint Detection and Grouping, NIPS, 2017
Alireza Fathi, Zbigniew Wojna, Vivek Rathod, Peng Wang, Hyun Oh Song, Sergio Guadarrama, Kevin P. Murphy, Semantic Instance Segmentation via Deep Metric Learning

Pixel-Pair Spherical Max-Margin Regression

We propose the module to learn a hyper-sphere (embedding space), such that

positive pairs have high cosine similarity;

negative pairs have low cosine similarity.

Pixel-Pair Spherical Max-Margin Regression

Why cosine similarity?

E. B. Saff and A. B. Kuijlaars. Distributing many points on a sphere. The mathematical intelligencer, 19(1):5–11, 1997.
L. Lovisolo ; E.A.B. da Silva, uniform distribution of points on a hyper-sphere with applications to vector bit-plane encoding, IEE Proc. Vision, Image and Signal Processing, 2001

Pixel-Pair Spherical Max-Margin Regression

Why cosine similarity?

1. scale-invariant to the length of feature vector;

E. B. Saff and A. B. Kuijlaars. Distributing many points on a sphere. The mathematical intelligencer, 19(1):5–11, 1997.
L. Lovisolo ; E.A.B. da Silva, uniform distribution of points on a hyper-sphere with applications to vector bit-plane encoding, IEE Proc. Vision, Image and Signal Processing, 2001

Pixel-Pair Spherical Max-Margin Regression

Why cosine similarity?

1. scale-invariant to the length of feature vector;

2. easy to analyze how to set hyper-parameters;

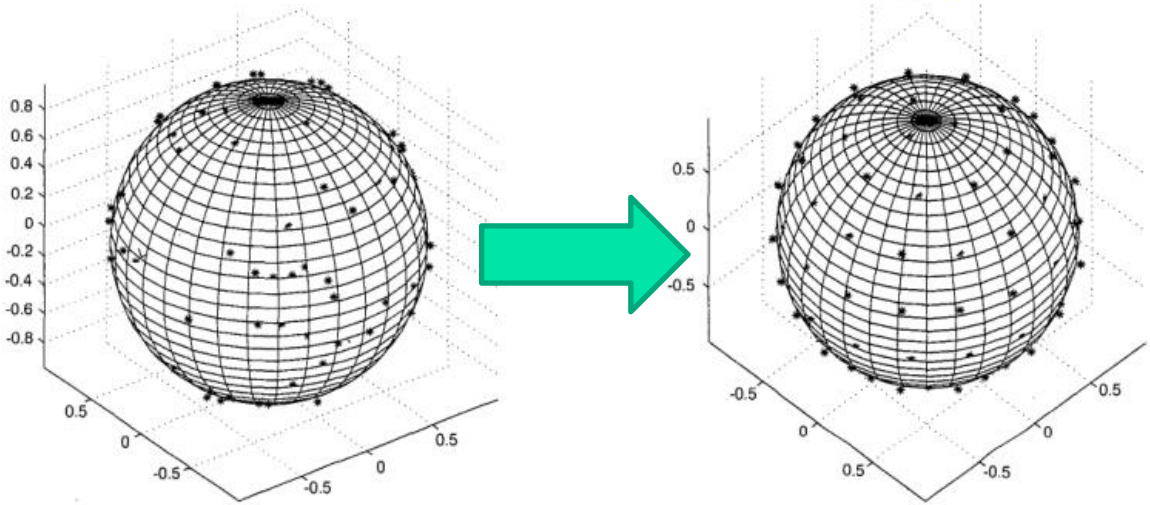
E. B. Saff and A. B. Kuijlaars. Distributing many points on a sphere. The mathematical intelligencer, 19(1):5–11, 1997.
L. Lovisolo ; E.A.B. da Silva, uniform distribution of points on a hyper-sphere with applications to vector bit-plane encoding, IEE Proc. Vision, Image and Signal Processing, 2001

Pixel-Pair Spherical Max-Margin Regression

Why cosine similarity?

1. scale-invariant to the length of feature vector;

2. easy to analyze how to set hyper-parameters;



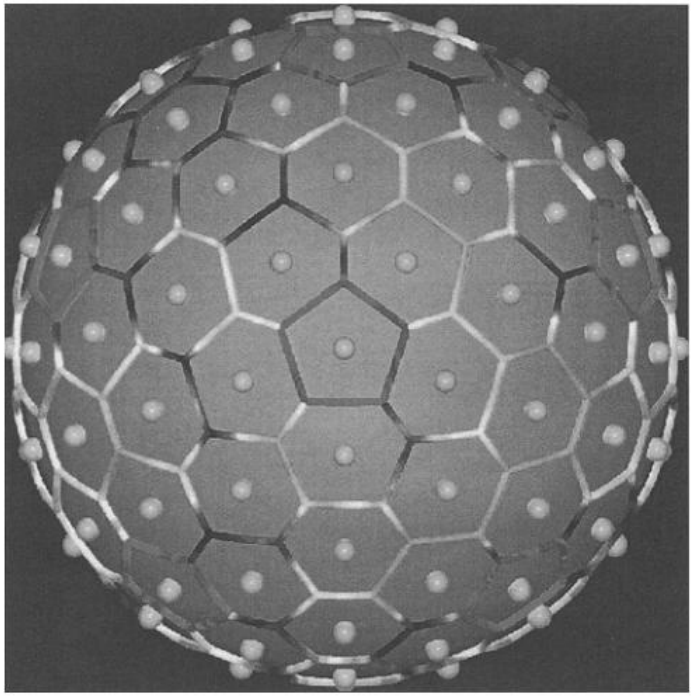


Figure 1. The 122 electrons in equilibrium and their Dirichlet cells.

UCIrvine
University of California, Irvine

E. B. Saff and A. B. Kuijlaars. Distributing many points on a sphere. The mathematical intelligencer, 19(1):5–11, 1997.
L. Lovisolo ; E.A.B. da Silva, uniform distribution of points on a hyper-sphere with applications to vector bit-plane encoding, IEE Proc. Vision, Image and Signal Processing, 2001

Pixel-Pair Spherical Max-Margin Regression

We use the calibrated cosine similarity as below

$$s_{ij} = \frac{1}{2} + \frac{\mathbf{x}_i^T \mathbf{x}_j}{2\|\mathbf{x}_i\|_2 \|\mathbf{x}_j\|_2}$$

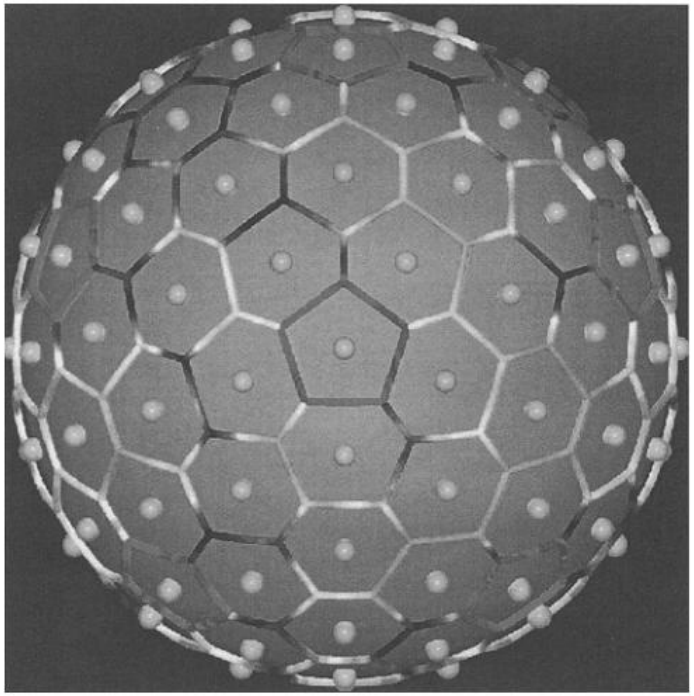


Figure 1. The 122 electrons in equilibrium and their Dirichlet cells.

Pixel-Pair Spherical Max-Margin Regression

We use the calibrated cosine similarity as below

$$s_{ij} = \frac{1}{2} + \frac{\mathbf{x}_i^T \mathbf{x}_j}{2\|\mathbf{x}_i\|_2 \|\mathbf{x}_j\|_2}$$

loss function contains postive and negative pairs

$$l = \frac{1}{|S|} \sum_{i,j \in S} w_{ij} \left(\mathbf{1}_{\{y_i = y_j\}} (1 - s_{ij}) + \mathbf{1}_{\{y_i \neq y_j\}} [s_{ij} - \alpha]_+ \right)$$

Pixel-Pair Spherical Max-Margin Regression

We use the calibrated cosine similarity as below

$$s_{ij} = \frac{1}{2} + \frac{\mathbf{x}_i^T \mathbf{x}_j}{2\|\mathbf{x}_i\|_2 \|\mathbf{x}_j\|_2}$$

loss function contains postive and negative pairs

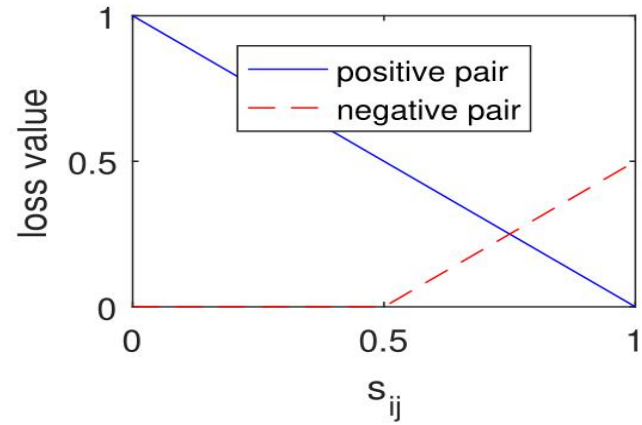
$$l = \frac{1}{|S|} \sum_{i,j \in S} w_{ij} \left(\mathbf{1}_{\{y_i = y_j\}} (1 - s_{ij}) + \mathbf{1}_{\{y_i \neq y_j\}} [s_{ij} - \alpha]_+ \right)$$

alpha is the margin, hyper parameter to be set.

Pixel-Pair Spherical Max-Margin Regression

We use the calibrated cosine similarity as below

$$s_{ij} = \frac{1}{2} + \frac{\mathbf{x}_i^T \mathbf{x}_j}{2\|\mathbf{x}_i\|_2 \|\mathbf{x}_j\|_2}$$



loss function contains postive and negative pairs

$$l = \frac{1}{|S|} \sum_{i,j \in S} w_{ij} \left(\mathbf{1}_{\{y_i = y_j\}} (1 - s_{ij}) + \mathbf{1}_{\{y_i \neq y_j\}} [s_{ij} - \alpha]_+ \right)$$

alpha is the margin, hyper parameter to be set.

Gradient is one, didn't penalize hard pixels in sensitive regions, say nearby boundary, segments, etc.

Pixel-Pair Spherical Max-Margin Regression

Important theories

Theorem 1 *For n vectors $\{\mathbf{x}_1, \ldots, \mathbf{x}_n\}$ on a hyper-sphere, i.e. $\|\mathbf{x}_i\|_2 = 1, \forall i = 1 \ldots n$, $\sum_{i \neq j} \mathbf{x}_i^T \mathbf{x}_j$ has lower bound as $-n$, i.e. $\sum_{i \neq j} \mathbf{x}_i^T \mathbf{x}_j \geq -n$.*

1. the loss has a lower bound, minimum;

2. the lower bound does not depend on the dimension of the embedding space.

Pixel-Pair Spherical Max-Margin Regression

2D case

Theorem 2 *For n vectors $\{\mathbf{x}_1, \ldots, \mathbf{x}_n\}$ on a circle, i.e. $\|\mathbf{x}_i\|_2 = 1, \forall i = 1 \ldots n$, when lower bound is achieved such that $\sum_{i \neq j} \mathbf{x}_i^T \mathbf{x}_j = -n$, then the minimum angle of any two vectors is* $\arccos \frac{2\pi}{n}$.

Pixel-Pair Spherical Max-Margin Regression

3D case

Lemma 4 *For n vectors $\{\mathbf{x}_1, \ldots, \mathbf{x}_n\}$ on a sphere (3D), i.e. $\|\mathbf{x}_i\|_2 = 1, \forall i = 1 \ldots n$, when lower bound is achieved such that $\sum_{i \neq j} \mathbf{x}_i^T \mathbf{x}_j = -n$, then the minimum angle θ of any two vectors is in the range for some constant $C > 0$, i.e. $\cos\theta = \min_{i,j}(\mathbf{x}_i^T \mathbf{x}_j)$:*

$$1 - \left(\frac{4\pi}{\sqrt{3}n}\right) \leq \cos\theta \leq 1 - \frac{1}{2}\left(\left(\frac{8\pi}{\sqrt{3}n}\right)^{\frac{1}{2}} - Cn^{-\frac{2}{3}}\right)^2 \quad (4)$$

Pixel-Pair Spherical Max-Margin Regression

high-dimensional case

https://en.wikipedia.org/wiki/N-sphere

Pixel-Pair Spherical Max-Margin Regression

high-dimensional case

Forget about it --

It is one of the mathematical challenges of the 21st century.

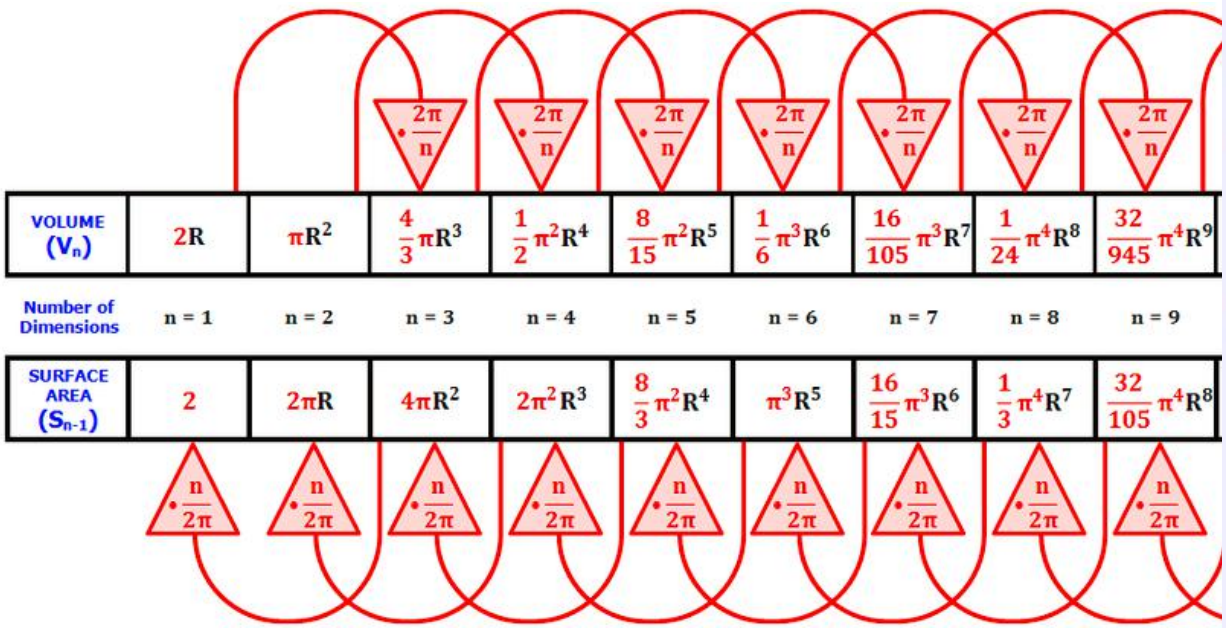
https://en.wikipedia.org/wiki/N-sphere

Pixel-Pair Spherical Max-Margin Regression

high-dimensional case

Forget about it --

It is one of the mathematical challenges of the 21st century.



$V_n(R)$ and $S_n(R)$ are the n-dimensional volume of the n-ball and the surface area of the n-sphere embedded in dimension n+1, respectively, of radius R.

The constants V_n and S_n (for the unit ball and sphere) are related by the recurrences:

$$V_0 = 1 \qquad V_{n+1} = S_n/(n+1)$$
$$S_0 = 2 \qquad S_{n+1} = 2\pi V_n$$

The surfaces and volumes can also be given in closed form:

$$S_n(R) = \frac{2\pi^{(n+1)/2}}{\Gamma(\frac{n+1}{2})} R^n$$

$$V_n(R) = \frac{\pi^{n/2}}{\Gamma(\frac{n}{2}+1)} R^n$$

where Γ is the gamma function. Derivations of these equations are given in this section.

https://en.wikipedia.org/wiki/N-sphere

Pixel-Pair Spherical Max-Margin Regression

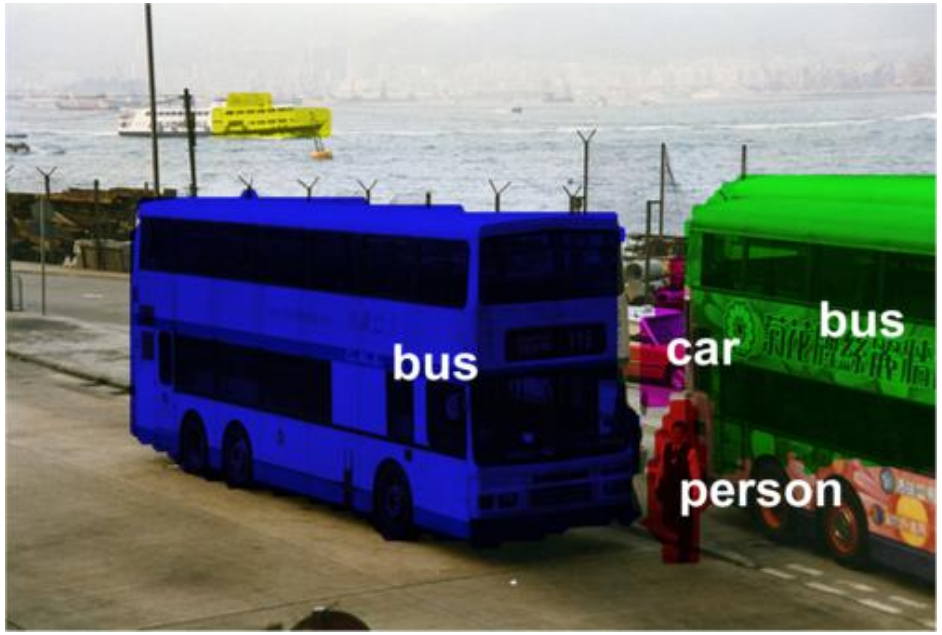
One more

Lemma 2 *Suppose there are n instances in the image to separate, and specifically, the ith instance has $|\mathcal{I}_i|$ pixels, of which the jth pixel is indexed by $\mathbf{x}_{ij}$; then* $\sum_{m \neq k,p,q} \mathbf{x}_{mp}^T \mathbf{x}_{kq} \geq -\sum_{i=1}^{n} |\mathcal{I}_i|^2$.

Pixel-Pair Spherical Max-Margin Regression

Last one -- Combination-aware Weighting

Lemma 3 *Suppose there are n instances in the image to separate, and specifically, the ith instance has $|\mathcal{I}_i|$ pixels, of which the jth pixel is indexed by $\mathbf{x}_{ij}$. We would like to weight the cross instance similarity loss to avoid large instance bias. Then,* $\sum_{m \neq k,p,q} \frac{1}{|\mathcal{I}_m|} \frac{1}{|\mathcal{I}_k|} \mathbf{x}_{mp}^T \mathbf{x}_{kq} \geq -n$.



Recurrent Mean Shift Grouping

From good embedding space to pixel labeling

How to get the instances?

How to group the pixels?

Recurrent Mean Shift Grouping

From good embedding space to pixel labeling

How to get the instances?

How to group the pixels?

k-means, k-medoids?

Recurrent Mean Shift Grouping

From good embedding space to pixel labeling

How to get the instances?

How to group the pixels?

k-means, k-medoids?

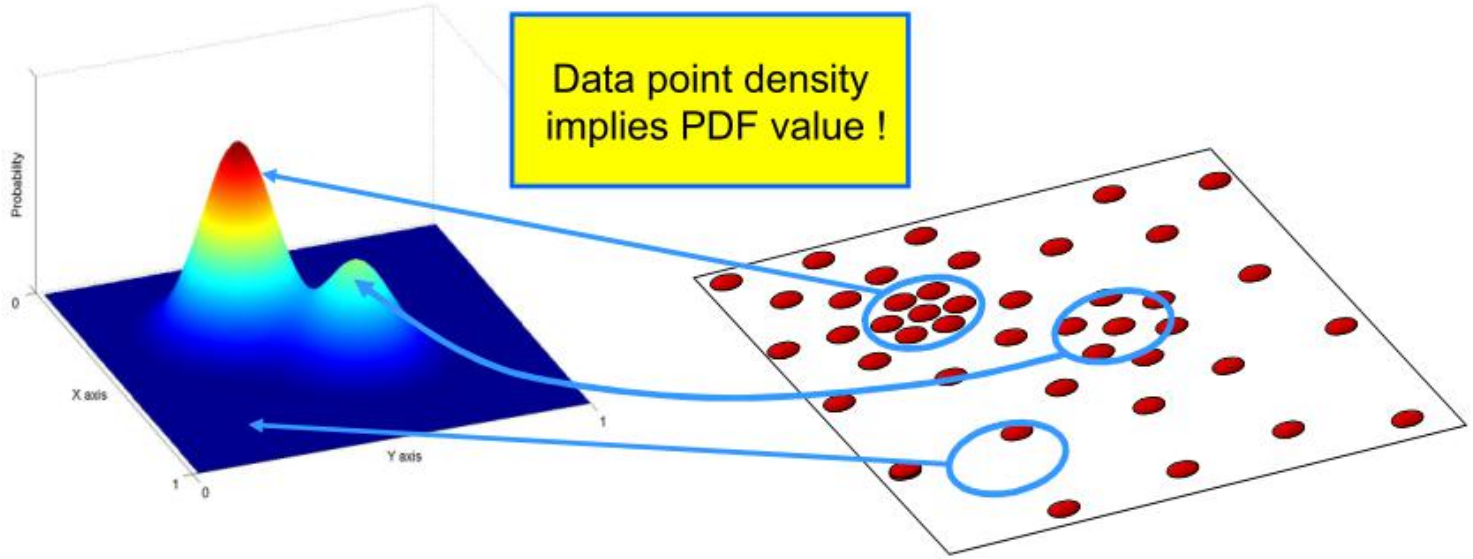
mean shift

Recurrent Mean Shift Grouping

mean shift

Non-Parametric Density Estimation

Assumption : The data points are sampled from an underlying PDF



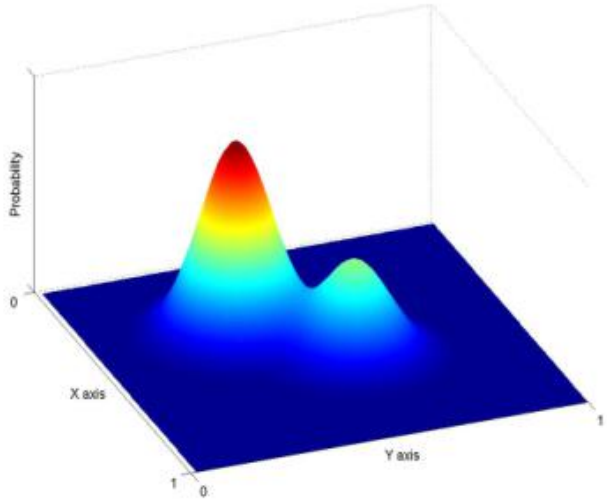
Assumed Underlying PDF

Real Data Samples

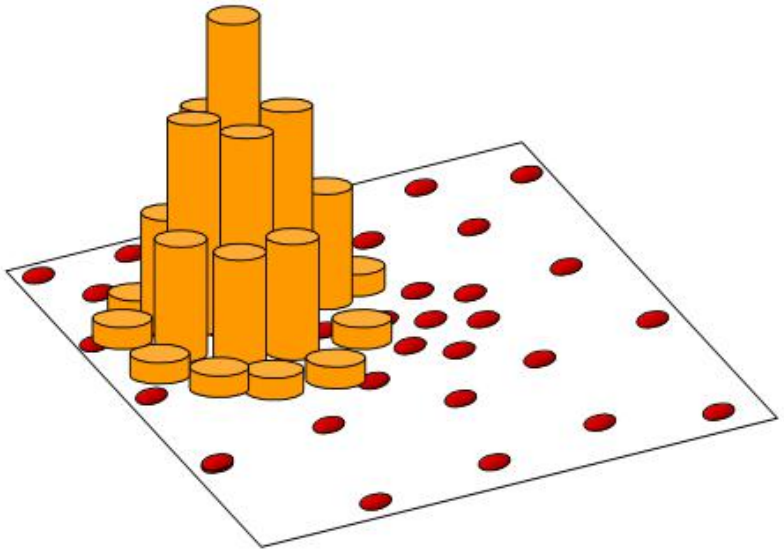
Recurrent Mean Shift Grouping

mean shift

Non-Parametric Density Estimation



Assumed Underlying PDF



Real Data Samples

Recurrent Mean Shift Grouping

mean shift

Consider a dataset $\{\mathbf{x}_n\}_{n=1}^N \subset \mathbb{R}^D$ and define a kernel density estimate

$$\hat{f}(\mathbf{x}) = \frac{1}{nh^d} \sum_{i=1}^{n} K\left(\frac{\mathbf{x} - \mathbf{x}_i}{h}\right).$$

mode seeking

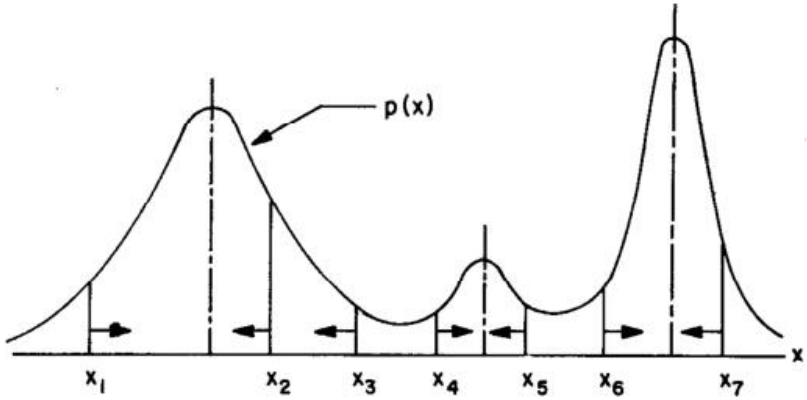


Fig. 1. Gradient mode clustering.

K. Fukunaga, L. Hostetler, The Estimation of the Gradient of a Density Function, with Applications in Pattern - Recognition, PAMI, 1975

Recurrent Mean Shift Grouping

mean shift

Consider a dataset $\{\mathbf{x}_n\}_{n=1}^N \subset \mathbb{R}^D$ and define a kernel density estimate

$$\hat{f}(\mathbf{x}) = \frac{1}{nh^d} \sum_{i=1}^{n} K\left(\frac{\mathbf{x} - \mathbf{x}_i}{h}\right).$$

Other than estimating the PDF directly, estimating the gradient --

$$\hat{\nabla} f_{h,K}(\mathbf{x}) \equiv \nabla \hat{f}_{h,K}(\mathbf{x}) = \frac{2c_{k,d}}{nh^{d+2}} \sum_{i=1}^{n} (\mathbf{x} - \mathbf{x}_i) k'\left(\left\|\frac{\mathbf{x} - \mathbf{x}_i}{h}\right\|^2\right).$$

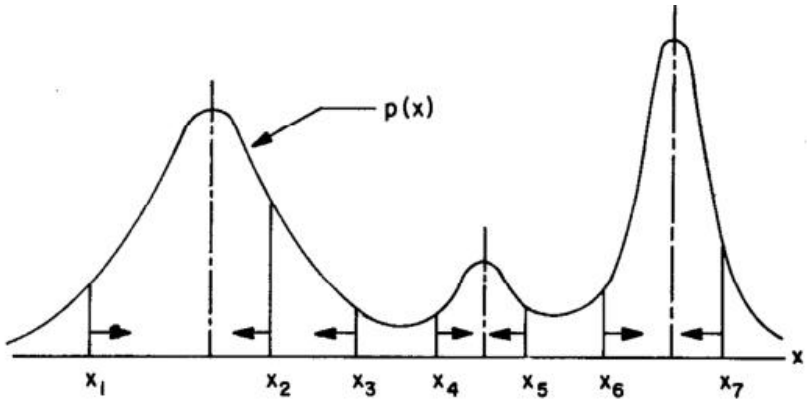


Fig. 1. Gradient mode clustering.

Recurrent Mean Shift Grouping

mean shift

$$\hat{\nabla} f_{h,K}(\mathbf{x}) \equiv \nabla \hat{f}_{h,K}(\mathbf{x}) = \frac{2c_{k,d}}{nh^{d+2}} \sum_{i=1}^{n} (\mathbf{x} - \mathbf{x}_i) k'\left(\left\|\frac{\mathbf{x} - \mathbf{x}_i}{h}\right\|^2\right).$$

We define the function

$$g(x) = -k'(x),$$

then

$$\begin{aligned}
&\hat{\nabla} f_{h,K}(\mathbf{x}) \\
&= \frac{2c_{k,d}}{nh^{d+2}} \sum_{i=1}^{n} (\mathbf{x}_i - \mathbf{x}) g\left(\left\|\frac{\mathbf{x} - \mathbf{x}_i}{h}\right\|^2\right) \\
&= \frac{2c_{k,d}}{nh^{d+2}} \left[\sum_{i=1}^{n} g\left(\left\|\frac{\mathbf{x} - \mathbf{x}_i}{h}\right\|^2\right)\right] \left[\frac{\sum_{i=1}^{n} \mathbf{x}_i g\left(\left\|\frac{\mathbf{x}-\mathbf{x}_i}{h}\right\|^2\right)}{\sum_{i=1}^{n} g\left(\left\|\frac{\mathbf{x}-\mathbf{x}_i}{h}\right\|^2\right)} - \mathbf{x}\right],
\end{aligned}$$

$\sum_{i=1}^{n} g\left(\left\|\frac{\mathbf{x}-\mathbf{x}_i}{h}\right\|^2\right)$ is assumed to be a positive number.

Recurrent Mean Shift Grouping

mean shift: iteratively updating by shifting the data by such an amount

The second term is the *mean shift*

$$\mathbf{m}_{h,G}(\mathbf{x}) = \frac{\sum_{i=1}^{n} \mathbf{x}_i g\left(\left\|\frac{\mathbf{x}-\mathbf{x}_i}{h}\right\|^2\right)}{\sum_{i=1}^{n} g\left(\left\|\frac{\mathbf{x}-\mathbf{x}_i}{h}\right\|^2\right)} - \mathbf{x},$$

Recurrent Mean Shift Grouping

mean shift: iteratively updating by shifting the data by such an amount

The second term is the *mean shift*

$$\mathbf{m}_{h,G}(\mathbf{x}) = \frac{\sum_{i=1}^{n} \mathbf{x}_i g\left(\left\|\frac{\mathbf{x}-\mathbf{x}_i}{h}\right\|^2\right)}{\sum_{i=1}^{n} g\left(\left\|\frac{\mathbf{x}-\mathbf{x}_i}{h}\right\|^2\right)} - \mathbf{x},$$

$$\mathbf{Y} = \mathbf{X} + \eta \mathbf{M}$$

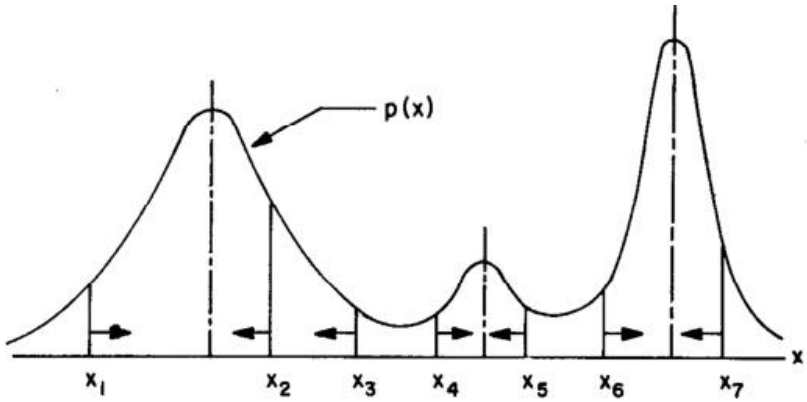


Fig. 1. Gradient mode clustering.

Recurrent Mean Shift Grouping

mean shift: iteratively updating by shifting the data by such an amount

The second term is the *mean shift*

$$\mathbf{m}_{h,G}(\mathbf{x}) = \frac{\sum_{i=1}^{n} \mathbf{x}_i g\left(\left\|\frac{\mathbf{x}-\mathbf{x}_i}{h}\right\|^2\right)}{\sum_{i=1}^{n} g\left(\left\|\frac{\mathbf{x}-\mathbf{x}_i}{h}\right\|^2\right)} - \mathbf{x},$$

Gaussian blurring mean-shift (GBMS) algorithm

the new iterate is the data average under the posterior probabilities given the current iterate:

repeat — Iteration loop

  for $m \in \{1, \ldots, N\}$ — For each data point

    $\forall n\colon p(n|\mathbf{x}_m) \leftarrow \dfrac{e^{-\frac{1}{2}\left\|\frac{\mathbf{x}_m-\mathbf{x}_n}{\sigma}\right\|^2}}{\sum_{n'=1}^{N} e^{-\frac{1}{2}\left\|\frac{\mathbf{x}_m-\mathbf{x}_{n'}}{\sigma}\right\|^2}}$ — Eq. (3a)

    $\mathbf{y}_m \leftarrow \sum_{n=1}^{N} p(n|\mathbf{x}_m)\mathbf{x}_n$ — One GMS step, eq. (3b)

  end

Recurrent Mean Shift Grouping

Gaussian blurring mean-shift (GBMS) algorithm

B. Gaussian blurring mean-shift (GBMS) algorithm

repeat	Iteration loop
for $m \in \{1, \ldots, N\}$	For each data point
$\forall n\colon p(n\|\mathbf{x}_m) \leftarrow \dfrac{e^{-\frac{1}{2}\left\|\frac{\mathbf{x}_m-\mathbf{x}_n}{\sigma}\right\|^2}}{\sum_{n'=1}^{N} e^{-\frac{1}{2}\left\|\frac{\mathbf{x}_m-\mathbf{x}_{n'}}{\sigma}\right\|^2}}$	Eq. (3a)
$\mathbf{y}_m \leftarrow \sum_{n=1}^{N} p(n\|\mathbf{x}_m)\mathbf{x}_n$	One GMS step, eq. (3b)
end	
$\forall m\colon \mathbf{x}_m \leftarrow \mathbf{y}_m$	Update whole dataset
until stop	See section 1
connected-components($\{\mathbf{x}_n\}_{n=1}^{N}$,`min_diff`)	Clusters

C. GBMS algorithm in matrix form

repeat	Iteration loop
$\mathbf{W} = \left(\exp\left(-\frac{1}{2}\left\|\frac{\mathbf{x}_m-\mathbf{x}_n}{\sigma}\right\|^2\right)\right)_{nm}$	Gaussian affinity matrix
$\mathbf{D} = \operatorname{diag}\left(\sum_{n=1}^{N} w_{nm}\right)$	Degree (normalising) matrix
$\mathbf{X} = \mathbf{X}\mathbf{W}\mathbf{D}^{-1}$	Update whole dataset
until stop	See section 1
connected-components($\{\mathbf{x}_n\}_{n=1}^{N}$,`min_diff`)	Clusters

Miguel A. Carreira-Perpinán, Fast Nonparametric Clustering with Gaussian Blurring Mean-Shift, ICML, 2006

Recurrent Mean Shift Grouping

Gaussian blurring mean-shift (GBMS) algorithm

It's guaranteed to converge without gradient vanishing or exploding

B. Gaussian blurring mean-shift (GBMS) algorithm

repeat	Iteration loop
for $m \in \{1, \ldots, N\}$	For each data point
$\forall n\colon p(n\|\mathbf{x}_m) \leftarrow \dfrac{e^{-\frac{1}{2}\left\|\frac{\mathbf{x}_m-\mathbf{x}_n}{\sigma}\right\|^2}}{\sum_{n'=1}^{N} e^{-\frac{1}{2}\left\|\frac{\mathbf{x}_m-\mathbf{x}_{n'}}{\sigma}\right\|^2}}$	Eq. (3a)
$\mathbf{y}_m \leftarrow \sum_{n=1}^{N} p(n\|\mathbf{x}_m)\mathbf{x}_n$	One GMS step, eq. (3b)
end	
$\forall m\colon \mathbf{x}_m \leftarrow \mathbf{y}_m$	Update whole dataset
until stop	See section 1
connected-components($\{\mathbf{x}_n\}_{n=1}^{N}$,`min_diff`)	Clusters

C. GBMS algorithm in matrix form

repeat	Iteration loop
$\mathbf{W} = \left(\exp\left(-\frac{1}{2}\left\|\frac{\mathbf{x}_m-\mathbf{x}_n}{\sigma}\right\|^2\right)\right)_{nm}$	Gaussian affinity matrix
$\mathbf{D} = \operatorname{diag}\left(\sum_{n=1}^{N} w_{nm}\right)$	Degree (normalising) matrix
$\mathbf{X} = \mathbf{X}\mathbf{W}\mathbf{D}^{-1}$	Update whole dataset
until stop	See section 1
connected-components($\{\mathbf{x}_n\}_{n=1}^{N}$,`min_diff`)	Clusters

Miguel A. Carreira-Perpinán, Fast Nonparametric Clustering with Gaussian Blurring Mean-Shift, ICML, 2006

Recurrent Mean Shift Grouping

Gaussian blurring mean-shift (GBMS) algorithm

It's guaranteed to converge without gradient vanishing or exploding

But, are the updated data still on the sphere?

B. Gaussian blurring mean-shift (GBMS) algorithm

repeat	Iteration loop
for $m \in \{1, \ldots, N\}$	For each data point
$\forall n\colon p(n\|\mathbf{x}_m) \leftarrow \dfrac{e^{-\frac{1}{2}\left\|\frac{\mathbf{x}_m-\mathbf{x}_n}{\sigma}\right\|^2}}{\sum_{n'=1}^{N} e^{-\frac{1}{2}\left\|\frac{\mathbf{x}_m-\mathbf{x}_{n'}}{\sigma}\right\|^2}}$	Eq. (3a)
$\mathbf{y}_m \leftarrow \sum_{n=1}^{N} p(n\|\mathbf{x}_m)\mathbf{x}_n$	One GMS step, eq. (3b)
end	
$\forall m\colon \mathbf{x}_m \leftarrow \mathbf{y}_m$	Update whole dataset
until stop	See section 1
connected-components($\{\mathbf{x}_n\}_{n=1}^{N}$,min_diff)	Clusters

C. GBMS algorithm in matrix form

repeat	Iteration loop
$\mathbf{W} = \left(\exp\left(-\frac{1}{2}\left\|\frac{\mathbf{x}_m-\mathbf{x}_n}{\sigma}\right\|^2\right)\right)_{nm}$	Gaussian affinity matrix
$\mathbf{D} = \operatorname{diag}\left(\sum_{n=1}^{N} w_{nm}\right)$	Degree (normalising) matrix
$\mathbf{X} = \mathbf{X}\mathbf{W}\mathbf{D}^{-1}$	Update whole dataset
until stop	See section 1
connected-components($\{\mathbf{x}_n\}_{n=1}^{N}$,min_diff)	Clusters

Miguel A. Carreira-Perpinán, Fast Nonparametric Clustering with Gaussian Blurring Mean-Shift, ICML, 2006

Recurrent Mean Shift Grouping

L2 normalization in the loop

Algorithm 1 : vMF Mean Shift

[Normalization] $||\boldsymbol{x}_i|| = 1, \forall i.$

for $i = 1$ **to** N **do**

[Init] Start from the i-th sample: $\boldsymbol{y}_i^{(0)} = \boldsymbol{x}_i$.

[Update] Update the target points (vectors) until convergence by the following formula:

$$\boldsymbol{y}_i^{(t+1)} \leftarrow \frac{\sum_j^N \boldsymbol{x}_j g(\boldsymbol{x}_j' \boldsymbol{y}_i^{(t)}; \kappa)}{||\sum_j^N \boldsymbol{x}_j g(\boldsymbol{x}_j' \boldsymbol{y}_i^{(t)}; \kappa)||}. \quad (6)$$

end for

[Postprocessing] Merge close convergent points $(\boldsymbol{y}_i^{(\infty)'} \boldsymbol{y}_j^{(\infty)} > 1 - \epsilon)$ into the same cluster.

It's guaranteed to converge without gradient vanishing or exploding.

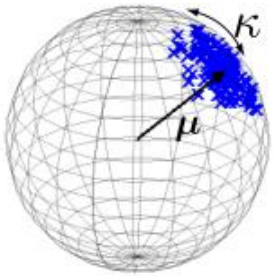


Figure 1. von Mises-Fisher distribution on a three-dimensional sphere.

Takumi Kobayashi, Nobuyuki Otsu, Von Mises-Fisher Mean Shift for Clustering on a Hypersphere, ICPR, 2010

Recurrent Mean Shift Grouping

running the von-Mises Fisher mean shift offline

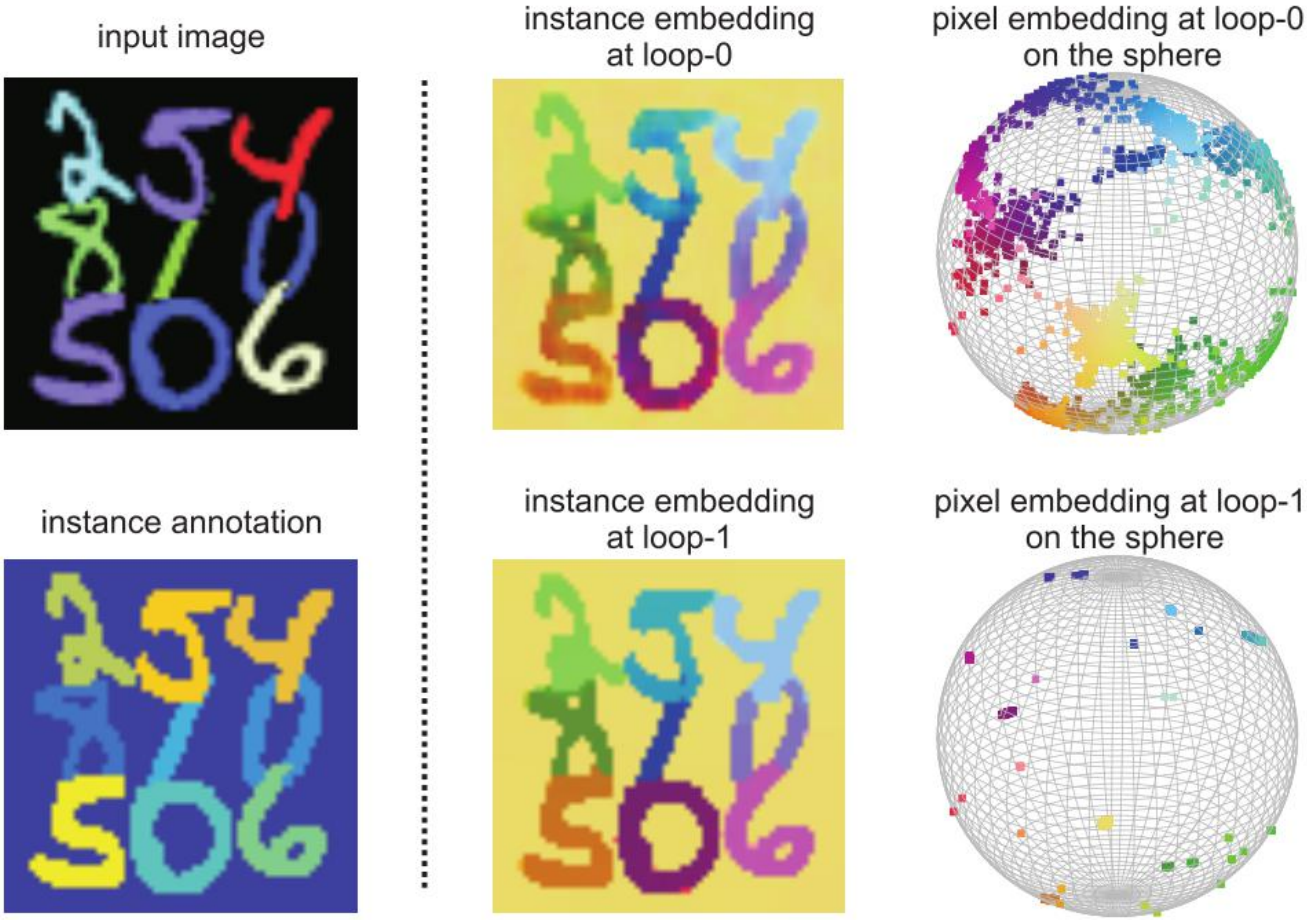


Figure 5: Demonstration. After 10 iterations of mean-shift

Recurrent Mean Shift Grouping

mean shift as recurrent module

Recurrent Mean Shift Grouping

mean shift as recurrent module

C. GBMS algorithm in matrix form

repeat

$\mathbf{W} = \left(\exp\left(-\frac{1}{2}\left\|\frac{\mathbf{x}_m - \mathbf{x}_n}{\sigma}\right\|^2\right)\right)_{nm}$

$\mathbf{D} = \mathrm{diag}\left(\sum_{n=1}^{N} w_{nm}\right)$

$\mathbf{X} = \mathbf{X}\mathbf{W}\mathbf{D}^{-1}$

until stop

connected-components($\{\mathbf{x}_n\}_{n=1}^{N}$,min_diff)

Let $\mathbf{X} \in \mathbb{R}^{p \times N}$, then we have cosine similarity graph $\mathbf{S} = \mathbf{X}^T\mathbf{X} \in \mathbb{R}^{N \times N}$. We can transform cosine similarity graph into Gaussian kernel affinity graph $\mathbf{G} = \exp(\delta^2(\mathbf{S} - 1))$.

calculating the degree vector $\mathbf{d} = \mathbf{G}^T\mathbf{1}$

elementwise inverse degree vector $\mathbf{q} = \mathbf{d}^{-1}$

diagonal matrix $\mathrm{diag}(\mathbf{q})$

the mean shift for all batch $\mathbf{X}$ is $\mathbf{M} = \mathbf{X}\mathbf{G}\mathrm{diag}(\mathbf{q}) - \mathbf{X}$

$$
\begin{aligned}
\mathbf{Y} =& \mathbf{X} + \eta\mathbf{M} \\
=& \mathbf{X} + \eta(\mathbf{X}\mathbf{G}\mathrm{diag}(\mathbf{q}) - \mathbf{X}) \\
=& \mathbf{X}(\eta\mathbf{G}\mathrm{diag}(\mathbf{q}) + (1-\eta)\mathbf{I})
\end{aligned}
$$

Recurrent Mean Shift Grouping

mean shift as recurrent module

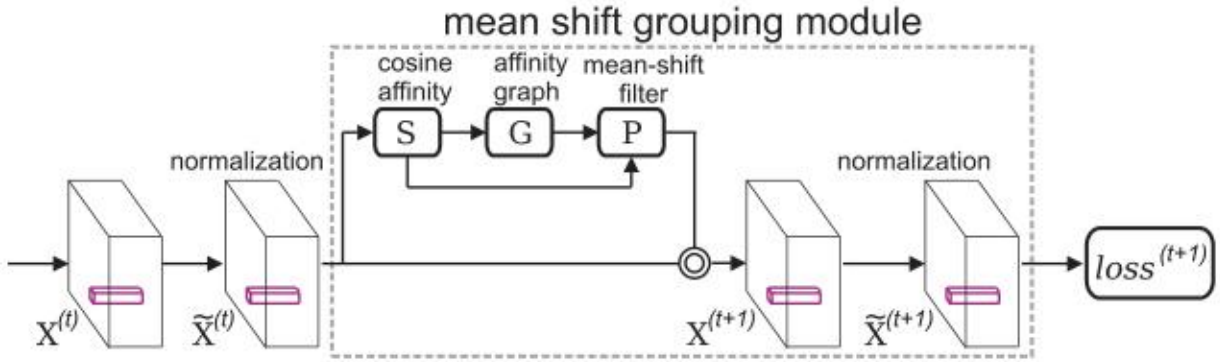


Figure 2: Mean shift grouping module.

$$\mathbf{S} = \mathbf{X}^T\mathbf{X}$$

$$\mathbf{G} = \exp(\delta^2(\mathbf{S} - 1)), \quad \mathbf{d} = \mathbf{G}^T\mathbf{1}, \quad \mathbf{q} = \mathbf{d}^{-1}$$

$$\mathbf{P} = \mathbf{G}\mathrm{diag}(\mathbf{q}), \text{ or } \mathbf{P} = (1-\eta)\mathbf{I} + \eta\mathbf{G}\mathrm{diag}(\mathbf{q})$$

$$\mathbf{Y} = \mathbf{X}\mathbf{P}$$

Recurrent Mean Shift Grouping

mean shift grouping in the loop

$$\mathbf{S} = \mathbf{X}^T\mathbf{X}$$

$$\mathbf{G} = \exp(\delta^2(\mathbf{S}-1)), \quad \mathbf{d} = \mathbf{G}^T\mathbf{1}, \quad \mathbf{q} = \mathbf{d}^{-1}$$

$$\mathbf{P} = \mathbf{G}\mathrm{diag}(\mathbf{q}), \text{ or } \mathbf{P} = (1-\eta)\mathbf{I} + \eta\mathbf{G}\mathrm{diag}(\mathbf{q})$$

$$\mathbf{Y} = \mathbf{X}\mathbf{P}$$

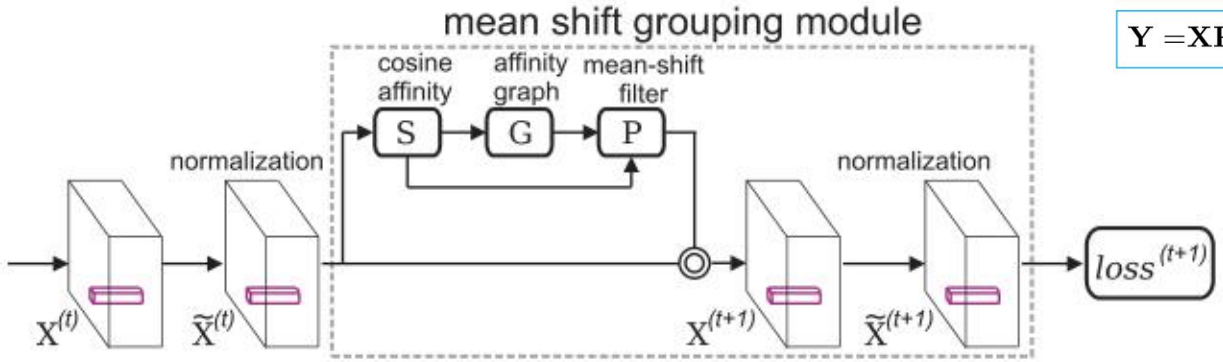
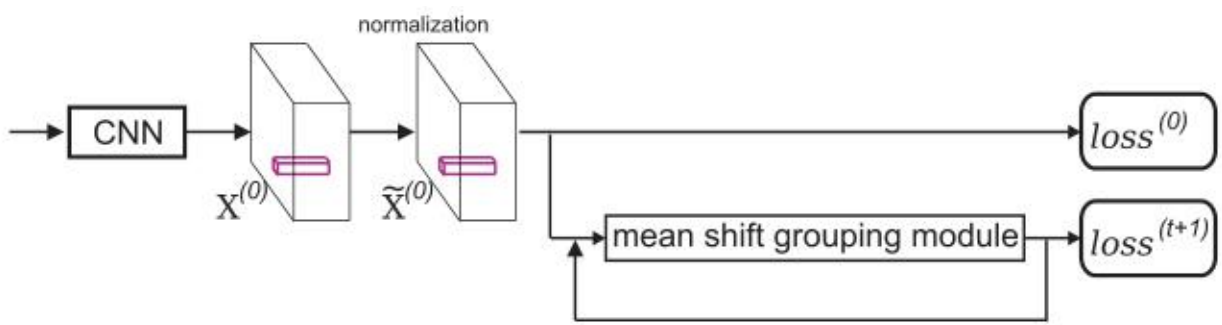


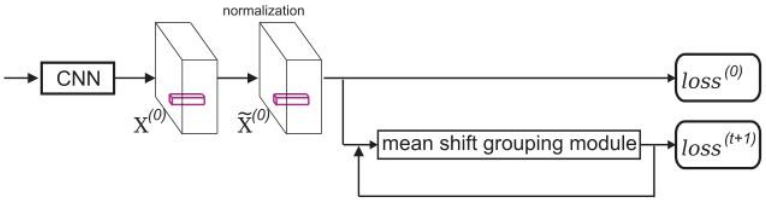
Figure 2: Mean shift grouping module.

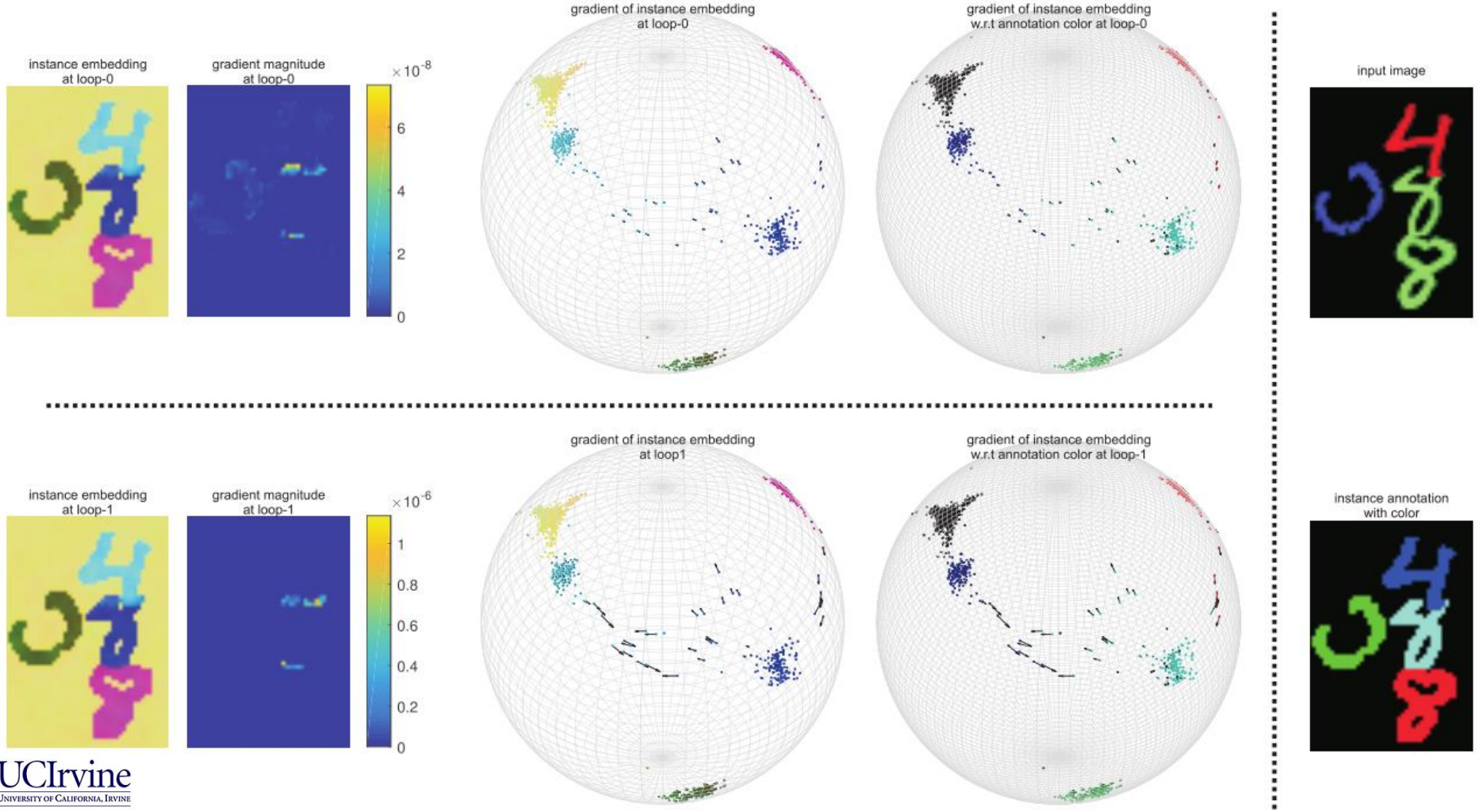


$$l^t = \frac{1}{|S|}\sum_{i,j\in S} w_{ij}\left(\mathbf{1}_{\{y_i=y_j\}}\|s_{ij}^t - 1\|_2^2 + \mathbf{1}_{\{y_i\neq y_j\}}[s_{ij}^t - \alpha]_+\right)$$

Recurrent Mean Shift Grouping

What does it mean by mean shift gradient?

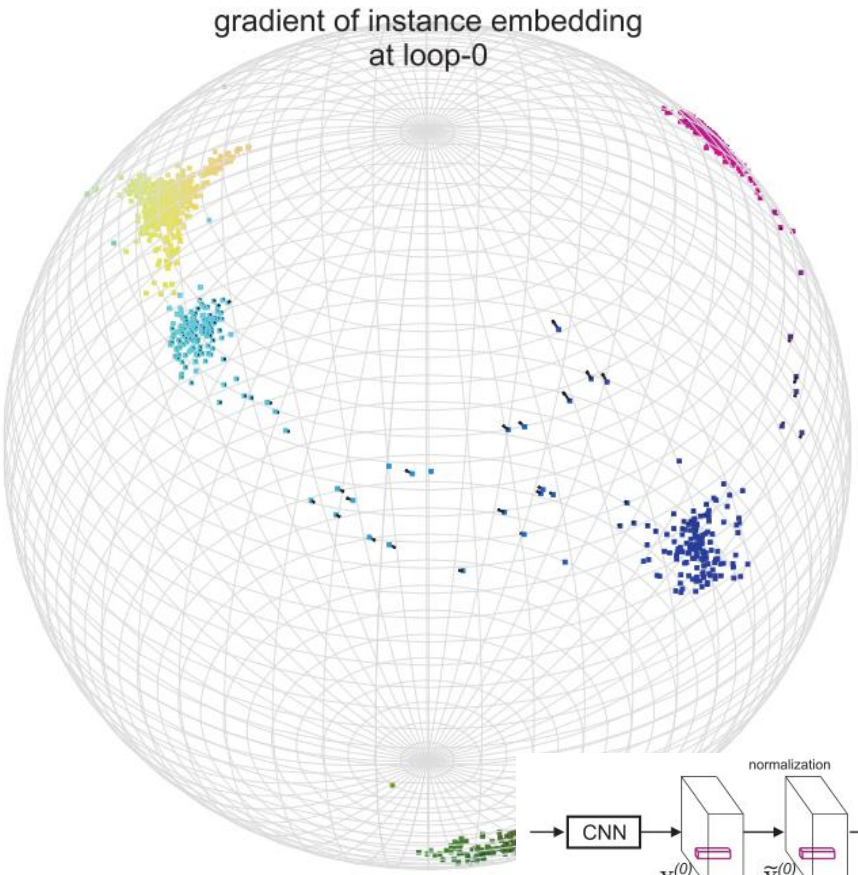


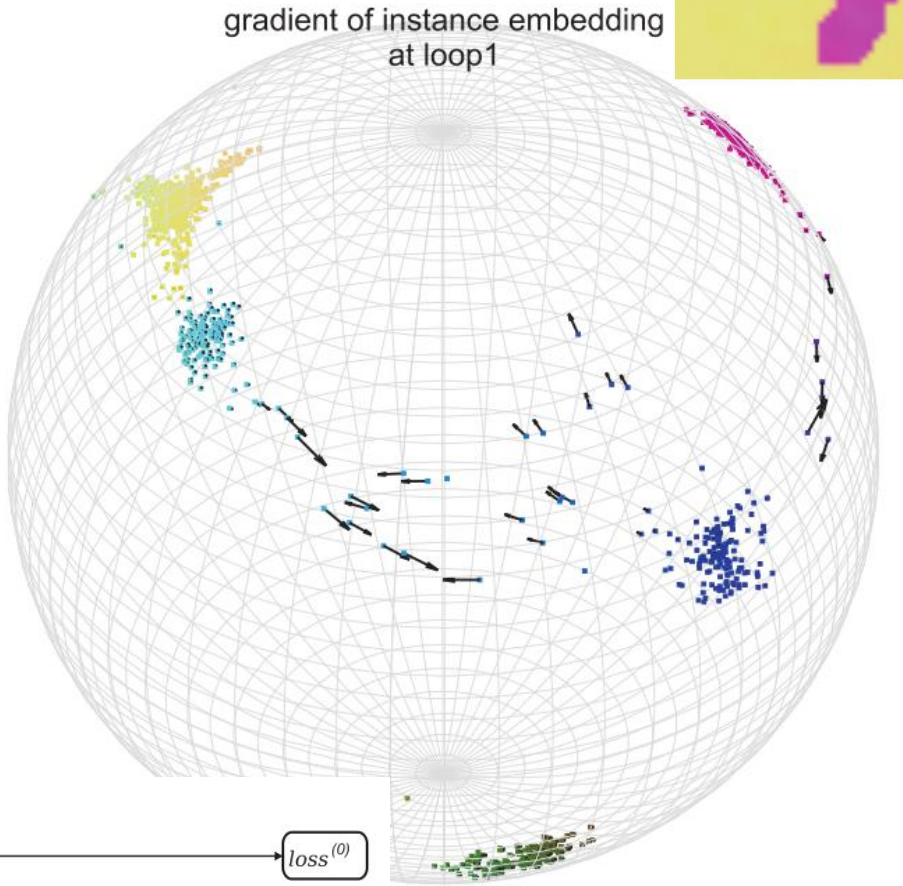


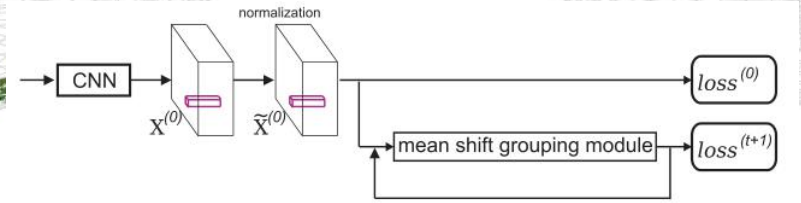
UCIrvine
University of California, Irvine

Recurrent Mean Shift Grouping

What does it mean by mean shift gradient?



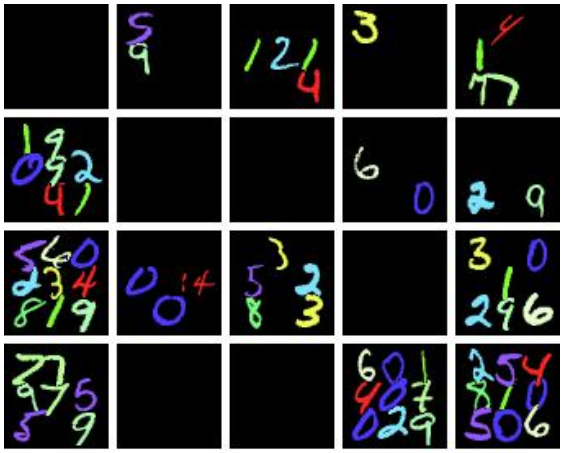




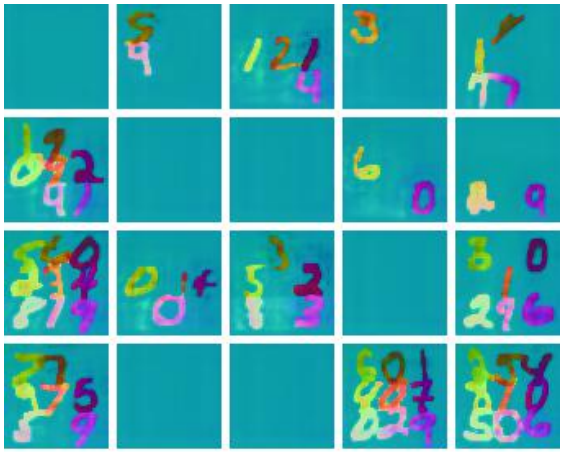
Recurrent Mean Shift Grouping

mean shift grouping in the loop

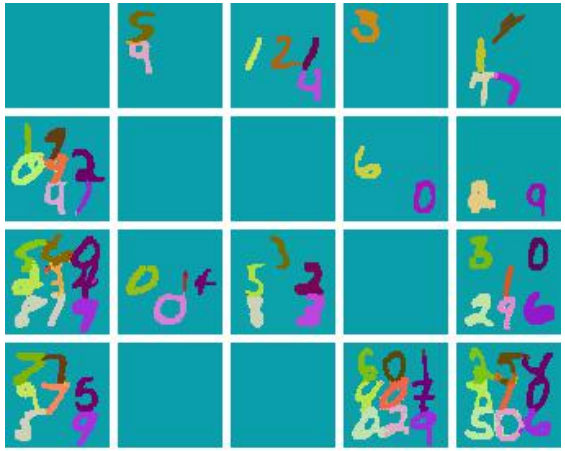
input image

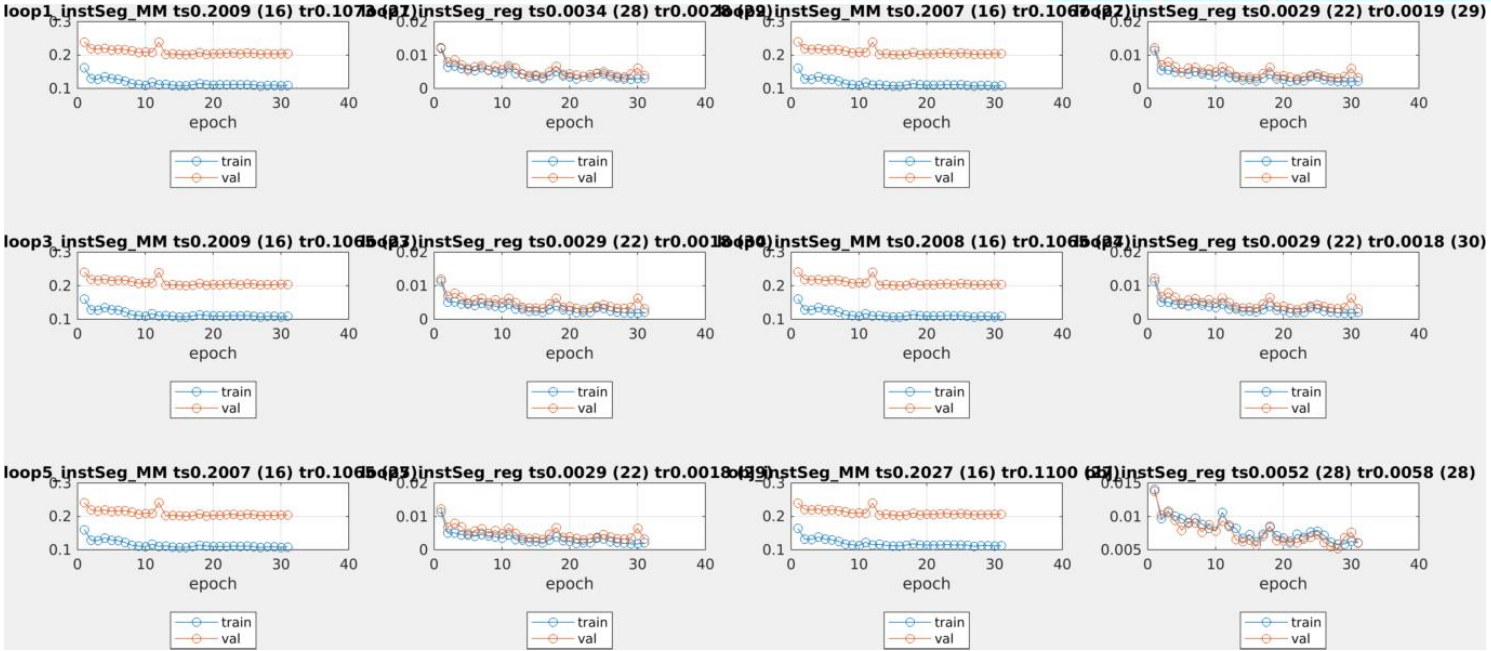


loop-0



loop-5





Learning to Group

Low-level vision:

edge, boundary, contour

Mid-level vision:

object proposal

High-level vision:

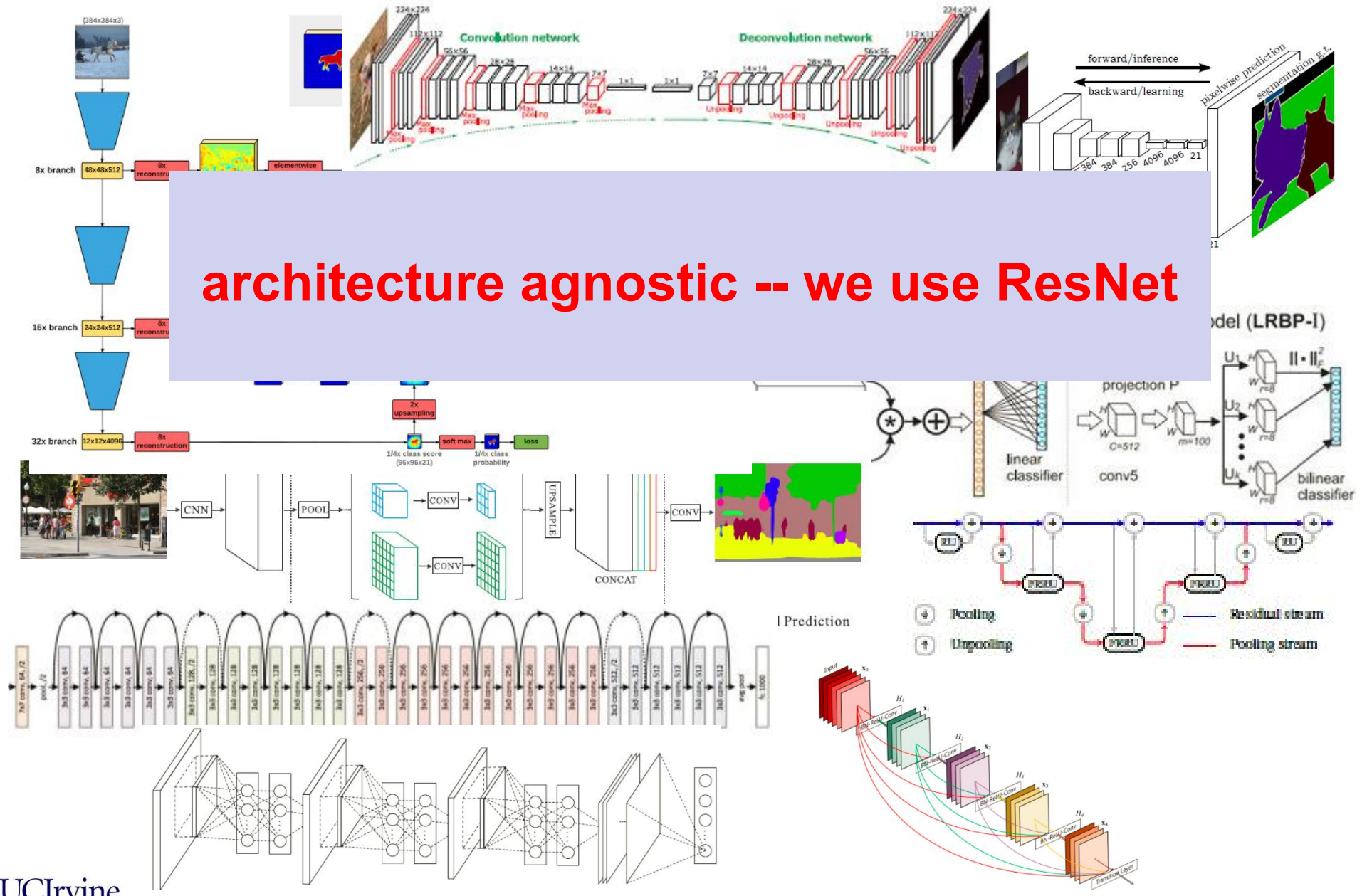
semantic segmentation

instance-level semantic segmentation

End-to-end trainable from data;

with the cross-entropy loss.

Backbone



architecture agnostic -- we use ResNet

K He, X Zhang, S Ren, J Sun, Deep Residual Learning for Image Recognition, CVPR, 2016

Experiment: Boundary Detection

boundary detection

one of most imbalanced problem, 85% pixels are non-boundary

1. learn the embedding space of 3-dim with our loss;
2. after convergence, adding logistic loss and fine-tuning;
3. averaging multiple outputs at resBlock2~5 followed by thinning (NMS).

Experiment: Boundary Detection

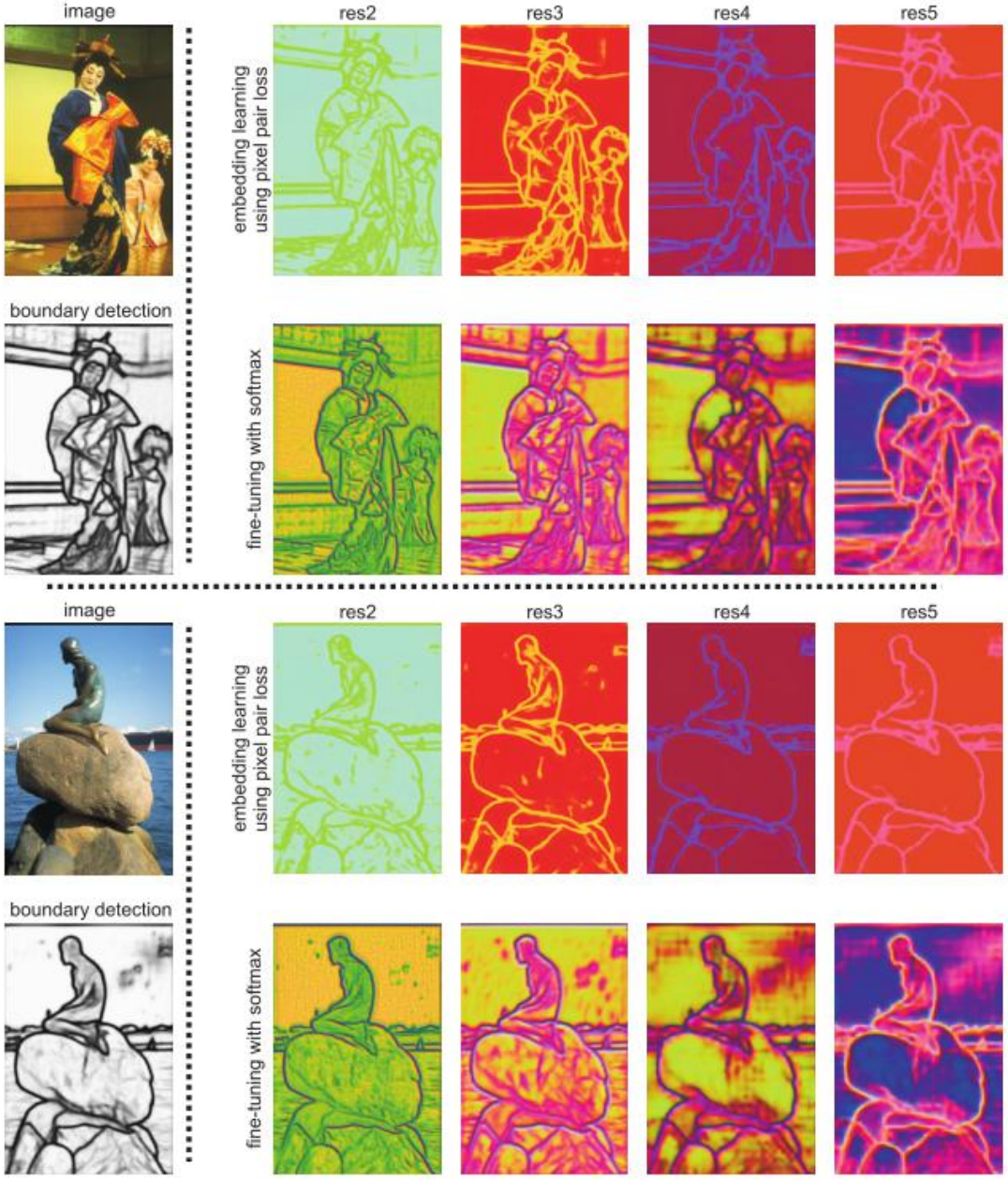
visualize the 3-dim embedding maps as rgb image

before&after fine-tuning with logistic loss

Experiment: Boundary Detection

visualize the 3-dim embedding maps as rgb image

before&after fine-tuning with logistic loss



Experiment: Boundary Detection

quantitative comparison

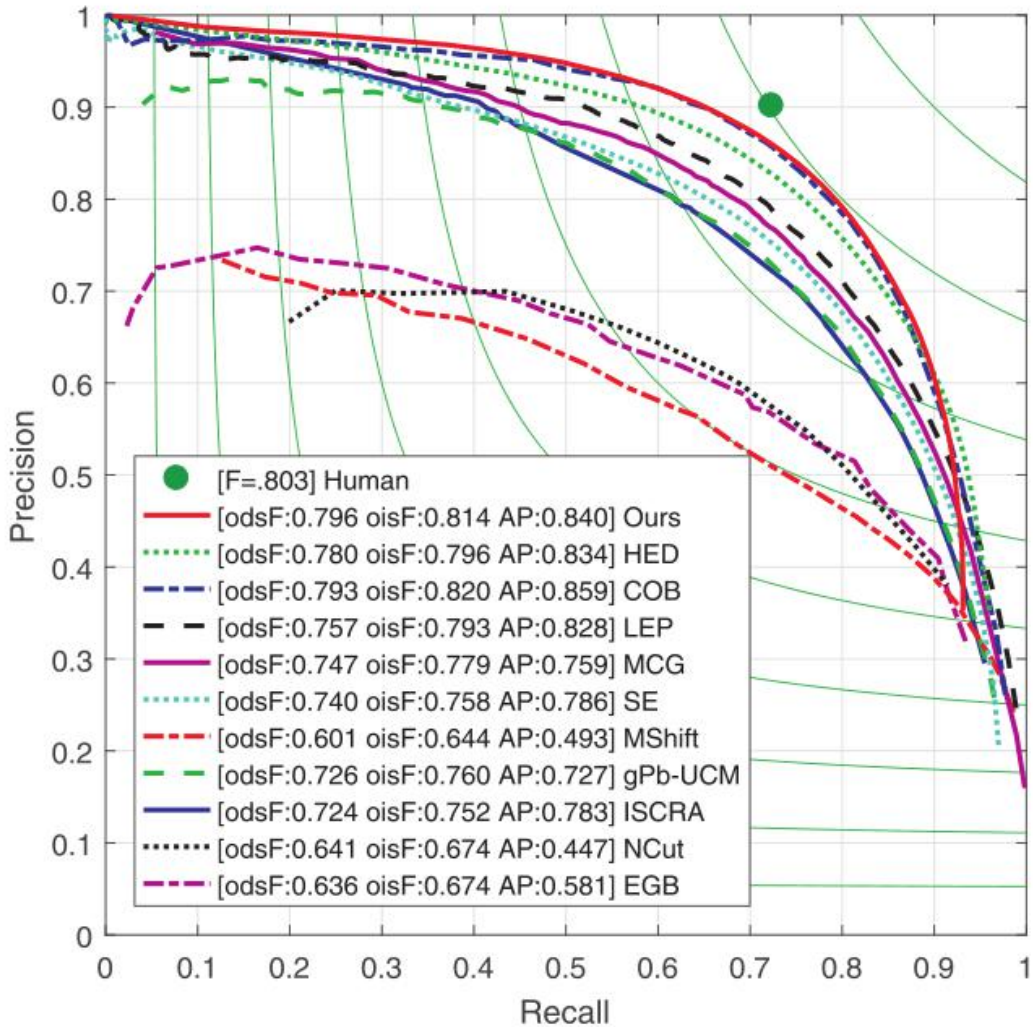
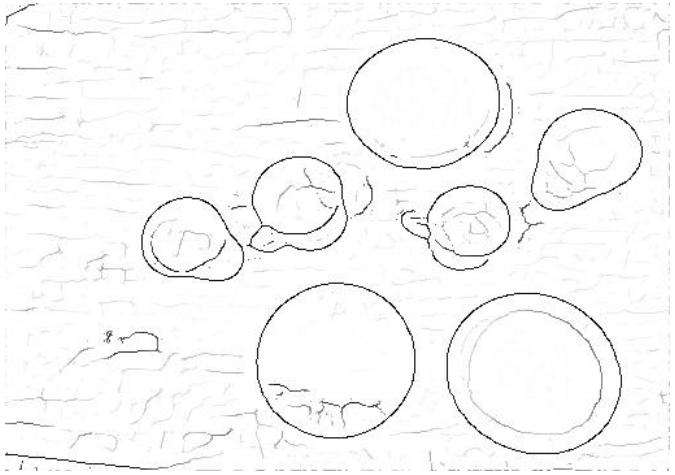


Figure 8: Boundary detection on BSDS500 test set.

Experiment: Boundary Detection

test image

input image

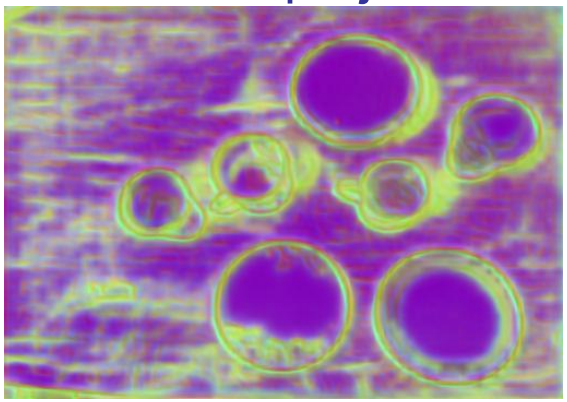


Experiment: Boundary Detection

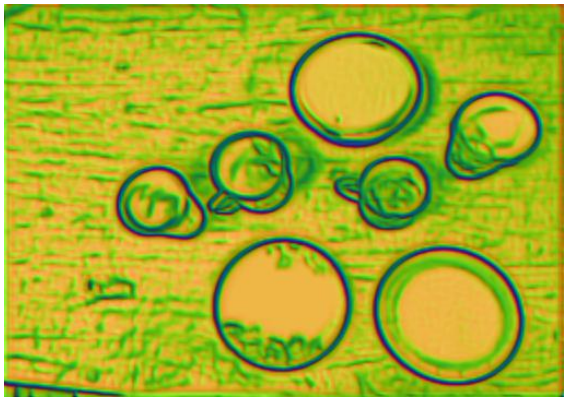
test image aesthetically colorful

input image

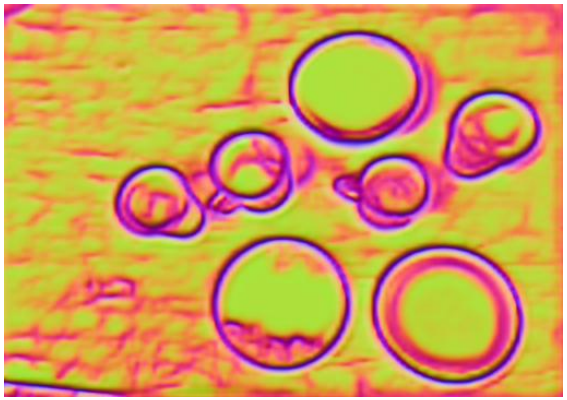
rand-proj



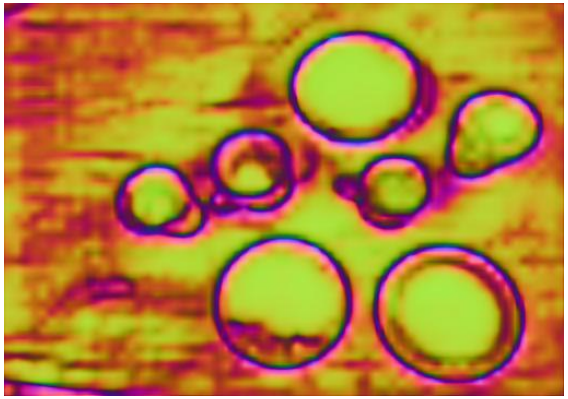
res2



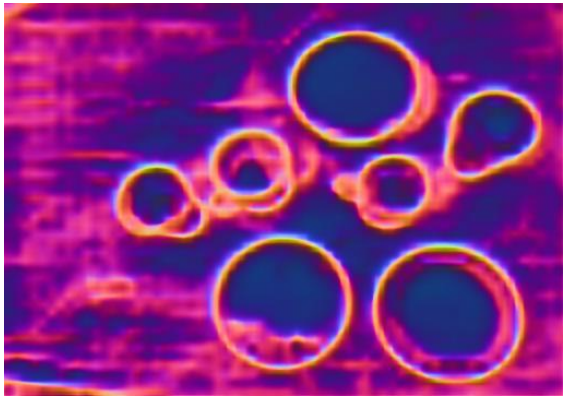
res3



res4

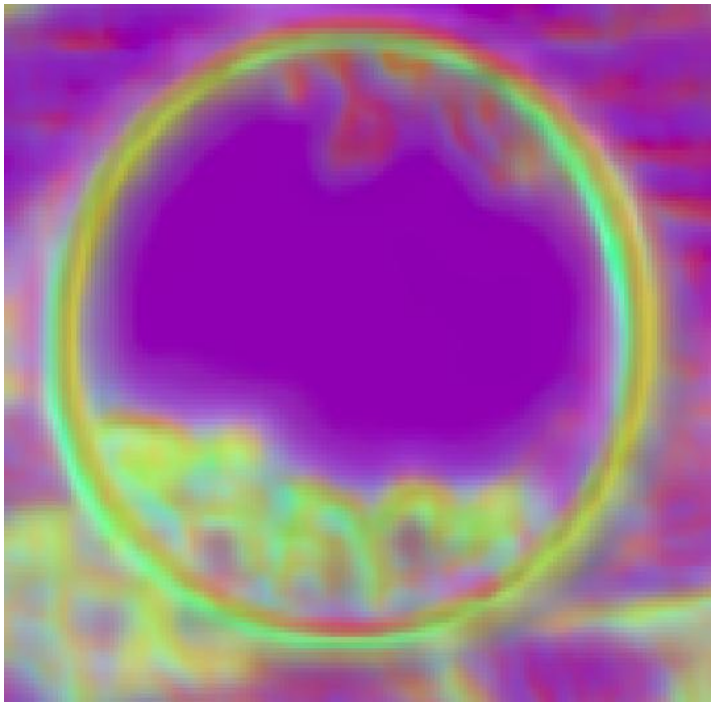


res5



Experiment: Boundary Detection

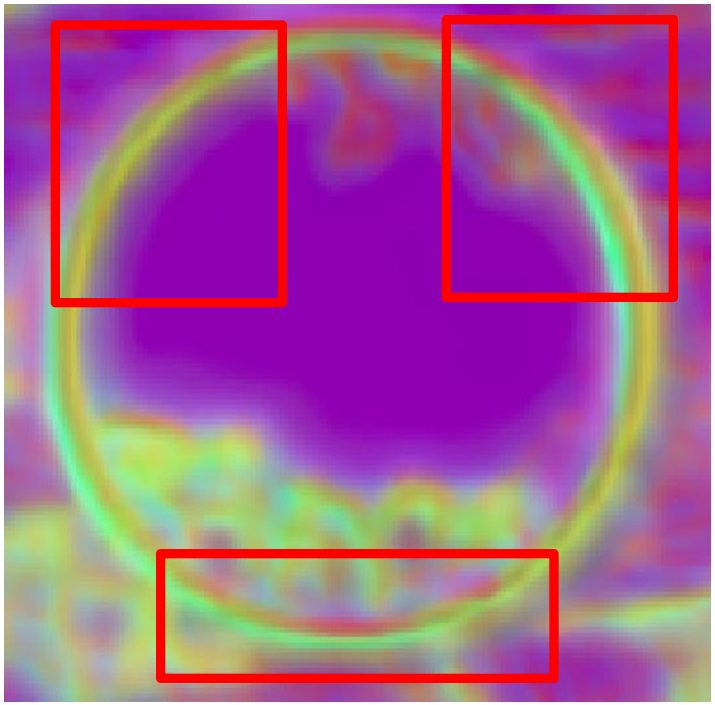
[zoom-in] encoding orientation, distance transform



Experiment: Boundary Detection

[zoom-in] encoding orientation, distance transform

the Mobius strip



UCIrvine
University of California, Irvine

Experiment: Object Proposal Detection

object proposal detection

- class-agnostic
- reduce search space for subsequential tasks, e.g. object detection
- the proposal framework is particularity suitable for this tasks

How suitable?

Experiment: Object Proposal Detection

object proposal detection -- How suitable?

Average Recall:
averaging recall at IoU=[0.5:0.05:0.95]

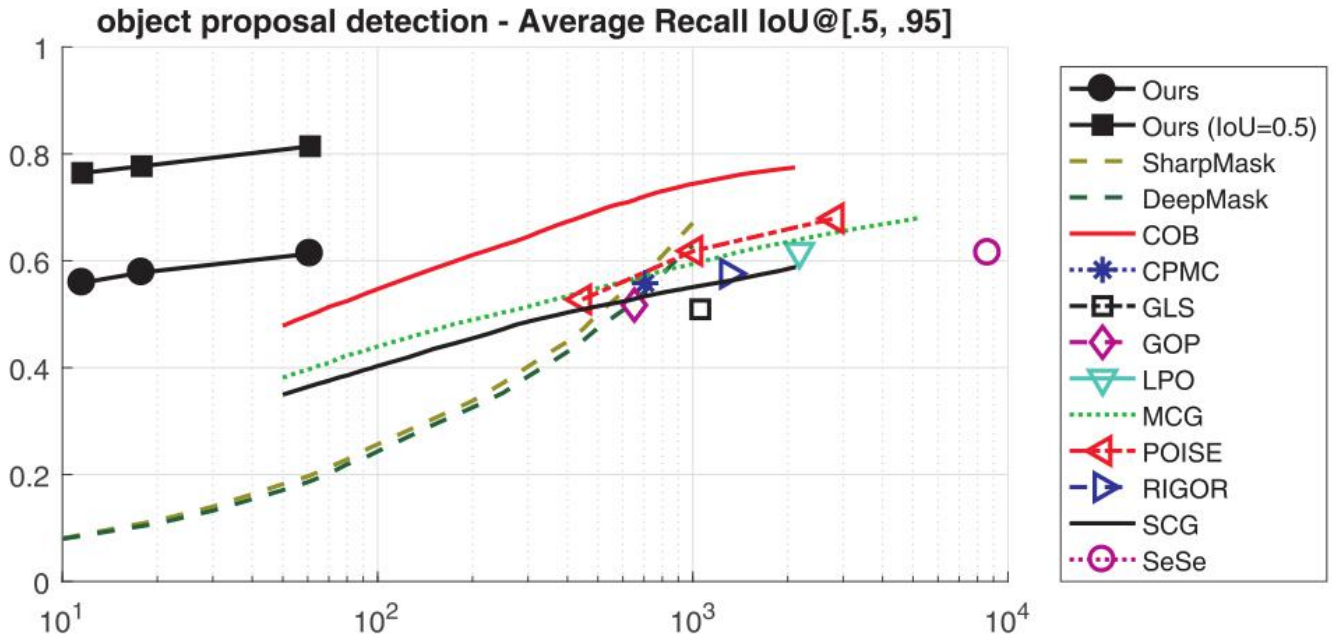
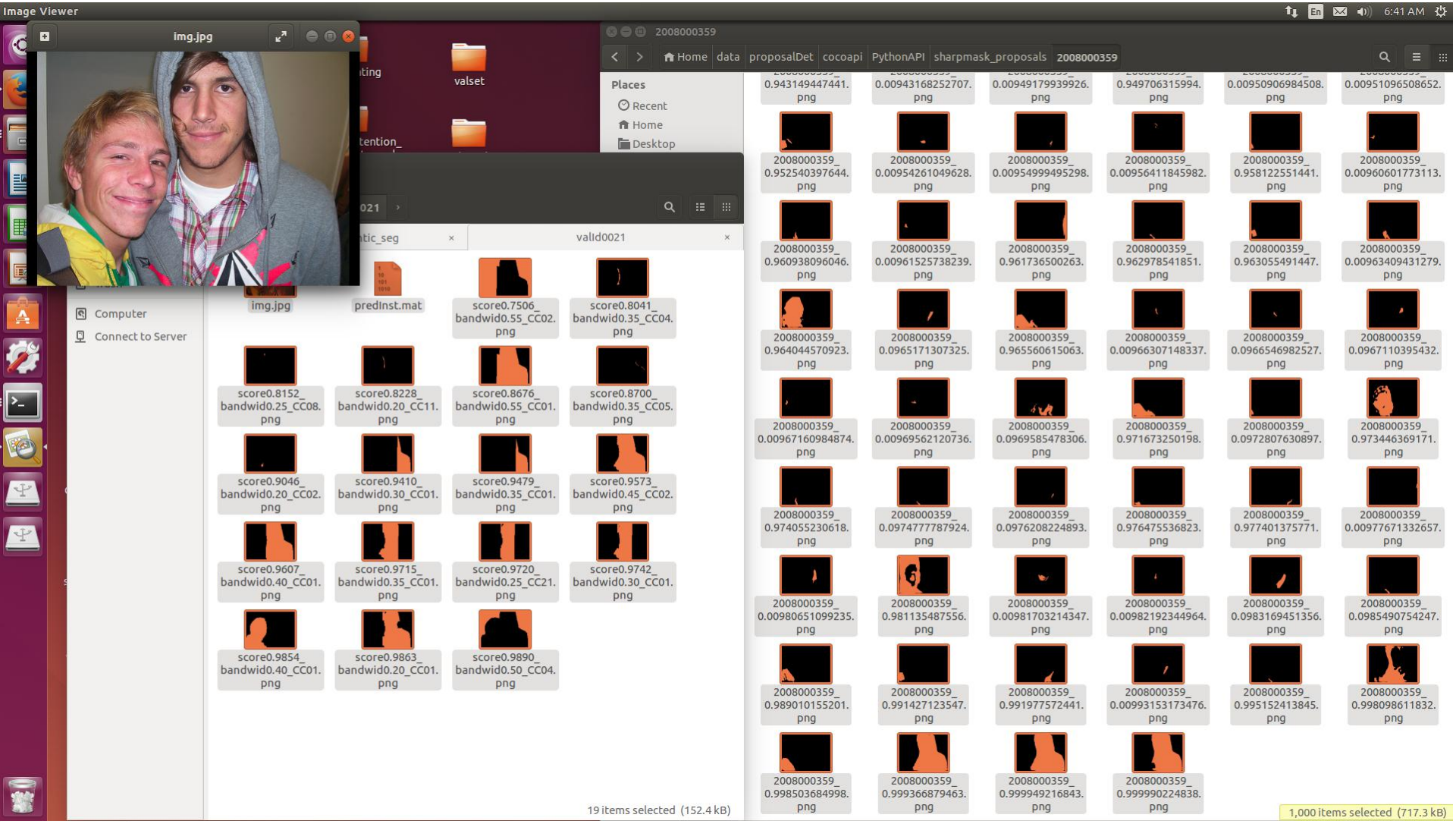


Figure 9: Segmented object proposals evaluation on PASCAL VOC 2012 validation set. We especially mark the performance of our method at IoU=0.5.

Achieving very high average recall (AR) with a dozen proposals per image!

Experiment: Object Proposal Detection

Quanlitatively: Ours vs. SharpMask



Experiment: Semantic Segmentation

semantic segmentation

with cross-entropy loss

The pixel pair loss can fill in

the “holes”.

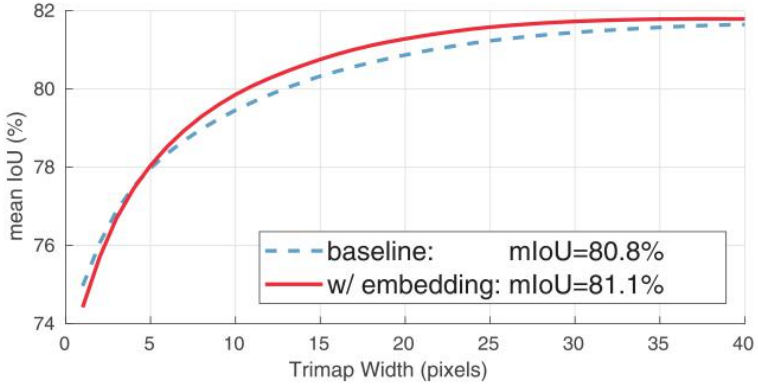


Figure 9: Semantic segmentaiton performance as a function of distance from ground-truth object boundaries comparing a baseline model trained with cross-entropy loss versus a model which also includes embedding loss.

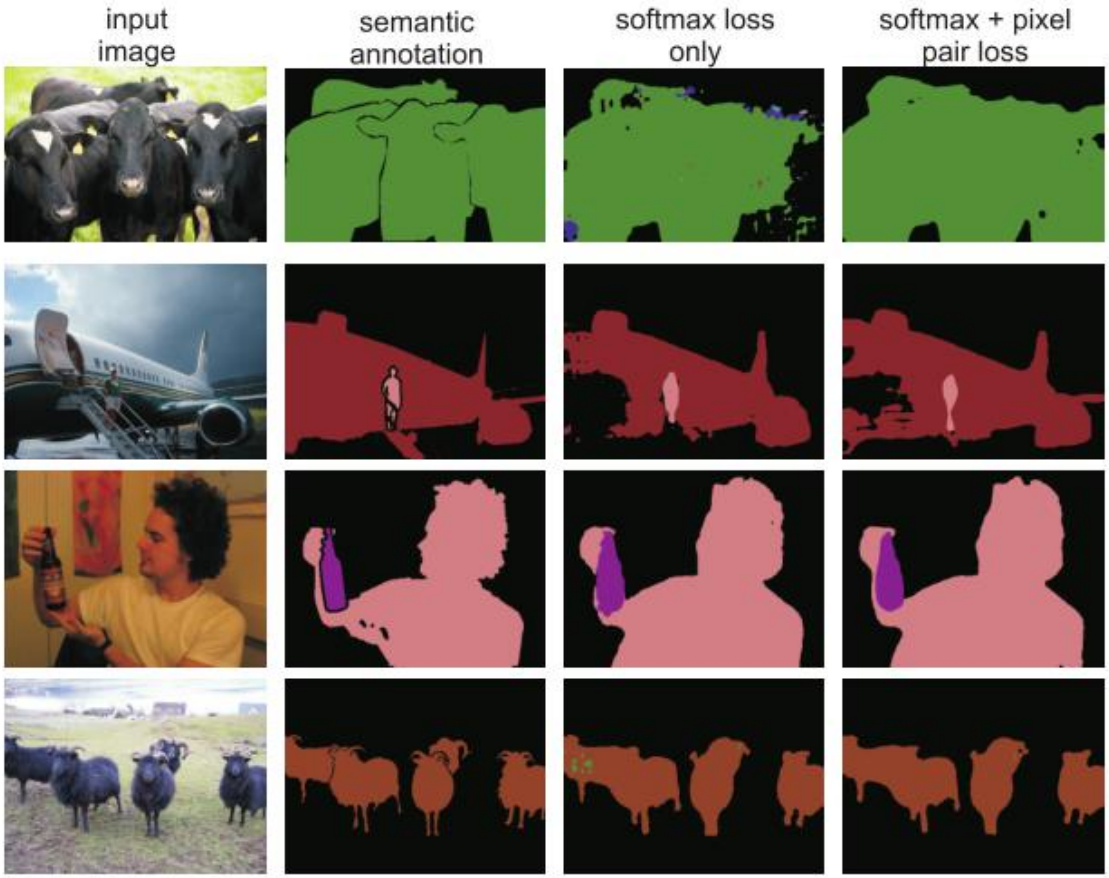


Table 2: Semantic Segmentation on PASCAL VOC 2012 validation dataset, measured by mIoU.

	w/o pixel pair loss	w/ pixel pair loss
mIoU	80.8	81.1

Experiment: Semantic Instance Segmentation

Instance-level semantic segmentation

Using the semantic segmentation result to vote for the semantic label within each object proposal

Experiment: Semantic Instance Segmentation

Instance-level semantic segmentation

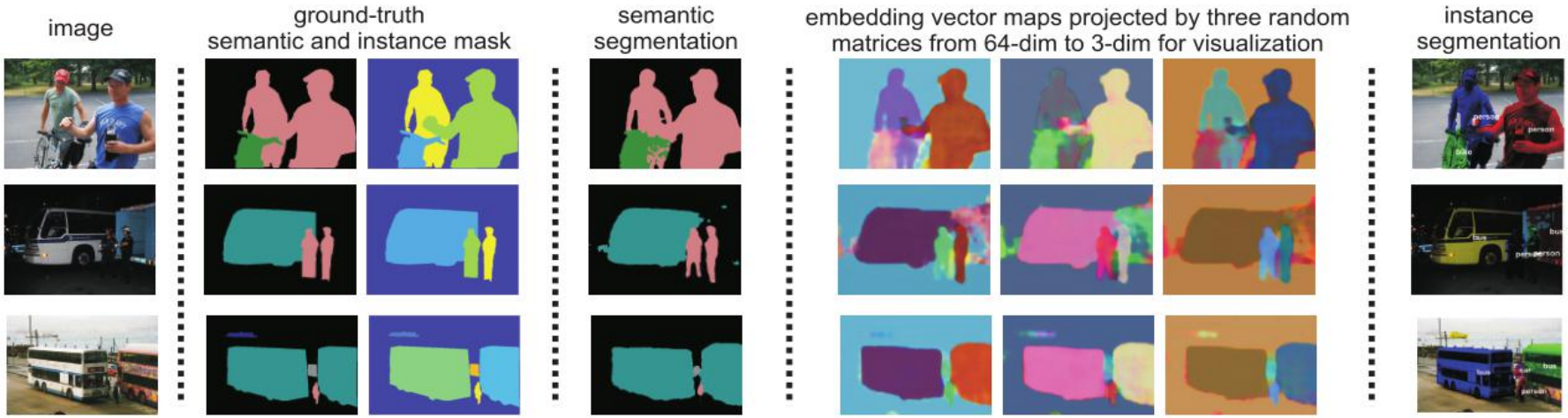


Figure 11: Visualization of generic/instance-level semantic segmentation result on random images from PASCAL VOC 2012 validation set.

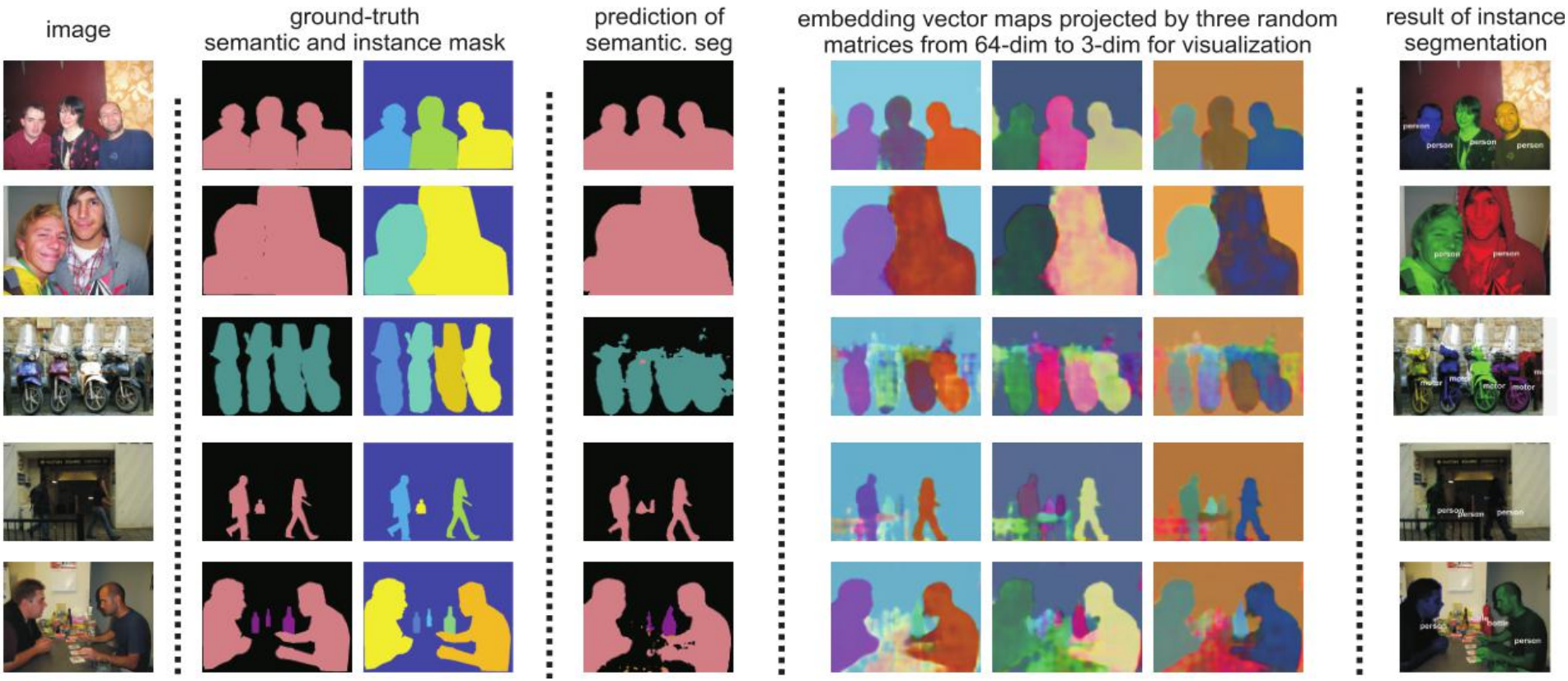
Method	plane	bike	bird	boat	bottle	bus	car	cat	chair	cow	table	dog	horse	motor	person	plant	sheep	sofa	train	tv	mean
SDS [31]	58.8	0.5	60.1	34.4	29.5	60.6	40.0	73.6	6.5	52.4	31.7	62.0	49.1	45.6	47.9	22.6	43.5	26.9	66.2	66.1	43.8
Chen et al. [12]	63.6	0.3	61.5	43.9	33.8	67.3	46.9	74.4	8.6	52.3	31.3	63.5	48.8	47.9	48.3	26.3	40.1	33.5	66.7	67.8	46.3
PFN [54]	76.4	15.6	74.2	54.1	26.3	73.8	31.4	92.1	17.4	73.7	48.1	82.2	81.7	72.0	48.4	23.7	57.7	64.4	88.9	72.3	58.7
MNC [17]	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	63.5
Li et al. [52]	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	65.7
R2-IOS [53]	87.0	6.1	90.3	67.9	48.4	86.2	68.3	90.3	24.5	84.2	29.6	91.0	71.2	79.9	60.4	42.4	67.4	61.7	94.3	82.1	66.7
Assoc. Embed. [64]	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	35.1
inst-DML [23]	69.7	1.2	78.2	53.8	42.2	80.1	57.4	88.8	16.0	73.2	57.9	88.4	78.9	80.0	68.0	28.0	61.5	61.3	87.5	70.4	62.1
Ours	85.9	10.0	74.3	54.6	43.7	81.3	64.1	86.1	17.5	77.5	57.0	89.2	77.8	83.7	67.9	31.2	62.5	63.3	88.6	74.2	64.5

Table 2: Instance-level segmentation comparison using APr metric at 0.5 IoU on the PASCAL VOC 2012 validation set.

Experiment: Semantic Instance Segmentation

Instance-level semantic segmentation

Using the semantic segmentation result to vote for the semantic label within each object proposal



Experiment: Semantic Instance Segmentation

Quanlitatively

div8 vs div4

Conclusion and Extension

- The framework is architecture-wise agnostic, conceptually simple, computationally efficient, practically effective, and theoretically abundant;

Conclusion and Extension

- The framework is architecture-wise agnostic, conceptually simple, computationally efficient, practically effective, and theoretically abundant;
- it can be purposed for boundary detection, object proposal detection, generic and instance-level segmentation, spanning low-, mid- and high-level vision tasks.

Conclusion and Extension

- The framework is architecture-wise agnostic, conceptually simple, computationally efficient, practically effective, and theoretically abundant;
- it can be purposed for boundary detection, object proposal detection, generic and instance-level segmentation, spanning low-, mid- and high-level vision tasks.
- Experiments demonstrate that the new framework achieves state-of-the-art performance on all these tasks.

Reference

1. R. Fisher. Dispersion on a sphere. In Proceedings of the Royal Society of London A: Mathematical, Physical and Engineering Sciences, volume 217, pages 295–305. The Royal Society, 1953.
2. E. B. Saff and A. B. Kuijlaars. Distributing many points on a sphere. The mathematical intelligencer, 19(1):5–11, 1997
3. K. Fukunaga and L. Hostetler. The estimation of the gradient of a density function, with applications in pattern recogni- tion. IEEE Transactions on information theory, 21(1):32–40, 1975
4. V. A. Epanechnikov. Non-parametric estimation of a multi-variate probability density. Theory of Probability & Its Applications, 14(1):153–158, 1969

Thanks