Recurrent Pixel Embedding for Grouping

Shu Kong
CS, ICS, UCI
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.
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:
  edge, boundary, contour

Mid-level vision:
  object proposal

High-level vision:
  semantic segmentation
  instance-level semantic segmentation

logistic loss
logistic loss for score
regression for location
logistic loss for mask&score
cross-entropy for category
A new framework consisting of two novel modules --
This framework is agnostic to architecture, so ignore deep learning for now!
A new framework consisting of two novel modules --

1. pixel-pair spherical max-margin regression

2. recurrent mean shift grouping
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
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
date back to Fisher Linear discriminant analysis (LDA)
date back to Fisher Linear discriminant analysis (LDA)

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

\[
J(w) = \frac{w^T S_B w}{w^T S_W w}
\]

where

\[
S_B = \sum_c (\mu_c - \bar{x})(\mu_c - \bar{x})^T
\]

\[
S_W = \sum_c \sum_{i \in c} (x_i - \mu_c)(x_i - \mu_c)^T
\]
What loss functions can we use at pixel-level?
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.
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.
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.

.....
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.
Why cosine similarity?


Why cosine similarity?

1. scale-invariant to the length of feature vector;
Why cosine similarity?

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

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

---


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;

---


We use the calibrated cosine similarity as below

\[ s_{ij} = \frac{1}{2} + \frac{x_i^T x_j}{2 \| x_i \|_2 \| x_j \|_2} \]
We use the calibrated cosine similarity as below

\[ s_{ij} = \frac{1}{2} + \frac{x_i^T x_j}{2 \| x_i \|_2 \| x_j \|_2} \]

loss function contains postive and negative pairs

\[ l = \frac{1}{|S|} \sum_{i,j \in S} w_{ij} \left( 1_{\{y_i = y_j\}} (1 - s_{ij}) + 1_{\{y_i \neq y_j\}} [s_{ij} - \alpha]_+ \right) \]
We use the calibrated cosine similarity as below

$$s_{ij} = \frac{1}{2} + \frac{x_i^T x_j}{2 \|x_i\|_2 \|x_j\|_2}$$

loss function contains positive and negative pairs

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

alpha is the margin, hyper parameter to be set.
We use the calibrated cosine similarity as below

\[ s_{ij} = \frac{1}{2} + \frac{x_i^T x_j}{2 \| x_i \|_2 \| x_j \|_2} \]

loss function contains postive and negative pairs

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

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.
Important theories

**Theorem 1** For $n$ vectors $\{x_1, \ldots, x_n\}$ on a hyper-sphere, i.e. $\|x_i\|_2 = 1$, $\forall i = 1 \ldots n$, $\sum_{i \neq j} x_i^T x_j$ has lower bound as $-n$, i.e. $\sum_{i \neq j} x_i^T x_j \geq -n$.

1. the loss has a lower bound, minimum;
2. the lower bound does not depend on the dimension of the embedding space.
2D case

**Theorem 2** For $n$ vectors $\{x_1, \ldots, x_n\}$ on a circle, i.e. $\|x_i\|_2 = 1, \forall i = 1 \ldots n$, when lower bound is achieved such that $\sum_{i \neq j} x_i^T x_j = -n$, then the minimum angle of any two vectors is $\arccos \frac{2\pi}{n}$. 
Lemma 4 For $n$ vectors $\{x_1, \ldots, x_n\}$ on a sphere (3D), i.e. $\|x_i\|_2 = 1, \forall i = 1 \ldots n$, when lower bound is achieved such that $\sum_{i \neq j} x_i^T x_j = -n$, then the minimum angle $\theta$ of any two vectors is in the range for some constant $C > 0$, i.e. $\cos \theta = \min_{i,j}(x_i^T x_j)$:

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

Forget about it --

It is one of the mathematical challenges of the 21st century.
high-dimensional case

Forget about it --

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

\[ V_n(R) \text{ and } S_n(R) \text{ are the } n\text{-dimensional volume of the } n\text{-ball and the surface area of the } n\text{-sphere embedded in dimension } n+1, \text{ respectively, of radius } R. \]

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

\[
V_0 = 1 \quad V_{n+1} = S_n / (n + 1) \\
S_0 = 2 \quad S_{n+1} = 2\pi V_n
\]

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

\[
S_n(R) = \frac{2\pi^{(n+1)/2}}{\Gamma\left(\frac{n+1}{2}\right)} R^n \\
V_n(R) = \frac{\pi^{n/2}}{\Gamma\left(\frac{n}{2} + 1\right)} R^n
\]

where \( \Gamma \) is the gamma function. Derivations of these equations are given in this section.
Lemma 2 Suppose there are $n$ instances in the image to separate, and specifically, the $i^{th}$ instance has $|I_i|$ pixels, of which the $j^{th}$ pixel is indexed by $x_{ij}$; then
\[ \sum_{m \neq k,p,q} x_m^T x_{kp} x_{kq} \geq - \sum_{i=1}^{n} |I_i|^2. \]
Lemma 3 Suppose there are \( n \) instances in the image to separate, and specifically, the \( i^{th} \) instance has \( |\mathcal{I}_i| \) pixels, of which the \( j^{th} \) pixel is indexed by \( x_{ij} \). We would like to weight the cross instance similarity loss to avoid large instance bias. Then, 

\[
\sum_{m \neq k, p, q} \frac{1}{|\mathcal{I}_m|} \frac{1}{|\mathcal{I}_k|} x_{mp}^T x_{kq} \geq -n.
\]
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
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
Consider a dataset \( \{x_n\}^N_{n=1} \subset \mathbb{R}^D \) and define a kernel density estimate

\[
\hat{f}(x) = \frac{1}{nh^d} \sum_{i=1}^{n} K\left(\frac{x - x_i}{h}\right).
\]

**mode seeking**

Fig. 1. Gradient mode clustering.
Recurrent Mean Shift Grouping

Consider a dataset \( \{x_n\}_{n=1}^{N} \subset \mathbb{R}^D \) and define a kernel density estimate

\[
\hat{f}(x) = \frac{1}{nh^d} \sum_{i=1}^{n} K\left(\frac{x - x_i}{h}\right).
\]

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

\[
\hat{\nabla} f_{h,K}(x) \equiv \nabla \hat{f}_{h,K}(x) = \frac{2c_{k,d}}{nh^{d+2}} \sum_{i=1}^{n} (x - x_i)k'\left(\frac{\|x - x_i\|^2}{h}\right).
\]
Recurrent Mean Shift Grouping

mean shift

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

We define the function

\[ g(x) = -k'(x), \]

then

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

\[ \sum_{i=1}^{n} g \left( \left\| \frac{x - x_i}{h} \right\|^2 \right) \] is assumed to be a positive number.
mean shift: iteratively updating by shifting the data by such an amount

The second term is the mean shift

\[ m_{h,G}(x) = \frac{\sum_{i=1}^{n} x_i g\left( \frac{x - x_i}{h} \right)}{\sum_{i=1}^{n} g\left( \frac{x - x_i}{h} \right)} - x, \]
Recurrent Mean Shift Grouping

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

The second term is the *mean shift*

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

\[
\mathbf{Y} = \mathbf{X} + \eta \mathbf{M}
\]

*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

\[
m_{h,G}(x) = \frac{\sum_{i=1}^{n} x_i g\left(\frac{\|x-x_i\|^2}{h}\right)}{\sum_{i=1}^{n} g\left(\frac{\|x-x_i\|^2}{h}\right)} - x,
\]

Gaussian blurring mean-shift (GBMS) algorithm

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

```
repeat
  for m ∈ {1, . . . , N}  
    \forall n: p(n|x_m) ← \frac{e^{-\frac{1}{2} \left\| \frac{x_m-x_n}{\sigma} \right\|^2}}{\sum_{n'=1}^{N} e^{-\frac{1}{2} \left\| \frac{x_m-x_{n'}}{\sigma} \right\|^2}}  \quad \text{Eq. (3a)}

  y_m ← \sum_{n=1}^{N} p(n|x_m)x_n  \quad \text{One GMS step, eq. (3b)}

end
```

Iteration loop

For each data point

Repeat

for each data point

compute posterior probability

y_m ← sum of posterior probabilities times data points

end
Gaussian blurring mean-shift (GBMS) algorithm

B. Gaussian blurring mean-shift (GBMS) algorithm

\[
\text{repeat} \quad \text{Iteration loop}
\]
\[
\text{for } m \in \{1, \ldots, N\} \quad \text{For each data point}
\]
\[
\forall n:\ p(n|x_m) \leftarrow \frac{e^{-\frac{1}{2} \left\| \frac{x_m - x_n}{\sigma} \right\|^2}}{\sum_{n' = 1}^{N} e^{-\frac{1}{2} \left\| \frac{x_m - x_{n'}}{\sigma} \right\|^2}} \text{ Eq. (3a)}
\]
\[
y_m \leftarrow \sum_{n=1}^{N} p(n|x_m)x_n \quad \text{One GMS step, eq. (3b)}
\]
\[
\forall m:\ x_m \leftarrow y_m \quad \text{Update whole dataset}
\]
\[\text{until stop} \quad \text{See section 1}
\]
\[\text{connected-components} \left( \{x_n\}_{n=1}^{N}, \text{min\_diff} \right) \quad \text{Clusters}
\]

C. GBMS algorithm in matrix form

\[
\text{repeat} \quad \text{Iteration loop}
\]
\[
W = \left( \exp \left( -\frac{1}{2} \left\| \frac{x_m - x_n}{\sigma} \right\|^2 \right) \right)_{nm} \quad \text{Gaussian affinity matrix}
\]
\[
D = \text{diag} \left( \sum_{n=1}^{N} w_{nm} \right) \quad \text{Degree (normalising) matrix}
\]
\[
X = XWD^{-1} \quad \text{Update whole dataset}
\]
\[\text{until stop} \quad \text{See section 1}
\]
\[\text{connected-components} \left( \{x_n\}_{n=1}^{N}, \text{min\_diff} \right) \quad \text{Clusters}
\]
**Recurrent Mean Shift Grouping**

Gaussian blurring mean-shift (GBMS) algorithm

**B. Gaussian blurring mean-shift (GBMS) algorithm**

\[ \text{repeat} \]
\[ \text{for } m \in \{1, \ldots, N\} \]
\[ \forall n: p(n|x_m) \leftarrow e^{-\frac{1}{2} \left\| \frac{x_m - x_n}{\sigma} \right\|^2} \frac{1}{\sum_{n' = 1}^{N} e^{-\frac{1}{2} \left\| \frac{x_m - x_{n'}}{\sigma} \right\|^2}} \text{ Eq. (3a)} \]
\[ y_m \leftarrow \sum_{n=1}^{N} p(n|x_m) x_n \text{ One GMS step, eq. (3b)} \]
\[ \forall m: x_m \leftarrow y_m \text{ Update whole dataset} \]
\[ \text{until stop} \]
\[ \text{connected-components(}\{x_n\}_{n=1}^{N}, \text{min_diff}) \text{ Clusters} \]

**C. GBMS algorithm in matrix form**

\[ \text{repeat} \]
\[ W = \left( \exp \left( -\frac{1}{2} \left\| \frac{x_m - x_n}{\sigma} \right\|^2 \right) \right)_{nm} \text{ Gaussian affinity matrix} \]
\[ D = \text{diag} \left( \sum_{n=1}^{N} w_{nm} \right) \text{ Degree (normalising) matrix} \]
\[ X = XWD^{-1} \text{ Update whole dataset} \]
\[ \text{until stop} \]
\[ \text{connected-components(}\{x_n\}_{n=1}^{N}, \text{min_diff}) \text{ Clusters} \]

It's guaranteed to converge without gradient vanishing or exploding.
Recurrent Mean Shift Grouping

Gaussian blurring mean-shift (GBMS) algorithm

B. Gaussian blurring mean-shift (GBMS) algorithm

\[
\text{repeat} \\
\quad \text{for } m \in \{1, \ldots, N\} \\
\quad \forall n: p(n|\mathbf{x}_m) \leftarrow \frac{e^{-\frac{1}{2} \left\| \frac{\mathbf{x}_m - \mathbf{x}_n}{\sigma} \right\|^2}}{\sum_{n'=1}^{N} e^{-\frac{1}{2} \left\| \frac{\mathbf{x}_m - \mathbf{x}_{n'}}{\sigma} \right\|^2}} \quad \text{Eq. (3a)} \\
\quad \mathbf{y}_m \leftarrow \sum_{n=1}^{N} p(n|\mathbf{x}_m) \mathbf{x}_n \quad \text{One GMS step, eq. (3b)} \\
\quad \forall m: \mathbf{x}_m \leftarrow \mathbf{y}_m \quad \text{Update whole dataset} \\
\text{until stop} \\
\text{connected-components}([\mathbf{x}_n]_{n=1}^N, \text{min}\_\text{diff}) \quad \text{Clusters}
\]

C. GBMS algorithm in matrix form

\[
\text{repeat} \\
\quad W = \left( \exp \left( -\frac{1}{2} \left\| \frac{\mathbf{x}_m - \mathbf{x}_n}{\sigma} \right\|^2 \right) \right)_{nm} \quad \text{Gaussian affinity matrix} \\
\quad D = \text{diag} \left( \sum_{n=1}^{N} w_{nm} \right) \quad \text{Degree (normalising) matrix} \\
\quad \mathbf{X} = \mathbf{X}WD^{-1} \quad \text{Update whole dataset} \\
\text{until stop} \\
\text{connected-components}([\mathbf{x}_n]_{n=1}^N, \text{min}\_\text{diff}) \quad \text{Clusters}
\]
Recurrent Mean Shift Grouping

L2 normalization in the loop

Algorithm 1: vMF Mean Shift

**[Normalization]** \(||x_i|| = 1, \forall i.\)**

```plaintext
for i = 1 to N do
  [Init] Start from the i-th sample: \(y_i^{(0)} = x_i.\)
  [Update] Update the target points (vectors) until convergence by the following formula:

\[
y_i^{(t+1)} \leftarrow \frac{\sum_j^N x_j g(x_j', y_i^{(t)}; \kappa)}{||\sum_j^N x_j g(x_j', y_i^{(t)}; \kappa)||}.
\] (6)
```

[Postprocessing] Merge close convergent points \((y_i^{(\infty)'}, y_j^{(\infty)}) > 1 - \epsilon\) 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
mean shift as recurrent module
Recurrent Mean Shift Grouping

mean shift as recurrent module

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

calculating the degree vector \( d = G^T 1 \)
elementwise inverse degree vector \( q = d^{-1} \)
diagonal matrix \( \text{diag}(q) \)
the mean shift for all batch \( X \) is \( M = XG\text{diag}(q) - X \)

\[
Y = X + \eta M
= X + \eta(XG\text{diag}(q) - X)
= X(\eta G\text{diag}(q) + (1 - \eta)I)
\]
mean shift as recurrent module

\[ S = X^T X \]

\[ G = \exp(\delta^2 (S - 1)) \]

\[ d = G^T 1 \]

\[ q = d^{-1} \]

\[ P = G \text{diag}(q) \text{, or } P = (1 - \eta)I + \eta G \text{diag}(q) \]

\[ Y = XP \]

Figure 2: Mean shift grouping module.
Recurrent Mean Shift Grouping

mean shift grouping in the loop

\[ S = X^T X \]

\[ G = \exp(\delta^2 (S - 1)) \]
\[ d = G^T 1 \]
\[ q = d^{-1} \]

\[ P = G \text{diag}(q), \text{ or } P = (1 - \eta)I + \eta G \text{diag}(q) \]

\[ Y = XP \]

Figure 2: Mean shift grouping module.

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

What does it mean by mean shift gradient?
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.
architecture agnostic -- we use ResNet
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).
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
object proposal detection

class-agnostic

reduce search space for subsequential tasks, e.g. object detection

the proposal framework is particularly suitable for this tasks

How suitable?
object proposal detection -- How suitable?

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

Quantitatively: Ours vs. SharpMask
Experiment: Semantic Segmentation

semantic segmentation with cross-entropy loss

The pixel pair loss can fill in the “holes”.

Figure 9: Semantic segmentation 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.

<table>
<thead>
<tr>
<th></th>
<th>w/o pixel pair loss</th>
<th>w/ pixel pair loss</th>
</tr>
</thead>
<tbody>
<tr>
<td>mIoU</td>
<td>80.8</td>
<td>81.1</td>
</tr>
</tbody>
</table>
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.

<table>
<thead>
<tr>
<th>Method</th>
<th>plane</th>
<th>bike</th>
<th>bird</th>
<th>boot</th>
<th>bottle</th>
<th>bus</th>
<th>car</th>
<th>cat</th>
<th>chair</th>
<th>cow</th>
<th>table</th>
<th>dog</th>
<th>horse</th>
<th>motor</th>
<th>person</th>
<th>plant</th>
<th>sheep</th>
<th>sofa</th>
<th>train</th>
<th>tv</th>
<th>mean</th>
</tr>
</thead>
<tbody>
<tr>
<td>SDS [31]</td>
<td>58.8</td>
<td>0.5</td>
<td>60.1</td>
<td>34.4</td>
<td>29.5</td>
<td>60.6</td>
<td>40.0</td>
<td>73.6</td>
<td>6.5</td>
<td>52.4</td>
<td>31.7</td>
<td>62.0</td>
<td>49.1</td>
<td>45.6</td>
<td>47.9</td>
<td>22.6</td>
<td>43.5</td>
<td>26.9</td>
<td>66.2</td>
<td>66.1</td>
<td>43.8</td>
</tr>
<tr>
<td>Chen et al. [12]</td>
<td>63.6</td>
<td>0.3</td>
<td>61.5</td>
<td>43.9</td>
<td>33.8</td>
<td>67.3</td>
<td>46.9</td>
<td>74.4</td>
<td>8.6</td>
<td>52.3</td>
<td>31.3</td>
<td>63.5</td>
<td>48.8</td>
<td>47.9</td>
<td>48.3</td>
<td>26.3</td>
<td>40.1</td>
<td>33.5</td>
<td>66.7</td>
<td>67.8</td>
<td>46.3</td>
</tr>
<tr>
<td>PFN [54]</td>
<td>76.4</td>
<td>15.6</td>
<td>74.2</td>
<td>54.1</td>
<td>26.3</td>
<td>73.8</td>
<td>91.4</td>
<td>92.1</td>
<td>17.4</td>
<td>73.7</td>
<td>48.1</td>
<td>82.2</td>
<td>81.7</td>
<td>72.0</td>
<td>48.4</td>
<td>23.7</td>
<td>57.7</td>
<td>64.4</td>
<td>88.9</td>
<td>72.3</td>
<td>58.7</td>
</tr>
<tr>
<td>MNC [17]</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>63.5</td>
</tr>
<tr>
<td>Li et al. [52]</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>66.7</td>
</tr>
<tr>
<td>R2-IOS [53]</td>
<td>87.0</td>
<td>6.1</td>
<td>90.3</td>
<td>67.9</td>
<td>48.4</td>
<td>86.2</td>
<td>68.3</td>
<td>90.3</td>
<td>24.5</td>
<td>84.2</td>
<td>29.6</td>
<td>91.0</td>
<td>71.2</td>
<td>79.9</td>
<td>60.4</td>
<td>42.4</td>
<td>67.4</td>
<td>61.7</td>
<td>94.3</td>
<td>82.1</td>
<td>65.7</td>
</tr>
<tr>
<td>Assoc. Embed. [64]</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>35.1</td>
</tr>
<tr>
<td>inst-DML [23]</td>
<td>69.7</td>
<td>1.2</td>
<td>78.2</td>
<td>53.8</td>
<td>42.2</td>
<td>80.1</td>
<td>57.4</td>
<td>88.8</td>
<td>16.0</td>
<td>73.2</td>
<td>57.9</td>
<td>88.4</td>
<td>78.9</td>
<td>80.0</td>
<td>68.0</td>
<td>28.0</td>
<td>61.5</td>
<td>61.3</td>
<td>87.5</td>
<td>70.4</td>
<td>62.1</td>
</tr>
<tr>
<td>Ours</td>
<td>85.9</td>
<td>10.0</td>
<td>74.3</td>
<td>54.6</td>
<td>43.7</td>
<td>81.3</td>
<td>64.1</td>
<td>86.1</td>
<td>17.5</td>
<td>77.5</td>
<td>57.0</td>
<td>89.2</td>
<td>77.8</td>
<td>83.7</td>
<td>67.9</td>
<td>31.2</td>
<td>62.5</td>
<td>63.3</td>
<td>88.6</td>
<td>74.2</td>
<td>64.5</td>
</tr>
</tbody>
</table>

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

Quantitatively

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.
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.


Thanks