Instance Similarity Learning for Unsupervised Feature Representation

Ziwei Wang[1,2,3], Yunsong Wang[1], Ziyi Wu[1], Jiwen Lu[1,2,3]*, Jie Zhou[1,2,3]
[1] Department of Automation, Tsinghua University, China
[2] State Key Lab of Intelligent Technologies and Systems, China
[3] Beijing National Research Center for Information Science and Technology, China

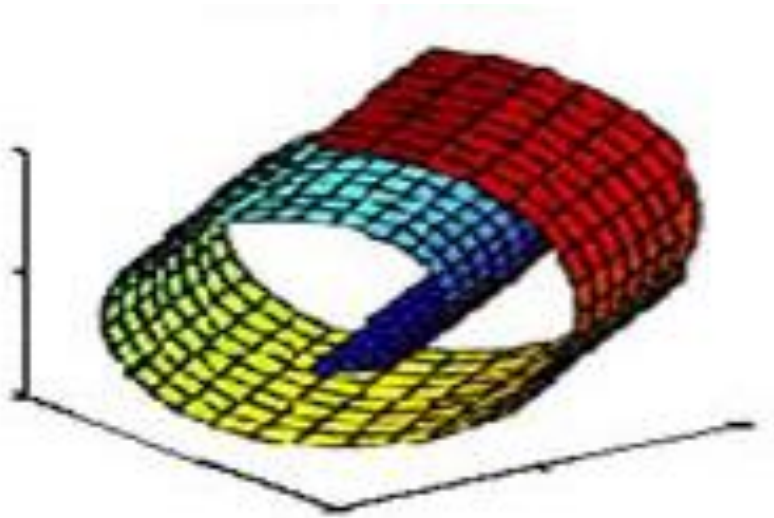
ICCV 2021

Background

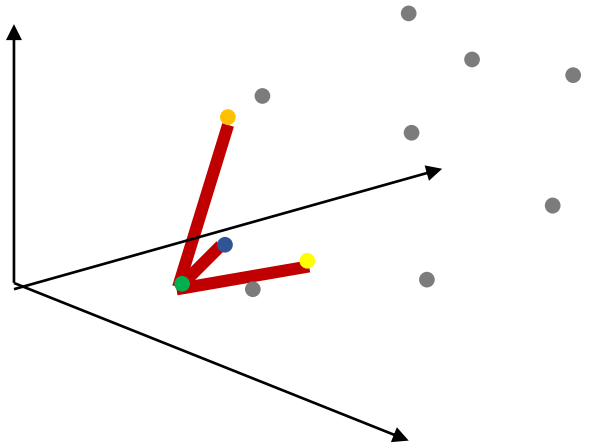
- unsupervised representation learning
  - Find the feature representation with unsupervised method
  - To provide information for downstream tasks (i.e. classification, object detection)
- contrastive learning
  - for an anchor: positive samples + negative samples
  - loss is about similaritis in feature space
  - difficulty: choice of positive/negative samples

Background

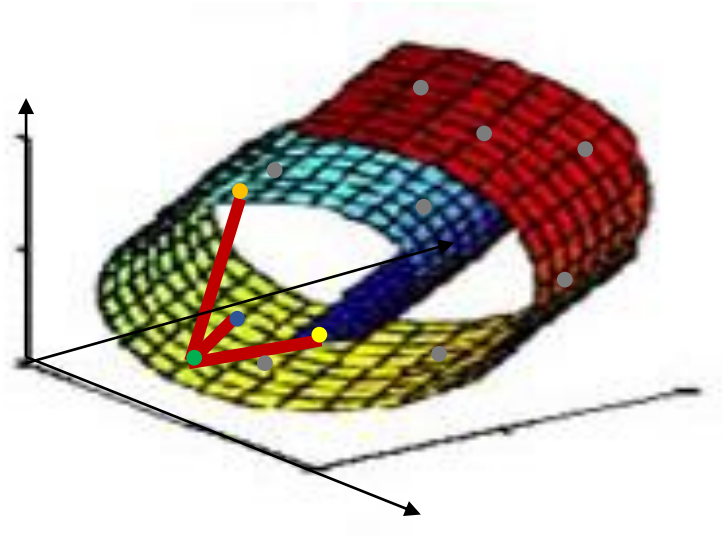
- manifold
  - 流形，这里我们简单理解为一个曲面，如球面是一个流形
  - 流形上的距离：测地距离，即在曲面上最短的距离。它的每个小邻域内的距离可看作欧式距离，但大区域内欧式距离不再适用



Motivation

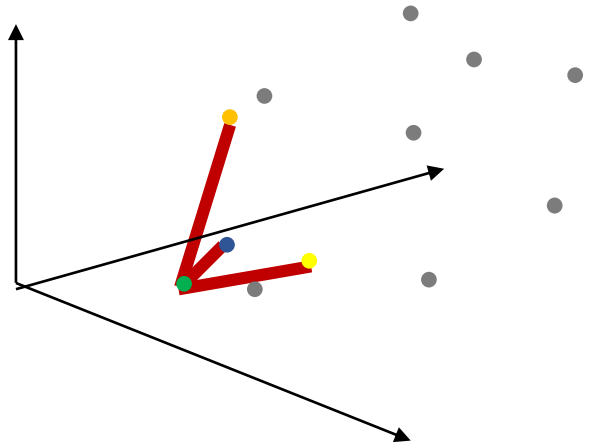


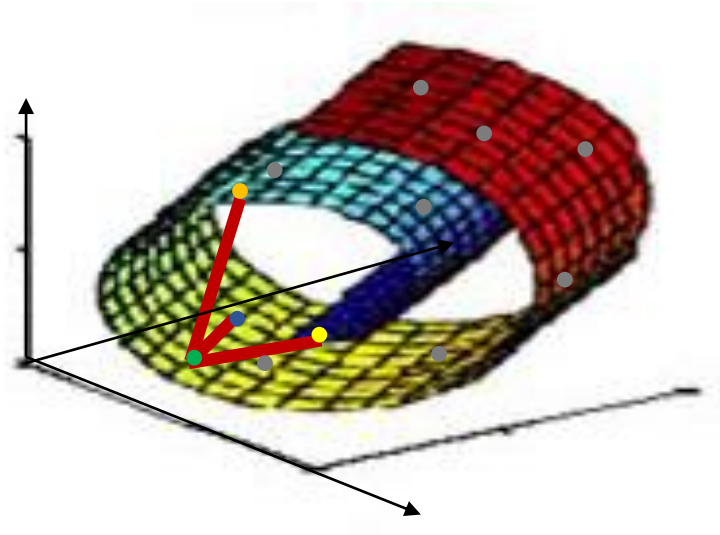
feature representation in R^3



features may on a manifold

Motivation





- the Euclidean distance in implicit feature manifold:
  - only reveals the true semantic similarity in **extremely small neighborhoods**
  - fails to provide the informative pseudo supervision for **large neighborhoods**

- try to:
  - mines the feature manifold in an unsupervised manner
  - utilizes a small neighborhood on the manifold to find positive samples

Method

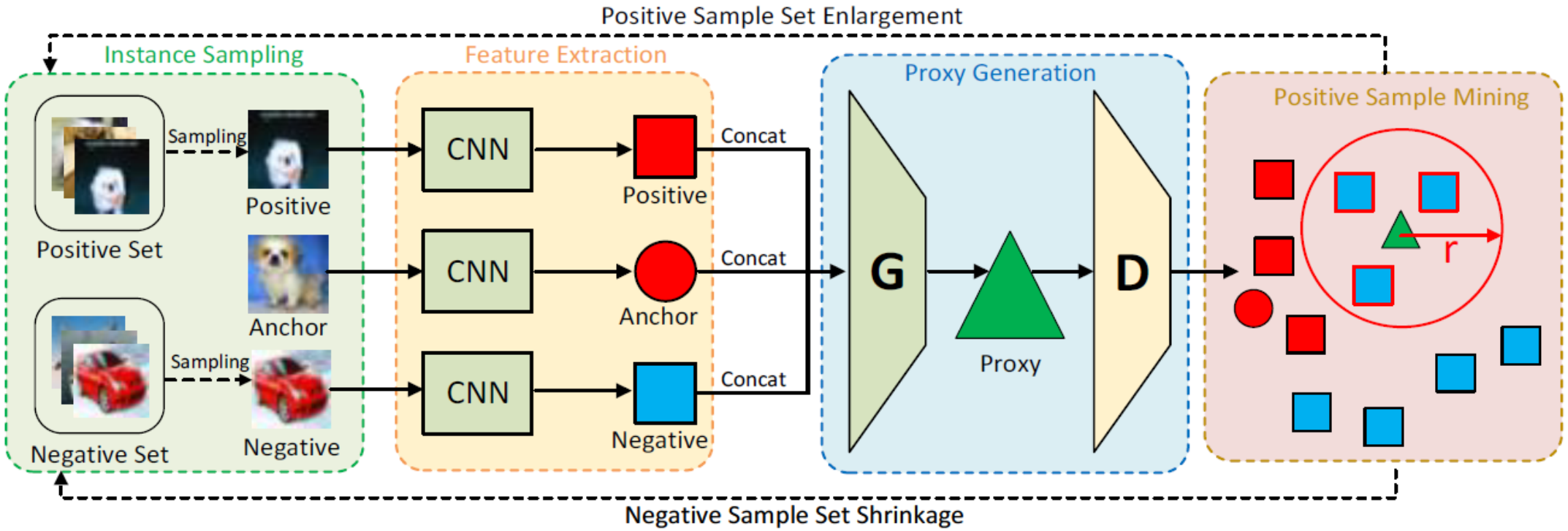


Figure 2. The pipeline of the instance similarity learning. For a given anchor, we first sample triplets from the mined positive set and the negative set, and then obtain the features via the convolutional neural networks. After concatenating the features of the anchor, the positive and the negative samples, we generate the proxy for feature manifold mining by the generator. The instances in the neighborhood of the proxy are removed from the negative set and added to the positive set if the proxy is semantically similar to the anchor, where the semantics similarity is predicted by the discriminator.

Method-Training of GAN

- i_{th} image as the anchor: feature triplet $\mathcal{T}_r = \{f_i, f_i^p, f_i^n\}$
- generator:
  - to give a proxy feature f_i^g of feature triplet $\mathcal{T}_r$ on the manifold
  - input: $\mathcal{T}_r$
  - output: proxy feature f_i^g
  - synthetic triplet: $\mathcal{T}_s^p = \{f_i, f_i^g, f_i^n\}$, $\mathcal{T}_s^n = \{f_i, f_i^p, f_i^g\}$
- discriminator:
  - to accurately classify the real triplet $\mathcal{T}_r$ and synthetic triplet $\mathcal{T}_s^p$, $\mathcal{T}_s^n$
  - input: $\mathcal{T}_r$/ $\mathcal{T}_s^p$/ $\mathcal{T}_s^n$
  - output: $D(\mathcal{T}_r)$(confidence score that the input triplet $\mathcal{T}_r$ is real)/$D(\mathcal{T}_s^p)$/ $D(\mathcal{T}_s^n)$

$$\min_G \max_D \mathcal{L}_{gan} = \log D(\mathcal{T}_r) + \log(1 - D(\mathcal{T}_s^p)) + \alpha \log(1 - D(\mathcal{T}_s^n))$$

to measure the semantic similarity between the proxy and the positives or negatives

Method-Training of GAN

- i_{th} image as the anchor: feature triplet $\mathcal{T}_r = \{f_i, f_i^p, f_i^n\}$
- generator:
  - to give a proxy representation f_i^g of anchor on the manifold
  - input: $\mathcal{T}_r$
  - output: proxy feature f_i^g
  - synthetic triplet: $\mathcal{T}_s^p = \{f_i, f_i^g, f_i^n\}$, $\mathcal{T}_s^n = \{f_i, f_i^p, f_i^g\}$
- discriminator:
  - to measure the semantic similarity between the proxy and positives/negatives
  - input: $\mathcal{T}_r$/ $\mathcal{T}_s^p$ / $\mathcal{T}_s^n$
  - output: $D(\mathcal{T}_r)$(confidence score that the input triplet $\mathcal{T}_r$ is real)/$D(\mathcal{T}_s^p)$/ $D(\mathcal{T}_s^n)$

$$\max_D \mathcal{L}_{gan} = \log D(\mathcal{T}_r) + \log(1 - D(\mathcal{T}_s^p)) + \alpha \log(1 - D(\mathcal{T}_s^n))$$

$D(\mathcal{T}_r) \uparrow$ $D(\mathcal{T}_s^n) \downarrow$ $D(\mathcal{T}_s^p) \downarrow$
accurately classify the real triplet $\mathcal{T}_r$ from $\mathcal{T}_s^n$ and $\mathcal{T}_s^p$

Method-Training of GAN

- i_{th} image as the anchor: feature triplet $\mathcal{T}_r = \{f_i, f_i^p, f_i^n\}$
- generator:
  - to give a proxy representation f_i^g of anchor on the manifold
  - input: $\mathcal{T}_r$
  - output: proxy feature f_i^g
  - synthetic triplet: $\mathcal{T}_s^p = \{f_i, f_i^g, f_i^n\}$, $\mathcal{T}_s^n = \{f_i, f_i^p, f_i^g\}$
- discriminator:
  - to measure the semantic similarity between the proxy and positives/negatives
  - input: $\mathcal{T}_r$/ $\mathcal{T}_s^p$/ $\mathcal{T}_s^n$
  - output: $D(\mathcal{T}_r)$(confidence score that the input triplet $\mathcal{T}_r$ is real)/$D(\mathcal{T}_s^p)$/ $D(\mathcal{T}_s^n)$

$$\min_G \quad \mathcal{L}_{gan} = \log(1 - D(\mathcal{T}_s^p)) + \alpha \log(1 - D(\mathcal{T}_s^n))$$

$D(\mathcal{T}_s^p) \uparrow$ $D(\mathcal{T}_s^n) \uparrow$
similar to f_i^n: enables active feature manifold exploration
similar to f_i^p: to mine more positive sets

set $\alpha = 1$

Method-Training of Feature Extractor

- for an anchor f_i , the optimal proxy $\boldsymbol{f}_i^{g*} = \arg\max_{\boldsymbol{f}_i^g} D(\mathcal{T}_s^p)$
- enlarge the positive sample set $\mathcal{P}_i$:

$$\boldsymbol{f}_j = \left\{ \boldsymbol{f}_j \mid \left\| \boldsymbol{f}_i^{g*} - \boldsymbol{f}_j \right\|_F < r, D(\mathcal{T}_s^P) > h \right\}$$

- $p_{ij} = \dfrac{\exp(\boldsymbol{f}_i^T \boldsymbol{f}_j / \tau)}{\sum_{k=1}^{N} \exp(\boldsymbol{f}_i^T \boldsymbol{f}_k / \tau)}$

- $\mathcal{L}_1 = -\sum_{i=1}^{N} \log\left(\sum_{\boldsymbol{f}_k \in \mathcal{P}_i} p_{ik} \right)$ to encourage the similarity between the anchor and all of its positive samples

Method-Training of Feature Extractor

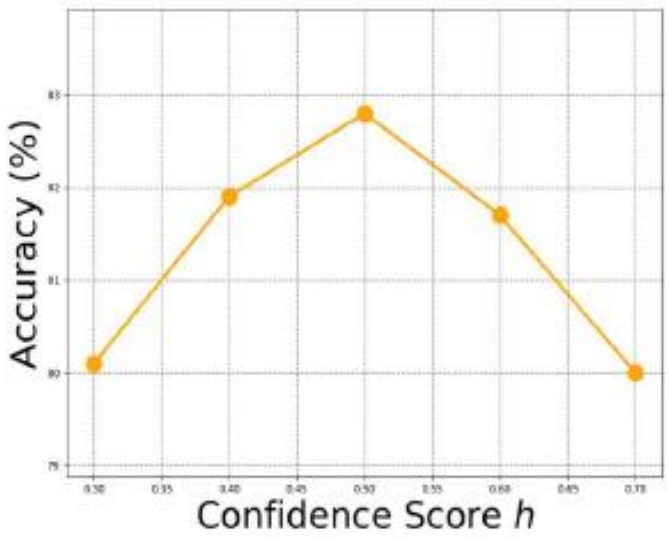
- hard positive $\boldsymbol{f}_i^{hard} = \min_{f_k \in \mathcal{P}_i} p_{ik}$
- p_{ik}^{hard}: similarity between $\boldsymbol{f}_i^{hard}$ and $\boldsymbol{f}_k$
- $\mathcal{L}_2 = \sum_{i=1}^{N} \sum_{k=1}^{N} p_{ik} \log \frac{p_{ik}}{p_{ik}^{hard}}$ to enhance the performance of hard positive
- $\mathcal{L} = \mathcal{L}_1 + \lambda \mathcal{L}_2$
- use memory bank in MoCo method to reduce computational cost

  $\hat{\boldsymbol{f}}_i = \eta \boldsymbol{f}_i + (1 - \eta) \hat{\boldsymbol{f}}_i$

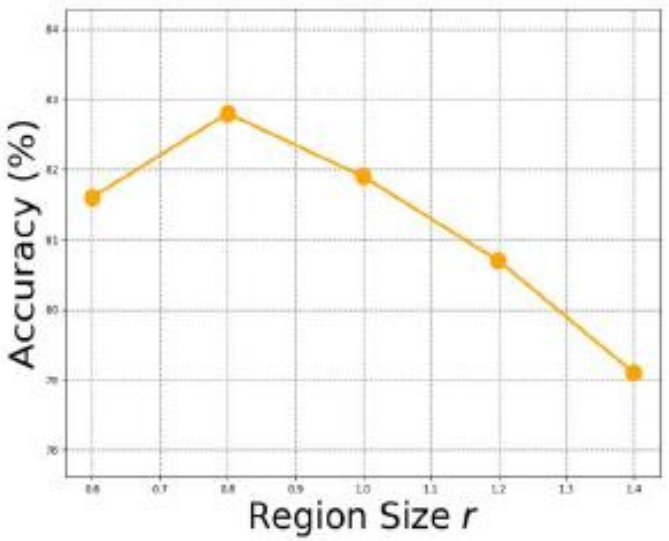
Experiment

- sample 5 triplets for a given anchor
- iterative training GAN and feature extractor for 4 rounds
- backbone of feature extractor: AlexNet, ResNet18 and ResNet50
- test method: accuracy of classification on CIFAR-10/CIFAR-100/SVHN/ImageNet
- classification method: Linear Classifier or weighted KNN

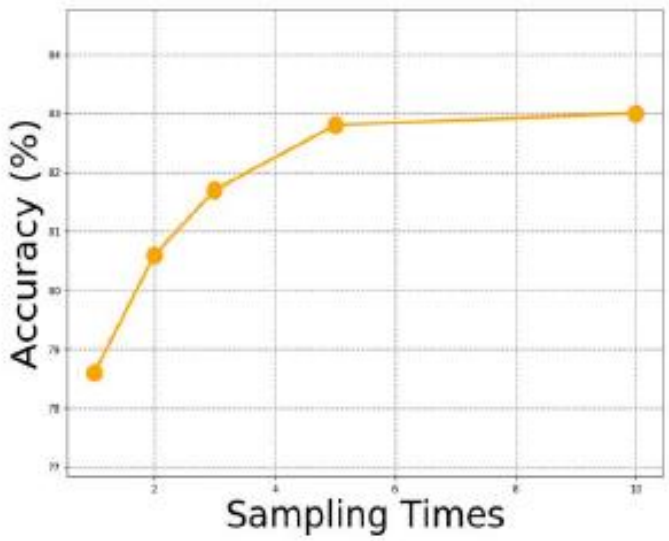
Experiment



(a)



(b)



(c)

Experiment

Architecture	Dataset / Classifier/Feat.	CIFAR10	CIFAR100	SVHN
		Weighted kNN / FC		
AlexNet	Random	34.5	12.1	56.8
	DeepCluster	62.3	22.7	84.9
	RotNet	72.5	32.1	77.5
	Instance	60.3	32.7	79.8
	AND	74.8	41.5	90.9
	ISL w/o HPE	**81.1**	**49.2**	**91.0**
	PAD	81.5	48.7	91.2
	ISL	**82.8**	**50.3**	**91.8**
ResNet18	Instance	80.8	40.1	92.6
	AND	86.3	48.1	93.1
	ISL w/o HPE	**87.0**	**52.1**	**93.9**
	ISL	**87.8**	**54.7**	**94.2**
ResNet50	Instance	81.8	42.3	92.9
	AND	87.6	49.0	93.2
	ISL w/o HPE	**88.3**	**56.7**	**94.0**
	ISL	**88.9**	**58.1**	**94.5**

Architecture	Classifier/Feat.	Linear Classifier / conv5		
AlexNet	Random	67.3	32.7	79.2
	DeepCluster	77.9	41.9	92.0
	RotNet	**84.1**	57.4	92.3
	Instance	70.1	39.4	89.3
	AND	77.6	47.9	**93.7**
	ISL w/o HPE	83.5	**58.5**	93.3
	PAD	84.7	58.6	93.2
	ISL	**85.8**	**60.1**	**93.9**
ResNet18	Instance	84.1	48.9	94.0
	AND	88.9	57.4	94.3
	ISL w/o HPE	**89.2**	**61.1**	**94.4**
	ISL	**90.7**	**63.5**	**94.5**
ResNet50	Instance	85.0	50.1	94.4
	AND	90.2	58.5	94.9
	ISL w/o HPE	**91.0**	**63.0**	**94.9**
	ISL	**91.5**	**65.9**	**95.2**

HPE: hard positive enhancement

Experiment

ImageNet

Classifier	Linear Classifier					kNN
Feature	conv1	conv2	conv3	conv4	conv5	FC
AlexNet						
Random	11.6	17.1	16.9	16.3	14.1	3.5
DeepCluster	13.4	32.3	**41.0**	39.6	38.2	26.8
RotNet	**18.8**	31.7	38.7	38.2	36.5	9.2
Instance	16.8	26.5	31.8	34.1	35.6	31.3
AND	15.6	27.0	35.9	39.7	37.9	31.3
PAD	-	-	-	-	38.6	35.1
LA	18.7	**32.7**	38.1	42.3	42.4	38.1
ISL	17.3	29.0	38.4	**43.3**	**43.5**	**38.9**
ResNet18						
DeepCluster	**16.4**	17.2	28.7	44.3	49.1	–
Instance	16.0	**19.9**	29.8	39.0	44.5	41.0
LA	9.1	18.7	**34.8**	48.4	52.8	45.0
ISL	15.3	19.1	32.7	**49.1**	**54.0**	**46.1**
ResNet50						
DeepCluster	18.9	27.3	36.7	52.4	44.2	–
LA	10.2	23.3	39.3	49.0	60.2	49.4
ISL	**17.3**	24.2	38.5	52.5	61.2	**50.2**
MoCo-v1*	15.7	22.9	40.6	50.8	60.6	37.7
MoCo-v2*	14.9	**28.4**	41.7	**52.9**	67.5	38.5
MoCo-v2+ISL*	13.2	27.1	**41.9**	51.7	**68.6**	40.1

HPE: hard positive enhancement

Experiment

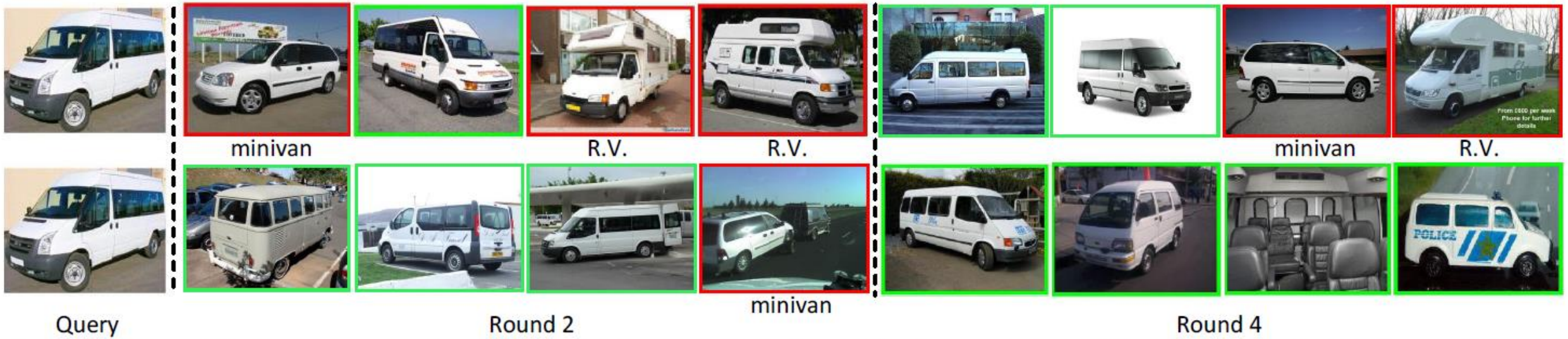


Figure 5. An example of positive sample mining via LA (top row) and our ISL (bottom row) in different rounds during training. The query image is from the minibus class. The images with green boxes represent the positives mined correctly and those with red boxes mean the images from other classes. More examples are visualized in the supplementary material.

Discussion

- 无需通过图像增强或预先任务来选择正样本
- 架构清晰，通过流形的概念来说明方法的优势
- 通过对特征空间的变换，来获得更多的信息（正样本的选择），思路值得借鉴

- 引入测地距离对比欧式距离来说明方法好，但并未对为什么自己方法得到的是测地距离有更多论证，有点直接套上的意味
- 流形这一概念也只是直接使用，实质上可以理解为在特征空间中找到一个新的表示，来提供更多的信息