HoughCL: Finding Better Positive Pairs in Dense Self-supervised Learning

Yunsung Lee[1,3], Teakgyu Hong[2], Han-Cheol Cho[2], Junbum Cha[2], Seungryong Kim[3]

Scatter Lab[1], Naver Clova AI Research[2], Korea University[3]

Correspondence to: Yunsung Lee {swack9751@korea.ac.kr}

Overview

Problem Statement

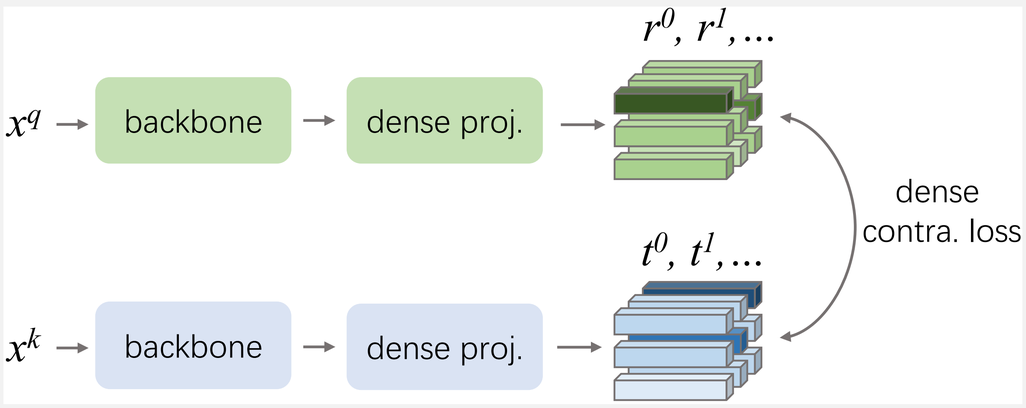
- Most of recent self-supervised visual representation learning only consider image-level embeddings lacking local information.
- Several recent works have learned representations from pixel-level densely embedded vectors.
- However, since pixel-level features are variant in augmentation, it is difficult to assign pixel-level positive pairs.

Our Contributions

- We introduce the pixel-level dense positive pairing method based on Hough geometric voting.
- Our method, Hough Contrastive Learning (HoughCL), can obtain better geometrical consistency in dense positive pairs.
- HoughCL does not require additional training parameters.
- Compared to previous works, our method shows better or comparable performance on dense prediction fine-tuning tasks.

Background

Dense Contrastive Learning (DenseCL)



The dense contrastive loss is defined as: (S^2 feature vectors in DenseCL)

$$\mathcal{L}_r = \frac{1}{S^2}\sum_s -\log\frac{\exp(r^s\cdot t^s_+/\tau)}{\exp(r^s\cdot t^s_+/\tau)+\sum_{t^s_-}\exp(r^s\cdot t^s_-/\tau)},$$

Dense Positive Pairs in DenseCL To obtain positive pixel pairs, they simply calculate the cosine similarity between pixel vectors and choose the positive pair which has the highest similarity value. This simple winner-takes-all method suffers from background clutter and outliers.

Method

Hough Contrastiv Learning (HoughCL) In our context, let $\mathcal{D} = (\mathcal{H}, \mathcal{H}')$ be two sets of dense projected features, and $m = (\mathbf{h}, \mathbf{h}')$ be a region vector match where $\mathbf{h}$ and $\mathbf{h}'$ are respectively elements of $\mathcal{H}$ and $\mathcal{H}'$. Given a Hough space $\mathcal{X}$ of possible offsets (image transformation) between the two dense projected features, the confidence for match m, $p(m|\mathcal{D})$, is computed as:

$$p(m|\mathcal{D}) \propto p(m_{\mathrm{a}})\sum_{\mathbf{x}\in\mathcal{X}} p(m_{\mathrm{g}}|\mathbf{x}) \sum_{m\in\mathcal{H}\times\mathcal{H}'} p(m_{\mathrm{a}})p(m_{\mathrm{g}}|\mathbf{x}),$$

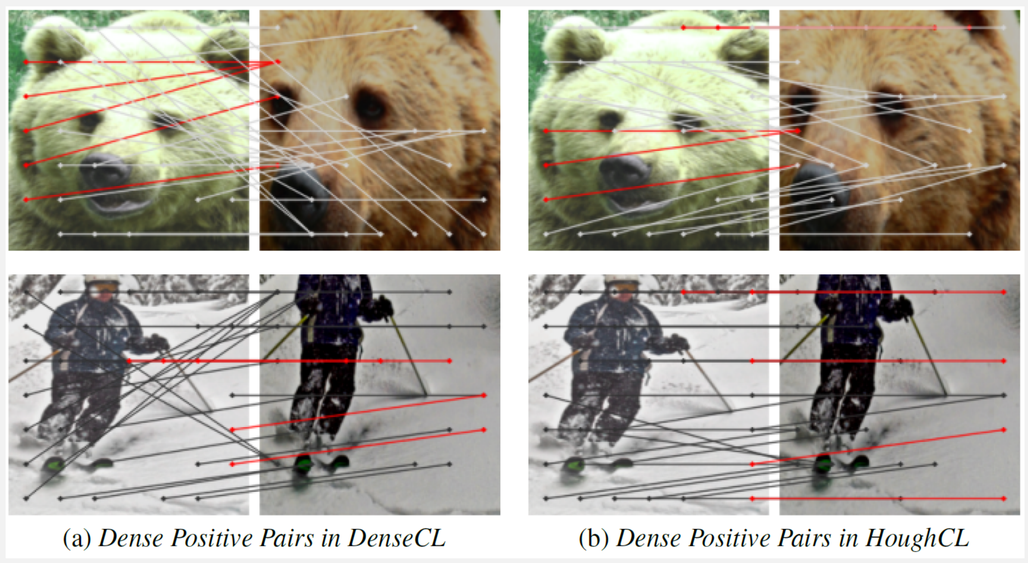
where $p(m_{\mathrm{a}})$ represents the confidence for similarity matching and $p(m_{\mathrm{g}}|\mathbf{x})$ is the confidence for geometric matching with an offset $\mathbf{x}$, measuring how close the offset induced by m is to $\mathbf{x}$, and implemented by a discretized Gaussian kernel centered on $\mathbf{x}$. By sharing the Hough space $\mathcal{X}$ for all matches, PHM efficiently computes the match confidence. Matching confidence is computed as the exponential cosine similarity, $p(m_{\mathrm{a}}) = \mathrm{ReLU}\left(\frac{\mathbf{f}\cdot\mathbf{f}'}{\|\mathbf{f}\|\|\mathbf{f}'\|}\right)^d$. The ReLU function clamps negative values to zero and the exponent $d \geq 2$ improves matching performance by suppressing noisy activations. We set $d = 3$ in our experiments.

Experiments - Pre-training

Pre-training Setup

- Architecture & Hyperparameters: Mostly follows DenseCL setup
- Datasets: TinyImageNet(200epoch), COCO(800epoch), ImageNet(200epoch)

Visualization of dense positive pairs in DenseCL and our HoughCL
Both methods are pre-trained 800 epochs on the COCO dataset and have a ResNet-50 backbone network. The dense positive pairs of HoughCL are geometrical consistent and robust against background clutter and outliers compared with DenseCL.
(The red line segments: 5 pairs with highest confidence, the gray line segments: 20 pairs with the lowest confidence.)



(a) *Dense Positive Pairs in DenseCL* (b) *Dense Positive Pairs in HoughCL*

Experiments - Fine-tuning

- PASCAL VOC Object Detection

Dataset	Method	AP	AP_{50}	AP_{75}
-	random init.†	32.8	59.0	31.6
Tiny ImageNet	MoCo v2	47.6	75.3	51.2
	DenseCL	47.5	74.6	51.2
	HoughCL	**50.5**	**76.9**	**55.0**
COCO	MoCo v2†	54.7	81.0	60.6
	DenseCL†	56.7	81.7	**63.0**
	HoughCL	**56.8**	**82.1**	**63.0**
ImageNet	super. IN†	54.2	81.6	59.8
	MoCo v2†	57.0	82.2	63.4
	DenseCL†	**58.7**	**82.8**	65.2
	HoughCL	58.5	82.6	**65.7**

- COCO Object Detection

Dataset	Method	AP	AP_{50}	AP_{75}
-	random init.†	32.8	59.0	31.6
Tiny ImageNet	MoCo v2	47.6	75.3	51.2
	DenseCL	47.5	74.6	51.2
	HoughCL	**50.5**	**76.9**	**55.0**
COCO	MoCo v2†	54.7	81.0	60.6
	DenseCL†	56.7	81.7	**63.0**
	HoughCL	**56.8**	**82.1**	**63.0**
ImageNet	super. IN†	54.2	81.6	59.8
	MoCo v2†	57.0	82.2	63.4
	DenseCL†	**58.7**	**82.8**	65.2
	HoughCL	58.5	82.6	**65.7**

- COCO Instance Segmentation

Dataset	Method	AP	AP_{50}	AP_{75}
-	random init.†	32.8	59.0	31.6
Tiny ImageNet	MoCo v2	47.6	75.3	51.2
	DenseCL	47.5	74.6	51.2
	HoughCL	**50.5**	**76.9**	**55.0**
COCO	MoCo v2†	54.7	81.0	60.6
	DenseCL†	56.7	81.7	**63.0**
	HoughCL	**56.8**	**82.1**	**63.0**
ImageNet	super. IN†	54.2	81.6	59.8
	MoCo v2†	57.0	82.2	63.4
	DenseCL†	**58.7**	**82.8**	65.2
	HoughCL	58.5	82.6	**65.7**