Exploring Cross-Image Pixel Contrast for Semantic Segmentation

Wenguan Wang\textsuperscript{1*}, Tianfei Zhou\textsuperscript{1*}, Fisher Yu\textsuperscript{1}, Jifeng Dai\textsuperscript{2}, Ender Konukoglu\textsuperscript{1}, Luc Van Gool\textsuperscript{1}

\textsuperscript{1}Computer Vision Lab, ETH Zurich \textsuperscript{2}SenseTime Research

https://github.com/tfzhou/ContrastiveSeg

Abstract

Current semantic segmentation methods focus only on mining “local” context, i.e., dependencies between pixels within individual images, by context-aggregation modules (e.g., dilated convolution, neural attention) or structure-aware optimization criteria (e.g., IoU-like loss). However, they ignore “global” context of the training data, i.e., rich semantic relations between pixels across different images. Inspired by recent advance in unsupervised contrastive representation learning, we propose a pixel-wise contrastive algorithm for semantic segmentation in the fully supervised setting. The core idea is to enforce pixel embeddings belonging to a same semantic class to be more similar than embeddings from different classes. It raises a pixel-wise metric learning paradigm for semantic segmentation, by explicitly exploring the structures of labeled pixels, which were rarely explored before. Our method can be effortlessly incorporated into existing segmentation frameworks without extra overhead during testing. We experimentally show that, with famous segmentation models (i.e., DeepLabV3, HRNet, OCR) and backbones (i.e., ResNet, HRNet), our method brings performance improvements across diverse datasets (i.e., Cityscapes, PASCAL-Context, COCO-Stuff, CamVid). We expect this work will encourage our community to rethink the current de facto training paradigm in semantic segmentation.

1. Introduction

Semantic segmentation, which aims to infer semantic labels for all pixels in an image, is a fundamental problem in computer vision. In the last decade, semantic segmentation has achieved remarkable progress, driven by the availability of large-scale datasets (e.g., Cityscapes [15]) and rapid evolution of convolutional networks (e.g., VGG [63], ResNet [32]) as well as segmentation models (e.g., fully convolutional network (FCN) [51]). In particular, FCN [51] is the cornerstone of modern deep learning techniques for segmentation, due to its unique advantage in end-to-end pixel-wise representation learning. However, its spatial invariance nature hinders the ability of modeling useful context among pixels (within images). Thus a main stream of subsequent effort delves into network designs for effective context aggregation, e.g., dilated convolution[80, 8, 9], spatial pyramid pooling[84], multi-layer feature fusion[58, 47] and neural attention [35, 24]. In addition, as the widely adopted pixel-wise cross entropy loss fundamentally lacks the spatial discrimination power, some alternative optimization criteria are proposed to explicitly address object structures during segmentation network training [40, 2, 86].

Basically, these segmentation models (excluding [37]) utilize deep architectures to project image pixels into a highly non-linear embedding space (Fig. 1(c)). However, they typically learn the embedding space that only makes use of “local” context around pixel samples (i.e., pixel dependencies within individual images), but ignores “global” context of the whole dataset (i.e., pixel semantic relations across images). Hence, an essential issue has been long ignored in the field: what should a good segmentation embedding space look like? Ideally, it should not only 1) address the categorization ability of individual pixel embeddings, but also 2) be well structured to address intra-class compactness and inter-class dispersion. With regard to 2), pixels

\*The first two authors contribute equally to this work.
from a same class should be closer than those from different classes, in the embedding space. Prior studies\cite{49, 60} in representation learning also suggested that encoding intrinsic structures of training data (i.e., 2) would facilitate feature discriminativeness (i.e., 1)). So we speculate that, although existing algorithms have achieved impressive performance, it is possible to learn a better structured pixel embedding space by considering both 1) and 2).

Recent advance in unsupervised representation learning \cite{12, 31} can be ascribed to the resurgence of contrastive learning – an essential branch of deep metric learning \cite{39}. The core idea is “learn to compare”: given an anchor point, distinguish a similar (or positive) sample from a set of dissimilar (or negative) samples, in a projected embedding space. Especially, in the field of computer vision, the contrast is evaluated based on image feature vectors; the augmented version of an anchor image is viewed as a positive, while all the other images in the dataset act as negatives.

The great success of unsupervised contrastive learning and our aforementioned speculation together motivate us to rethink the current de facto training paradigm in semantic segmentation. Basically, the power of unsupervised contrastive learning roots from the structured comparison loss, which takes the advantage of the context within the training data. With this insight, we propose a pixel-wise contrastive algorithm for more effective dense representation learning in the fully supervised setting. Specifically, in addition to adopting the pixel-wise cross entropy loss to address class discrimination (i.e., property 1)), we utilize a pixel-wise contrastive loss to further shape the pixel embedding space, through exploring the structural information of labeled pixel samples (i.e., property 2)). The idea of the pixel-wise contrastive loss is to compute pixel-to-pixel contrast: enforce embeddings to be similar for positive pixels, and dissimilar for negative ones. As the pixel-level categorical information is given during training, the positive samples are the pixels belonging to a same class, and the negatives are the pixels from different classes (Fig. 1(d)). In this way, the global property of the embedding space can be captured (Fig. 1(e)) for better reflecting intrinsic structures of training data and enabling more accurate segmentation predictions.

With our supervised pixel-wise contrastive algorithm, two novel techniques are developed. First, we propose a region memory bank to better address the nature of semantic segmentation. Faced with huge amounts of highly-structured pixel training samples, we let the memory store pooled features of semantic regions (i.e., pixels with a same semantic label from a same image), instead of pixel-wise embeddings only. This leads to the pixel-to-region contrast, as a complementary for the pixel-to-pixel contrast strategy. Such memory design allows us to access more representative data samples during each training step and fully explore structural relations between pixels and semantic-level segments, i.e., pixels and segments belonging to a same class should be close in the embedding space. Second, we propose different sampling strategies to make better use of informative samples and let the segmentation model pay more attention to those segmentation-hard pixels. Previous works have confirmed that hard negatives are crucial for metric learning \cite{39, 60, 62}, and our study further reveals the importance of mining both informative negatives/positives and anchors in this supervised, dense image prediction task.

In a nutshell, our contributions are three-fold:

- We propose a supervised, pixel-wise contrastive learning method for semantic segmentation. It lifts current image-wise training strategy to an inter-image, pixel-to-pixel paradigm. It essentially learns a well structured pixel semantic embedding space, by making full use of the global semantic similarities among labeled pixels.
- We develop a region memory to better explore the large visual data space and support to further calculate pixel-to-region contrast. Integrated with pixel-to-pixel contrast computation, our method exploits semantic correlations among pixels, and between pixels and semantic regions.
- We demonstrate that more powerful segmentation models with better example and anchor sampling strategies could be delivered instead of selecting random pixel samples.

Our method can be seamlessly incorporated into existing segmentation networks without any changes to the base model and without extra inference burden during testing (Fig. 2). Hence, our method shows consistently improved intersection-over-union segmentation scores over challenging datasets (i.e., Cityscapes \cite{15}, PASCAL-Context \cite{53}, COCO-Stuff\cite{5} and CamVid\cite{3}), using state-of-the-art segmentation architectures (i.e., DeepLabV3 \cite{9}, HRNet \cite{65} and OCR \cite{81}) and standard backbones (i.e., ResNet \cite{32}, HRNet \cite{65}). The impressive results shed light on the promises of metric learning in dense image prediction tasks. We expect this work to provide insights into the critical role of global pixel relationships in segmentation network training, and foster research on the open issues raised.
2. Related Work

Our work draws on existing literature in semantic segmentation, contrastive learning and deep metric learning. For brevity, only the most relevant works are discussed.

Semantic Segmentation. FCN [51] greatly promotes the advance of semantic segmentation. It is good at end-to-end dense feature learning, however, only perceiving limited visual context with local receptive fields. As strong dependencies exist among pixels in an image and these dependencies are informative about the structures of the objects [70], how to capture such dependencies becomes a vital issue for further improving FCN. A main group of follow-up effort attempts to aggregate multiple pixels to explicitly model context, for example, utilizing different sizes of convolutional/pooling kernels or dilation rates to gather multi-scale visual cues [80, 84, 8, 9], building image pyramids to extract context from multi-resolution inputs, adopting the Encoder-Decoder architecture to merge multi-layer features [38, 47, 66], applying CRF to recover detailed structures [50, 87], and employing neural attention [67, 29] to directly exchange context between paired pixels [10, 55, 36, 24]. Apart from investigating context-aggregation network modules, another line of work turns to designing context-aware optimization objectives [40, 2, 86], i.e., directly verify segmentation structures during training, to replace the pixel-wise cross entropy loss.

Though impressive, these methods only address pixel dependencies within individual images, neglecting the global context of the labeled data, i.e., pixel semantic correlations across different training images. Through a pixel-wise contrastive learning formulation, we map pixels in different categories to more distinctive features. The learned pixel features are not only discriminative for semantic classification within images, but also, more crucially, across images.

Contrastive Learning. Recently, the most compelling methods for learning representations without labels have been unsupervised contrastive learning [55, 34, 73, 13, 12], which significantly outperformed other pretext task-based alternatives [43, 26, 18, 54]. With a similar idea to exemplar learning [19], contrastive methods learn representations in a discriminative manner by contrasting similar (positive) data pairs against dissimilar (negative) pairs. A major branch of subsequent studies focuses on how to select the positive and negative pairs. For image data, the standard positive pair sampling strategy is to apply strong perturbations to create multiple views of each image data [73, 12, 31, 34, 6]. Negative pairs are usually randomly sampled, but some hard negative example mining strategies [41, 57, 38] were recently proposed. In addition, to store more negative samples during contrast computation, fixed [73] or momentum updated [52, 31] memories are adopted. Some latest studies [41, 33, 71] also confirm label information can assist contrastive learning based image-level pattern pre-training.

We raise a pixel-to-pixel contrastive learning method for semantic segmentation in the fully supervised setting. It yields a new training protocol that explores global pixel relations in labeled data for regularizing segmentation embedding space. Though a few concurrent works also address contrastive learning in dense image prediction [67, 7, 69], the ideas are significantly different. First, they typically consider contrastive learning as a pre-training step for dense image embedding. Second, they simply use the local context within individual images, i.e., only compute the contrast among pixels from augmented versions of a same image. Third, they do not notice the critical role of metric learning in complementing current well-established pixel-wise cross-entropy loss based training regime (cf. §3.2).

Deep Metric Learning. The goal of metric learning is to quantify the similarity among samples using an optimal distance metric. Contrastive loss [28] and triplet loss [60] are two basic types of loss functions for deep metric learning. With a similar spirit of increasing and decreasing the distance between similar and dissimilar data samples, respectively, the former one takes pairs of sample as input while the latter is composed of triplets. Deep metric learning [22] has proven effective in a wide variety of computer vision tasks, such as image retrieval [64] and face recognition [60].

Although a few prior methods address the idea of metric learning in semantic segmentation, they only account for the local content from objects [29] or instances [16, 1, 22, 42]. It is worth noting [37] also explores cross-image information of training data, i.e., leverage perceptual pixel groups for non-parametric pixel classification. Due to its clustering based metric learning strategy, [37] needs to retrieve extra labeled data for inference. Differently, our core idea, i.e., exploit inter-image pixel-to-pixel similarity to enforce global constraints on the embedding space, is conceptually novel and rarely explored before. It is executed by a compact training paradigm, which enjoys the complementary advantages of unary, pixel-wise cross-entropy loss and pair-wise, pixel-to-pixel contrast loss, without bringing any extra inference cost or modification to the base network during deployment.

3. Methodology

Before detailing our supervised pixel-wise contrastive algorithm for semantic segmentation (§3.2), we first introduce the contrastive formulation in unsupervised visual representation learning and the notion of memory bank (§3.1).

3.1. Preliminaries

Unsupervised Contrastive Learning. Unsupervised visual representation learning aims to learn a CNN encoder $f_{CNN}$ that transforms each training image $I$ to a feature vector $v = f_{CNN}(I) \in \mathbb{R}^D$, such that $v$ best describes $I$. To achieve this goal, contrastive approaches conduct training by distinguishing a positive (an augmented version of an-
contrastive learning, called InfoNCE \[27, 55\], takes the following form:

$$\mathcal{L}_{i}^{\text{NCE}} = -\log \frac{\exp(v \cdot v^+/\tau)}{\exp(v \cdot v^+/\tau) + \sum_{v' \in \mathcal{N}_i} \exp(v \cdot v' / \tau)},$$ \hspace{1cm} (1)

where \(v^+\) is an embedding of a positive for \(i\), \(\mathcal{N}_i\) contains embeddings of negatives, \(\cdot \cdot\cdot\) denotes the inner (dot) product, and \(\tau > 0\) is a temperature hyper-parameter. Note that all the embeddings in the loss function are \(\ell_2\)-normalized.

**Memory Bank.** As revealed by recent studies [73, 13, 31], a large set of negatives (i.e., \(|\mathcal{N}_i|\)) is critical in unsupervised contrastive representation learning. As the number of negatives is limited by the mini-batch size, recent contrastive methods utilize large, external memories as a bank to store more navigate samples. Specifically, some methods [73] directly store the embeddings of all the training samples in the memory, however, easily suffering from asynchronous update. Some others choose to keep a queue of the last few updates. Some others have a large set of negatives (\(|\mathcal{N}_i| = \mathcal{N}_i\) from several images randomly drawn from the training set excluding \(I\)), based on the principle of similarity between samples. A popular loss function for contrastive learning, called InfoNCE [27, 55], takes the following form:

$$\mathcal{L}_{i}^{\text{NCE}} = -\log \frac{\exp(v \cdot v^+ / \tau)}{\sum_{v' \in \mathcal{N}_i} \exp(v \cdot v' / \tau)} + \frac{\exp(v \cdot v^- / \tau)}{\sum_{v' \in \mathcal{N}_i} \exp(v \cdot v' / \tau)},$$ \hspace{1cm} (2)

where \(v^+\) and \(v^-\) denote pixel embedding collections of the positive and negative samples, respectively, for pixel \(i\). Note that the positive/negative samples and the anchor \(i\) are not restricted to being from a same image. As Eq. (3) shows, the purpose of such pixel-to-pixel contrast loss design is to learn an embedding space, by pulling the same class pixel samples close and by pushing different class samples apart.

**Pixel-Wise Cross-Entropy Loss.** In the context of semantic segmentation, each pixel \(i\) of an image \(I\) has to be classified into a semantic class \(c \in \mathcal{C}\). Current approaches typically cast this task as a pixel-wise classification problem. Specifically, let \(f_{\text{FCN}}\) be an FCN encoder (e.g., ResNet [32]), that produces a dense feature \(I \in \mathbb{R}^{H \times W \times D}\) for \(I\), from which the pixel embedding \(i \in \mathbb{R}^D\) of \(i\) can be derived (i.e., \(i \in I\)). Then a segmentation head \(f_{\text{SEG}}\) maps \(I\) into a categorical score map \(Y = f_{\text{SEG}}(I) \in \mathbb{R}^{H \times W \times |\mathcal{C}|}\). Further let \(y = [y_i, \cdots, y_{|\mathcal{C}|}] \in \mathbb{R}^{|\mathcal{C}|}\) be the unnormalized score vector (termed as logit) for pixel \(i\), derived from \(Y\), i.e., \(y \in Y\). Given \(y\) for pixel \(i\) w.r.t its groundtruth label \(c \in \mathcal{C}\), the cross-entropy loss is optimized with softmax (cf. Fig. 3):

$$\mathcal{L}_{i}^{\text{CE}} = -\mathbf{1}_c \log(\text{softmax}(y)), \hspace{1cm} (2)$$

where \(\mathbf{1}_c\) denotes the one-hot encoding of \(c\), the logarithm is defined as element-wise, and \(\text{softmax}(y) = \frac{\exp(y_c)}{\sum_{c'=1}^{|\mathcal{C}|} \exp(y_{c'})}\). Such training objective design mainly suffers from two limitations. 1) It penalizes pixel-wise predictions independently but ignores relationship between pixels [86], 2) Due to the use of \(\text{softmax}\), the loss only depends on the relative relation among logits and cannot directly supervise on the learned representations [56]. These two issues were rarely noticed; only a few structure-aware losses are designed to address 1), by considering pixel affinity [40], optimizing intersection-over-union measurement [2], or maximizing the mutual information between the groundtruth and prediction map [86]. Nevertheless, these alternative losses only consider the dependencies between pixels within an image (i.e., local context), regardless of the semantic correlations between pixels across images (i.e., global structure).

**Pixel-to-Pixel Contrast.** In this work, we develop a pixel-wise contrastive learning method that addresses both 1) and 2), through regularizing the embedding space and exploring the global structures of training data. We first extend Eq. (1) to our supervised, dense image prediction setting. Basically, the data samples in our contrastive loss computation are training image pixels. In addition, for a pixel \(i\) with its groundtruth semantic label \(\bar{c}\), the positive samples are other pixels also belonging to the class \(\bar{c}\), while the negatives are the pixels belonging to the other classes \(\mathcal{C}\setminus \bar{c}\). Our supervised, pixel-wise contrastive loss is defined as:

$$\mathcal{L}_{i}^{\text{NCE}} = -\frac{1}{|\mathcal{P}_i|} \sum_{i' \in \mathcal{P}_i} \log \frac{\exp(i \cdot i' / \tau)}{\sum_{i'' \in \mathcal{N}_i} \exp(i \cdot i'' / \tau) + \sum_{i'' \in \mathcal{N}_i} \exp(i \cdot i'' / \tau)},$$ \hspace{1cm} (3)

where \(\mathcal{P}_i\) and \(\mathcal{N}_i\) denote pixel embedding collections of the positive and negative samples, respectively, for pixel \(i\). Note that the positive/negative samples and the anchor \(i\) are not restricted to being from a same image. As Eq. (3) shows, the purpose of such pixel-to-pixel contrast loss design is to learn an embedding space, by pulling the same class pixel samples close and by pushing different class samples apart.

The pixel-wise cross-entropy loss in Eq. (2) and our contrastive loss in Eq. (3) are complementary to each other; the former lets segmentation networks learn discriminative
ory is built with size $|C| \times \bar{s}$ via a variant of pixel embeddings. The training images and $|C|$ region embeddings are absorbed from image segments (i.e., a memory bank that stores more representative embeddings). The under-sampled pixel embeddings are too sparse to fully make use of massive data to learn good representations. However, the latter helps to regularize the embedding space with image segments.

Pixel-to-Region Contrast. As stated in §3.1, memory is a critical technique that helps contrastive learning to make use of massive data to learn good representations. However, since there are vast numbers of pixel samples in our dense prediction setting and most of them are redundant (i.e., sampled from harmonious object regions), directly storing all the training pixel samples, like traditional memory [12], will greatly slow down the learning process. Maintaining several last batches in a queue, like [68, 13, 31], is also not a good choice, as recent batches only contain a limited number of images, reducing the diversity of pixel samples. Thus we choose to maintain a pixel queue per category. For each category, only a small number, i.e., $V$, of pixels are randomly selected from each image in the latest mini-batch, and pulled into the queue, with a size of $T \gg V$. In practice we find this strategy is very efficient and effective, but the under-sampled pixel embeddings are too sparse to fully capture image content. Therefore, we further build a region memory bank that stores more representative embeddings absorbed from image segments (i.e., semantic regions).

Specifically, for a segmentation dataset with a total of $N$ training images and $|C|$ semantic classes, our region memory is built with size $|C| \times N \times D$, where $D$ is the dimension of pixel embeddings. The $(\bar{c}, n)$-th element in the region memory is a $D$-dimensional feature vector obtained by average pooling all the embeddings of pixels labeled as $\bar{c}$ category in the $n$-th image. The region memory brings two advantages: 1) store more representative “pixel” samples with low memory consumption; and 2) allow our pixel-wise contrastive loss (cf. Eq. (3)) to further explore pixel-to-region relations. With regard to 2), when computing Eq. (3) for an anchor pixel $\bar{i}$ belonging to $\bar{c}$ category, stored region embeddings with the same class $\bar{c}$ are viewed as positives, while the region embeddings with other classes $\bar{C}\setminus \bar{c}$ are negatives. For the pixel memory, the size is $|C| \times (N+T) \times D$. Therefore, for the whole memory (denoted as $\mathcal{M}$), the total size is $|C| \times (N+T) \times D$. We examine the design of $\mathcal{M}$ in §4.2. In the following sections, we will not distinguish pixel and region embeddings in $\mathcal{M}$, unless otherwise specified.

**Hard Example Sampling.** Prior research [60, 39, 41, 57, 38] found that, in addition to loss designs and the amount of training samples, the discriminating power of the training samples is crucial for metric learning. Considering our case, the gradient of the pixel-wise contrastive loss (cf. Eq. (3)) w.r.t. the anchor embedding $\bar{i}$ can be given as:

$$\frac{\partial L_{\text{NCE}}^i}{\partial \bar{i}} = -\frac{1}{B_1} \sum_{i \in P_{\bar{i}}} \sum_{i' \in P_{\bar{i}'}} \left( \frac{1-p_{\bar{i}+\bar{i}'}}{\tau} - \sum_{i' \in P_{\bar{i}'}} p_{\bar{i}'-\bar{i}} \cdot \frac{1}{\tau} \right),$$

where $p_{\bar{i}+\bar{i}'} \in [0, 1]$ denotes the matching probability between a positive/negative $\bar{i}+\bar{i}'$ and the anchor $\bar{i}$, i.e., $p_{\bar{i}+\bar{i}'} = \frac{\exp(\bar{i}^{\top} \bar{i}' / \tau)}{\sum_{i' \in P_{\bar{i}'}} \exp(\bar{i}^{\top} \bar{i}' / \tau)}$. We view the negatives with dot products (i.e., $\bar{i} \cdot \bar{i}'$) closer to 1 to be harder, i.e., where $p_{\bar{i}'-\bar{i}}$ are similar to the anchor $\bar{i}$. Similarly, the positives with dot products (i.e., $\bar{i} \cdot \bar{i}'$) closer to −1 are considered as harder, i.e., positives which are dissimilar to $\bar{i}$. We can find that, harder negatives bring more gradient contributions, i.e., $p_{\bar{i}'-\bar{i}}$, than easier negatives. This principle also holds true for positives, whose gradient contributions are 1 − $p_{\bar{i}+\bar{i}'}$. Kalantidis et al. [38] further indicate that, as training progresses, more and more negatives become too simple to provide significant contributions to the unsupervised contrastive loss (cf. Eq. (1)). This also happens in our supervised setting (cf. Eq. (3)), for both negatives and positives. To remedy this problem, we propose the following sampling strategies:

- **Hardest Example Sampling.** Inspired by hardest negative mining in metric learning [4], we first design a “hardest example sampling” strategy: for each anchor pixel embedding $\bar{i}$, only sampling top-$K$ hardest negatives and positives from the memory bank $\mathcal{M}$, for the computation of the pixel-wise contrastive loss (i.e., $L_{\text{NCE}}^i$ in Eq. (3)).

- **Semi-Hard Example Sampling.** Some studies propose to make use of harder negatives, as optimizing with the hardest negatives for metric learning likely leads to bad local minima [60, 74, 23]. Thus we further design a “semi-hard example sampling” strategy: for each anchor embedding $\bar{i}$, we first collect top $10\%$ nearest negatives (resp. top

![Figure 4: Visualization of features learned with (left) the pixel-wise entropy loss (i.e., $L_{\text{CE}}^i$ in Eq. (2)) and (right) our pixel contrast based optimization objective (i.e., $L_{\text{SEG}}^i$ in Eq. (4)) on Cityscapes val [15]. Features are colored according to class labels. As seen, the proposed $L_{\text{SEG}}^i$ begets a well-structured semantic feature space.](image-url)
Joint Loss

- Segmentation-Aware Hard Anchor Sampling. Rather than mining informative positive and negative examples, we develop an anchor sampling strategy. We treat the categorization ability of an anchor embedding as its importance during contrastive learning. This leads to “segmentation-aware hard anchor sampling”: the pixels with incorrect predictions, i.e., $c \neq \hat{c}$, are treated as hard anchors. For the contrastive loss computation (cf. Eq. (3)), half of the anchors are randomly sampled and half are the hard ones. This anchor sampling strategy enables our contrastive learning to focus more on the pixels hard for classification, delivering more segmentation-aware embeddings.

In practice, we find “semi-hard example sampling” strategy performs better than “hardest example sampling”. In addition, after employing “segmentation-aware hard anchor sampling” strategy, the segmentation performance can be further improved. See §4.2 for related experiments.

3.3. Detailed Network Architecture

Our algorithm has five major components (cf. Fig.3):

- FCN Encoder, $f_{\text{FCN}}$, which maps each input image $I$ into dense embeddings $I = f_{\text{FCN}}(I) \in \mathbb{R}^{H \times W \times D}$. In our algorithm, any FCN backbones can be used to implement $f_{\text{FCN}}$ and we test two commonly used ones, i.e., ResNet [32] and HRNet [65], in our experiments.

- Segmentation Head, $f_{\text{SEG}}$, that projects $I$ into a score map $Y = f_{\text{SEG}}(I) \in \mathbb{R}^{H \times W \times |C|}$. We conduct evaluations using different segmentation heads in mainstream methods (i.e., DeepLabv3 [9], HRNet [65], and OCR [81]).

- Project Head, $f_{\text{PROJ}}$, that maps each high-dimensional pixel embedding $i \in I$ into a 256-d $\ell_2$-normalized feature vector [12], for the computation of the contrastive loss $\mathcal{L}_{\text{NCE}}$. $f_{\text{PROJ}}$ is implemented as two $1 \times 1$ convolutional layers with ReLU. Note that the project head is only applied during training and is removed at inference time. Thus it does not introduce any changes to the segmentation network or extra computational cost in deployment.

- Memory Bank, $\mathcal{M}$, which consists of two parts that store pixel and region embeddings, respectively. For each training image, we sample $V = 10$ pixels per class. For each class, we set the size of the pixel queue as $T = 10V$. The memory bank is also discarded after training.

- Joint Loss, $\mathcal{L}_{\text{SEG}}$ (cf. Eq. (4)), that takes the power of representation learning (i.e., $\mathcal{L}_{\text{CE}}$ in Eq. (2)) and metric learning (i.e., $\mathcal{L}_{\text{NCE}}$ in Eq. (3)) for more distinct segmentation feature learning. In practice, we find our method is not sensitive to the coefficient $\lambda$ (e.g., when $\lambda \in \{0.1, 1\}$) and empirically set $\lambda$ as 1. For $\mathcal{L}_{\text{NCE}}$ in Eq. (3), we set the temperature $\tau$ as 0.1. For sampling, we find “semi-hard example sampling” + “segmentation-aware hard anchor sampling” performs the best and set the numbers of sampled instances (i.e., $K$) as 1,024 and 2,048 for positive and negative, respectively. For each mini-batch, 50 anchors are sampled per category (half are randomly sampled and the other half are segmentation-hard ones).

4. Experiment

4.1. Experimental Setup

Datasets. Our experiments are conducted on four datasets:

- Cityscapes [15] has 5,000 finely annotated urban scene images, with 2,975/500/1,524 for train/val/test. The segmentation performance is reported on 19 challenging categories, such as person, sky, car, and building.

- PASCAL-Context [53] contains 4,998 and 5,105 images in train and test splits, respectively, with precise annotations of 59 semantic categories.

- COCO-Stuff [5] consists of 10,000 images gathered from COCO [48]. It is split into 9,000 and 1,000 images for train and test. It provides rich annotations for 80 object classes and 91 stuff classes.


Training. As mentioned in §3.3, various backbones (i.e., ResNet [32] and HRNet [65]) and segmentation networks (i.e., DeepLabv3 [9], HRNet [65], and OCR [81]) are exploited in our experiments to thoroughly validate the proposed algorithm. We follow conventions [65, 81, 14, 76] for training hyper-parameters. For fairness, we initialize all backbones using corresponding weights pretrained on ImageNet [59], with the remaining layers being randomly initialized. For data augmentation, we use color jitter, horizontal flipping and random scaling with a factor in [0.5, 2]. We use SGD as our optimizer, with a momentum 0.9 and weight decay 0.0005. We adopt the polynomial annealing policy [9] to schedule the learning rate, which is multiplied by $(1 - \frac{\text{iter}}{\text{total iter}})^{power}$ with $power = 0.9$. Moreover, for Cityscapes, we use a mini-batch size of 8, and an initial learning rate of 0.01. All the training images are augmented by random cropping from 1024×2048 to 512×1024. For the experiments on test, we follow [65] to train the model for 100K iterations. Note that we do not use any extra training data (e.g., Cityscapes coarse [15]). For PASCAL-Context and COCO-Stuff, we opt a mini-batch size of 16, an initial learning rate of 0.001, and crop size of 520×520. We train for 60K iterations over their train sets. For CamVid, we train the model for 6K iterations, with batch size 16, learning rate 0.02 and original image size.

Testing. Following general protocol [65, 81, 61], we average the segmentation results over multiple scales with flipping, i.e., the scaling factor is 0.75 to 2.0 (with intervals of 0.25) times of the original image size. Note that, during testing, there is no any change or extra inference step intro-
Table 1: Comparison of different contrastive mechanisms on Cityscapes val [15]. See §4.2 for more details.

<table>
<thead>
<tr>
<th>Memory</th>
<th>Backbone</th>
<th>mIoU (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Baseline (w/o contrast)</td>
<td>HRNetV2-W48</td>
<td>78.1</td>
</tr>
<tr>
<td>Intra-Image Contrast</td>
<td>HRNetV2-W48</td>
<td>78.9 (+0.8)</td>
</tr>
<tr>
<td>Inter-Image Contrast</td>
<td>HRNetV2-W48</td>
<td>81.0 (+2.9)</td>
</tr>
</tbody>
</table>

Table 2: Comparison of different memory bank designs on Cityscapes val [15]. See §4.2 for more details.

<table>
<thead>
<tr>
<th>Memory</th>
<th>Backbone</th>
<th>mIoU (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Baseline (w/o contrast)</td>
<td>HRNetV2-W48</td>
<td>78.1</td>
</tr>
<tr>
<td>Mini-Batch (w/o memory)</td>
<td>HRNetV2-W48</td>
<td>79.8 (+1.7)</td>
</tr>
<tr>
<td>Pixel Memory</td>
<td>HRNetV2-W48</td>
<td>80.5 (+2.6)</td>
</tr>
<tr>
<td>Region Memory</td>
<td>HRNetV2-W48</td>
<td>80.2 (+2.1)</td>
</tr>
<tr>
<td>Pixel + Region Memory</td>
<td>HRNetV2-W48</td>
<td>81.0 (+2.9)</td>
</tr>
</tbody>
</table>

Table 3: Comparison of different hard example sampling strategies on Cityscapes val [15]. See §4.2 for more details.

<table>
<thead>
<tr>
<th>Sampling Anchor</th>
<th>Pos./Neg.</th>
<th>Backbone</th>
<th>mIoU (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Random</td>
<td></td>
<td>HRNetV2-W48</td>
<td>79.3 (+1.2)</td>
</tr>
<tr>
<td>Hardest</td>
<td></td>
<td>HRNetV2-W48</td>
<td>79.4 (+1.3)</td>
</tr>
<tr>
<td>Semi-Hard</td>
<td></td>
<td>HRNetV2-W48</td>
<td>80.1 (+2.0)</td>
</tr>
<tr>
<td>Seg.-aware hard</td>
<td>Random</td>
<td>HRNetV2-W48</td>
<td>80.2 (+2.1)</td>
</tr>
<tr>
<td></td>
<td>Hardest</td>
<td>HRNetV2-W48</td>
<td>80.5 (+2.4)</td>
</tr>
<tr>
<td></td>
<td>Semi-Hard</td>
<td>HRNetV2-W48</td>
<td>81.0 (+2.9)</td>
</tr>
</tbody>
</table>

4.2. Diagnostic Experiment

We first study the efficacy of our core ideas and essential model designs, over Cityscapes val [15]. We adopt HRNet [65] as our base segmentation network (denoted as “Baseline (w/o contrast)” in Tables 1-3). To perform extensive ablation experiments, we train each model for 40K iterations while keeping other hyper-parameters unchanged.

Inter-Image vs. Intra-Image Pixel Contrast. We first investigate the effectiveness of our core idea of inter-image pixel contrast. As shown in Table 1, additionally considering cross-image pixel semantic relations (i.e., “Inter-Image Contrast”) in segmentation network learning leads to a substantial performance gain (i.e., 2.9%), compared with “Baseline (w/o contrast)” in Table 2. In addition, we develop another baseline, “Intra-Image Contrast”, which only samples pixels from same images during the contrastive loss (i.e., $L_{NCE}$ in Eq. (5)) computation. The results in Table 1 suggest that, although “Intra-Image Contrast” also boosts the performance over “Baseline (w/o contrast)” (i.e., 78.1% → 78.9%), “Inter-Image Contrast” is more favored.

Memory Bank. We next validate the design of our memory bank. The results are summarized in Table 2. Based on “Baseline (w/o contrast)”, we first derive a variant, “Mini-Batch w/o memory”: only compute pixel contrast within each mini-batch, without outside memory. It gets 79.8% mIoU. We then provision this variant with pixel and region memories separately, and observe consistent performance gains (79.8% → 80.5% for pixel memory and 79.8% → 80.2% for region memory). This verifies that i) leveraging more pixel samples during contrastive learning leads to better pixel embeddings; and ii) both pixel-to-pixel and pixel-to-region relations are informative cues. Finally, after using both the two memories, a higher score (i.e., 81.0%) is achieved, revealing i) the effectiveness of our memory design; and ii) necessity of comprehensively considering both pixel-to-pixel contrast and pixel-to-region contrast.

Hard Example Mining. Table 3 presents a comprehensive examination of various hard example mining strategies proposed in §3.2. Our main observations are the following: i) For positive/negative sampling, mining meaningful pixels (i.e., “hardest” or “semi-hard” sampling), rather than “random” sampling, is indeed useful; ii) Hence, “semi-hard” sampling is more favored, as it improves the robustness of training by avoiding overfitting outliers in the training set. This corroborates related observations in unsupervised setting [72] and indicates that segmentation may benefit from more intelligent sample treatment; and iii) For anchor sampling, “seg.-aware hard” strategy further improves the performance (i.e., 80.1% → 81.0%) over “random” sampling only. This suggests that exploiting task-related signals in supervised metric learning may help develop better segmentation solutions, which has remained relatively untapped.

4.3. Comparison to State-of-the-Arts

Cityscapes [15]. Table 4 lists the scores on Cityscapes test, under two widely used training settings [65] (trained over train or train+val). Our method brings impressive gains over 3 strong baselines (i.e., DeepLabV3, HRNetV2, and OCR), and sets a new state-of-the-art.

PASCAL-Context [53]. Table 5 presents comparison results on PASCAL-Context test. Our approach improves the performance of base networks by solid margins (i.e., 54.0 → 55.1 for HRNetV2, 56.2 → 57.2 for OCR). This is particularly impressive considering the fact that improvement on this extensively-benchmarked dataset is very hard.

COCO-Stuff [5]. Table 6 reports performance comparison of our method against seven competitors on COCO-Stuff test. As we find that OCR+Ours yields a mIoU of 41.0%, which leads to a promising gain of 0.5% over its counterpart (i.e., OCR with a 40.5% mIoU). Besides, HR-
Backbone | ResNet-101 | mIoU (%) |
--- | --- | --- |
HRNetV2-W48 | 57.2 (+1.0) | 54.1 |
HRNetV2-W48 | 54.4 | 52.6 |
D-ResNet-101 | 53.2 | 55.3 |
D-ResNet-101 | 54.4 | 52.6 |
HRNetV2-W48 | 55.1 (+1.1) | 55.3 |
HRNetV2-W48 | 56.2 | 57.2 (+1.0) |


Fig. 5 depicts qualitative comparisons of OCR+Ours against OCR over representative examples from three datasets (i.e., Cityscapes, PASCAL-Context, COCO-Stuff). As seen, our method is capable of producing more accurate segments across various challenge scenarios.

5. Conclusion and Discussion

In this paper, we propose a new supervised learning paradigm for semantic segmentation, enjoying the complementary advantages of unary classification and structured metric learning. Through pixel-wise contrastive learning, it investigates global semantic relations between training pixels, guiding pixel embeddings towards cross-image category-discriminative representations that eventually improve the segmentation performance. Our method generates promising results and shows great potential in a variety of dense prediction tasks, such as pose estimation [89, 21] and body parsing [88, 20]. It also comes with new challenges, in particular regarding smart data sampling, metric learning loss design, class rebalancing during training, and multi-layer feature contrast. Given the massive number of technique breakthroughs over the past few years, we expect a flurry of innovation towards these promising directions.

Acknowledgment This work was supported by Zhejiang Lab’s Open Fund (No. 2020AA3AB14) and CCF-Baidu Open Fund.
References

[15] Marius Cordts, Mohamed Omran, Sebastian Ramos, Timo Rehfeld, Markus Enzweiler, Rodrigo Benenson, Uwe Franke, Stefan Roth, and Bernt Schiele. The cityscapes dataset for semantic urban scene understanding. In CVPR, 2016. 1, 2, 5, 6, 7, 8
[34] R Devon Hjelm, Alex Fedorov, Samuel Lavoie-Marchildon, Karan Grewal, Phil Bachman, Adam Trischler, and Yoshua Bengio. Learning deep representations by mutual information estimation and maximization. In ICLR, 2019. 3
[37] Jyh-Jing Hwang, Stella X Yu, Jianbo Shi, Maxwell D
Unsupervised feature learning via non-parametric instance discrimination. In *CVPR*, 2018. 3, 4


[81] Li Zhang, Xiangtai Li, Anurag Arnab, Kuiyuan Yang, Yunhai Tong, and Philip HS Torr. Dual graph convolutional network for semantic segmentation. In *BMVC*, 2019. 8

