Unsupervised Representation for Semantic Segmentation by Implicit Cycle-Attention Contrastive Learning

Bo Pang[1], Yizhuo Li[1], Yifan Zhang[1], Gao Peng[1], Jiajun Tang[1], Kaiwen Zha[2], Jiefeng Li[1], Cewu Lu[1]*

[1] Shanghai Jiao Tong University
[2] Massachusetts Institute of Technology
{pangbo, liyizhuo, zhangyf_sjtu, penggao, yelantingfeng, ljf_likit, lucewu}@sjtu.edu.cn
kzha@mit.edu

Abstract

We study the unsupervised representation learning for the semantic segmentation task. Different from previous works that aim at providing unsupervised pre-trained backbones for segmentation models which need further supervised fine-tune, here, we focus on providing representation that is only trained by unsupervised methods. This means models need to directly generate pixel-level, linearly separable semantic results. We first explore and present two factors that have significant effects on segmentation under the contrastive learning framework: 1) the difficulty and diversity of the positive contrastive pairs, 2) the balance of global and local features. With the intention of optimizing these factors, we propose the cycle-attention contrastive learning (CACL). CACL makes use of semantic continuity of video frames, adopting unsupervised cycle-consistent attention mechanism to implicitly conduct contrastive learning with difficult, global-local-balanced positive pixel pairs. Compared with baseline model MoCo-v2 and other unsupervised methods, CACL demonstrates consistently superior performance on PASCAL VOC (+4.5 mIoU) and Cityscapes (+4.5 mIoU) datasets.

Introduction

Semantic segmentation which densely assigns semantic labels for every pixel in an image is a fundamental and important visual task with wide range of application scenarios. Deep learning (LeCun, Bengio, and Hinton 2015) under the framework of supervised learning together with large annotated datasets (Deng et al. 2009; Carreira and Zisserman 2017; Lin et al. 2014) has taken computer vision to new heights of accuracy in the last decade and semantic segmentation (Long, Shelhamer, and Darrell 2015; Chen et al. 2017; Sun et al. 2019) has been ready for commercial usage in the supervised setting. Nevertheless, obtaining annotations required by supervised methods is expensive and costs significant amounts of time (Asano et al. 2020; Fabbri et al. 2018). Different scenarios need annotating different specific data, making the application of models very expensive. To this end, in this paper, we study the unsupervised method to train deep semantic segmentation models.

Inspired by contrastive learning (Chen et al. 2020a; He et al. 2020; Grill et al. 2020) that makes great progress in image-level classification task, we choose the unsupervised contrastive representation learning framework to solve semantic segmentation task which is essentially a pixel-level classification task. Some of the previous works tried to advance the contrastive method from image-level to pixel-level and their solution can be summarised into two main directions. One is to design better pre-training methods which are more suitable for dense tasks (Wang et al. 2020; Xie et al. 2020). This strategy needs to fine-tune models with supervision on downstream tasks (semantic segmentation) after the pre-training progress. The other one adopts a two-stage scheme (Xiong et al. 2021; Van Gansbeke et al. 2021). It takes advantage of optical flow (Sun et al. 2018; Teed and Deng 2020) or saliency estimation (Qin et al. 2019; Nguyen et al. 2019) methods to generate preliminary intermediate results, then applies the contrastive framework to finish the representation learning. This strategy adopts complicated and ad-hoc techniques from other fields to get the intermediate results, some of which are trained by synthesized data in a supervised manner (Teed and Deng 2020). This makes the validity of models not transparent and the two stages is not compact and grace for representation learning.

In this paper, to this end, we aim at designing an end-to-

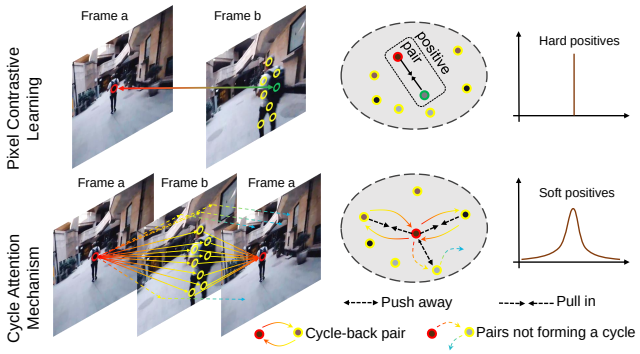

Figure 1: **Illustration of the cycle-attention scheme.** Different from the conventional contrastive method that assigns positive and negative pairs, the cycle-attention scheme implicitly pulls in and pushes out the samples by maximizing the probability of "cycle back". Compared to the hard positive pairs adopted by conventional contrastive methods, cycle-attention scheme forms soft positive relationships. Best viewed in color.

*Cewu Lu is the corresponding author.

end unsupervised representation learning method for the semantic segmentation task based on the contrastive learning framework to directly generate linearly separable semantic results. To this end, we first define and analyse two important factors under the contrastive framework: 1) *the difficulty and diversity of the positive contrastive pairs*: We discuss what is a difficult positive pair for semantic segmentation and how different methods affect the difficulty and diversity. 2) *the balance between local and global semantics*: We discuss whether a pixel-level task needs global information and the relationship between them.

With the results of the above analysis, we propose the cycle-attention contrastive learning (CACL) for semantic segmentation. Inspired by the cycle-consistency mechanism designed for the semi-supervised tracking task, we propose the cycle-attention scheme (see Fig. 1). In CACL, three stand-alone unsupervised heads work together to learn representations. In addition to the cycle-attention head, a pixel contrastive head and a global contrastive head are designed to capture local and global features. The cycle-attention head conducts contrastive learning in an implicit manner and takes charge of digging more diverse positive samples and balancing global features with local ones by maintaining the cycle consistency, leading to automatically generating hierarchical soft positive relationships.

The proposed end-to-end CACL method is effective and pretty simple. We evaluate it on PASCAL-VOC (Everingham et al. 2010) and Cityscapes (Cordts et al. 2016) datasets under linear protocol setting to verify results' linear separability. Extensive experiments show that CACL works robustly and we observe consistent performance improvements over baselines. Compared to image-level contrastive methods, our CACL outperforms MoCo-v2 by **4.5** mIoU on VOC and Cityscapes. Compared with two-stage methods, CACL also achieves better results in a much simpler compact structure. We hope this one-stage method will provide the community with new insights.

Related Work

Unsupervised Representation Learning Earlier methods for unsupervised representation learning often involve handcrafted pretext such as denoising (Vincent et al. 2008), colorization (Iizuka, Simo-Serra, and Ishikawa 2016; Larsson, Maire, and Shakhnarovich 2017; Zhang, Isola, and Efros 2016), inpainting (Pathak et al. 2016), and jigsaw solving (Noroozi and Favaro 2016; Noroozi et al. 2018). There are more pretext tasks avaliable for video representation learning such as ordering (Fernando et al. 2017; Misra, Zitnick, and Hebert 2016; Wei et al. 2018), motion estimation (Agrawal, Carreira, and Malik 2015; Jayaraman and Grauman 2015; Tung et al. 2017; Liu et al. 2018), and future frame prediction (Lotter, Kreiman, and Cox 2016; Mathieu, Couprie, and LeCun 2015; Srivastava, Mansimov, and Salakhudinov 2015; Vondrick, Pirsiavash, and Torralba 2016a,b). The representations learned by these pretext tasks are relatively limited in terms of the performance on downstream tasks. Clustering methods using pseudo labels or prototypes in training (Caron et al. 2018, 2020a) have also been explored for unsupervised representation learning.

Contrastive method for representation learning has been quite popular recently for its simplicity and effectiveness (Oord, Li, and Vinyals 2018; Hjelm et al. 2018; Zhuang, Zhai, and Yamins 2019; Tian, Krishnan, and Isola 2019; Bachman, Hjelm, and Buchwalter 2019; Chen et al. 2020a; He et al. 2020; Grill et al. 2020; Caron et al. 2020b). The performance of its representations on downstream tasks has been comparable to that of supervised learning. These method are often implemented via a large batch size (Chen et al. 2020a), a memory bank (Wu et al. 2018) or a negative queue (He et al. 2020). Some recent works don't need negative sample, which use the Siamese network and asymmetry structures (Grill et al. 2020; Chen and He 2020). Our method follows the contrastive framework and aims at making contrastive learning feasible for end-to-end solving the pixel-level dense task, semantic segmentation.

Unsupervised Semantic Segmentation Some recent works (Ji, Henriques, and Vedaldi 2019; Hwang et al. 2019; Mirsadeghi, Royat, and Rezatofighi 2021) present clustering objective to train a neural network to discover clusters that accurately match semantic classes. These works maximize the discrete mutual information between augmented views to learn a clustering function. (Xiong et al. 2021; Van Gansbeke et al. 2021) adopt optical flow or saliency estimation methods to build two-stage methods.

Our unsupervised segmentation method is also related to another line of recent works on unsupervised scene decomposition (Locatello et al. 2020; Burgess et al. 2019; Greff et al. 2019). These methods represent a scene in terms of a collection of latent variables with the same representational format, perform an iterative encoding-decoding step followed by a comparison in pixel space.

Self-supervised Temporal Correspondence Learning temporal correspondence is a vital topic for unsupervised visual representation learning in videos and can be formulated as dense tracking or flow estimation.

Dense tracking is to predict mask in latter frames given the mask in the first frame. Many self-supervised methods have been developed to avoid costly human labour of annotations (Zhu et al. 2020; Tokmakov, Alahari, and Schmid 2017; Han, Xie, and Zisserman 2020; Oh et al. 2019; Voigtlaender et al. 2019; Wang et al. 2019). (Wang, Jabri, and Efros 2019; Jabri, Owens, and Efros 2020) use cycle-consistency as pretext task. However, current approaches based on cycle-consistency often lack pixel-level semantics and utilize image patches as nodes, making it difficult to apply on segmentation task.

Optical flow estimation problem has also been widely explored for pixel level temporal correspondence (Lucas, Kanade et al. 1981; Sun et al. 2018; Teed and Deng 2020). However, self-supervised optical flow estimation is still challenging and flow-based methods often suffer when faced with long-range correspondence.

Method

In this paper, we aim to provide representation for semantic segmentation in an end-to-end unsupervised way. Before introducing our solution, let's first analyze two important factors when applying contrastive learning to pixel level tasks.

Analysis from Image to Pixel Level

Contrastive learning is designed to maximize the representation similarity of two "views" that contain common semantics based on the premise that the representation does not collapse. For the image-level classification task, the mainstream method to generate positive pairs of "view" is to apply different augmentations on the same element (image). The high dimensionality of each image view guarantees that the pairs are distant from each other in the input space and this high diversity makes the model able to learn the common semantics from noise. When directly applying the conventional contrastive learning to pixel-level tasks, each pixel as the contrasting element does not contains high-level semantics and the low dimensionality makes it difficult to get pixel-level views with enough diversity for contrasting. In order to adapt the contrastive framework to pixel-level, we need to consider the following two questions:

Do we need global semantics or not? Too little information can be carried by a single pixel. Thus, it needs to be combined with surrounding pixels to form high-level features. Although a multi-layer convolutional network with large receptive fields can capture local features containing relatively high-level semantics, the contrastive learning taking a pixel as the basic contrast unit forces them to compete with each other, making it difficult to gradually learn the larger-scale higher-level features. Thus, to adjust contrastive learning to the pixel level, we need to focus on global semantics, as well as the local features to boost the performance. How to balance them is an important factor to build effective segmentation models (details in experiment section).

What is a difficult pair of pixel-level views? Also because of the low dimensionality of pixel-level views, as mentioned above, it is not feasible to directly generate diverse pairs by augmentation. Samples with insufficient diversity will lead to myopic models that fail to learn semantics effectively (Grill et al. 2020; Wallace and Hariharan 2020; Azabou et al. 2021). We need to explore another definition of difficult diverse pixel pairs, besides augmentation, for pixel-level tasks. This kind of pixel-to-pixel correspondence naturally exists between video frames. It is a feasible plan to take videos into the framework and simply adopting an optical flow algorithm (Sun et al. 2018; Teed and Deng 2020) can provide us these pixel correspondences. However, this kind of correspondence cannot generate view pairs with the largest diversity because it only links the same position of the same object as the pairs. For semantic segmentation, the most difficult kind is the pairs between the different positions of different individuals of the same category. How to generate them under the unsupervised framework and apply them to contrastive learning is the problem we want to solve. Experiments about the difficulty of views are shown below.

Cycle-Attention Contrastive Learning

Overview We treat the unsupervised segmentation task as the pixel-level representation learning and follow the contrastive learning framework to train the network. Contrastive learning is one of the self-supervised learning algorithms. Given a pair of positive samples x, x_+ and several negative ones $\{x_-\}$, contrastive learning makes the similarity of positive samples' representations z, z_+ closers than the negative ones $\{z_-\}$ to learn effective features:

$$target = -\log \frac{\exp(\mathrm{sim}(z, z_+)/\tau)}{\sum_{z' \in \{z_-\} \cup z_+} \exp(\mathrm{sim}(z, z')/\tau)}, \quad (1)$$

where τ denotes the temperature hyper-parameter and similarity function sim is usually defined as cosine similarity.

Commonly, the image-level contrastive learning method adopts different augmentations of the same image as the positive samples and different images as negative samples. However, as mentioned above, for pixel-level settings, another approach is needed to generate diverse positive samples. We employ videos as input, taking advantage of the relationships between video frames as basics. Given a video V_i and its frames $\{v_i^n\}$, to better handle the pixel level contrastive task, we design three heads above the backbone $f(\cdot)$ (see Fig. 2): the pixel head g_P taking charge of local semantics, the global head g_G for global features, and most importantly, the cycle-attention head g_C adopting cycle-consistency loss to implicit conduct contrastive learning for digging difficult diverse positive pairs and balance local features with global ones. The calculation flow is:

$$\begin{aligned} h &= f(v) \\ u,\ p,\ c &= g_G(h),\ g_P(h),\ g_C(h) \end{aligned} \quad (2)$$

In the following, we will introduce the three heads in detail.

Global and Pixel Head Global and pixel heads are multi-layered convolutional networks that map the representation of the backbone into the contrastive target space as (Chen et al. 2020a; He et al. 2020) do. Global head utilizes a global average pooling (GAP) layer to get global features and directly applies the contrastive loss adopting cosine similarity $\cos$ as the similarity measurement to guide backbones learning global semantics just as image-level contrastive learning frameworks do:

$$\mathcal{L}_G = -\log \frac{\exp(\cos(u, u_+)/\tau)}{\sum_{u' \in \{u_-\} \cup u_+} \exp(\cos(u, u')/\tau)}, \quad (3)$$

where u and u_+ are features of the frames from the same video and $\{u_-\}$ is a set of features from other video frames.

For the pixel head, p, p_+, and $\{p_-\}$ are treat as three sets of pixels $\{p^i\}$, $\{p_+^i\}$, and $\{p_-^i\}$. Similarly, features in $\{p_-^i\}$ can all be treated as negatives since they are pixels from other videos. While positive pairs are generated by matching pixels between the two frames $\{p^i\}$ and $\{p_+^i\}$ from the same video into matched pair (p^i, p_+^i). As videos naturally contain matchup between frames' pixels, we adopt global head's features which contain preliminary semantics to generate the affinity matrix A and guide the pixel head to learn pixel-level semantics:

$$\begin{aligned} A_{ij} &= \cos(u^i, u_+^j), \\ p_+^m &= p_+^{\mathrm{argmax}_j(A_{ij})} \\ \mathcal{L}_P &= -\log \frac{\exp(\cos(p^i, p_+^m)/\tau)}{\sum_{p^k \in \{p_-^i\} \cup p_+^m} \exp(\cos(p^i, p^k)/\tau)}, \end{aligned} \quad (4)$$

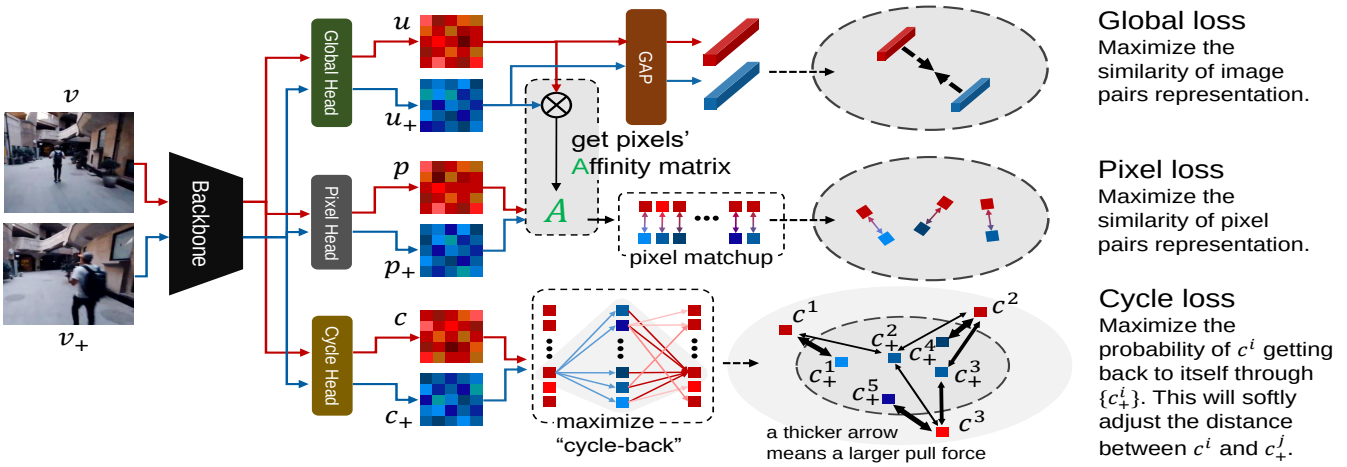

Figure 2: **The pipeline of our CACL model.** Given two frames v and v_+ from the same video, three heads adopt the output of the same backbone as input and map it to their own target spaces. Global head employs the global average pooling (GAP) to get global features and contrast them to learn the global semantics. Pixel head adopts the affinity matrix from global head to get the matchup between two sets of pixels $\{p^i\}$ and $\{p^i_+\}$. Pixel-level contrasting is conducted between them. The cycle head first maps $\{c^i\}$ to $\{c^i_+\}$ to get the similarity matrix between them and together with the mapping back process, we get the probability distribution of where c^i getting back (in $\{c^i\}$). The cycle loss maximizes the probability of back to themselves, which will softly adjust the distance between the pixels in $\{c^i\}$ and $\{c^i_+\}$ in the target space.

where u^i and u^j_+ are pixels in u and u_+.

These two heads preliminarily form a hierarchical structure that modifies the contrastive learning framework into pixel level. But the pixel positive pairs rely too much on the semantics learned by the global head. And the positive pairs are not difficult enough as Fig. 3 shows.

Cycle-Attention Head As mentioned above, the combination of global and pixel head still cannot find difficult enough positive pixel pairs, because the most similar pixels in two frames usually have little diversity. And the one-way dependence between the global and pixel head make it hard to balance them well. Inspired by (Jabri, Owens, and Efros 2020), we propose the cycle-attention head.

The same as the other two heads, the structure of the cycle-attention head is a multi-layer convolutional network that maps features to the target space. With the feature matrix $c \in \mathbb{R}^{H\times W,k}$ and $c_+ \in \mathbb{R}^{H\times W,k}$ of two frames in the same video, where k is the number of channels, the cycle-attention head forms a cycle mapping $\mathcal{M}$ through two jumpings between $\{c^i\}$ and $\{c^i_+\}$:

$$\mathcal{M} = \mathrm{softmax}(\langle c, c_+\rangle / \tau) \times \mathrm{softmax}(\langle c_+, c\rangle / \tau) \quad (5)$$

Note that the features $c^i, c^i_+ \in \mathbb{R}^k$ of each pixel are $l2$-normalized k-dimensional vectors, thus we adopt the inner product to denote the calculation of cosine similarity and the $\mathrm{softmax}$ function is applied on the last dimension. The $\mathrm{softmax}$ function changes the similarity into transition probabilities and we call each transition as a "jump". After two jumpings, $\mathcal{M} \in \mathbb{R}^{H\times W,H\times W}$, the similarity matrix among all the pairs of pixels in $\{c^i\}$, can be seen as the jump back distribution. A great representation will make most pixels jump back to themselves (called "cycle-back") to maintain the cycle-consistency (Jabri, Owens, and Efros 2020; Wang, Jabri, and Efros 2019), thus the optimization target can be written as:

$$\mathcal{L}_C = \mathrm{CrossEntropy}(\mathcal{M}, \mathbb{I}) \quad (6)$$

where $\mathrm{CrossEntropy}$ is cross entropy loss function and $\mathbb{I} \in \mathbb{R}^{H\times W,H\times W}$ is an identity matrix.

Analysis This cycle-consistency loss actually implicitly conducts the contrastive learning. It will form soft pixel-level positive and negative relationships among $\{c^i\}$ and $\{c^i_+\}$. Intuitively, to achieve a higher probability to cycle back through $\{c^i_+\}$, the optimization process will make c^i closer to a subset of $\{c^i_+\}$ that is far from other pixels in $\{c^i\}$ and at the same time, make c^i keep away from the other subsets of $\{c^i_+\}$ that are close to other c^j (see supplementary file for detailed explanation). Because of the different distances between c^i and all the pixels in $\{c^i_+\}$, the pull and push forces applied on c^i also satisfy a soft distribution (see the lower right of Fig. 2: a thicker arrow means a larger pull force). It is these automatic pull and push forces that make the cycle head works in an implicitly contrastive manner.

Managing the mentioned two problems Since these soft optimization signals exist among all the pairs of pixels, compared to the conventional contrastive methods that assign hard positive pairs, the cycle-attention head has the opportunity to implicitly dig positive pairs with high diversity (see Fig. 3). And due to the pixel-to-image-to-pixel calculation manner instead of the direct pixel-to-pixel contrasting, the cycle-attention head won't overly focus on local features and can find a better balance between the global and pixel features (see supplementary files for a detailed explanation). By the way, during training, c^i will gradually gets close to a group of pixels in $\{c^i_+\}$ to maximize the cycle-back probability. This can be seen as c^i gradually attending to a part of the frame. Thus, we call it the "cycle-attention" head.

According to (Chen et al. 2020a), we add negative samples in the cycle process to enhance the performance. The

new cycle path to calculate $\mathcal{M}$ is $\{c^i\} \rightarrow \{c^i_+\} \rightarrow (\{c^i\} + \{c^i_-\})$. Further more, another problem is that the above optimization target of $\mathcal{M}$ is a one-hot vector for each c^i to guide the pixels only jumping back to themselves, but actually many pixels in the same frame have relevant semantics. To this end, we propose a small trick that modifies the one-hot target to $\mathrm{softmax}(\langle u, u_+ \rangle / \tau)$ and adopt the KL-divergence as the loss function. That is to say we adopt the global head to describe the relationships among pixels in the frame, and we call the trick "soft cycle".

In summary, for the whole proposed CACL method, the total training loss is: $\mathcal{L} = \mathcal{L}_G + \mathcal{L}_P + \mathcal{L}_C$.

Pixel Shuffle Due to the requirement of keeping the size of the input, the convolutional network employs padding. This makes the calculation flow of pixels in different positions different because pixels closer to the edge of images are convolved with more paddings. Thus, the whole network has an opportunity to leak the position information of pixels and according to this position information, the network can more easily finish the pixel contrastive task at the edges. Even worse, the optimization target of the cycle-attention head can be totally solved by the position information, making it unable to learn useful semantics.

To this end, we propose the pixel shuffle trick. It adopts two random rolling distances (a, b) in two directions to modify the input image $X \in \mathbb{R}^{H,W}$:

$$X_{i,j} = X_{(i+a) \bmod H,\ (j+b) \bmod W} \tag{7}$$

After the forward propagation, we roll the output back. Pixel shuffle solves the position leak problem by making pixels at every position have the same chance to contact with zero paddings. In the ablation study, we analyze its effectiveness.

Implementation Details

Contrastive algorithm Our CACL framework can be applied to various kinds of contrastive learning algorithms. In our experiments, we follow MoCo (He et al. 2020) to implement our model. There is an online and a momentum version of networks in the model. z is generated by the online network that updates its parameters by the optimizer. z_- and z_+ are generated by the momentum network which updates parameters in a momentum manner. The momentum network is only used in the training phase. $\{z_-\}$ is stored in a large first-in-first-out queue that updates with the training process. The three heads in CACL maintain their own queues respectively. All the hyper-parameters in CACL related to MoCo are the same with MoCo-v2 (Chen et al. 2020b).

Network structure As for the network structure, we utilize ResNet (He et al. 2016) as our backbone f. For the semantic segmentation task, we remove the downsampled layers in the last two stages and adopt the dilated convolution layer (Chen et al. 2017), following (Zhao et al. 2017). Thus, the whole downsample rate is 8. The structures of the three heads are the same. Both are MLP containing two convolution layers with an 1×1 kernel which can be seen as the pixel-level linear layer with the same number of parameters. The number of output channels for the first layer keeps the same as its input channels and it is followed by a ReLU activation layer to build the non-linear structure. The second layer's output channel is 128, the same as MoCo-v2.

Augmentation We adopt almost the same augmentation as MoCo-v2 (Chen et al. 2020b) to preprocess input frames. The exclusive difference lies that v and v_+ are two crops at the same position of two different frames, instead of two crops generated from different positions of the same image.

Experiments

In this section, we evaluate the proposed CACL on PASCAL-VOC (Everingham et al. 2010) and Cityscapes (Cordts et al. 2016) datasets. In ablation study, we also provide the experimental supports for the analysis mentioned above.

Setup

Dataset The unsupervised training is conducted on ImageNet (Deng et al. 2009) and Kinetics-400 (Carreira and Zisserman 2017) datasets. Kinetics-400 is a large-scale video dataset that contains 240k training videos covering 400 action categories. We train the models on their training splits. The experimental evaluations are conducted on two commonly used benchmarks, PASCAL-VOC (Everingham et al. 2010) and Cityscapes (Cordts et al. 2016) datasets. The training and evaluation splits in PASCAL-VOC contain about 10k augmented images covering 20 classes, and 3.5k images covering 19 classes in Cityscapes.

Training Setup We train our CACL on ImageNet and Kinetics with 16 GPU and the batch size on each GPU is 32. For each video in Kinetics, we randomly choose one pair of frames with stride 8. The optimizer we adopt for training is LARS (You, Gitman, and Ginsburg 2017) with SGD. The momentum is set to 0.9, while $1e^{-6}$ for weight decay and 0.001 for the trust coefficient of LARS. The initial learning rate is 1.0 and decayed with a cosine schedule scheme (Loshchilov and Hutter 2016). The contrastive dimension and temperature τ are 128 and 0.07 just as MoCo (He et al. 2020). For CACL, the size of the queue for negative samples is 16384, a quarter of MoCo to reduce the complexity. To reduce the information leak caused by batch normalization, we adopted shuffle-bn proposed in MoCo.

Evaluation Setup To evaluate the representation quality learned by the unsupervised models, we adopt the following two evaluation protocols:

Linear Protocol: Linear evaluation that tests models' linear separability is a frequently-used protocol to evaluate the learned representation (Chen et al. 2020a; He et al. 2020). For all the models, we remove their projection heads and take the final output of the backbone as the learned representation. We take the frozen representation as to the input of a linear classifier and train it with the ground truth of evaluation datasets. The linear classifier is one convolutional layer with a kernel of size 1×1. We train the classifier for 300 epochs under 0.6 lr and the weight decay is set to 0.

Limited Sample Supervision: We also test the model's performance with small numbers of data with annotations to

Table 1: **Ablations on Cityscapes**. Experiments are mainly based on ResNet18. The unsupervised models are trained on Kinetics. ResNet50 backbones adopt initial parameter weights from MoCo-v2 trained on ImageNet. "pix head", "glo head", and "cyc head" denote pixel, global, and cycle-attention heads.

(a) **Balance between local and global information** They need to work together and the cycle-attention head can better balance them.

backbone	pix head	glo head	cyc head	mIoU
Res18	✓	✗	✗	28.7
Res18	✗	✓	✗	33.2
Res18	✗	✓	✓	34.3
Res18	✓	✓	✗	35.9
Res18	✓	✓	✓	38.2
Res50	✗	✓	✗	44.5
Res50	✓	✓	✗	46.1
Res50	✓	✓	✓	49.1

(b) **Difficulty of views** Harder views lead to better results. The cycle head implicitly generates the hardest ones.

type of pos pair	stride	mIoU
only augmentation	0	27.5
cross video frames	2	28.4
cross video frames	4	29.7
cross video frames	8	33.1
decide by global head	4	31.6
decide by global head	8	35.9
+ cycle-attention head	4	34.5
+ cycle-attention head	8	38.2

(c) **Key elements in CACL** Pixel shuffle helps to avoid shortcut. Soft cycle boosts the performance a little.

backbone	elements	mIoU
Res18	simple model	34.2
Res18	+ pixel shuffle trick	37.1
Res18	+ soft cycle trick	38.2
R50	simple model	45.6
R50	+ pixel shuffle trick	47.3
R50	+ soft cycle trick	49.1

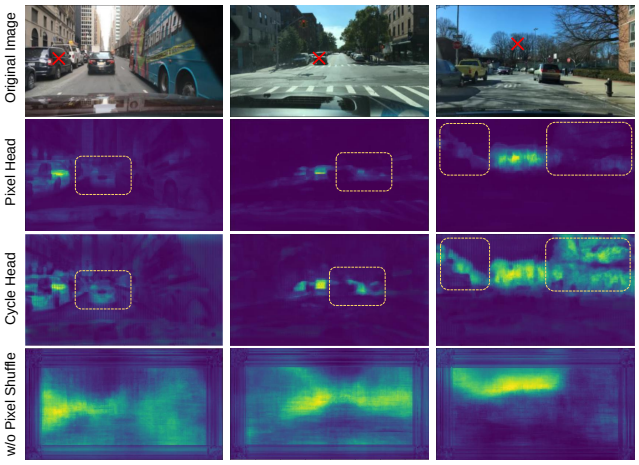

Figure 3: **Visualization**. Given a target pixel (noted by red crosses) from the frame and models under the early stage of training, we visualize the similarities between representation of the target pixel and the pixels in the next 8th frame. We can see that the cycle-attention head digs more difficult positive pairs that link different parts from different objects. In addition, in the last row, we show that without pixel shuffle, there will be shortcuts in cycle-attention head.

evaluate the representation. The same as the linear protocol, the evaluated model is the pre-trained backbone with a randomly initialized linear classifier. Here, we do not freeze the backbone, instead, train the whole network end-to-end with small number of samples. We fine-tune the models for 1000 epochs on dataset under 0.3 learning rate.

Ablation Study

This section provides ablation studies about the proposed CACL on Cityscapes dataset. All models are trained on Kinetics-400 in an unsupervised manner. In addition, we give out some experimental supports on the analysis conclusions mentioned above.

Balance between local and global information After a brief analysis, we consider that local and global features need to work cooperatively to learn powerful representations. To this end, we design the pixel head that utilizes information from the global head as the learning guide to balance local-global relationships. Meanwhile, the proposed cycle-attention head implicitly learning the soft affinity matrix among pixel and global features further enhances their balance. In Tab. 1a, we report the performances of different settings adopting different levels of contrastive learning. Note that when only adopting pixel head, the positive pair of views are simply the pixels at the same position in two video frames. From the experiment results, we can see that a stand-alone pixel head cannot get high performance. (Wang et al. 2020) draws similar conclusions to ours. Focusing too much on local information makes it difficult for models to learn large-scale high-level semantics. Adopting global and local heads together outperforms the performance of two heads used independently. This confirms the conclusions of our previous analysis. Furthermore, after adding the cycle-attention head, the further enhanced interaction between local and global features boosts the performance once again (+ 2.6 mIoU in average), just as expected.

Difficulty of pixel-level positive views The positive pair of views that possess enough diversity is one of the key factors for the contrastive learning to learn effective representations. Previous works adopt complex augmentations to generate image-level positive pairs with high diversity. However, utilizing augmentations is not an effective method for pixel-level views as mentioned above. From the results listed in Tab. 1b, we can see that directly adopting augmentation indeed achieves a relative poor performance. To increase the difficulty (diversity), we first take advantage of the relationship between video frames. The larger the stride (in reasonable range) between two frames, the pixels at the same position of the two frames have larger diversity. It can be seen that, the performance is gradually improved. Next, we matching the pixels between the two frames by the features generated by the global head to dig positives from different parts of the object, instead of the same part. The performance boosts once again. Finally, we add cycle-attention head. The "one-to-all" contrastive scheme makes it possible to generate pairs from different parts of different objects in the same category (see Fig. 3), further increasing the difficulty and diversity. As expected, this setting achieves the best performance (+ 2.6 mIoU in average).

Some key elements in CACL To avoid the information leak caused by padding in convolutions layers, we design the pixel shuffle trick. Tab. 1c shows it effectiveness. It is seen

Table 2: **Linear protocol results on Cityscapes**. "mACC", "IN", and "2 heads" respectively denote mean accuracy, ImageNet, and pixel head + global head.

method	train data	mAcc	mIoU
ResNet50 + supervised	ImageNet		43.8
ResNet50 + SimCLR	ImageNet		39.9
ResNet50 + BYOL	ImageNet		38.2
ResNet50 + MoCo-v2	ImageNet		44.5
ResNet50 + VINCE	R2V2 (Gordon et al. 2020)		26.7
ResNet50 + FlowE	BDD100K (Yu et al. 2018)		45.6
ResNet50 + DenseCL	IN + Kinetics-400	51.6	45.3
ResNet50 + PixPro	IN + Kinetics-400	51.8	45.4
ResNet50 + SetSim	IN + Kinetics-400	52.6	46.3
ResNet18 + MoCo-v2	Kinetics-400	40.0	33.2
ResNet18 + 2 heads	Kinetics-400	42.6	35.9
ResNet18 + CACL	Kinetics-400	45.7	38.2
ResNet50 + MoCo-v2	IN + Kinetics-400	50.7	44.6
ResNet50 + 2 heads	IN + Kinetics-400	53.7	46.1
ResNet50 + CACL	IN + Kinetics-400	**57.4**	**49.1**

that without pixel shuffle the performance of CACL is even worse than the two-heads model. This is because the target of the cycle-attention head can be solved with only the position information, which invalidates the cycle-attention head and also affects the learning of representations. Pixel shuffle solves this problem successfully and visualization is shown in Fig. 3. To solve this problem we also try to adopt padding of "reflect" mode, but it does not solve the problem completely. This may be because the receptive field of the network is larger than the input image, thus, the model can still recognize the repeated part as the boundary. We also design to use soft cycle trick for CACL, instead of only jumping back to self. This trick provides models with more hierarchical optimization signals and this thought is also the key idea of knowledge distillation (Hinton, Vinyals, and Dean 2015). Results show that it considerably improves performance.

Main Results

In this section, we compare our method with recent strong baselines on Cityscapes and PASCAL-VOC datasets. We adopt mIoU and mean accuracy as our metrics.

Linear Protocol Tab. 2 reports the performance on Cityscapes. When adopting ResNet50 as the backbone, we load the weights of MoCo-v2 pre-trained on ImageNet as the initial weight and unsupervised train the proposed model on the video dataset Kinetics-400. For a fair comparison, we also train the baseline MoCo-v2 under this setting and it can be seen that the introduction of Kinetics dataset is not a significant factor affecting the performance. In the first place, the contrastive learning frameworks are competitive with the supervised method on semantic segmentation under the linear protocol. Compared to the DenseCL (Wang et al. 2020), PixPro (Xie et al. 2020), and SetSim (Wang et al. 2021), models similarly with a global and a pixel head, our two heads setting achieves competitive results with much simpler design, mainly due to the global-local interaction through the affine matrix A. Compared with the image-level contrastive models, our CACL improves the performance by **4.5** mIoU and 6.7 mAcc. As an end-to-end method, our CACL outperforms the two-stage model FlowE (Xiong et al. 2021) that adopts off-the-shelf optical flow results.

Tab. 3 reports the performance on PASCAL-VOC. The training setting is the same as the Cityscapes mentioned above. We can see that compared to other unsupervised learning methods, contrastive learning frameworks generate better representation and achieve much higher performances. But the image-level models are still not competitive with the supervised method. Our CACL achieves the state-of-the-art results, **4.5** mIoU higher than the baseline, and also outperforms the supervised method, revealing the effectiveness of the cycle-attention.

Table 3: **Linear protocol results on PASCAL-VOC**. "mACC", "IN", and "2 heads" respectively denote mean accuracy, ImageNet, and pixel head + global head.

method	train data	mAcc	mIoU
Co-Occurence (Isola et al. 2015)	VOC		13.5
CMP (Zhan et al. 2019)	YouTube		16.5
Colorization (Zhang, Isola, and Efros 2016)	ImageNet		25.5
IIC (Ji, Henriques, and Vedaldi 2019)	coco-stuff		28.0
Feature Clustering	VOC		35.2
Instance Discrimination (Wu et al. 2018)	ImageNet		26.8
ResNet50 + SimCLR (Chen et al. 2020a)	ImageNet	57.1	46.3
ResNet50 + MoCo-v2 (Chen et al. 2020b)	ImageNet	59.4	48.2
ResNet50 + InfoMin (Tian et al. 2020)	ImageNet	58.8	48.0
ResNet50 + Supervised	ImageNet	64.8	50.3
ResNet50 + MoCo-v2 (Chen et al. 2020b)	IN + Kinetics	59.4	48.1
ResNet50 + CACL	IN + Kinetics	**64.8**	**52.6**

Table 4: **Results (mIoU) with limited samples on Cityscapes**. "2 heads" means pixel head + global head. Models are trained on Kinetics. In this group of experiments, we adopt ResNet18 as the backbone.

	1%	5%	10%	100%
MoCo-v2	24.8	35.0	42.0	62.6
2 heads	31.4 *(+6.6)*	37.5 *(+2.5)*	45.6 *(+3.6)*	63.3 *(+0.7)*
CACL	34.3 *(+9.5)*	43.1 *(+8.1)*	47.8 *(+5.8)*	64.5 *(+1.9)*

Limited sample supervision Besides the linear protocol, we also adopt the limited sample supervision protocol to verify the proposed method. This experiment can reveal the transferability of the learned representation to downstream tasks with a small number of data. We randomly sample a subset of the target dataset and end-to-end fine-tune the model pre-trained by unsupervised methods. The results are shown in Tab. 4. We pre-train the models on Kinetics and report the performance fine-tuned under 1%, 5%, and 10% training samples of Cityscapes. Results show that CACL outperforms the strong baseline and the performance gap becomes larger when the number of samples goes smaller.

Conclusion

In this paper, we first analyze some key points to design a pixel-level contrastive framework for dense tasks, based on which we propose the Cycle-Attention Contrastive Learning (CACL) method, an end-to-end model for completely unsupervised semantic segmentation. CACL introduces the cycle consistency into the contrastive framework which is able to dig difficult positive pairs in an implicit manner and better balance the local and global features. Experimental results show that the proposed CACL model considerably outperforms the strong baselines on PASCAL-VOC and Cityscapes datasets by 4.5 mIoU. We hope CACL can provide new insight for community to build unsupervised contrastive-learning-based semantic segmentation models.

Acknowledgements

This work is supported in part by the National Key R&D Program of China, No. 2017YFA0700800, National Natural Science Foundation of China under Grants 61772332 and Shanghai Qi Zhi Institute, SHEITC (018-RGZN-02046).

References

Agrawal, P.; Carreira, J.; and Malik, J. 2015. Learning to see by moving. In *ICCV*, 37–45.

Asano, Y. M.; Patrick, M.; Rupprecht, C.; and Vedaldi, A. 2020. Labelling unlabelled videos from scratch with multi-modal self-supervision. *arXiv preprint*.

Azabou, M.; Azar, M. G.; Liu, R.; Lin, C.-H.; Johnson, E. C.; Bhaskaran-Nair, K.; Dabagia, M.; Hengen, K. B.; Gray-Roncal, W.; Valko, M.; et al. 2021. Mine Your Own vieW: Self-Supervised Learning Through Across-Sample Prediction. *arXiv preprint*.

Bachman, P.; Hjelm, R. D.; and Buchwalter, W. 2019. Learning representations by maximizing mutual information across views. *arXiv preprint*.

Burgess, C. P.; Matthey, L.; Watters, N.; Kabra, R.; Higgins, I.; Botvinick, M.; and Lerchner, A. 2019. Monet: Unsupervised scene decomposition and representation. *arXiv preprint*.

Caron, M.; Bojanowski, P.; Joulin, A.; and Douze, M. 2018. Deep clustering for unsupervised learning of visual features. In *ECCV*, 132–149.

Caron, M.; Misra, I.; Mairal, J.; Goyal, P.; Bojanowski, P.; and Joulin, A. 2020a. Unsupervised learning of visual features by contrasting cluster assignments. *arXiv preprint*.

Caron, M.; Misra, I.; Mairal, J.; Goyal, P.; Bojanowski, P.; and Joulin, A. 2020b. Unsupervised learning of visual features by contrasting cluster assignments. *arXiv preprint*.

Carreira, J.; and Zisserman, A. 2017. Quo vadis, action recognition? a new model and the kinetics dataset. In *CVPR*, 6299–6308.

Chen, L.-C.; Papandreou, G.; Kokkinos, I.; Murphy, K.; and Yuille, A. L. 2017. Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs. *TPAMI*, 40(4): 834–848.

Chen, T.; Kornblith, S.; Norouzi, M.; and Hinton, G. 2020a. A simple framework for contrastive learning of visual representations. In *ICML*, 1597–1607. PMLR.

Chen, X.; Fan, H.; Girshick, R.; and He, K. 2020b. Improved baselines with momentum contrastive learning. *arXiv preprint*.

Chen, X.; and He, K. 2020. Exploring Simple Siamese Representation Learning. *arXiv preprint*.

Cordts, M.; Omran, M.; Ramos, S.; Rehfeld, T.; Enzweiler, M.; Benenson, R.; Franke, U.; Roth, S.; and Schiele, B. 2016. The cityscapes dataset for semantic urban scene understanding. In *CVPR*, 3213–3223.

Deng, J.; Dong, W.; Socher, R.; Li, L.-J.; Li, K.; and Fei-Fei, L. 2009. Imagenet: A large-scale hierarchical image database. In *CVPR*, 248–255. Ieee.

Everingham, M.; Van Gool, L.; Williams, C. K.; Winn, J.; and Zisserman, A. 2010. The pascal visual object classes (voc) challenge. *International journal of computer vision*, 88(2): 303–338.

Fabbri, M.; Lanzi, F.; Calderara, S.; Palazzi, A.; Vezzani, R.; and Cucchiara, R. 2018. Learning to detect and track visible and occluded body joints in a virtual world. In *ECCV*, 430–446.

Fernando, B.; Bilen, H.; Gavves, E.; and Gould, S. 2017. Self-supervised video representation learning with odd-one-out networks. In *CVPR*, 3636–3645.

Gordon, D.; Ehsani, K.; Fox, D.; and Farhadi, A. 2020. Watching the world go by: Representation learning from unlabeled videos. *arXiv preprint*.

Greff, K.; Kaufman, R. L.; Kabra, R.; Watters, N.; Burgess, C.; Zoran, D.; Matthey, L.; Botvinick, M.; and Lerchner, A. 2019. Multi-object representation learning with iterative variational inference. In *ICML*, 2424–2433. PMLR.

Grill, J.-B.; Strub, F.; Altché, F.; Tallec, C.; Richemond, P. H.; Buchatskaya, E.; Doersch, C.; Pires, B. A.; Guo, Z. D.; Azar, M. G.; et al. 2020. Bootstrap your own latent: A new approach to self-supervised learning. *arXiv preprint*.

Han, T.; Xie, W.; and Zisserman, A. 2020. Self-supervised co-training for video representation learning. *arXiv preprint*.

He, K.; Fan, H.; Wu, Y.; Xie, S.; and Girshick, R. 2020. Momentum contrast for unsupervised visual representation learning. In *CVPR*, 9729–9738.

He, K.; Zhang, X.; Ren, S.; and Sun, J. 2016. Deep residual learning for image recognition. In *CVPR*, 770–778.

Hinton, G.; Vinyals, O.; and Dean, J. 2015. Distilling the knowledge in a neural network. *arXiv preprint*.

Hjelm, R. D.; Fedorov, A.; Lavoie-Marchildon, S.; Grewal, K.; Bachman, P.; Trischler, A.; and Bengio, Y. 2018. Learning deep representations by mutual information estimation and maximization. *arXiv preprint*.

Hwang, J.-J.; Yu, S. X.; Shi, J.; Collins, M. D.; Yang, T.-J.; Zhang, X.; and Chen, L.-C. 2019. Segsort: Segmentation by discriminative sorting of segments. In *ICCV*, 7334–7344.

Iizuka, S.; Simo-Serra, E.; and Ishikawa, H. 2016. Let there be color! Joint end-to-end learning of global and local image priors for automatic image colorization with simultaneous classification. *ACM Transactions on Graphics*, 35(4): 1–11.

Isola, P.; Zoran, D.; Krishnan, D.; and Adelson, E. H. 2015. Learning visual groups from co-occurrences in space and time. *arXiv preprint*.

Jabri, A.; Owens, A.; and Efros, A. A. 2020. Space-Time Correspondence as a Contrastive Random Walk. In *NeurIPS*.

Jayaraman, D.; and Grauman, K. 2015. Learning image representations tied to ego-motion. In *ICCV*, 1413–1421.

Ji, X.; Henriques, J. F.; and Vedaldi, A. 2019. Invariant information clustering for unsupervised image classification and segmentation. In *ICCV*, 9865–9874.

Larsson, G.; Maire, M.; and Shakhnarovich, G. 2017. Colorization as a proxy task for visual understanding. In *CVPR*, 6874–6883.

LeCun, Y.; Bengio, Y.; and Hinton, G. 2015. Deep learning. *nature*, 521(7553): 436–444.

Lin, T.-Y.; Maire, M.; Belongie, S.; Hays, J.; Perona, P.; Ramanan, D.; Dollár, P.; and Zitnick, C. L. 2014. Microsoft coco: Common objects in context. In *ECCV*, 740–755. Springer.

Liu, S.; Zhong, G.; De Mello, S.; Gu, J.; Jampani, V.; Yang, M.-H.; and Kautz, J. 2018. Switchable temporal propagation network. In *ECCV*, 87–102.

Locatello, F.; Weissenborn, D.; Unterthiner, T.; Mahendran, A.; Heigold, G.; Uszkoreit, J.; Dosovitskiy, A.; and Kipf, T. 2020. Object-centric learning with slot attention. *arXiv preprint*.

Long, J.; Shelhamer, E.; and Darrell, T. 2015. Fully convolutional networks for semantic segmentation. In *CVPR*, 3431–3440.

Loshchilov, I.; and Hutter, F. 2016. Sgdr: Stochastic gradient descent with warm restarts. *arXiv preprint*.

Lotter, W.; Kreiman, G.; and Cox, D. 2016. Deep predictive coding networks for video prediction and unsupervised learning. *arXiv preprint*.

Lucas, B. D.; Kanade, T.; et al. 1981. An iterative image registration technique with an application to stereo vision. In *IJCAI*. Vancouver, British Columbia.

Mathieu, M.; Couprie, C.; and LeCun, Y. 2015. Deep multi-scale video prediction beyond mean square error. *arXiv preprint*.

Mirsadeghi, S. E.; Royat, A.; and Rezatofighi, H. 2021. Unsupervised Image Segmentation by Mutual Information Maximization and Adversarial Regularization. *IEEE Robotics and Automation Letters*, 6(4): 6931–6938.

Misra, I.; Zitnick, C. L.; and Hebert, M. 2016. Shuffle and learn: unsupervised learning using temporal order verification. In *ECCV*, 527–544. Springer.

Nguyen, D. T.; Dax, M.; Mummadi, C. K.; Ngo, T. P. N.; Nguyen, T. H. P.; Lou, Z.; and Brox, T. 2019. Deepusps: Deep robust unsupervised saliency prediction with self-supervision. *arXiv preprint*.

Noroozi, M.; and Favaro, P. 2016. Unsupervised learning of visual representations by solving jigsaw puzzles. In *ECCV*, 69–84. Springer.

Noroozi, M.; Vinjimoor, A.; Favaro, P.; and Pirsiavash, H. 2018. Boosting self-supervised learning via knowledge transfer. In *CVPR*, 9359–9367.

Oh, S. W.; Lee, J.-Y.; Xu, N.; and Kim, S. J. 2019. Video object segmentation using space-time memory networks. In *ICCV*, 9226–9235.

Oord, A. v. d.; Li, Y.; and Vinyals, O. 2018. Representation learning with contrastive predictive coding. *arXiv preprint*.

Pathak, D.; Krahenbuhl, P.; Donahue, J.; Darrell, T.; and Efros, A. A. 2016. Context encoders: Feature learning by inpainting. In *CVPR*, 2536–2544.

Qin, X.; Zhang, Z.; Huang, C.; Gao, C.; Dehghan, M.; and Jagersand, M. 2019. Basnet: Boundary-aware salient object detection. In *CVPR*, 7479–7489.

Srivastava, N.; Mansimov, E.; and Salakhudinov, R. 2015. Unsupervised learning of video representations using lstms. In *ICML*, 843–852. PMLR.

Sun, D.; Yang, X.; Liu, M.-Y.; and Kautz, J. 2018. Pwc-net: Cnns for optical flow using pyramid, warping, and cost volume. In *CVPR*, 8934–8943.

Sun, K.; Zhao, Y.; Jiang, B.; Cheng, T.; Xiao, B.; Liu, D.; Mu, Y.; Wang, X.; Liu, W.; and Wang, J. 2019. High-resolution representations for labeling pixels and regions. *arXiv preprint*.

Teed, Z.; and Deng, J. 2020. Raft: Recurrent all-pairs field transforms for optical flow. In *ECCV*, 402–419. Springer.

Tian, Y.; Krishnan, D.; and Isola, P. 2019. Contrastive multiview coding. *arXiv preprint*.

Tian, Y.; Sun, C.; Poole, B.; Krishnan, D.; Schmid, C.; and Isola, P. 2020. What makes for good views for contrastive learning. *arXiv preprint*.

Tokmakov, P.; Alahari, K.; and Schmid, C. 2017. Learning video object segmentation with visual memory. In *ICCV*, 4481–4490.

Tung, H. F.; Tung, H.; Yumer, E.; and Fragkiadaki, K. 2017. Self-supervised Learning of Motion Capture. *CoRR*.

Van Gansbeke, W.; Vandenhende, S.; Georgoulis, S.; and Van Gool, L. 2021. Unsupervised Semantic Segmentation by Contrasting Object Mask Proposals. *arXiv preprint*.

Vincent, P.; Larochelle, H.; Bengio, Y.; and Manzagol, P.-A. 2008. Extracting and composing robust features with denoising autoencoders. In *ICML*, 1096–1103.

Voigtlaender, P.; Chai, Y.; Schroff, F.; Adam, H.; Leibe, B.; and Chen, L.-C. 2019. Feelvos: Fast end-to-end embedding learning for video object segmentation. In *CVPR*, 9481–9490.

Vondrick, C.; Pirsiavash, H.; and Torralba, A. 2016a. Anticipating visual representations from unlabeled video. In *CVPR*, 98–106.

Vondrick, C.; Pirsiavash, H.; and Torralba, A. 2016b. Generating videos with scene dynamics. *arXiv preprint*.

Wallace, B.; and Hariharan, B. 2020. Extending and analyzing self-supervised learning across domains. In *ECCV*, 717–734. Springer.

Wang, Q.; Zhang, L.; Bertinetto, L.; Hu, W.; and Torr, P. H. 2019. Fast online object tracking and segmentation: A unifying approach. In *CVPR*, 1328–1338.

Wang, X.; Jabri, A.; and Efros, A. A. 2019. Learning correspondence from the cycle-consistency of time. In *CVPR*, 2566–2576.

Wang, X.; Zhang, R.; Shen, C.; Kong, T.; and Li, L. 2020. Dense Contrastive Learning for Self-Supervised Visual Pre-Training. *arXiv preprint*.

Wang, Z.; Li, Q.; Zhang, G.; Wan, P.; Zheng, W.; Wang, N.; Gong, M.; and Liu, T. 2021. Exploring Set Similarity for Dense Self-supervised Representation Learning. *arXiv preprint*.

Wei, D.; Lim, J. J.; Zisserman, A.; and Freeman, W. T. 2018. Learning and using the arrow of time. In *CVPR*, 8052–8060.

Wu, Z.; Xiong, Y.; Yu, S. X.; and Lin, D. 2018. Unsupervised feature learning via non-parametric instance discrimination. In *CVPR*, 3733–3742.

Xie, Z.; Lin, Y.; Zhang, Z.; Cao, Y.; Lin, S.; and Hu, H. 2020. Propagate Yourself: Exploring Pixel-Level Consistency for Unsupervised Visual Representation Learning. *arXiv preprint*.

Xiong, Y.; Ren, M.; Zeng, W.; and Urtasun, R. 2021. Self-Supervised Representation Learning from Flow Equivariance. *arXiv preprint*.

You, Y.; Gitman, I.; and Ginsburg, B. 2017. Large batch training of convolutional networks. *arXiv preprint*.

Yu, F.; Xian, W.; Chen, Y.; Liu, F.; Liao, M.; Madhavan, V.; and Darrell, T. 2018. Bdd100k: A diverse driving video database with scalable annotation tooling. *arXiv preprint*, 2(5): 6.

Zhan, X.; Pan, X.; Liu, Z.; Lin, D.; and Loy, C. C. 2019. Self-supervised learning via conditional motion propagation. In *CVPR*, 1881–1889.

Zhang, R.; Isola, P.; and Efros, A. A. 2016. Colorful image colorization. In *ECCV*, 649–666. Springer.

Zhao, H.; Shi, J.; Qi, X.; Wang, X.; and Jia, J. 2017. Pyramid scene parsing network. In *CVPR*, 2881–2890.

Zhu, F.; Zhang, L.; Fu, Y.; Guo, G.; and Xie, W. 2020. Self-supervised Video Object Segmentation. *arXiv preprint*.

Zhuang, C.; Zhai, A. L.; and Yamins, D. 2019. Local aggregation for unsupervised learning of visual embeddings. In *ICCV*, 6002–6012.