SELF-SUPERVISED DENSE VISUAL REPRESENTATION LEARNING

by

Timoteos Onur Özçelik

B.S., Computer Engineering, Galatasaray University, 2019

B.S., Industrial Engineering, Galatasaray University, 2020

Submitted to the Institute for Graduate Studies in
Science and Engineering in partial fulfillment of
the requirements for the degree of
Master of Science

Graduate Program in Computer Engineering

Boğaziçi University

2024

ACKNOWLEDGEMENTS

I would like to thank my supervisors Prof. Lale Akarun and Berk Gökberk for their consistent support and guidance during my master education. Without them I would not have made it through my masters degree!

I would like to express my gratitude to my family who encouraged and supported me in every way throughout my life. When I was unwell, they met my every need with love and enabled me to complete this process in a healthy way. I am also very grateful to my friend Yusuf Peker for his friendship, love and support from my first day in the hospital until now.

Special thanks to Specialist Dr. Yener Georgi Yassıbaş, Dr. Ayşe Ülkü ASLAN GÜVEN, Physiotherapist Mr. Fehmi and Hydrotherapist Mrs. Belma, who helped me recover with their love and services from the first day I got sick.

ABSTRACT

SELF-SUPERVISED DENSE VISUAL REPRESENTATION LEARNING

In this thesis, we study on self-supervised dense visual representation learning. We propose three different approaches to be used in dense tasks such as object detection and instance segmentation. In the first one, we propose consistency regularization at both image-level and pixel-level to alleviate the false negative pairing problem in the contrastive learning. In the second, we combine two positive-pair matching strategies implemented at the pixel-level, based on similarity and distance, to eliminate the weaknesses of these strategies. In the third one, we adapt the relational self-supervised framework (ReSSL) proposed in image-level to dense-level. Contrary to common practice, we pre-trained these approaches on the COCO dataset, not on ImageNet, and evaluated them on the Pascal VOC object detection and COCO object detection and instance segmentation tasks. In our experiments, we observed that consistency regularization, using two separate pixel-level matching strategies together, and stimulating ReSSL to the pixel-level lead to an increase in performance, both quantitatively and qualitatively.

ÖZET

KENDİ KENDİNE DENETİMLİ YOĞUN GÖRSEL TEMSİL ÖĞRENİMİ

Bu tezde, kendi kendine denetimli yoğun görsel temsil öğrenimi üzerine çalışıyoruz. Nesne algılama ve örnek segmentasyonu gibi yoğun görevlerde kullanılmak üzere üç farklı yaklaşım öneriyoruz. İlkinde, kontrastlı öğrenmedeki yanlış negatif eşleştirme sorununu hafifletmek için hem görüntü hem de piksel düzeyinde tutarlılık düzenlemesi öneriyoruz. İkincisinde, bu stratejilerin zayıflıklarını ortadan kaldırmak için piksel düzeyinde uygulanan benzerlik ve mesafeye dayalı iki pozitif çift eşleştirme stratejisini birleştiriyoruz. Üçüncüsünde, görüntü seviyesinde önerilen ilişkisel öz denetimli çerçeveyi (ReSSL) yoğun seviyeye uyarlıyoruz. Yaygın uygulamanın aksine, bu yaklaşımları ImageNet üzerinde değil COCO veri kümesi üzerinde ön eğitime tabi tuttuk ve Pascal VOC nesne algılama ve COCO nesne algılama ve örnek segmentasyonu görevleri üzerinde değerlendirdik. Deneylerimizde, tutarlılık düzenlemesi, iki ayrı piksel düzeyinde eşleştirme stratejisinin birlikte kullanılması ve ReSSL'in piksel düzeyine uyarılmasının hem nicel hem de nitel olarak performansta artışa yol açtığını gözlemledik.

TABLE OF CONTENTS

ACKNOWLEDGEMENTS . . . iii
ABSTRACT . . . iv
ÖZET . . . v
LIST OF FIGURES . . . viii
LIST OF TABLES . . . ix
LIST OF SYMBOLS . . . x
LIST OF ACRONYMS/ABBREVIATIONS . . . xi
1. INTRODUCTION . . . 1
1.1. Contributions . . . 2
1.2. Organization of the Thesis . . . 4
2. RELATED WORK . . . 5
2.1. Hand-crafted Predictive Pretext Tasks . . . 5
2.2. Instance Discrimination-based Methods . . . 5
2.3. Contrastive Learning-based Methods . . . 7
2.4. Distillation-based Methods . . . 10
2.5. Clustering-based Methods . . . 12
2.6. Information Theory-based Methods . . . 14
2.7. Graph-partitioning-based Methods . . . 15
2.8. Masked Image Modeling (MIM)-based Methods . . . 17
3. METHODOLOGY . . . 19
3.1. Baselines . . . 19
3.1.1. Dense Contrastive Learning (DenseCL) . . . 19
3.1.2. Pixel-Level Contrastive Learning (PixContrast) . . . 21
3.1.3. Consistent Contrast (CO2) . . . 22
3.1.4. Relational Self-supervised Learning (ReSSL) . . . 23
3.2. Proposed Methods . . . 25
3.2.1. DenseCL + Consistency . . . 25
3.3. DenseCL + PixContrast . . . 26
3.4. Dense Relational Self-supervised Learning . . . 27

4. EXPERIMENTS AND RESULTS 29
4.1. Datasets 29
4.1.1. MS COCO 29
4.1.2. Pascal VOC 29
4.2. Evaluation Metrics 30
4.3. Pre-training Setup 31
4.4. Evaluation Setup 32
4.5. Quantitative Results 32
4.5.1. Main Results 32
4.5.2. Ablation Study 34
4.5.3. Qualitative Results 36
5. CONCLUSION 42
REFERENCES 44

LIST OF FIGURES

Figure 2.1. Overall scheme of (a) early pretext-task based SSL and (b) Pretext-Invariant Learning (PIRL). 7

Figure 3.1. Illustration of DenseCL's parallel global and dense level representation learning branches seperately. 20

Figure 3.2. Overview of ReSSL's relational framework. 24

Figure 4.1. Visualization of the 5 most similar images according to DenseCL + Consistency. The images in the first column are the images whose 5 most similars are searched. 37

Figure 4.2. Visualization of the 5 most similar images according to DenseCL + PixContrast. The images in the first column are the images whose 5 most similars are searched. 38

Figure 4.3. Visualization of the 5 most similar images according to Pixel-Level ReSSL. The images in the first column are the ones for which we are seeking the 5 most similar images. The images to their right are those 5 most similar ones. 39

Figure 4.4. Visualization of Attention Maps. 41

LIST OF TABLES

Table 3.1. Weak and strong augmentation sets. 24

Table 4.1. Object detection results on Pascal VOC. 'CC' and 'IN' indicate the pre-training models trained on COCO and ImageNet, respectively. '*' stands for re-implementation based on the official code or the paper if no official code is available. 33

Table 4.2. Object detection and instance segmentation results on COCO. 'CC' and 'IN' indicate the pre-training models trained on COCO and ImageNet, respectively. '*' stands for re-implementation based on the official code or the paper if no official code is available. 34

Table 4.3. Ablation study on the impact of α and τ parameters on the performance of DenseCL + Consistency. 35

Table 4.4. Ablation study on the effect of single or multiple positive matching by distance on DenseCL + PixContrast. 35

Table 4.5. Ablation study of the impact of designing ReSSL at pixel-level only or at both pixel-level and image-level. 'IL' and 'PL' indicate image-level and pixel-level, respectively. 36

LIST OF SYMBOLS

$\mathbf{D}_{KL}$	KL-divergence
f	Feature Map
g	Embedding
$\mathbf{H}$	Entropy
k	Key
$\mathbf{L}_{\text{con}}$	Consistency Loss
$\mathbf{L}_{\text{InfoNCE}}$	Contrastive Loss
q	Query
α	Balance coefficient between contrastive and consistency losses
λ	Balance coefficient between image-level and pixel-level losses
τ	Temperature

LIST OF ACRONYMS/ABBREVIATIONS

AP	Average Precision
AUC	Area Under Curve
BERT	Bidirectional Encoder Representations from Transformers
CLS	Classification Token
CNN	Convolutional Neural Networks
CO2	Consistent Contrast
CV	Computer Vision
DenseCL	Dense Contrastive Learning
DINO	Distillation with No Objects
EM	Expectation-Maximization
FIFO	First-In-First-Out
FP	False Positive
GPT	Generative Pre-Training
GPU	Graphics Processing Unit
IoU	Intersection over Union
KL-Divergence	Kullback-Liebler Divergence
MIM	Masked Image Modelling
MLM	Masked Language Modelling
MLP	Multi-Layer Perceptron
MoCo	Momentum Contrast
MS COCO	Microsoft Common Objects in Context
NCE	Noise Contrastive Estimation
NLP	Natural Language Processing
NTXent	Normalized Temperature Scaled Cross Entropy Loss
P	Precision
Pascal VOC	PASCAL Visual Object Classes
PIRL	Pretext-Invariant Representation Learning
PixContrast	Pixel-Level Contrastive Learning

PixPro	Pixel-Level Distillation
ProtoNCE	Prototypical Noise Contrastive Estimation
R	Recall
ReLU	Rectified Linear Unit
ResNet	Residual Network
ReSSL	Relational Self-supervised Learning Framework
SGD	Stochastic Gradient Descent
SimCLR	Simple Contrastive Learning
SSL	Self-supervised Learning
TP	True Positive
ViT	Vision Transformers

1. INTRODUCTION

Visual representation learning is a fundamental area of research in computer vision (CV) that uses the same backbone models as transfer learning to acquire general-purpose representations of visual data for addressing various problems such as classification, detection, and segmentation. In the past, models aiming for this objective were trained in a supervised manner, such as ResNets [1] trained on the ImageNet [2] dataset. The success of this approach relied on the availability of annotated datasets created by human experts. However, it appears that the limits of this learning method have been reached, as more extensive datasets are required to improve the quality of learned models. This is where unsupervised or self-supervised learning comes in, with the aim of surpassing the current performance levels for two main reasons. Firstly, it is challenging to prepare larger datasets suitable for supervised learning due to the significant manpower required. In other words, these datasets need to be carefully curated, which is a labor-intensive process. On the other hand, unsupervised or self-supervised learning methods can operate on uncurated datasets. Secondly, the prepared datasets and the supervised learning approach may not be sufficient for solving real-life problems. Indeed, supervised learning is constrained by the level of supervision available.

Self-supervised learning (SSL), like unsupervised learning, learns from the data itself without any external supervision. However, unlike unsupervised learning, self-supervised learning extracts a supervisory signal from the data itself while learning. Recently, SSL methods have gained a lot of popularity. In particular, the success of SSL models in natural language processing (NLP) (e.g., GPT [3], BERT [4]) has led to the widespread use of this approach in vision. As a matter of fact, the supervisory signal that these methods can generate for learning is more informative than that of supervised approaches.

1.1. Contributions

In this thesis, we perform self-supervised visual representation learning at the dense level, i.e. at the pixel level, on the COCO dataset collected for evaluating models in dense prediction tasks such as object detection, where objects are classified as well as their positions are predicted by bounding boxes, semantic segmentation, where each pixel is classified as belonging to a specific class, instance segmentation, where each pixel which belongs to an object is assigned to the corresponding object and classified as belonging to a specific class or keypoint detection, where people are detected and their keypoints are localized.

Earlier works in self-supervised visual representation learning methods work at the image-level and are trained on ImageNet [2] which is an object centric, well-curated dataset. Since these techniques work at the image level, the features they learn are basically at the image level. Therefore, they have limited success in dense prediction tasks. Besides, ImageNet is limited in generating supervised signals and evaluating models compared to scene-centric datasets such as COCO. Since ImageNet is an object-centric dataset, it can be assumed that there is only one object in the image, which is not suitable for dense tasks as much as COCO. Therefore, it is more challenging and realistic to train and evaluate at the dense level, or in other words, at the pixel level, on datasets like COCO that contain on average more than one object in the images. The number of studies in the literature that work at dense-level and pre-train and evaluate on COCO are limited.

We follow contrastive learning [5–9], which is the dominant approach in self-supervised visual representation learning. In this approach, the images are augmented twice and two views, query and key, are obtained from each image. These two views from the same image are also called positive pairs. On the other hand, views from different images become negative pairs for each other. In other words, the views from each separate instance form a separate class. The goal is that the feature embeddings of positive pairs are close to each other and negative pairs are far apart.

This matching applies at the image level. At the pixel level, views will be represented by multiple patches, so the question is how to match them. Some of the proposed approaches match these patches based on their feature similarity [5] or distance from each other [10]. We follow either matching by similarity or a combination of both.

However, there is a fundamental error in the contrastive learning approach called false-negative pairing. The problem is that views coming from different images are designated as negative pairs for each other regardless of any similarity. Negative-paired views may come from images with the same semantics, such as two different cat images. In response, approaches such as k-nearest neighbor detection to increase the set of positive pairs [11], hard negative-mining [12] or false-negative cancellation [13] to eliminate false negatives in the negative pair set , or consistency regularization [14] that considers the relationship between negative pairs have been proposed in the literature. We use consistency regularization to tackle this problem.

Furthermore, we follow the relational self-supervised framework (ReSSL) [15] as a separate model and extend it to the pixel-level. This approach does not deal with contrastive learning due to the need for negative-pairing and hence the problem of false-negative pairing. However, it aims to make positive pairs similar to each other only through similarity relations with other views.

The contributions of this thesis can be listed as follows:

(i) Pixel-level contrastive learning with consistency regularization: We regularize contrastive learning with consistency loss at the pixel-level to alleviate false-negative pairing problem.
(ii) Combination of different pairing strategies at the pixel-level: We combine two matching, similarity and distance-based, approaches at the pixel-level to overcome their weaknesses.
(iii) Extension of relational self-supervised framework to the pixel-level: We extend the relational self-supervised framework at the image-level to the pixel-level.

1.2. Organization of the Thesis

The rest of the thesis is organized as follows: In Chapter 2, we review the literature, present the existing approaches. In Chapter 3, we describe the baseline models and explain our proposed models in detail. In Chapter 4, we present the experiments that we have performed to learn visual representations, and discuss their results. Lastly, we share our final thoughts about our work in Chapter 5.

2. RELATED WORK

Early research in self-supervised visual representation learning focused on the concept of pretext tasks, where proxy hand-crafted predictive tasks are used to learn image representations. In recent years, there has been a major breakthrough and different approaches have been proposed. One way to classify recent work is to group it into six categories: instance discrimination, contrastive learning, distillation, clustering, information theory, graph-partitioning, and masked image modeling (MIM) based methods.

2.1. Hand-crafted Predictive Pretext Tasks

Some of the proposed pretext tasks include colorization from grayscale images [16, 17], predicting rotation angle [18], predicting the relative position of two patches (regions) sampled from an image [19], and solving jigsaw puzzles [20].

2.2. Instance Discrimination-based Methods

In an early study, Dosovitskiy *et al.* [21] converted the unsupervised representation learning process into a supervised classification problem by using an instance discrimination pretext task. Each image in the dataset, along with its augmented views by image transformations (e.g. resizing, cropping, etc.), was treated as a separate surrogate class and this allowed the problem to be solved with supervised classification techniques. To form surrogate classes, they first detected and cropped areas where the object(s) might be found in each image based on gradient values in the regions, and then resized those areas to a fixed size. Subsequently, multiple data augmentations were applied to the obtained images. Thus, each image and its augmented views formed surrogate classes. However, this approach led to a significant increase in complexity due to the softmax operation in the classification layer as the number of classes increases in parallel with the growth of data.

Wu *et al.* [22] observed that supervised models can automatically capture shared semantic similarities across classes. Based on this observation, they explored how this might translate beyond classes to the instance level and addressed the complexity issue in [21] by introducing non-parametric softmax with Noise Contrastive Estimation (NCE) [23]. The class prototypes, in this case, the instance prototypes were stored in the memory bank for each instance and updated in each iteration. To learn a visual representation, the instance encoding and its corresponding feature vector in memory were drawn close while pushing away from the representations of other instances by minimizing the negative log-likelihood of the non-parametric softmax with temperature.

Misra and Maaten [6] noted that predictive pretext tasks such as jigsaw puzzles lead to representations meaningful for these pretext tasks since these tasks focus on capturing discriminative features relevant to only these pretext tasks. This hinders learning general-purpose representations for classification, segmentation and detection, which is the purpose of self-supervised visual representation learning.

A paradigm shift has therefore occurred with the use of Siamese [24] type two-branch networks, where images are augmented (distorted) twice with image transformations, and augmented views are fed to two separate branches of Siamese network. In this scheme of learning, the goal is to pull representations produced by the views of the same image closer (i.e. maximize the agreement between the representations of the views) in the embedding space (see Figure 2.1).

However, it is important to note that merely pulling representations closer together can lead to what is known as collapsed or trivial solutions. In these situations, the representations become constant and lack variation. This uniformity in representations negates the purpose of the learning process as it does not allow for the differentiation of data points based on their unique characteristics. Consequently, there will be no meaningful learning in the end, rendering the whole process ineffective. Several approaches have been proposed in the literature to address this issue, including contrastive learning, distillation, clustering, and information theory-based methods.

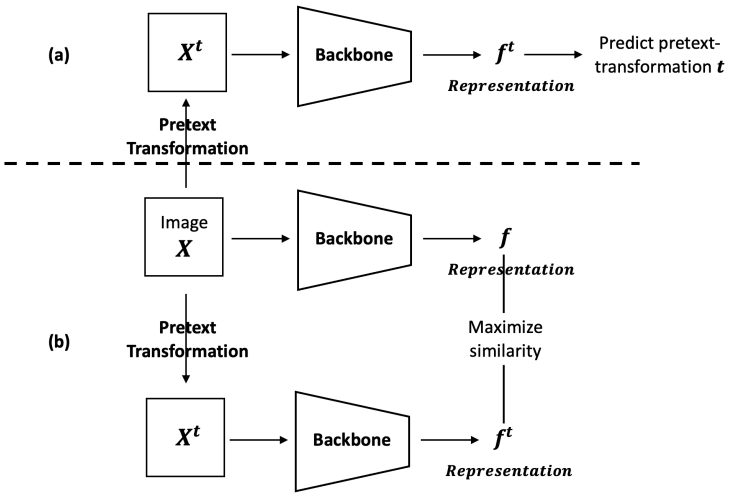


Figure 2.1. Overall scheme of (a) early pretext-task based SSL and (b) Pretext-Invariant Learning (PIRL).

2.3. Contrastive Learning-based Methods

Contrastive learning, following instance discrimination, is built upon the idea that the representations of the query and the key, which are two augmented views of the same image that are also called positive pairs need to be pulled closer while the negative pairs which are representations of the different images' augmented views need to pushed away from each other in the latent space with contrastive loss. That is, it uses negative sampling to avoid collapsed solutions. Thus, the collapsed solution problem is avoided, since the representations different images are kept far apart.

Many researchers followed this approach at the image-level. Misra and Maaten proposed Pretext invariant representations (PIRL) [6]. Chen *et al.* used contrastive learning (SimCLR) [7]. He *et al.* [8], and Chen *et al.* (MoCov2) [9] used momentum contrast (MoCo, MoCov2). While PIRL distorted images with predefined pretext tasks (e.g. permuting image patches as in jigsaw puzzle), SimCLR, MoCo, and MoCov2 did so with image augmentations (e.g. resizing, cropping, color jittering etc.). The impact of augmentations on the quality of representations was examined in Chen *et al.* [7]. The augmentation approaches proposed in this study, besides the ones in the work of He *et al.* [8] were utilized in the work of Chen *et al.* [9].

In the overall architecture, PIRL and MoCo used only the encoder and calculated the loss based on the representations produced by encoder(s), while SimCLR and MoCov2 added a non-linear projection layer (an MLP head) on top of the encoder and calculated the loss based on the projections (embeddings). In addition, PIRL and SimCLR used a single encoder and passed the views through the same single encoder, while MoCo and MoCov2 used a teacher-student framework, passing the query through the encoder and the key through the momentum encoder, which shares the same architecture as the encoder but updates as the exponential average of the encoder instead of being updated by back-propagation like the encoder. Thanks to this momentum encoder, MoCo and MoCov2 did not need to use memory banks or large batches like PIRL and SimCLR. Instead, they used a First-In-First-Out (FIFO) style dynamic queue that is updated with the outputs of the momentum encoder, and this is where the negative pairs come from.

The contrastive loss used in these methods is termed as *InfoNCE* in [25] or *NT-Xent* (the normalized temperature-scaled cross entropy loss) in [7] and defined as

$$L = \sum_{i=1}^{B} -\log\left(\frac{\exp(q.k_+/\tau)}{\exp(q.k_+/\tau) + \sum_{j=1}^{K} \exp(q.k_j/\tau)}\right), \tag{2.1}$$

where q is the embedding of query, k_+ is the embedding of key, which is the positive pair of q, k_j is the jth negative sample, K is the number of negative samples, and B is the batch size.

Although these methods provide representations invariant to transformations (pretext or image augmentation), they are based on two naïve assumptions: views from different images, regardless of their content, form negative pairs and are considered equally negative. These are naive assumptions because the fact that images come from different images (instances) does not guarantee that they are negative pairs. Consider two separate cat images from the ImageNet [2], a dataset commonly used in SSL. In basic contrastive learning, the views from these images are considered to be negative pairs with each other, just as an image of a car is a negative pair with an image of a cat. Also, in this approach there is a binary understanding: two views are either positive

or negative pairs. However, in reality there is no such black-and-white (either positive or negative) relationship between images. Some images are more similar to each other than others, in other words, some images are more positive (or negative) pairs than others.

To alleviate this problem, Wei *et al.* [14] introduced consistency loss, which aims to keep the probability distributions of the similarity of positive pairs to the negative views consistent among themselves, while Dwibedi *et al.* [11] and Huynh *et al.* [13] manipulated the positive-pair and negative-pair sets with false negative identification and computed the contrastive loss function over the newly formed matchings. Dwibedi *et al.* [11] did so by creating support sets containing the k-nearest neighbors of the positive pair of each view in the embedding space and matching the nearest-neighbor of the query with the key, whereas Huynh *et al.* applied more augmentation to the images, obtaining more than two views per image, called support views, and used these views and their k most similar views.

To leverage representations meaningful for dense tasks such as detection and segmentation beyond the image-level tasks, Wang *et al.* [5], Pinheiro *et al.* [26], and Wang *et al.* [27] employed contrastive learning at the pixel-level along with the image-level contrastive learning. To this end, pixels in the query view must be paired with the pixels in the key view of the same image. Wang *et al.* [5] matched pixels as positive pairs based on their embedding similarities and performed dense contrastive learning (DenseCL). Pinheiro *et al.* [26] matched pixels of query and key views as positive pairs based on their distance on the initial image. However, Wang *et al.* [27] claimed that matching based on feature similarity would be noisy due to background pixels, while matching based on distance would lead to semantic inconsistency. Therefore, [27] formed instead pixel-set including pixels whose attention score are above a certain threshold and their nearest-neighbors and computed contrastive loss considering each pixel in these sets as positive-pairs between views.

2.4. Distillation-based Methods

Another way to prevent collapsed solutions is through distillation. In this approach, views are predicted from each other rather than being contrasted. Unlike contrastive learning, distillation does not require negative pairing, as the objective is solely centered on prediction. Although distillation does not offer a theoretical guarantee for avoiding collapsed solutions, it has been experimentally shown to do so through architectural and algorithmic methods such as using asymmetric architecture, stop-gradient operation, centering (mean normalization), and sharpening.

Grill *et al.* [28], Chen and He [29], Koohpayegani *et al.* [30], and Tejankar *et al.* [31] used two asymmetric networks, the online network (student) and the target network (teacher), to transform the nature of the learning objective into a prediction task where the online network has an extra head to predict the output of the target network.

Grill *et al.* [28] experimentally demonstrated that applying stop-gradient operation on the target network in the forward propagation and updating its parameters as an exponential moving average of the online network rather than these networks sharing weights, together with the asymmetric architecture, hinders collapsed solutions. Chen and He [29] argued whether or not sharing weights prevents collapsed solutions and concluded that the stop-gradient operation, along with the asymmetric architecture is adequate to avoid collapsed solutions even if the online and the target networks share weights.

The study of properties acquired by Vision Transfomers (ViT) [32], trained in a supervised manner, has motivated the use of Transformers in SSL. Caron *et al.* [33] trained a ViT with a distillation objective (DINO). Although DINO did not use symmetric networks (i.e. there is no prediction head in the online network), it still did not collapse into constant solutions. DINO empirically showed that by centering and sharpening the target network's outputs with mean normalization and temperature scaling in softmax operations before computing the negative cross-entropy loss lets the

network avoid collapsed solutions. In addition, it was observed that DINO's attention maps, associated with the class token ([CLS]) in the last layer, give hints on the salient foreground regions in an image.

Although distillation does not face the false negative-pairing and does not require any negative pairing at all, the fact that each view has only one positive pair is still problematic as mentioned before for the contrastive learning. Therefore, [30, 31] adopted the memory bank [8] and referring to the mean shift clustering, Koohpayegani *et al.* [30] shifted the embeddings of the online network towards the mean of k-nearest neighbors' of its positive pair (target embedding) in the memory bank. On the other hand, Tejankar *et al.* [31] obtained two probability distributions indicating the similarity between the embeddings of the online and the target networks and the embeddings in the memory bank and computed symmetric Kullback-Leibler (KL) divergence loss between these probability distributions.

Reminding the false negative pairing problem in contrastive learning and restrictiveness of the equipartition constraint in the recent clustering approaches, Zheng *et al.* [15] proposed relational self-supervised learning framework (ReSSL) that ensures consistency between positive pairs over their relationships with the memory bank. This relational consistency is established using entropy loss between similarity probability distributions computed between the positive pairs and the memory bank. Besides, It showed that the use of weak (less) augmentation on the target side improved its performance.

In their detailed study, Xie *et al.* looked closely at pixel-level contrastive learning and distillation. They developed a method of matching pixels as positive pairs based on their geometric distances. This innovative approach allowed them to examine images on a more refined level. They applied both contrastive learning, or PixContrast, and distillation, or PixPro, at the pixel level, aiming to better understand image characteristics. In addition to pixel-level analysis, they also examined images as a whole, offering a broad view of contrastive learning and distillation techniques.

2.5. Clustering-based Methods

Earlier clustering approaches [34–36] assigned pseudo-labels to each image by iteratively clustering their embeddings and then classified them by considering these pseudo-labels as their labels with standard supervised classifiers.

Caron *et al.* [34] and Asano *et al.* [35] followed this approach in an off-line fashion, where clustering is performed over the whole dataset and then classification is performed, while Zhan *et al.* [36] followed the same approach in an online manner, where joint clustering and feature learning occur simultaneously on each batch.

Caron *et al.* [34] used k-means for clustering. However, k-means has no constraint on the size of clusters, which will eventually lead to a collapsed solution where all the representations are assigned to one cluster and the rest are empty clusters, making it impossible to minimize loss in the classification step. To avoid this problem, when a cluster becomes empty, they randomly select a non-empty cluster and add a small perturbation to its center to generate a new cluster and apportion the selected cluster's data points to the new cluster to remove the empty cluster. Moreover, in the classification phase, they sample instances uniformly from each classes if there is imbalance among them.

In other respects, Asano *et al.* [35] used Sinkhorn-Knopp to cluster data points, which is an optimal transport problem technique that ensures equipartitioning among clusters. This equipartition constraint helps to prevent the collapsed solution problem that arises in [34] with k-means since the possibility of all instances being assigned to a single cluster is eliminated, but introduces a bias on the cluster sizes, as each cluster must have an equal number of data points.

To employ online clustering, Zhan *et al.* [36] used two dynamic memories: sample memory and centroid memory, to store representations of the entire dataset with their pseudo-labels and cluster centroids, respectively. These memories were initialized via global clustering with k-means. In each iteration, respectively, they performed forward

propagation to obtain image representations, then applied back-propagation based on pseudo-labels in the sample memory, updated the sample memory as moving-average, assigned new pseudo-labels with k-means clustering, and finally updated the centroids as the average of the data points in the clusters. To avoid trivial solutions problem, the classification loss was re-weighted according to the number of samples in each cluster, the clusters whose sizes are lower than a threshold were eliminated, these clusters' samples were assigned into new clusters, and the largest cluster was splited into two clusters with k-means.

Li *et al.* [37] combined contrastive learning with clustering to overcome the false negative-pairing problem in contrastive learning. For representation learning, they introduced an Expectation-Maximization (EM) framework in which k-means clustering is performed to find prototypes that are the centers of clusters in step E and optimization of network parameters with the proposed contrastive loss ProtoNCE in step M.

In another work, Caron *et al.* [38] combined distillation with clustering. They adapted Sinkhorn-Knopp for soft clustering by adjusting the constraints in the algorithm to be able to bring learning online while maintaining the equipartition constraint to avoid collapsed solutions. In addition, they introduced code vectors as products of feature (representation) vectors and prototype vectors and a swapping prediction objective where code of the key view is predicted from the representation of the query view.

Some recent studies [39, 40] segmented images through clustering and obtained object-level segmentation masks rejecting the naive assumption of image-level methods that images are monolithic instances (i.e. there is only one object in images).

The work of Henaff *et al.* [39] presented an approach consisting of two kinds of networks was proposed: object discovery network and object representation networks for discovering object(s) and learning representations, respectively. Object discovery network takes the spanning view, which is the smallest crop of the image that spans

the augmented views taken by the object representation networks and produces feature maps and spatially clusters them with k-means to segment image into k non-overlapping binary masks. Since the spanning view spans the other two views, a mapping from the object discovery network to the object representation networks is performed on the segmentation masks, and mask-pooled representations of these two views are computed through this mapping. Object representation networks consist of an online network and a target network. A contrastive loss between the outcome of these two networks was calculated and the online networks' parameters were updated using gradient over the loss and the target network's and the object discovery network's parameters as the exponential moving average of the online network with different learning rates.

Wen *et al.* [40] employed prototypical clustering similar to [35] and [38], but operated at the pixel-level to learn object-level representations using learnable prototypes, or "slots". They followed a two-step procedure: semantic grouping and representation learning. Semantic grouping was achieved by assigning each pixel to a cluster with respect to the slots and representation learning was accomplished by enforcing consistency at both object- and image-level. Object-level consistency was realized by contrasting slots and image-level consistency by contrasting groups shared by two views rather than views directly. These groups' representations were obtained through an inverse augmentation process. Meanwhile, pre-training was realized on scene-centric datasets COCO and COCO+ [41] besides ImageNet [2] which is an object-centric dataset to explore the full-potential of self-supervised learning.

2.6. Information Theory-based Methods

Information theory-based methods exploit informative statistics such as entropy, cross-entropy, cross-correlation (cross-covariance), invariance, variance, and covariance and introduce new loss functions to learn representations without requiring specific designs such as asymmetric network, stop-gradient operation, or momentum encoder to avoid the collapsed solution.

Zbontar *et al.* [42] proposed a loss function that aims to have a cross-correlation matrix of the positive pairs embeddings close to the identity matrix. This prevents redundancy between features in order to avoid collapsed solutions, while ensuring that positive pairs are similar to each other.

Wang *et al.* [43] introduced TWIST loss function that ensures consistent representations between positive-pairs, and sharp and diverse representations over feature dimensions, which addresses each of these properties in itself. To satisfy these properties, TWIST loss minimizes the KL-divergence between positive pairs for consistency, and the sum of the entropy of each embedding along the batch dimension for sharpness and maximize the entropy of the sum of embeddings.

Bardes *et al.* [44] came up with another loss function that comprises three terms: invariance, variance, and covariance. Invariance is calculated as mean-squared error between embeddings and thus procures positive pairs to be invariant. Variance is calculated as a hinge loss of standard deviation of embeddings along the batch dimension and hence forces each embedding to be variant. Covariance is calculated as the sum of the squared off-diagonal coefficients of the covariance matrix and so encourages each feature dimension to be uncorrelated to each other. In the subsequent study, Bardes *et al.* [45] adapted this loss function to the pixel-level by calculating each loss at the pixel-level along with the image-level.

2.7. Graph-partitioning-based Methods

Assuming the existence of at least one object in an image, graph-partitioning methods aim to localize the seeds where the salient (foreground) object is located. To that end, each method draws on the attention maps of DINO in the last layer based on the observation that DINO's attention maps, associated with the class token ([CLS]), in the last layer, give hints on the salient foreground regions in an image and provides a suitable environment for unsupervised object discovery (object localization) and subsequent detection and segmentation by displaying images as graphs where nodes are patches of images and edges are connections (similarities) between patches.

In this respect, Simeoni *et al.* [46] primarily constructed a binary similarity graph between patches, where an edge exists if the dot product similarity of two patches' feature maps (key tokens of the patches in the last layer) is positive. To localize the foreground object, they selected the initial seed - the patch with the lowest degree in the graph - based on the assumption that the foreground object spans a smaller area than the background, then expanded the seed selection by selecting another k patches with the lowest degree and keeping the ones having positive similarity with the initial seed, based on the assumption of high intra-similarity (between foreground-foreground and background-background) and low inter-similarity (between foreground-background). Finally, The localized foreground objects were used for class-agnostic and class-aware detection.

Wang *et al.* [47] constructed a binary (epsilon or one) similarity graph, or feature affinity matrix, based on the cosine similarity of the feature maps of the patches (key tokens) in the last layer. Spectral decomposition with standard normalized cut was then applied to obtain the eigenvectors of the feature affinity matrix. To detect the salient (foreground) object, which is assumed to be less connected to the entire graph than the background, the Fiedler eigenvector was selected and a threshold at the mean was applied to bisect the graph. The largest connected component covering the patch corresponding to the maximum absolute value of the Fiedler eigenvector was identified as the foreground (salient) object, since the absolute values indicate the likelihood of there being an object in the patches.

Melas-Kyriazi *et al.* [48] constructed a similarity graph (feature affinity matrix) by combining color and semantic consistency between patches. The former was obtained from the image itself and the latter from the last attention layer of DINO. Then, spectral decomposition with the standard normalized cut over the graph was employed to obtain eigenvectors of the feature affinity matrix. Thus, to localize the foreground object, the Fiedler eigenvector was selected similar to [47], and threshold at zero was performed on this eigenvector to bisect the graph. In this bisection, after keeping the largest connected components in both sections, the smaller section among the two was assumed to be the foreground object. Besides, to segment the image, first eigenvectors

were clustered, then the pixels of the image were updated according to the clustering assignments in the previous step, and finally, the entire dataset was fed to the DINO model and the output feature maps were clustered across the dataset.

In contrast to their peers, Van *et al.* [49] used two different affinity graphs: an objectness graph and a similarity adjacency graph. The objectness graph's edges were constructed according to the dot product similarity of the CLS token and keys of each patch. Apart from these objectness scores, the patch (node) with the highest score was selected as the initial seed, in other words source path, and the seed was expanded with another k-patch with the highest score. Then, class-agnostic masks were generated such that seeds and patches with positive dot-product similarity with seeds are 1 (foreground) and the rest are 0 (background), and a binary segmentation model was trained on these masks. For class-aware mask generation, k-means clustering was performed on the CLS tokens of the patches that are foreground in the images, so that k-class masks were obtained and a Mask R-CNN model was trained on these masks.

Wang *et al.* [50] expanded upon prior research [47] to advance graph-partitioning techniques. They constructed a similarity graph and dissected it using the normalized cut method, bisection was based on the Fiedler eigenvector's mean value. To identify the foreground object in the bisected graph, two criteria were used: the foreground must contain the patch with the maximum absolute Fiedler eigenvector value and should have less than two corners. If these conditions aren't met, the bisection is reversed. They also introduced an iterative process to find multiple objects, where patches with detected objects are disconnected from the graph in each iteration. This approach enhances the technique's ability to identify and isolate multiple objects.

2.8. Masked Image Modeling (MIM)-based Methods

MIM-based generative methods consist of two main parts: an encoder and a decoder. The encoder compresses the input and obtains a representation in the latent space, while the decoder aims to reconstruct the original input from the representation.

Early studies harnessed convolutional neural networks (CNN). Pathak *et al.* [51] whitened a region of pixels in images and regenerated it by inpainting based on the context of non-whitened pixels. In another study, Vincent *et al.* [52] added noise to the images and reconstructed the noiseless images.

With the rise of Transformers [53] and Masked Language Modelling (MLM) in NLP (e.g., GPT [3], BERT [4]) recent works drewed upon ViTs [32] for MIM.

Chen *et al.* [54] proposed two ways of designing MIM with Transformer architecture: predicting the next pixel from the previous ones in an auto-regressive fashion as in [3] and the masked pixels from the unmasked ones as in [4]. Zhou *et al.* articulated MIM as distillation problem by predicting the output of the teacher network from the student network which takes masked images as input. Besides unlike [54] they operated at the patch level, thus reduced the emerging computational complexity.

Li *et al.* [55] and He *et al.* [56] disputed different masking strategies. Li *et al.* [55] performed masking depending on the attention of patches and masked images without losing essential patches to be able to reconstruct them. On the other hand, He *et al.* [56] followed a simpler approach and masked patches randomly as [54]. Yet, they hypothesised that words in languages cover highly semantic information by themselves unlike pixels or patches in images which can be inferred from neighbor pixels or patches (i.e., heavy spatial redundancy in images). Consequently, they masked out a high portion of patches (75%).

3. METHODOLOGY

In this chapter, we will first describe our baseline models in detail and examine their components and functionality. We will then also discuss the changes we have made to improve our proposed models, along with the reasons why. By analyzing both the base models and our proposed models, we aim to provide a comprehensive understanding of the changes our models offer.

3.1. Baselines

We have used four methods as baselines: DenseCL, PixContrast, CO2, and ReSSL. Some of these methods have been chosen because they work at the pixel-level and some have only been tested at the image-level. Our main objective was to enhance and refine these methods, taking into account their strengths and weaknesses, in order to achieve even better performance and accuracy in visual representation learning at the pixel-level.

3.1.1. Dense Contrastive Learning (DenseCL)

As mentioned above, DenseCL performs contrastive learning at the pixel-level along with the image-level. At the image-level, DenseCL follows MoCov2 [9] and consists of an backbone and a (global) projection head. To extend this approach to the pixel-level representation learning, a dense projection head is introduced in parallel to the global projection head (see Figure 3.1). The backbone is the model transferred to solve downstream tasks and produces spatial features of size H×W to preserve the pixel-level information. The global projection head provides representation learning at the image level and the dense projection head provides representation learning at the dense-level, i.e. pixel-level. Both heads share a single backbone.

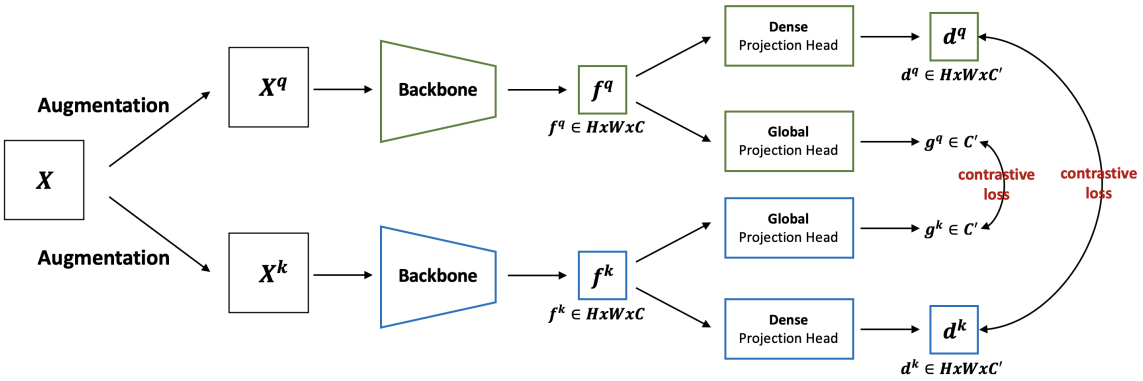


Figure 3.1. Illustration of DenseCL's parallel global and dense level representation learning branches seperately.

The global projection head is identical to the projection head in [9], which is a two-layer MLP with Rectified Linear Unit (ReLU) activation. On the other head, the dense projection head is a two-layer CNN with ReLU activation. The MLP layers in the global projection head are replaced with 1x1 convolutions in the dense projection head.

DenseCL adopts ResNet50 as the backbone. It operates on 224x224 images and produces 7x7 spatial feature maps. Since the global projection head works at the image-level, it primarily employs global pooling on these 7x7 feature maps to have a single, global feature vector for each image. On the other hand, the dense projection head works at the pixel-level. Therefore it does not employ any pooling and operates on the spatial feature maps.

Moreover, DenseCL uses the same data augmentations as in MoCov2. It first applies 224 × 224-pixel random resize cropping, then performs random color jittering, random gray-scale conversion and gaussian blurring and finally employs random horizontal flip.

For the image-level contrastive learning, two augmented views of the same image, query and key, are regarded as positive-pairs. For the pixel-level contrastive learning, it is not as direct as the image-level since there are 49 (7x7) feature vectors generated by

each view. DenseCL matches pixels as positive-pairs according to their cosine similarity. Each pixel in the query is matched with the pixel in the corresponding key that has the highest cosine similarity to it.

Besides the queue in MoCov2, DenseCL stores another First-In-First-Out (FIFO) style dynamic queue. The length of both queue are set to 65536. The first queue is used to keep global features, while the second queue is used to keep dense features. The first queue is updated with the output of the global projection head, while the second queue is updated with the globally pooled features of the dense projection head's output. The feature vectors stored in the first and second queues are used as negative samples for the image-level and the pixel-level contrastive learning, respectively.

DenseCL applies the InfoNCE contrastive loss for both the image and pixel levels to pull positive-pairs closer while pushing negative-pairs away with a temperature value of 0.2. In overall, DenseCL equally weighs these two losses.

What distinguishes DenseCL from other pixel-level contrastive learning methods in general is that it has been pre-trained on both ImageNet and COCO datasets. This may seem like a simple distinction, but it is not. While it is reasonable to pre-train the image-level contrastive learning methods only on ImageNet, which is an object-centric dataset, it is crucial to pre-train and evaluate the pixel-level contrastive learning approaches on scene-centric datasets such as COCO to observe their performance.

3.1.2. Pixel-Level Contrastive Learning (PixContrast)

PixContrast, like DenseCL, carries out contrastive learning not only at the image-level but also at the pixel-level. This approach of learning at two different levels simultaneously enhances the model's capability to understand and interpret the image more accurately. When it comes to the architecture, PixContrast and DenseCL share a striking similarity. Both the models are composed of the same three key components: a backbone (ResNet50), a global projection head (two-layer MLP), and a dense projection head (two-layer CNN).

The only important difference between them is the pixel matching strategy. DenseCL does not take into account the distance of pixels from each other. In other words, the position of the pixels does not matter when matching them. The only criterion considered is the similarity of the pixels to each other.

On the other hand, PixContrast matches pixels as positive pairs based on their distance to each other. For this, the positional information of the augmented views is needed. This is achieved by returning the x, y coordinates of the top left corner of the view and the width and height of the view according to the crop and horizontal flip that occurs during data augmentation. These values are normalized between 0 and 1.

The distances of the points inside the intersection of keys and queues coming from the same instance are found, these distances are normalized by dividing them by the length of the diagonal of a square of the feature vector, and finally the points with distances below a threshold are matched as positive pairs. Based on these matchings, the contrastive loss is minimized, with equal weighting at the image level and pixel level.

3.1.3. Consistent Contrast (CO2)

Aiming to alleviate the false negative pairing problem in contrastive learning, CO2 introduces a consistency loss. Contrastive learning uses one-hot labelling and is not concerned with the similarity of negative pairs to each other. For this reason, CO2 computes two separate probability distributions based on the similarity of the query and the key to negative pairs. The similarity between the query q and the negatives n_i ($i \in \{1, ..., K\}$) is defined as

$$\mathbf{q}_i = \frac{\exp(q.n_i/\tau)}{\sum_{k=1}^{K} \exp(q.n_k/\tau)}, \tag{3.1}$$

and the similarity between the positive pair p of the query q and the negatives n_i is formulated as

$$\mathbf{p}_i = \frac{\exp(p.n_i/\tau)}{\sum_{k=1}^{K} \exp(p.n_k/\tau)}, \tag{3.2}$$

where τ is the temperature to control the sharpness of the distributions. Consistency loss is defined as the symmetric KL-divergence loss of these distributions:

$$L_{con} = \frac{1}{2} D_{KL}(\mathbf{q}||\mathbf{p}) + \frac{1}{2} D_{KL}(\mathbf{p}||\mathbf{q}), \tag{3.3}$$

and the overall loss becomes:

$$L = L_{\text{InfoNCE}} + \alpha L_{\text{con}}, \tag{3.4}$$

where α is the coefficient used to combine the losses.

3.1.4. Relational Self-supervised Learning (ReSSL)

Most SSL methods do not deal with relationships between instances. For example, distillation only considers positive pairs. On the other hand, contrastive learning represents only binary relations: views are either positive pairs or negative pairs. Clustering assigns a single label to each instance. However, these labels do not describe the similarity relationship between instances. This is another manifestation of the false negative pairing problem mentioned earlier.

ReSSL therefore introduces a relational self-supervised framework where no direct comparison, prediction or clustering over embeddings takes place. In this framework, relationships are not depicted with binary labels. Instead, relations are represented in terms of the similarity of embeddings (see Figure 3.2).

Architecturally, ReSSL adopts the teacher-student framework consisting of backbone and projection head networks. These networks share the same architecture, except that the student's parameters are updated by backpropagation, while the teacher's parameters are updated as an exponential moving average of the student's. It also keeps a queue where previous embeddings are stored, similar to MoCov2. However, the embeddings stored in this queue are not seen as negative-pairs. They are used to model the relationships between instances.

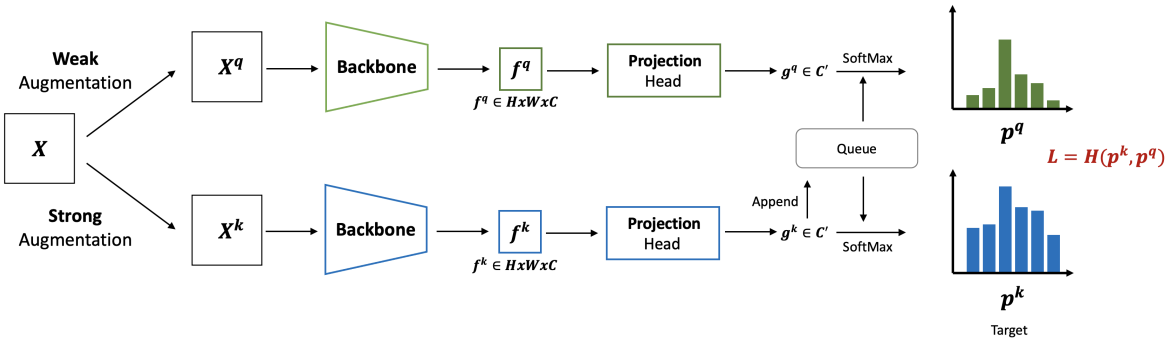


Figure 3.2. Overview of ReSSL's relational framework.

Unlike common practice, ReSSL initially augments the instances twice with two different augmentations, weak and strong (contrastive). The weak augmentation set consists only of random resize cropping and random horizontal flipping. It does not contain any color distortion or blurring. The reason behind this choice is not to move the key view too far away from the original image and thus keep the output of the teacher network more stable. The contrastive augmentation set consists of all the augmentations used in MoCov2. The use of extra augmentations in the student branch is to provide the augmentation-invariant property.

Table 3.1. Weak and strong augmentation sets.

	Weak	**Strong**
Crop	✓	✓
Horizontal Flip	✓	✓
Color jitter		✓
Grayscale		✓
Gaussian Blur		✓

The weakly augmented view (key) is fed to the teacher network, while the strongly augmented view (query) is fed to the student network. Thus, two embeddings are obtained from the same instance. Then, ReSSL calculates relationship distributions of embeddings of the key and the query with the embeddings stored in the queue applying softmax with temperature over their dot-product similarity. The relationship

of the query q and the key p with the ith negative in the queue is formulated in the Equations (3.1) and (3.2), respectively.

In addition to using weaker augmentation, a lower value of temperature was chosen in the calculation of key's relation distribution compared to the query's. Using a lower temperature for the teacher network provides a sharper distribution as a target.

Since the relation distributions between embeddings are used instead of embeddings, the objective becomes maximizing consistency between the relation distribution of positive pairs. For this reason, KL-divergence loss is minimized, which is defined as

$$D_{KL}(\mathbf{p^k}||\mathbf{p^q}) = H(p^k, p^q) - H(p^k), \tag{3.5}$$

where $H()$ is the entropy. Since p^k is the target and the gradients only flow backwards in the student branch, symmetric KL-divergence is not preferred, $H(p^k)$ is not taken into account and $H(p^k, p^q)$ is just minimized.

3.2. Proposed Methods

We propose three models for learning pixel-level visual representation. The first two of these models try to improve performance by applying pixel-level contrastive learning, while the last one adapts the relational SSL framework implemented at the image-level to the pixel-level.

3.2.1. DenseCL + Consistency

As mentioned above, DenseCL has performed contrastive learning at the pixel-level. However, it did not provide a solution to the negative pairing problem either at the image-level or at the pixel-level. Therefore, in our first proposal, we tackle this problem and we regularize the contrastive loss both at the image-level and at the pixel-level with the consistency loss introduced in CO2. CO2 employs consistency regularization at the image-level, while our proposed model DenseCL + Consistency also performs it at the pixel level.

DenseCL + Consistency consists of a backbone, a global projection head, and a dense projection head. It also keeps a queue for negative samples. We augment the images twice to get query and key views. To process them, we follow the teacher-student structure and feed query into the student's backbone model and key into the teacher's backbone model to obtain representations of these views. After that, these representations are passed through two parallel projection heads, the global and dense projection heads, to obtain their global and dense embeddings. We follow DenseCL and use a two-layer MLP as the global projection head and a two-layer CNN for the dense projection head.

These embeddings are brought closer together at both image and pixel level, while they are moved away from the embeddings stored in the queue with the contrastive loss (2.1). We also compute the similarity distributions of these embeddings in relation to the negative samples according to the Equations (3.1) and (3.2), and similarize these distributions with symmetric KL-divergence loss (3.3) at both image and pixel level. The overall loss, therefore, is formulated as

$$L = \lambda(L_{\text{InfoNCE}} + \alpha L_{\text{con}})_{image} + (1-\lambda)(L_{\text{InfoNCE}} + \alpha L_{\text{con}})_{pixel}, \tag{3.6}$$

where λ is the coefficient to balance the image-level and pixel-level losses and α is the coefficient to balance the contrastive and consistency losses. After the loss calculation, the student's parameters are updated by backpropagation, while the teacher's parameters are updated as an exponential moving average of the student's.

3.3. DenseCL + PixContrast

In our second proposal, we aim to observe the performance impact of combining two separate matching strategies at dense-level. The previously mentioned DenseCL and PixContrast approaches follow two separate matching strategies at the pixel level. DenseCL's similarity-based matching strategy is not optimal due to both the weakness of the model's matching in the first epochs and the noise caused by the large number of background pixels. On the other hand, since PixContrast only matches spatially close pixels, it ignores the existence of semantically similar but spatially distant pixels. We

therefore combine these two matching strategies, similarity-based and distance-based matching.

Our model follows the same architecture with DenseCL. The only difference between them is in the matching strategy, which manifests itself in the overall loss function. Taking into account the distance-based matching at the pixel level, we add another contrastive loss term. As a result, the overall loss is defined as

$$L = \lambda_1 L_g + (1 - \lambda_1)(\lambda_2 L_{dense} + (1 - \lambda_2) L_{pix}), \tag{3.7}$$

where L_g is the image-level contrastive loss, L_{dense} and L_{pix} are the pixel-level contrastive loss based on similarity-based and distance-based matching strategies, respectively, λ_1 is the coefficient to proportion the image-level and pixel-level losses, and λ_2 is the coefficient to proportion the contrastive loss based on similarity-based and distance-based matching strategies.

3.4. Dense Relational Self-supervised Learning

As our third proposal, we extend ReSSL to the pixel level. ReSSL only deals with the relationship between views at the image-level. We adapt this idea to the pixel-level by adding the second projection head used in the pixel-level models, the dense projection head. Despite its simplicity, this addition does not introduce much memory or temporal complexity.

As in ReSSL, we follow the teacher-student structure. We apply two separate augmentations to the images, one weak and one strong, and generate the key and query views, respectively. These views first pass through the backbone and then through the global projection head and dense projection head in parallel. Thus, we get two separate feature maps for each view, one at the image-level and one at the pixel-level. At the image-level this corresponds to a single-dimensional feature, while at the pixel-level this corresponds to H*W dimensional spatial features.

Adapting the relational framework to the pixel-level does not only require the addition of a dense projection head. This only allows spatial features to be obtained. However, a matching between these spatial features is necessary. For this we follow DenseCL and perform matching based on similarities. We also keep a second queue at the pixel-level to calculate the relation distributions.

We compute the relation distributions with non-parametric softmax. Then, we calculate the entropy loss between student and teacher branches with the relation distributions from the teacher branch as the target, as in ReSSL. As a result, the total loss is the combination of the image-level and pixel-level entropies.

4. EXPERIMENTS AND RESULTS

In this chapter, we first describe the datasets used during the pre-training and evaluation of the proposed models. Then, we will present the metrics used for performance evaluation and the configurations defined for the pre-training and evaluation phases. Finally, we will discuss the results of the models quantitatively and qualitatively.

4.1. Datasets

Since we aim to learn semantically-rich dense visual representation during the pre-training and evaluation of the models, we use COCO and Pascal VOC datasets prepared for dense prediction tasks.

4.1.1. MS COCO

COCO is a large-scale scene-centric dataset. It is designed to train models for object detection, segmentation, keypoint detection and captioning. Therefore, unlike object-centric datasets, there are often multiple objects in the same image. In COCO, 80 object categories are labeled. We are using the 2017 version of the dataset, first published in 2014. In this version, the training, validation and testing sections contain 118K, 5K and 41K images, respectively. The training-validation set of 123K images contains a total of 896K labeled objects, corresponding to an average of 7.3 objects per image. We use this dataset for both pre-training and evaluation in the object detection and instance segmentation tasks by fine-tuning our pre-trained models.

4.1.2. Pascal VOC

Pascal VOC is another scene-centric dataset. It is designed to train object detection, semantic segmentation, and classification models. Compared to the COCO dataset, it is a smaller dataset both in terms of data size and number of object cat-

egories labeled. In total, 20 object categories are defined. The training set consists of 1,464 images and the validation set consists of 1,449 images. The test set is private. We use this dataset for evaluation of our pre-trained models in the object detection task by fine-tuning.

4.2. Evaluation Metrics

We follow the standard average precision (AP), AP50, and AP75 metrics for evaluating our models in the object detection and instance segmentation tasks. IoU (Intersection over Union), which is the ratio of the intersection of the prediction and the ground-truth to their union is a commonly used measure of image detection and segmentation tasks. If the IoU is over a threshold, the detection is deemed to be successful. Different thresholds may be used for different tasks. AP (or mAP) is the mean of the average precision values at 10 IoU thresholds with increments of 0.05 from 0.50 to 0.95. AP50 and AP75 are the average precision values over all object categories at 0.50 and 0.75 IoU thresholds, respectively.

Precision and recall are two measures used to measure the performance of retrieval tasks. Precision (P) is the ratio of correct predictions to the total number of predictions and the recall (R) is the number of correct predictions divided by the number of ground-truth objects. Defined formally, $P = TP/(TP + FP)$ and $R = TP/(TP + FN)$, where TP is the number of correctly predicted objects, FP is the number of incorrectly predicted objects and FN is the number of unpredicted objects. Since there is a trade-off between precision and recall in retrieval tasks, a precision-recall curve is used for evaluations.

Object-detection and instance segmentation models make their predictions with a confidence score in order to use them in real-world applications. The precision-recall curve is generated based on the IoU threshold and the confidence values of the model's predictions. The AP at the specified IoU threshold is the area under this precision-recall curve (AUC).

In order to draw the precision-recall curve, the first step is to sort the predictions from high to low according to their confidence values. For a prediction to be true, the IoU value must be higher than the threshold. This is necessary but not sufficient by itself. If there are multiple predictions for the same object, only the one with the highest IoU value is considered true. Otherwise, predictions with other conditions are considered false. After this correction, the precision and recall values are updated and the precision-recall curve is drawn according to these values.

4.3. Pre-training Setup

For COCO pre-training including both baseline and our models, we closely followed DenseCL and used the same hyper-parameters. Unless otherwise stated, we used stochastic gradient descent (SGD) as the optimizer with an initial learning rate of 0.03, weight decay of 0.0001 and momentum of 0.9. We run the pre-training on a 4-GPU server for 800 epochs, with 64 samples per GPU, for a total mini-batch size of 256.

Our proposed models consist of a backbone, a global projection head and a dense projection head. Among the pixel-level ReSSL models, the model that solely considers relations at the pixel-level consists of a backbone and a dense projection head. We adopted ResNet-50 as the backbone, two-layer MLP as the global projection head, and two-layer CNN as the dense projection head. We specified 2048 and 128 as the feature dimension of the projection heads. After pre-training, we only use the student branch's backbone for downstream tasks.

In terms of data augmentation, we use several techniques such as random resize cropping, color jittering, grayscaling, gaussian blurring, and horizontal flipping, applicable to all models except pixel-level ReSSL models. For these, we use random resize cropping, horizontal flipping, and normalization, known as 'weak augmentation' in ReSSL. The temperature value in all softmax computations is 0.04, but it's adjusted to 0.1 for ReSSL-based student models.

4.4. Evaluation Setup

For evaluation, we fine-tuned our models on COCO and Pascal VOC and measured their performance on COCO object detection, COCO instance segmentation, and VOC object detection. When evaluating on Pascal VOC object detection, we fine-tuned a Faster R-CNN [57] on the VOC trainval07+12 for 24k iterations with initial learning rate of 0.02 and tested on the VOC test2007 set. We used the output of the fourth layer of the ResNet backbone (C4-backbone). When evaluation on COCO object detection and instance segmentation, we fine-tuned a Mask R-CNN [58] detector on the COCO train2017 split for 90k iterations with initial learning rate of 0.02 and evaluated on the COCO val2017 split. We used the output of the first four layers of the ResNet backbone as Feature Pyramid Network (FPN) [59].

4.5. Quantitative Results

In this section we will discuss the main quantitative results on COCO detection, COCO instance segmentation, and VOC object detection with ablation studies to determine some hyper-parameters.

4.5.1. Main Results

For the task of object detection within Pascal VOC (see Table 4.1), the application of consistency regularization notably enhances the AP50 metric by +0.2. By combining both similarity-based and distance-based matching strategies, our suggested technique exceeds PixContrast's performance, although it slightly trails behind the DenseCL method. The relational framework (ReSSL) doesn't perform as well when pre-trained on the COCO dataset. Yet, extending this framework to not only the image level but also the pixel level, as we suggest, considerably boosts performance (+29.3 AP, +32 AP50, and +27.7 AP75). Furthermore, when comparing the performance levels, it turns out that, contrary to the general trend, ReSSL (CC) outperforms ReSSL (IN) when operating at the pixel level.

Consistency regularization improves performance in COCO's object detection and instance segmentation tasks, as shown in Table 4.2 (+0.3 AP, +0.5 AP50, and +0.1 AP75). The combined approach of DenseCL + PixContrast outperforms both individual methods in these tasks. Adapting the relational framework to pixel-level also improves performance over the image-level approach. Furthermore, pixel-level ReSSL surpasses both DenseCL and PixContrast without requiring negative pairs.

Table 4.1. Object detection results on Pascal VOC. 'CC' and 'IN' indicate the pre-training models trained on COCO and ImageNet, respectively. '*' stands for re-implementation based on the official code or the paper if no official code is available.

Pre-train	**AP**	**AP50**	**AP75**
DenseCL (IN)	58.7	82.8	65.2
PixContrast (IN)	58.1	82.4	64.5
DenseCL (CC)	56.7	81.7	63.0
PixContrast (CC*)	56	81.7	62.5
ReSSL (CC*)	26.7	49.6	24.3
DenseCL + Consistency (CC)	56.3	81.9	62.2
DenseCL + PixContrast (CC)	56.5	81.4	62.7
Pixel-Level ReSSL (CC)	56.0	81.6	62.0

The results can also be evaluated according to the dataset used for pre-training. Pre-training on ImageNet consistently outperforms COCO for contrastive learning. The reasons behind this difference are related to the nature of the datasets and the pre-training configurations:

- ImageNet is larger than COCO, with $\sim$ 118 thousand and $\sim$ 1.28 million images, respectively,
- the pre-training in ImageNet is longer than in COCO, $\sim$ 368 thousand and 1 million iterations respectively, and,

- ImageNet is more suitable for image-level contrastive learning than COCO. Since ImageNet is an object-centric dataset, data augmentations are more likely to result in two views with similar semantics than COCO. Since the images in COCO contain multiple objects from different classes, it is not guaranteed that the views resulting from the data augmentations are semantically positive-pair at the image-level.

Table 4.2. Object detection and instance segmentation results on COCO. 'CC' and 'IN' indicate the pre-training models trained on COCO and ImageNet, respectively. '*' stands for re-implementation based on the official code or the paper if no official code is available.

Pre-train	Object Detection			Instance Segmentation		
	AP	**AP50**	**AP75**	**AP**	**AP50**	**AP75**
DenseCL (IN)	40.3	59.9	44.3	36.4	57.0	39.2
PixContrast (IN)	38.8	-	-	-	-	-
DenseCL (CC)	39.6	59.3	43.3	35.6	56.5	38.4
PixContrast (CC*)	39.2	59.1	42.8	35.3	56.2	37.9
ReSSL (CC*)	29.0	46.3	31.2	26.8	43.5	28.2
DenseCL + Consistency (CC)	39.9	59.8	43.7	36.0	57.0	38.5
DenseCL + PixContrast (CC)	39.9	59.5	43.7	36.0	56.9	38.7
Pixel-Level ReSSL (CC)	39.8	59.5	43.6	35.9	56.5	38.7

4.5.2. Ablation Study

We examine hyperparameters and configurations in our methods. First, we study the impact of balance (α) and temperature (τ) hyperparameters, as proposed in [14], on DenseCL + Consistency performance. Second, we observe DenseCL + PixContrast's performance based on single or multiple matching in the PixContrast part. Third, we explore the pixel-level relational framework design in two ways: considering both pixel and image-level relations, and considering only pixel-level relations.

Table 4.3. Ablation study on the impact of α and τ parameters on the performance of DenseCL + Consistency.

τ	α	VOC Detection			COCO Detection			COCO Segmentation		
		AP	**AP50**	**AP75**	**AP**	**AP50**	**AP75**	**AP**	**AP50**	**AP75**
0.04	0.3	55.8	81.3	62.2	39.7	59.6	43.7	35.9	56.7	38.5
	1	55.1	81.5	61.0	39.7	59.5	43.4	35.9	56.8	38.5
	10	56.3	81.9	62.2	39.7	59.8	43.3	35.9	56.7	38.6
0.05	0.3	56.1	81.5	62.3	39.3	59.0	43.2	35.4	56.1	38.0
	1	55.1	81.5	60.6	39.9	59.8	43.7	36.0	57.0	38.5
	10	55.3	81.7	61.2	27.4	44.2	29.0	25.3	41.5	24.6

In Table 4.3, we observe that the third configuration gives better performance in Pascal VOC object detection, while the first configuration gives better performance in COCO object detection and instance segmentation.

Table 4.4. Ablation study on the effect of single or multiple positive matching by distance on DenseCL + PixContrast.

Matching	VOC Detection			COCO Detection			COCO Segmentation		
	AP	**AP50**	**AP75**	**AP**	**AP50**	**AP75**	**AP**	**AP50**	**AP75**
Single	56.5	81.4	62.7	39.8	59.5	43.6	35.9	56.5	38.5
Multiple	55.9	81.4	62.2	39.9	59.5	43.7	36.0	56.9	38.7

As can be seen in Table 4.4, single matching performs better in Pascal VOC object detection, while multiple matching performs better in COCO object detection and COCO instance segmentation.

Table 4.5 shows that for dense visual representation learning, using the image level in addition to the pixel level significantly improves performance. In other words, for object detection and instance segmentation tasks, understanding the image at the global level positively affects the understanding at the local level.

Table 4.5. Ablation study of the impact of designing ReSSL at pixel-level only or at both pixel-level and image-level. 'IL' and 'PL' indicate image-level and pixel-level, respectively.

IL	PL	VOC Detection			COCO Detection			COCO Segmentation		
		AP	**AP50**	**AP75**	**AP**	**AP50**	**AP75**	**AP**	**AP50**	**AP75**
	✓	43.7	70.2	46.3	33.3	51.3	36.2	30.4	48.7	32.6
✓	✓	56.0	81.6	62.0	39.8	59.5	43.6	35.9	56.5	38.7

4.5.3. Qualitative Results

In our study, we also undertake comprehensive qualitative analyses, the purpose of which is to compare and contrast the performance level of our newly proposed models with those of the existing baseline models. Our initial step in this process involves assessing whether our proposed models are effective in accurately identifying the similarities that exist between different images. To meet our goal, we handpicked ten diverse images from a range of different categories that are included within the COCO validation set. We then identified the top five most similar images by calculating and visualizing similarities in the latent space.

The results of this process are clearly depicted in both Figure 4.1 and Figure 4.2. It is evident from these figures that our proposed methods, which we have named DenseCL + Consistency and DenseCL + PixContrast, and which are both based on the principles of contrastive learning, are successful in identifying similarities in images that belong to the same semantic class.

Furthermore, these methods also exhibit an effective ability to differentiate between various objects (for instance, clocks, people, elephants) and activities (such as baseball, skiing, and eating), to a significant degree. A comparison of the first and the last lines reveals that the DenseCL + Consistency approach has a slight advantage over the DenseCL + PixContrast approach.

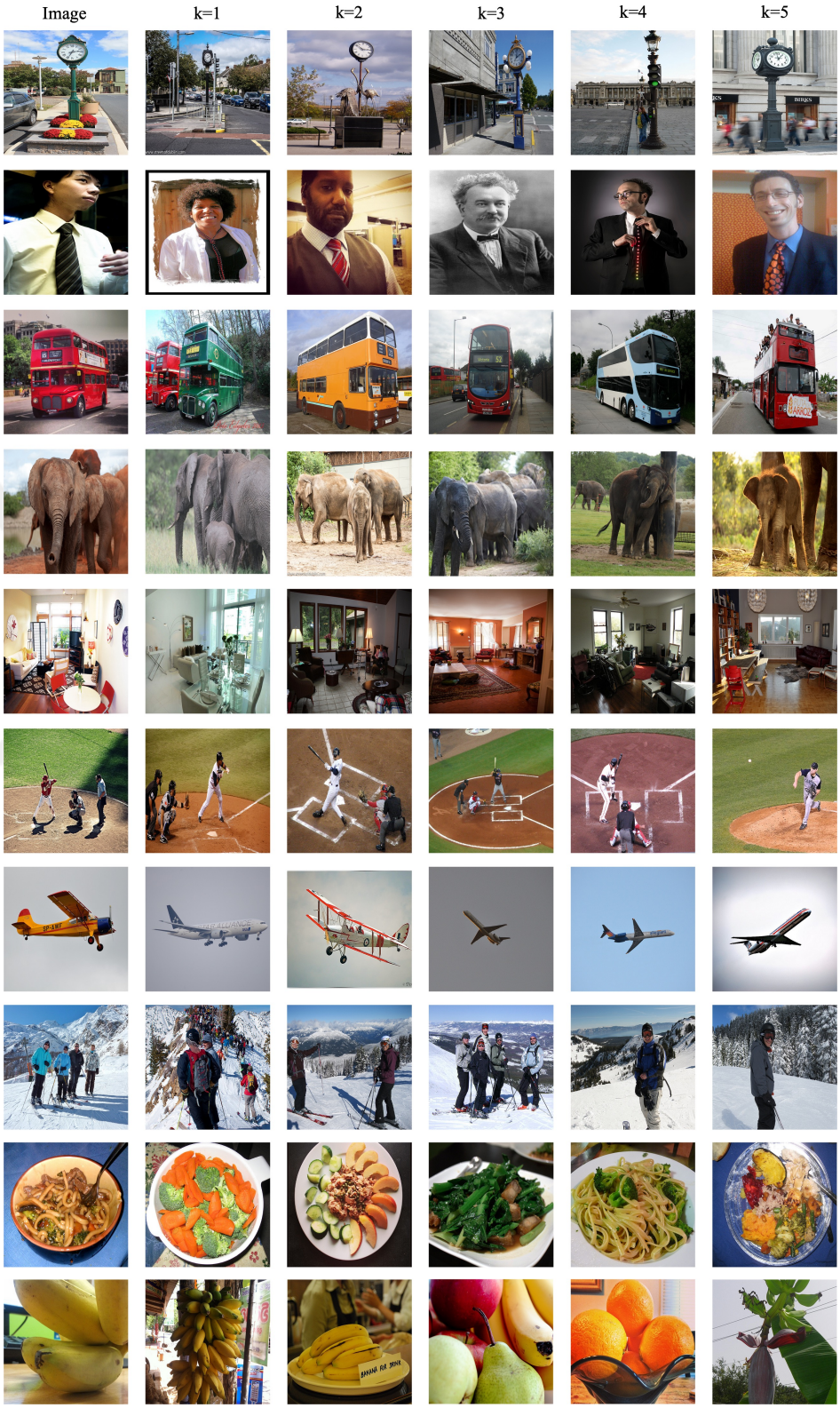


Figure 4.1. Visualization of the 5 most similar images according to DenseCL + Consistency. The images in the first column are the images whose 5 most similars are searched.

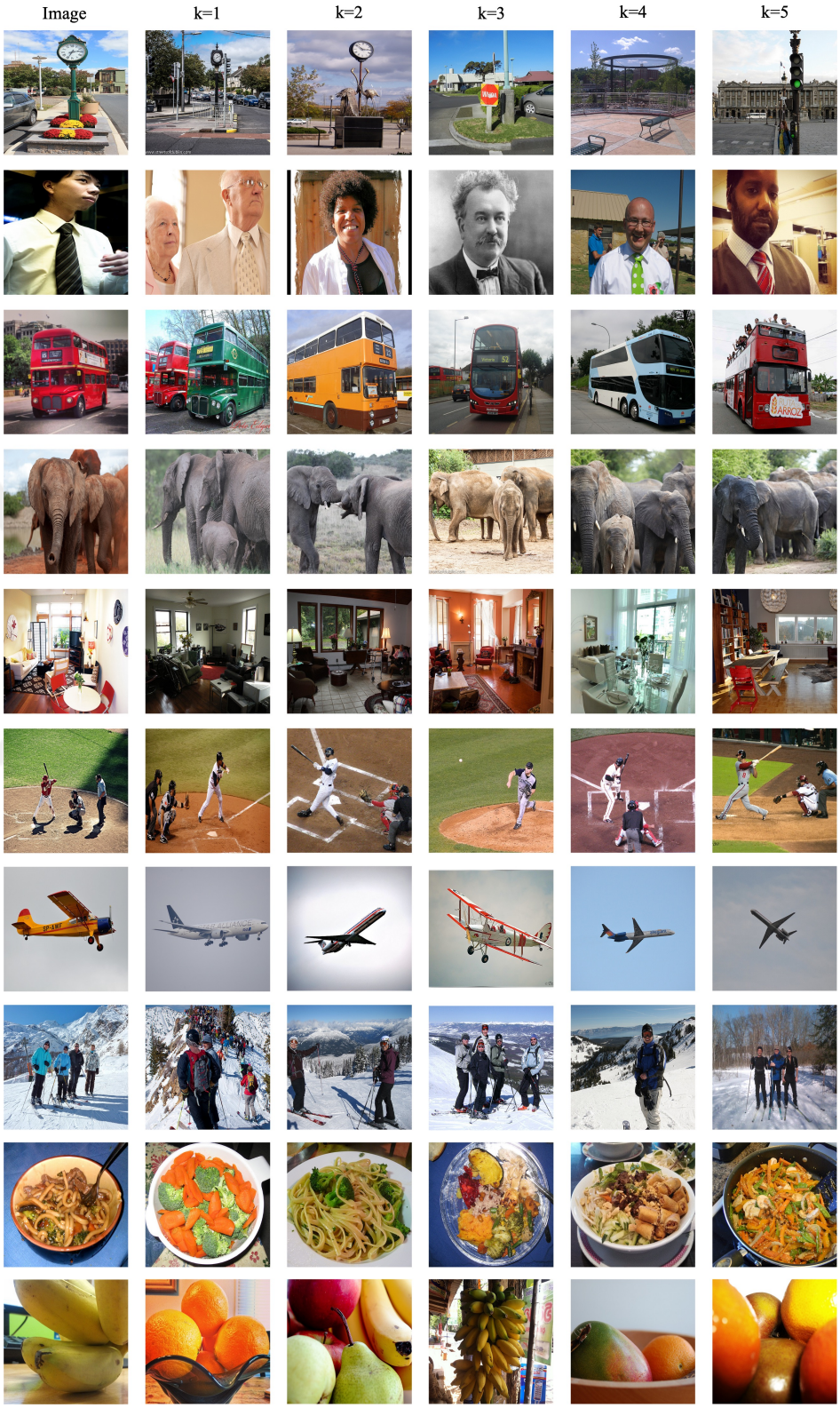


Figure 4.2. Visualization of the 5 most similar images according to DenseCL + PixContrast. The images in the first column are the images whose 5 most similars are searched.

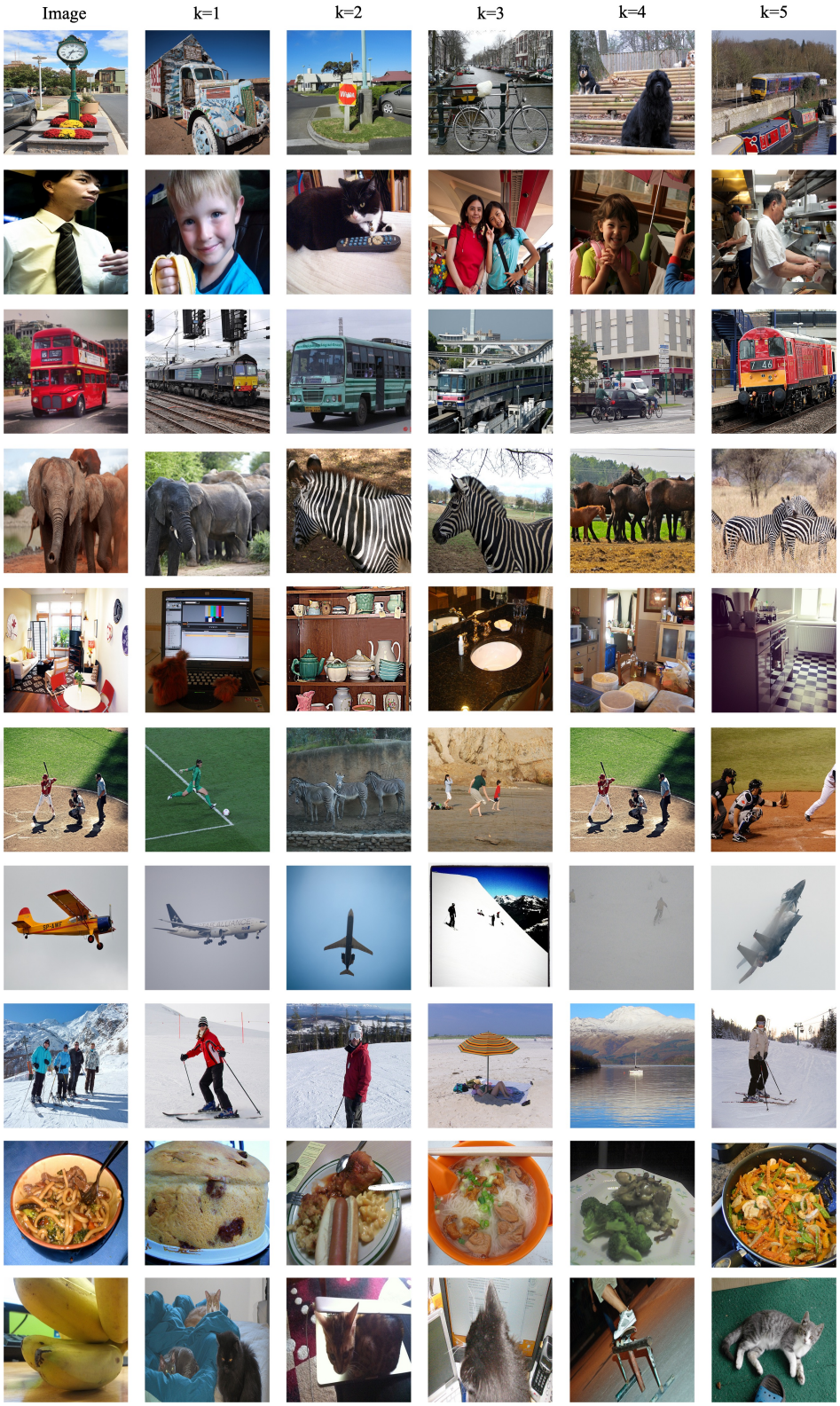


Figure 4.3. Visualization of the 5 most similar images according to Pixel-Level ReSSL. The images in the first column are the ones for which we are seeking the 5 most similar images. The images to their right are those 5 most similar ones.

In a surprising turn of events, the pixel-level ReSSL model's performance does not quite live up to initial expectations, particularly when placed in a qualitative comparison with other models. Despite delivering comparable quantitative results on the COCO dataset, as clearly illustrated in Figure 4.3, this model seems to fall short in certain areas. A potential problem that might be causing this could be the model's seeming inclination to place an over-emphasis on the low-level features in the background, such as snow or the color white. This issue comes into sharp focus in specific examples like the airplane and ski images where it's particularly noticeable. Additionally, the model seems to struggle when tasked with managing broader semantic classes, as is evident in cases like the bus-train and elephant-zebra images. At its worst, there are moments when the model seems to fail entirely, as is the case with the banana example.

In an effort to progress our understanding and improve future model performance, we carry out an all-encompassing, detailed analysis of the areas within images that our models primarily concentrate on. By incorporating the use of visual aids, we aim to deepen our comprehension of the attention maps that are generated by our models. These visual tools are indispensable in highlighting the areas that the models deem to be of significance or importance. We achieve this through the application of a gradient-based visual interpreter, GradCam, as mentioned in [60].

As illustrated qualitatively in Figure 4.4, the attention maps produced by our proposed models, specifically DenseCL + Consistency and DenseCL + PixContrast, are rich with valuable information that aids in the segmentation of an image. Broadly speaking, these models are equipped with the ability to concentrate on object areas in a way that DenseCL and PixContrast may not be capable of. Conversely, the Pixel-Level ReSSL model falls short in its qualitative object identification abilities, despite the fact that it delivers similar quantitative results when juxtaposed with the other models we proposed, as evidenced in Table 4.1 and Table 4.2. Further inspection of the attention maps brings to light that Pixel-Level ReSSL has a tendency to focus on the corners of the images, as opposed to the objects themselves.

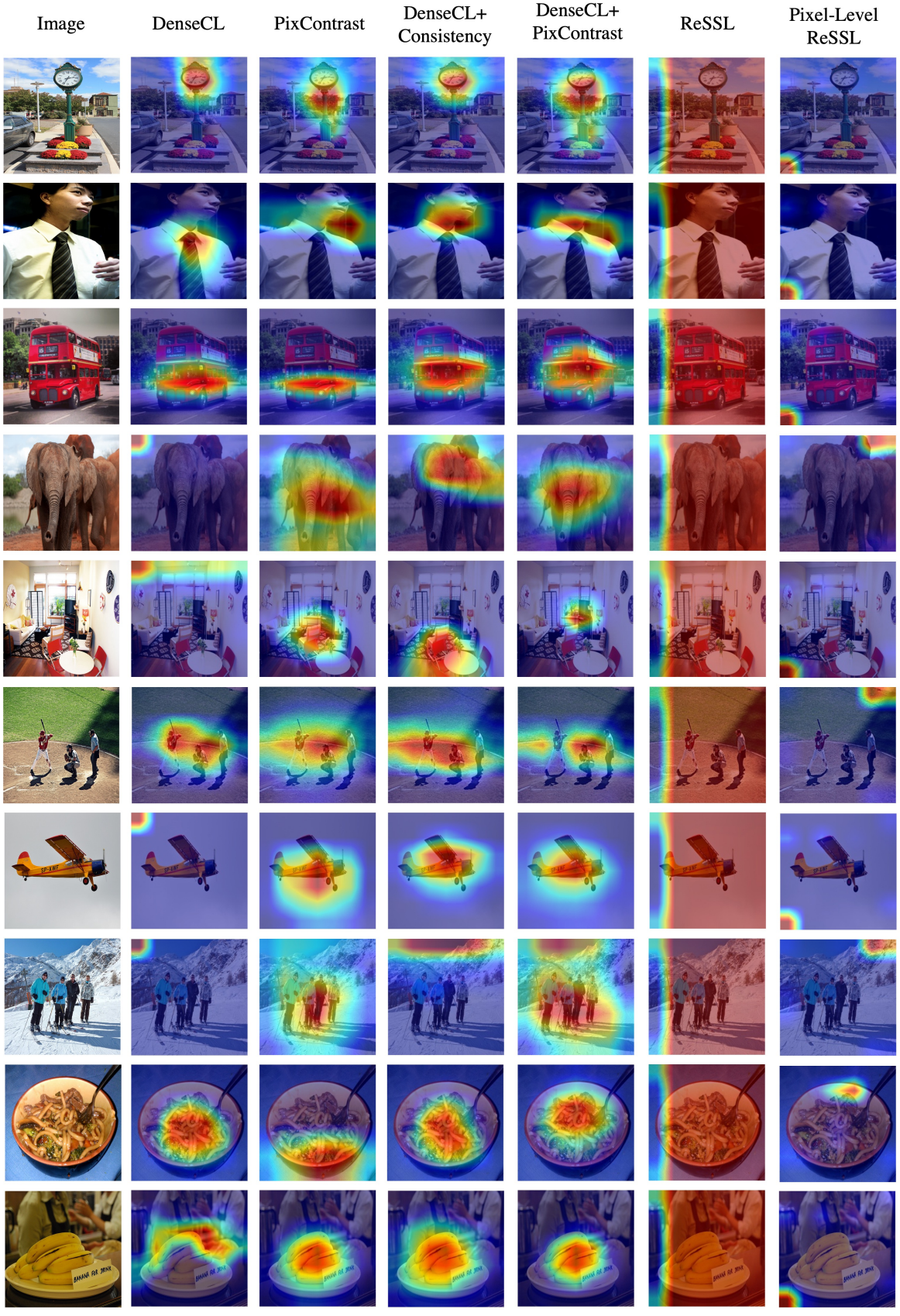


Figure 4.4. Visualization of Attention Maps.

5. CONCLUSION

Self-supervised Visual Representation Learning has gained popularity in recent years, and models pre-trained in this way have outperformed their supervised counterparts on many tasks (e.g. classification). While SSVRL ostensibly avoids the costly labeling problem, in practice most models are pre-trained on the well-organized object-centric ImageNet dataset. Moreover, most of the studies have been performed at the image level. Therefore, the full potential of this approach has not been reached, especially at the dense level (or pixel level).

The most prominent methods in the literature in this area are contrastive learning methods. However, this approach treats different instances as separate classes, independent of their semantic content. This problem is called the false negative matching problem. In addition, when contrastive learning is applied at the dense level, it is an important question how pixels are matched as positive-pairs.

In this thesis, we perform SSVRL at the dense level and try to provide solutions to the aforementioned problems. We first attempt to improve the DenseCL and PixContrast models that perform contrastive learning at the dense level. For this purpose, we try consistency regularization, which takes into account the similarity between instances and is introduced at the image-level in response to the false-negative pairing problem, on DenseCL at the dense level. In addition, we realized an approach that can be considered as a combination of DenseCL and PixContrast, where the matching strategies of both approaches are applied together. In second step, we adapted the relational self-supervised learning framework presented in image-level to dense-level.

In our experiments, we did our pre-training on COCO and not on ImageNet. Since we aim for dense-level representation learning, we performed our evaluations on Pascal VOC and COCO datasets on object detection and instance segmentation tasks, which are dense tasks.

DenseCL + Consistency provides +0.2 AP50 improvement in Pascal VOC object detection compared to DenseCL. In COCO, this improvement was consistently realized in object detection (+0.3 AP, +0.5 AP50, and +0.4 AP75) and instance segmentation (+0.3 AP, +0.5 AP50, and +0.1 AP75) across all metrics. DenseCL + PixContrast surpassed PixContrast on both datasets. However, it outperformed DenseCL in both tasks in the COCO dataset. Adapting ReSSL to dense-level also significantly improved the performance of the relational-framework on both datasets.

Our experiments show that the false negative pairing problem in contrastive learning is mitigated by taking into account the relationships between instances at the image level and pixels at the dense level with consistency regularization. On the other hand, since image-level helps to understand images at the global level and pixel-level helps to understand groups of pixels at the local level, the combination of the two has a significant impact on the performance of the relational framework.

We also evaluated the performance of the models both in terms of whether they were able to capture similarities between images and whether they were able to focus on the objects in the images. Our proposed DenseCL + Consistency and DenseCL + PixContrast models not only outperformed DenseCL and PixContrast models quantitatively, but also qualitatively. The Pixel-Level ReSSL model shows similar quantitative results but struggles to capture similarities between images and objects in images.

However, the fact that the models trained with COCO still underperform the models trained with ImageNet suggests that the methods should be better designed for scene-centric datasets. For example, after crop data augmentation, the views obtained in ImageNet are guaranteed to have similar semantic content, whereas this cannot be guaranteed in COCO. Therefore, the relationships between positive-pairs and negative-pairs need to be handled more precisely. Although this has been achieved to some extent in this study with consistency regularization and relation framework, it is open to further improvement in future studies.

REFERENCES

1. He, K., X. Zhang, S. Ren and J. Sun, "Deep Residual Learning for Image Recognition", *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Las Vegas, USA, pp. 770–778, 2016.

2. Deng, J., W. Dong, R. Socher, L.-J. Li, K. Li and L. Fei-Fei, "Imagenet: A Large-Scale Hierarchical Image Database", *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Miami, USA, pp. 248–255, 2009.

3. Radford, A., N. Karthik, S., Tim and S. Ilya, "Improving Language Understanding by Generative Pre-training", `https://cdn.openai.com/`, accessed on May 23, 2023.

4. Devlin, J., M.-W. Chang, K. Lee and K. Toutanova, "Bert: Pre-training of Deep Bidirectional Transformers for Language Understanding", arXiv:1810.04805, 2018.

5. Wang, X., R. Zhang, C. Shen, T. Kong and L. Li, "Dense Contrastive Learning for Self-Supervised Visual Pre-training", *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, Nashville, USA, pp. 3024–3033, 2021.

6. Misra, I. and L. v. d. Maaten, "Self-Supervised Learning of Pretext-Invariant Representations", *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, Seattle, USA, pp. 6707–6717, 2020.

7. Chen, T., S. Kornblith, M. Norouzi and G. Hinton, "A Simple Framework for Contrastive Learning of Visual Representations", *International Conference on Machine Learning (ICML)*, Vienna, Austria, pp. 1597–1607, 2020.

8. He, K., H. Fan, Y. Wu, S. Xie and R. Girshick, "Momentum Contrast for Unsupervised Visual Representation Learning", *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, Seattle, USA, pp. 9729–9738, 2020.

9. Chen, X., H. Fan, R. Girshick and K. He, "Improved Baselines with Momentum Contrastive Learning", arXiv:2003.04297, 2020.

10. Xie, Z., Y. Lin, Z. Zhang, Y. Cao, S. Lin and H. Hu, "Propagate Yourself: Exploring Pixel-Level Consistency for Unsupervised Visual Representation Learning", *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, Nashville, USA, pp. 16684–16693, 2021.

11. Dwibedi, D., Y. Aytar, J. Tompson, P. Sermanet and A. Zisserman, "With a Little Help from My Friends: Nearest-Neighbor Contrastive Learning of Visual Representations", *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, Nashville, USA, pp. 9588–9597, 2021.

12. Zhu, W., J. Liu and Y. Huang, "HNSSL: Hard Negative-Based Self-Supervised Learning", *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, Vancouver, Canada, pp. 4777–4786, 2023.

13. Huynh, T., S. Kornblith, M. R. Walter, M. Maire and M. Khademi, "Boosting Contrastive Self-Supervised Learning with False Negative Cancellation", *IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, Waikoloa, USA, pp. 2785–2795, 2022.

14. Wei, C., H. Wang, W. Shen and A. Yuille, "CO2: Consistent Contrast for Unsupervised Visual Representation Learning", arXiv:2010.02217, 2020.

15. Zheng, M., S. You, F. Wang, C. Qian, C. Zhang, X. Wang and C. Xu, "ReSSL: Relational Self-Supervised Learning with Weak Augmentation", *Advances in Neural Information Processing Systems (NeurIPS)*, pp. 2543–2555, 2021.

16. Zhang, R., P. Isola and A. A. Efros, "Colorful Image Colorization", *European Conference on Computer Vision (ECCV)*, Amsterdam, Netherlands, pp. 649–666, 2016.

17. Vondrick, C., A. Shrivastava, A. Fathi, S. Guadarrama and K. Murphy, "Track-

ing Emerges by Colorizing Videos", *European Conference on Computer Vision (ECCV)*, Munich, Germany, pp. 391–408, 2018.

18. Gidaris, S., P. Singh and N. Komodakis, "Unsupervised Representation Learning by Predicting Image Rotations", arXiv:1803.07728, 2018.

19. Doersch, C., A. Gupta and A. A. Efros, "Unsupervised Visual Representation Learning by Context Prediction", *IEEE International Conference on Computer Vision (CVPR)*, Boston, USA, pp. 1422–1430, 2015.

20. Noroozi, M. and P. Favaro, "Unsupervised Learning of Visual Representations by Solving Jigsaw Puzzles", *European Conference on Computer Vision (EECV)*, Amsterdam, Netherlands, pp. 69–84, 2016.

21. Dosovitskiy, A., J. T. Springenberg, M. Riedmiller and T. Brox, "Discriminative Unsupervised Feature Learning with Convolutional Neural Networks", *Advances in Neural Information Processing Systems (NIPS)*, Montreal, Canada, p. 766–774, 2014.

22. Wu, Z., Y. Xiong, S. X. Yu and D. Lin, "Unsupervised Feature Learning via Non-Parametric Instance Discrimination", *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Salt Lake City, USA, pp. 3733–3742, 2018.

23. Gutmann, M. and A. Hyvärinen, "Noise-Contrastive Estimation: A New Estimation Principle for Unnormalized Statistical Models", *13th International Conference on Artificial Intelligence and Statistics (AISTATS)*, Journal of Machine Learning Research Workshop and Conference Proceedings, Sardinia, Italy, pp. 297–304, 2010.

24. Koch, G., R. Zemel, R. Salakhutdinov *et al.*, "Siamese Neural Networks for One-Shot Image Recognition", *International Conference on Machine Learning (ICML), Deep Learning Workshop*, Lille, France, 2015.

25. Oord, A. v. d., Y. Li and O. Vinyals, "Representation Learning with Contrastive

Predictive Coding", arXiv:1807.03748, 2018.

26. O Pinheiro, P. O., A. Almahairi, R. Benmalek, F. Golemo and A. C. Courville, "Unsupervised Learning of Dense Visual Representations", *Advances in Neural Information Processing Systems (NeurIPS)*, Online, pp. 4489–4500, 2020.

27. Wang, Z., Q. Li, G. Zhang, P. Wan, W. Zheng, N. Wang, M. Gong and T. Liu, "Exploring Set Similarity for Dense Self-supervised Representation Learning", *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, New Orleans, USA, pp. 16590–16599, 2022.

28. Grill, J.-B., F. Strub, F. Altché, C. Tallec, P. Richemond, E. Buchatskaya, C. Doersch, B. Avila Pires, Z. Guo, M. Gheshlaghi Azar *et al.*, "Bootstrap Your Own Latent A New Approach to Self-Supervised Learning", *Advances in Neural Information Processing Systems (NeurIPS)*, Online, pp. 21271–21284, 2020.

29. Chen, X. and K. He, "Exploring Simple Siamese Representation Learning", *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, Nashville, USA, pp. 15750–15758, 2021.

30. Koohpayegani, S. A., A. Tejankar and H. Pirsiavash, "Mean Shift for Self-Supervised Learning", *IEEE/CVF International Conference on Computer Vision (CVPR)*, Nashville, USA, pp. 10326–10335, 2021.

31. Tejankar, A., S. A. Koohpayegani, V. Pillai, P. Favaro and H. Pirsiavash, "ISD: Self-Supervised Learning by Iterative Similarity Distillation", *IEEE/CVF International Conference on Computer Vision (CVPR)*, Nashville, USA, pp. 9609–9618, 2021.

32. Dosovitskiy, A., L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly *et al.*, "An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale", arXiv:2010.11929, 2020.

33. Caron, M., H. Touvron, I. Misra, H. Jégou, J. Mairal, P. Bojanowski and A. Joulin,

"Emerging Properties in Self-Supervised Vision Transformers", *IEEE/CVF International Conference on Computer Vision (ICCV)*, Nashville, USA, pp. 9650–9660, 2021.

34. Caron, M., P. Bojanowski, A. Joulin and M. Douze, "Deep Clustering for Unsupervised Learning of Visual Features", *European Conference on Computer Vision (ECCV)*, Munich, Germany, pp. 132–149, 2018.

35. Asano, Y. M., C. Rupprecht and A. Vedaldi, "Self-Labelling via Simultaneous Clustering and Representation Learning", arXiv:1911.05371, 2019.

36. Zhan, X., J. Xie, Z. Liu, Y.-S. Ong and C. C. Loy, "Online Deep Clustering for Unsupervised Representation Learning", *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, Seattle, USA, pp. 6688–6697, 2020.

37. Li, J., P. Zhou, C. Xiong and S. C. Hoi, "Prototypical Contrastive Learning of Unsupervised Representations", arXiv:2005.04966, 2020.

38. Caron, M., I. Misra, J. Mairal, P. Goyal, P. Bojanowski and A. Joulin, "Unsupervised Learning of Visual Features by Contrasting Cluster Assignments", Online, pp. 9912–9924, 2020.

39. Hénaff, O. J., S. Koppula, E. Shelhamer, D. Zoran, A. Jaegle, A. Zisserman, J. Carreira and R. Arandjelović, "Object Discovery and Representation Networks", *European Conference on Computer Vision (ECCV)*, Tel Aviv, Israel, pp. 123–143, 2022.

40. Wen, X., B. Zhao, A. Zheng, X. Zhang and X. Qi, "Self-Supervised Visual Representation Learning with Semantic Grouping", *Advances in Neural Information Processing Systems (NeurIPS)*, New Orleans, USA, pp. 16423–16438, 2022.

41. Lin, T.-Y., M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár and C. L. Zitnick, "Microsoft COCO: Common Objects in Context", *European*

Conference on Computer Vision (ECCV), Zurich, Switzerland, pp. 740–755, 2014.

42. Zbontar, J., L. Jing, I. Misra, Y. LeCun and S. Deny, "Barlow Twins: Self-Supervised Learning via Redundancy Reduction", *International Conference on Machine Learning (ICML)*, Online, pp. 12310–12320, 2021.

43. Wang, F., T. Kong, R. Zhang, H. Liu and H. Li, "Self-Supervised Learning by Estimating Twin Class Distribution", *IEEE Transactions on Image Processing*, Vol. 32, No. 1, pp. 2228–2236, 2023.

44. Bardes, A., J. Ponce and Y. LeCun, "VICReg: Variance-Invariance-Covariance Regularization for Self-Supervised Learning", arXiv:2105.04906, 2021.

45. Bardes, A., J. Ponce and Y. LeCun, "VICRegL: Self-Supervised Learning of Local Visual Features", *Advances in Neural Information Processing Systems (NeurIPS)*, New Orleans, USA, pp. 8799–8810, 2022.

46. Siméoni, O., G. Puy, H. V. Vo, S. Roburin, S. Gidaris, A. Bursuc, P. Pérez, R. Marlet and J. Ponce, "Localizing Objects with Self-Supervised Transformers and no Labels", arXiv:2109.14279, 2021.

47. Wang, Y., X. Shen, S. X. Hu, Y. Yuan, J. L. Crowley and D. Vaufreydaz, "Self-Supervised Transformers for Unsupervised Object Discovery using Normalized Cut", *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, New Orleans, USA, pp. 14543–14553, 2022.

48. Melas-Kyriazi, L., C. Rupprecht, I. Laina and A. Vedaldi, "Deep Spectral Methods: A Surprisingly Strong Baseline for Unsupervised Semantic Segmentation and Localization", *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, New Orleans, USA, pp. 8364–8375, 2022.

49. Van Gansbeke, W., S. Vandenhende and L. Van Gool, "Discovering Object Masks with Transformers for Unsupervised Semantic Segmentationn", arXiv:2206.06363,

2022.

50. Wang, X., R. Girdhar, S. X. Yu and I. Misra, "Cut and Learn for Unsupervised Object Detection and Instance Segmentation", *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, Vancouver, Canada, pp. 3124–3134, 2023.

51. Pathak, D., P. Krahenbuhl, J. Donahue, T. Darrell and A. A. Efros, "Context Encoders: Feature Learning by Inpainting", *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Las Vegas, USA, pp. 2536–2544, 2016.

52. Vincent, P., H. Larochelle, Y. Bengio and P.-A. Manzagol, "Extracting and Composing Robust Features with Denoising Autoencoders", *International Conference on Machine Learning (ICML)*, Helsinki, Finland, pp. 1096–1103, 2008.

53. Vaswani, A., N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser and I. Polosukhin, "Attention Is All You Need", *Advances in Neural Information Processing Systems (NIPS)*, Montreal, Canada, pp. 5998–6008, 2017.

54. Chen, M., A. Radford, R. Child, J. Wu, H. Jun, D. Luan and I. Sutskever, "Generative Pretraining from Pixels", *International Conference on Machine Learning (ICML)*, Online, pp. 1691–1703, 2020.

55. Li, Z., Z. Chen, F. Yang, W. Li, Y. Zhu, C. Zhao, R. Deng, L. Wu, R. Zhao, M. Tang *et al.*, "MST: Masked Self-Supervised Transformer for Visual Representation", *Advances in Neural Information Processing Systems (NeurIPS)*, pp. 13165–13176, 2021.

56. He, K., X. Chen, S. Xie, Y. Li, P. Dollár and R. Girshick, "Masked Autoencoders Are Scalable Vision Learners", *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, New Orleans, USA, pp. 16000–16009, 2022.

57. Girshick, R., "Fast R-CNN", *IEEE International Conference on Computer Vision*

(CVPR), Boston, USA, pp. 1440–1448, 2015.

58. He, K., G. Gkioxari, P. Dollár and R. Girshick, "Mask R-CNN", *IEEE International Conference on Computer Vision (CVPR)*, Honolulu, USA, pp. 2961–2969, 2017.

59. Lin, T.-Y., P. Dollár, R. Girshick, K. He, B. Hariharan and S. Belongie, "Feature Pyramid Networks for Object Detection", *IEEE International Conference on Computer Vision (CVPR)*, Honolulu, USA, pp. 2117–2125, 2017.

60. Selvaraju, R. R., M. Cogswell, A. Das, R. Vedantam, D. Parikh and D. Batra, "Grad-CAM: Visual Explanations from Deep Networks via Gradient-based Localization", *IEEE International Conference on Computer Vision (CVPR)*, Honolulu, USA, pp. 618–626, 2017.