Leveraging Image Mixtures for Unsupervised Detection Pre-training

Prudence Lam
Worcester Polytechnic Institute
{plam}@wpi.edu

Abstract

Self-supervised learning (SSL) aims to learn discriminative representations from images without the use of human annotations. However, using SSL for network pre-training often leads to poor transfer performance on object detection due to their reliance on high-level feature maps. Moreover, SSL methods designed for detection often sacrifice performance on image classification. Inspired by the recent success of image mixtures in supervised networks, we propose an SSL strategy that utilizes both image mixtures and their components to improve localization ability, while maintaining competitive image classification accuracy. Our experiments on CIFAR-10, as well the visualization of attention maps, yield promising results. Our code is available at https://github.com/p-lam/mix-ssl.

1. Introduction

In recent times, self-supervised learning (SSL) has gained considerable attention for its ability to outperform supervised pre-training for various downstream tasks [1, 2, 4, 6, 7, 10, 16]. These methods encourage representations to remain invariant to data augmentations, typically by maximizing the similarity between embedding vectors from augmented views of the same image. Due to the possibility of trivial solutions (i.e. the model maps all inputs to the same vector) studies have proposed a variety of approaches. Contrastive learning, such as Chen et al. [3] and He et al. [7], employs the use of ”negative” samples to repulse similar (”positive”) samples. Clustering-based approaches [2] enforce cluster assignments for different image augmentations to remain consistent. BYOL [6] and SimSiam [4] remove the need for negative samples using ”tricks” such as momentum encoders and stop-gradient operations. More recently, [1, 16] utilize information-maximization techniques to reduce redundant information between embeddings.

Despite the success of SSL for pretraining, there remains a gap in performance between image classification and dense prediction tasks, such as object detection [5, 8] and semantic segmentation. Many of these SSL techniques build upon the task of instance discrimination, which treats each image in a dataset and its transformations as a separate class. As a result, they transfer well to image classification tasks, which identify global instances from high-level feature maps, but fail to address local instances critical to object detection.

In this paper, we address this limitation by introducing an SSL strategy that exploits the relationship between local and global image patches. Specifically, we take advantage of the local and global crops used in image mixtures, namely CutMix [15]. Recent studies have proven that image mixtures can improve the localization ability of supervised networks [9] and promote label smoothing in SSL networks [11]. We hypothesize that the combination of CutMix image representations with their patch components can prompt networks to learn more consistent representations between different image scales.

Our contributions can be summarized as follows:

- We utilize representations from global and local image crops used in CutMix augmentations to account for relationships between different image views.
- We experiment with two different mixing strategies in SSL joint-embedding architectures, namely between branches or within each mini-batch of training.
- Our proposed methods perform competitively in downstream classification and detection tasks and can be implemented using just a few lines of code.

2. Related Work

2.1. Contrastive Learning

Contrastive learning aims to maximize the similarity between ”positive” pairs of images while minimizing the similarity between ”negative” pairs. Many contrastive learning approaches utilize the InfoNCE loss [13], which approximates the mutual information between a pair of images by comparing a positive pair with a batch of negative pairs. These approaches typically use data augmentations to define invariance between embeddings. One well-known example is SimCLR [3], which augments an image twice to form a positive pair, and pairs it with all other images in a batch to form negative ones. More recent SSL methods adopt a *joint embedding* architecture, training a pair of networks to produce similar embeddings for different transformations of the same image. Momentum Contrast [7], or MoCo, treats contrastive learning as a dictionary lookup problem, training a ”query” encoder using negative samples, or ”keys”, from a memory bank. We will utilize the MoCo architecture [7] for learning representations in this study.

2.2. CutMix

Proposed by Yun et al., CutMix [15] is a data augmentation method and regularization strategy for deep neural networks. CutMix creates new training images by randomly cropping one image and filling the crop space with patches from the other image, with the new label being the proportion of how much of each image remains. Eq 1 shows the algorithm for combining the images, with (x_A, y_A) being the first image and label and (x_B, y_B) being the second image. M is the binary mask that indicates the cutout and the fill-in regions from the two randomly drawn images and λ is the combination ratio.

$$\begin{aligned} \tilde{x} &= \boldsymbol{M} \odot x_A + (\mathbf{1} - \boldsymbol{M}) \odot x_B \\ \tilde{y} &= \lambda y_A + (1 - \lambda)\, y_B \end{aligned} \tag{1}$$

More recently, a study by [11] showed that image mixtures, including CutMix, can promote softened similarity distances between embeddings in contrastive learning. When applied to SSL networks, it encouraged label smoothing across positive and negative pairs, and provided a performance boost on classification.

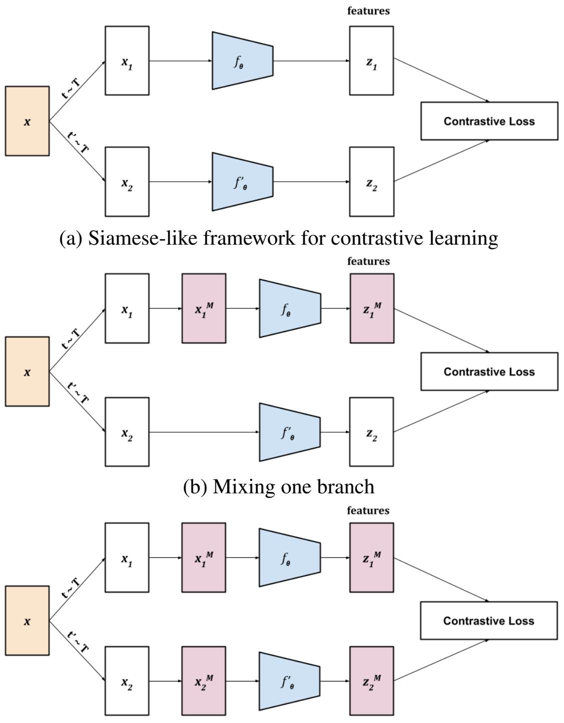
3. Proposed Method

In this section, we first present the architectures explored in this background, namely the inclusion of CutMix in one or both networks. Then, we present the design of extracting the global and local features from a mixed image. Finally, we describe the implementation details and loss functions of our approach.

3.1. Mixture Frameworks

We define an image x as having two augmentations, x_1 and x_2. In a typical siamese-like framework for contrastive learning, each image is fed through an encoder f_θ to produce a latent representation. The representations are projected to produce a final representation vector, z, on which contrastive loss can be performed (see Fig 1a). Let us also define a mixed image transformation as x^M. Inspired by [11], we experiment with two different mixture paradigms:

Mixing in one branch (Fig1b): According to Shen et al., simply mixing an image in one branch (i.e. x_1) achieves greater efficiency than mixing both branches, as it requires only one additional forward pass. In extracting local and global patches, however, this method allows only for the comparison between global ↔ global and local ↔ global crops, or global ↔ global and local ↔ local crops, as the augmented images will be mixed with others not derived from the same image.



(a) Siamese-like framework for contrastive learning

(b) Mixing one branch

(c) Mixing both branches

Figure 1. **Different types of mixture paradigms**

Mixing in both branches (Fig1c): In this case, both x_1 and x_2 are mixed (see Fig 1(c)). We select this as the main strategy in our work, due to the method’s ability to compare global ↔ global, local ↔ local, and global ↔ local features.

3.2. Proposed Approaches

We reference MoCo [7], which consists of a memory bank for negative keys $\{k_0, k_1, k_2\}$, a backbone network, and several projection heads. Each query image is denoted as q. MoCo aims to train the query encoder, `encoder_q` with the key encoder, `encoder_k` using the same architecture for both. `encoder_k` is updated using a momentum update of the query encoder. For updating `encoder_q`, MoCo utilizes the InfoNCE loss [13], expressed as:

$$\mathcal{L}_q = -log \frac{exp(q \cdot k_+ / \tau)}{\sum_{i=0}^{K} exp(q \cdot k_i / \tau)} \tag{2}$$

where the similarity between a positive pair (q and k_+) is calculated through their dot product, and τ is a temperature

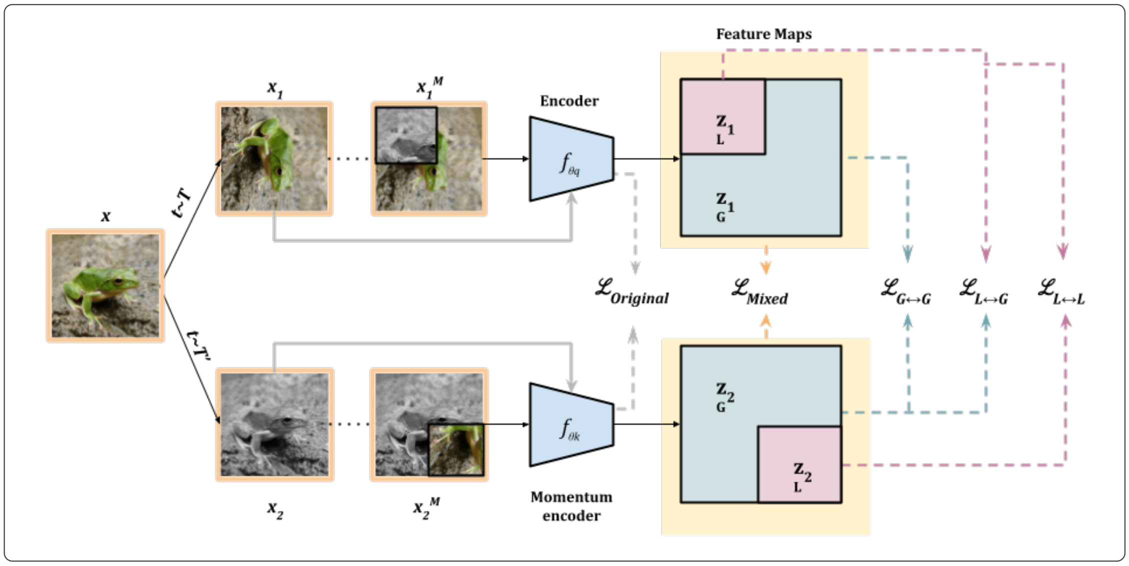


Figure 2. **The overall pipeline of Approach 1.** An image is augmented twice, and the augmentations are mixed with each other using CutMix [15]. The augmentations and their corresponding mixtures are encoded, and from it, the average feature inside and outside the cropped region are computed. The network makes use of three memory banks of negative samples, namely for the augmentations, the global features, and the local features.

hyper-parameter for scaling the weights of hard negatives [14].

3.2.1 Direct mixing between branches

Our first approach involves mixing x_1 with x_2 with CutMix, such that x_1^M has a patch replaced by x_2, and vice versa. The overall architecture is illustrated in Figure 2. Both the transformed images and the image mixtures are fed through an encoder, which outputs their respective latent representations. For the representations of x_1 and x_2, we linearly project them and calculate their loss using the InfoNCE equation from Eq2. We denote this as our "original" loss, $\mathcal{L}_{original}$. We additionally keep a running queue of "original" negative images that is updated by mini-batches of x_2. Furthermore, we calculate a contrastive loss for the mixed image, formulated as:

$$\mathcal{L}_{Mixed} = -log \frac{exp(q^M \cdot k_+/\tau)}{\sum_{i=0}^{K} exp(q^M \cdot k_i/\tau)} \tag{3}$$

Given a feature map, the network upsamples it to the original input size with bilinear interpolation. The average feature inside and outside of the cropped region is computed by applying a binary mask to the upsampled feature map. The global ↔ global contrastive loss can be written as:

$$\mathcal{L}_{G\leftrightarrow G} = -log \frac{exp(q^G \cdot k_+^G/\tau)}{\sum_{i=0}^{K} exp(q^G \cdot k_i^G/\tau)} \tag{4}$$

Similarly, the local ↔ local and local ↔ global loss are expressed as in Eq(3) and (4) respectively.

$$\mathcal{L}_{L\leftrightarrow L} = -log \frac{exp(q^L \cdot k_+^L/\tau)}{\sum_{i=0}^{K} exp(q^L \cdot k_i^L/\tau)} \tag{5}$$

$$\mathcal{L}_{L\leftrightarrow G} = -log \frac{exp(q^L \cdot k_+^G/\tau)}{\sum_{i=0}^{K} exp(q^L \cdot k_i^G/\tau)} \tag{6}$$

To compute Eq 4, 5, and 6, we keep two dynamic queues for negative samples of local and global images respectively, resulting in three individual memory banks for the network. Given the above loss terms, we define the complete loss function as:

$$\begin{aligned}\mathcal{L}_{final} = L_{original} + \lambda(\mathcal{L}_{mixed}) + \\ (1-\lambda)(\mathcal{L}_{\mathcal{L}\leftrightarrow\mathcal{G}} + \mathcal{L}_{\mathcal{L}\leftrightarrow\mathcal{L}} + \mathcal{L}_{\mathcal{L}\leftrightarrow\mathcal{L}})\end{aligned} \tag{7}$$

where λ is a hyperparameter to balance out the effect of the mixed terms with the global and local terms on the original loss.

3.2.2 Mixing within each Mini-Batch of Training

We utilize the self-mixing strategy proposed by [11] to evaluate the efficacy of local and global features. In each mini-batch of training, the first image is directly mixed with

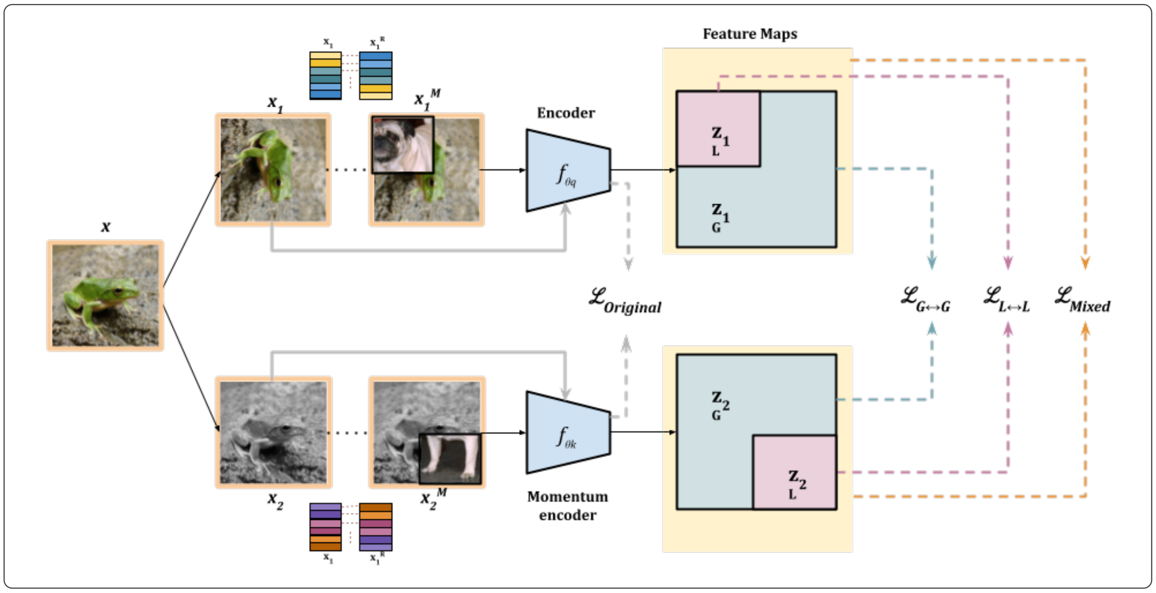


Figure 3. **The overall pipeline of Approach 2.** An image is augmented twice, and the augmentations are mixed with an image within the batch. Similar to Approach 1, the average features inside and outside the cropped region are computed from the CutMix embeddings, and three memory banks are used to store the negative pairs for the original, global, and local images. One key difference is that the loss function in Approach 2 omits the relationship between local ↔ global crops, as the crops are no longer semantically similar.

the last image in the batch, the second image with the penultimate image, and so forth. This method maintains a consistent distance between the vector encodings of the original image samples and their mixtures. Unlike the previous approach, this network cannot include a local ↔ global loss term, as the local and global patches of the query and keys come from different images. The full architecture for this approach is shown in Fig 3.

We reflect this mixing approach in the loss function by computing the reverse orders of each image batch. Doing so allows us to consider each crop in the mixture as a ”global” crop and a ”local” crop individually. Similar to the previous approach, we calculate each loss term using the InfoNCE loss. We define the final loss term as:

$$\mathcal{L}_{final} = L_{original} + \lambda(\mathcal{L}_{mixed} + \mathcal{L}_{\mathcal{G}\leftrightarrow\mathcal{G}} + \mathcal{L}_{\mathcal{L}\leftrightarrow\mathcal{L}}) + (1-\lambda)(\mathcal{L}^{R}_{mixed} + \mathcal{L}^{\mathcal{R}}_{\mathcal{G}\leftrightarrow\mathcal{G}} + \mathcal{L}^{\mathcal{R}}_{\mathcal{L}\leftrightarrow\mathcal{L}}) \quad (8)$$

where $\mathcal{L}^{R}_{mixed}$, $\mathcal{L}^{\mathcal{R}}_{\mathcal{G}\leftrightarrow\mathcal{G}}$, and $\mathcal{L}^{\mathcal{R}}_{\mathcal{L}\leftrightarrow\mathcal{L}}$ are equivalent to Eq (3), (4), and (5), but with queries in their reverse order. Similarly, we keep a dynamic queue for original, local, and global negative samples.

4. Experiment

4.1. Implementation Details

We studied unsupervised training of our model on CIFAR10, which contains 50,000 32 × 32 images drawn from 10 classes. We follow the parameters of MoCo in our experiments, including an SGD optimizer with a weight decay of 0.0001 and a momentum of 0.9. To save computation time, we selected a mini-batch size of 512, and trained a ResNet-18 backbone for 1000 epochs at a learning rate of 3×10^{-3} with a cosine annealing schedule.

Evaluation Protocol We evaluate our network using the weighted K-Nearest Neighbors (KNN) classification method from [14]. For each query, we extract the top 200 most similar images used to predict the label. We selected several well-known baselines for comparison, including SimCLR, BYOL, and MoCo, the baseline architecture for our models. Additionally, we tested MoCo with just the mixture loss ($L_{final} = L_{original} + \lambda L_{mixed}$), found in both Approach 1 and 2, as well as MoCo with Un-Mix, which our paper was inspired by.

5. Results

5.1. Image Classification on CIFAR-10

As shown in Table 1, MoCo with Un-Mix achieved the greatest classification accuracy of 91.34%. However, our models' top-1 accuracies are comparable, with 90.15% for Approach 1 and 89.97% for Approach 2. Additionally, they perform marginally better than MoCo. Interestingly enough, MoCo with just the mixture loss performs comparably to MoCo with Un-Mix, suggesting that self-mixing within mini-batches is not always more beneficial.

Table 1. 5-nearest neighbors classification results with a ResNet-18 backbone. Like MoCo, we trained using 1000 epochs and an asymmetric loss, and evaluated with $k = 200$ in KNN monitor.

Method	**CIFAR-10**
	5-nn
SimCLR [3]	88.42
BYOL [6]	94.20
MoCo [7]	88.97
MoCo + CutMix	91.31
MoCo + Un-Mix [11]	91.34
Approach 1 (ours)	90.15
Approach 2 (ours)	89.97

5.2. Visualization Results

Since existing baselines for SSL performance on downstream object detection tasks consist of ImageNet pre-training, we opted to save computation and visualize the attention maps of the final layer of our pre-trained encoders instead. Figure 4 shows the visualization of these maps for the MoCo and MoCo+CutMix baselines as well as our two approaches. Overall, our methods appear to be more sensitive to background noise. For example, in the image of the horse, MoCo locates just the head, whereas both Approach 1 and 2 react strongly to the head, trees, fence, and clouds. Additionally, our methods appear to perform more poorly on easier localization tasks (i.e. the first image of the bird), but respond to multiple object instances when present (i.e. the image of the deer).

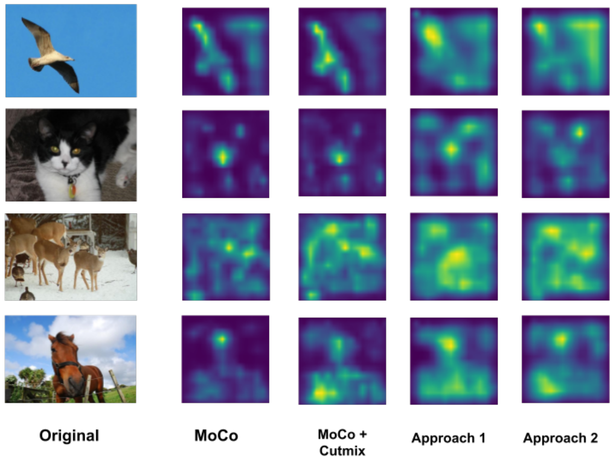


Figure 4. Attention maps generated by our baselines, MoCo and MoCo + CutMix, in addition to our models

6. Discussion

Our work seeks to combine image mixtures with their local and global image views to improve the localization ability of SSL methods for downstream object detection tasks. We proposed two different approaches to creating image mixtures, namely mixing between branches and within each mini-batch of training.

For image classification, our incorporation of local and global image features did not appear to have as much of an impact as CutMix alone. In particular, MoCo performed better with just the original and mixture loss, in contrast to combinations of losses between local and global image patches. One reason could be that the addition of loss terms introduces unwanted noise to the softened distance losses imposed by CutMix.

In terms of object detection, our methods react more strongly to background noise for simpler tasks, i.e. only one object is present. However, for images with multiple class instances, our approaches can localise more objects than MoCo. Thus, we see incorporating local and global patch losses as beneficial for future work on SSL for object detection.

7. Summary and Future Work

Our work analyzed the viability of combining the representations of mixture operations with their local and global parts. From experiments on CIFAR-10, we have shown that our approaches perform competitively on image classification. Based on their attention maps, we can infer that our methods improve localization to some extent. Additionally, the inclusion of the mixture loss alone appears to improve MoCo drastically for both downstream tasks, more so than either of our approaches. We believe that future work on larger datasets is required to justify our methods. We also believe that using other mixture techniques, such as SaliencyMix [12], may help incorporate more semantic information into pre-training techniques.

References

[1] Adrien Bardes, Jean Ponce, and Yann LeCun. Vicreg: Variance-invariance-covariance regularization for self-supervised learning. In *ICLR*, 2022. 1

[2] Mathilde Caron, Ishan Misra, Julien Mairal, Priya Goyal, Piotr Bojanowski, and Armand Joulin. Unsupervised learning of visual features by contrasting cluster assignments. 2020. 1

[3] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. In *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pages 1597–1607. PMLR, 13–18 Jul 2020. 1, 5

[4] Xinlei Chen and Kaiming He. Exploring simple siamese representation learning. *arXiv preprint arXiv:2011.10566*, 2020. 1

[5] M. Everingham, S. M. A. Eslami, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman. The pascal visual ob-

ject classes challenge: A retrospective. *International Journal of Computer Vision*, 111(1):98–136, Jan. 2015. 1

[6] Jean-Bastien Grill, Florian Strub, Florent Altché, Corentin Tallec, Pierre H. Richemond, Elena Buchatskaya, Carl Doersch, Bernardo Avila Pires, Zhaohan Daniel Guo, Mohammad Gheshlaghi Azar, Bilal Piot, Koray Kavukcuoglu, Rémi Munos, and Michal Valko. Bootstrap your own latent: A new approach to self-supervised learning, 2020. 1, 5

[7] Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross Girshick. Momentum contrast for unsupervised visual representation learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2020. 1, 2, 5

[8] Tsung-Yi Lin, Michael Maire, Serge Belongie, Lubomir Bourdev, Ross Girshick, James Hays, Pietro Perona, Deva Ramanan, C. Lawrence Zitnick, and Piotr Dollár. Microsoft coco: Common objects in context. 1

[9] Runji Liu, Ying Chen, Jiasheng Wang, and Zhaojin Guo. Attentive mix: An efficient data augmentation method for object detection. In *2021 7th International Conference on Computer and Communications (ICCC)*, pages 770–774, 2021. 1

[10] Ishan Misra and Laurens van der Maaten. Self-supervised learning of pretext-invariant representations. *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 6706–6716, 2020. 1

[11] Zhiqiang Shen, Zechun Liu, Zhuang Liu, Marios Savvides, Trevor Darrell, and Eric Xing. Un-mix: Rethinking image mixtures for unsupervised visual representation learning. 2022. 1, 2, 3, 5

[12] A F M Shahab Uddin, Mst. Sirazam Monira, Wheemyung Shin, TaeChoong Chung, and Sung-Ho Bae. Saliencymix: A saliency guided data augmentation strategy for better regularization. In *International Conference on Learning Representations*, 2021. 5

[13] Aäron van den Oord, Yazhe Li, and Oriol Vinyals. Representation learning with contrastive predictive coding. *ArXiv*, abs/1807.03748, 2018. 1, 2

[14] Zhirong Wu, Yuanjun Xiong, Stella Yu, and Dahua Lin. Unsupervised feature learning via non-parametric instance-level discrimination, 2018. 3, 4

[15] Sangdoo Yun, Dongyoon Han, Seong Joon Oh, Sanghyuk Chun, Junsuk Choe, and Youngjoon Yoo. Cutmix: Regularization strategy to train strong classifiers with localizable features. In *International Conference on Computer Vision (ICCV)*, 2019. 1, 2, 3

[16] Jure Zbontar, Li Jing, Ishan Misra, Yann LeCun, and Stephane Deny. Barlow twins: Self-supervised learning via redundancy reduction. In Marina Meila and Tong Zhang, editors, *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pages 12310–12320. PMLR, 18–24 Jul 2021. 1