Masking Improves Contrastive Self-Supervised Learning for ConvNets, and Saliency Tells You Where

Zhi-Yi Chin[1]* Chieh-Ming Jiang[1]* Ching-Chun Huang[1] Pin-Yu Chen[2] Wei-Chen Chiu[1]
[1] National Yang Ming Chiao Tung University [2] IBM Research
{joycenerd.cs09, nax1016.cs10, chingchun, walon}@nycu.edu.tw, pin-yu.chen@ibm.com

Abstract

While image data starts to enjoy the simple-but-effective self-supervised learning scheme built upon masking and self-reconstruction objective thanks to the introduction of tokenization procedure and vision transformer backbone, convolutional neural networks as another important and widely-adopted architecture for image data, though having contrastive-learning techniques to drive the self-supervised learning, still face the difficulty of leveraging such straightforward and general masking operation to benefit their learning process significantly. In this work, we aim to alleviate the burden of including masking operation into the contrastive-learning framework for convolutional neural networks as an extra augmentation method. In addition to the additive but unwanted edges (between masked and unmasked regions) as well as other adverse effects caused by the masking operations for ConvNets, which have been discussed by prior works, we particularly identify the potential problem where for one view in a contrastive sample-pair the randomly-sampled masking regions could be overly concentrated on important/salient objects thus resulting in misleading contrastiveness to the other view. To this end, we propose to explicitly take the saliency constraint into consideration in which the masked regions are more evenly distributed among the foreground and background for realizing the masking-based augmentation. Moreover, we introduce hard negative samples by masking larger regions of salient patches in an input image. Extensive experiments conducted on various datasets, contrastive learning mechanisms, and downstream tasks well verify the efficacy as well as the superior performance of our proposed method with respect to several state-of-the-art baselines. Our code is publicly available at: https://github.com/joycenerd/Saliency-Guided-Masking-for-ConvNets

1. Introduction

The recent renaissance of deep learning techniques has brought a magic leap to various fields, such as computer vision, natural language processing, and robotics. Learning from a large-scale labeled/supervised dataset, which is one of the key factors leading to the success of deep learning, however, has now turned out to be a significant limitation on its extensions to more fields. In addition to the expensive cost of time and human resources to collect training datasets for different tasks and their corresponding labels, the supervised learning scenario typically would suffer from the issue of overfitting on the training dataset, thus leading to worse generalizability of the learnt models. These problems bring challenges for the application of deep learning techniques but also give rise to the research topic of self-supervised learning, wherein it aims to learn to extract informative feature representations from an unlabelled dataset via leveraging the underlying structure of data and building the supervisory signals from the data itself. The discovered representations are typically more general and can be further utilized or fine-tuned to various downstream tasks.

Without loss of generality, self-supervised learning has firstly made great success in natural language processing, where the autoregressive modeling (i.e. predicting the next word given the previous words) and masked modeling (i.e. masking operation to randomly mask a portion of words in a text, coupled with a self-reconstruction objective to predict those masked words) bring up the powerful language models such as GPT [26] and BERT [7]. Nevertheless, the direct adaptation of such techniques (especially masking and self-reconstruction) to image data [24] only contributes to slight improvement (at least not as significant as what happens in the field of natural language processing), in which such a predicament was later relieved with the help of vision transformers [8] (e.g. the influential work from masked autoencoder (MAE) [15] and the related ones such as SimMIM [33], BEiT [1], and iBOT [37]). In contrast to vision transformers which enable the application of masking operation and its coupled self-reconstruction objective on the

*These authors contributed equally to this work self-supervised learning for vision data, another dominant architecture over the last decade for computer vision field, i.e. convolutional neural networks, has difficulty incorporating the random masking operation (on image patches), because the resultant edges between masksed and unmasked regions could cause problems for learning convolution kernels, and the nature of performing convolutions on regular grids also hinder it from adopting positional embeddings or masked tokens as the typical transformer models [15].

In turn, the most popular self-supervised learning scenarios nowadays for convolution neural networks come from contrastive learning – given one sample image, two different views of it are respectively created by two different augmentations. The contrastive objective which attracts the views from the same image (known as positive pair/views) while repelling the ones from distinct images (respectively, negative pair/views) drives the learning of feature extractor (i.e. encoder) to capture the crucial features invariant to augmentations. Hence, how to design good positive and negative views with augmentations [12,25] plays an important role in the success of contrastive learning, in which the design choices for augmentations typically are highly dependent on the characteristics of the image data domain (i.e. more domain-specific). From such a point of view, including the less domain-specific augmentations would definitely be able to benefit the versatility and the flexibility of the corresponding contrastive learning algorithms, thus the fundamentals of masking (as being one of the most straightforward and general operations) consequently come into our sight: *Are we able to include masking as an extra augmentation method into contrastive self-supervised learning framework with convolutional neural networks as its backbone?*

We are not the first to ask such a question. Two prior works (i.e. **MSCN** [18] and **ADIOS** [30]) proposed to tackle the issues "how to mask" and "learning where to mask" respectively. Nevertheless, on one hand, the improvements provided by these prior works are relatively insignificant, thus showing this topic is still under-explored; on the other hand, we highlight the potential issue that: if the masking is performed in a completely random manner, there exists a chance where all the masked patches fall on either the foreground or the background objects, in which the contrastive objective upon positive pair under such case could be detrimental for the overall model learning (e.g. attracting two views where one is completely background while the other still owns most of the foreground).

To address this potential issue, we propose to particularly include **saliency** as a prior before performing masking. That is, we suggest that the masked patches should be evenly distributed to an image's foreground objects and background, regardless of the masking ratio. To this end, we introduce ***random masking with saliency constraint*** as an augmentation method for the contrastive self-supervised learning framework, in which the feature extractor is built upon convolutional neural networks. Basically, we split the entire input image into the foreground objects and the background, followed by performing random masking on them separately, and three different masking strategies are provided to handle the parasitic edges stemming from masking. Moreover, we also introduce hard negative samples by masking more salient patches of the original input image, where these hard negative samples are experimentally shown to bring an extra boost to our proposed method. Lastly, we further discover that masking only one branch (to be specific, masking only the query branch when processing the positive pairs) of the contrastive learning framework (usually also known as *siamese network*) provides better performance than masking both branches due to the effects in terms of sample variance that masking brings, which also well corroborates the statement claimed in [31]. Our main contributions in this work are summarized as follows,

- We propose a saliency masking augmentation method for the contrastive self-supervised learning framework with convolutional neural networks as backbones, where the saliency information is utilized to guide the random masking applied on the foreground and background regions individually.
- Three masking strategies are proposed to tackle parasitic edges between masked and unmasked regions, in which hard negative samples can also be created by masking more salient patches to achieve further improvement.
- From the perspective of manipulating the difference in terms of variance between two branches of the siamese network, we propose to apply masking augmentation solely on the query branch when processing positive pairs to benefit the model training.

2. Related work

Self-Supervised Learning (SSL) aims to learn a feature encoder for extracting representations from unlabeled data via the help of pretext tasks (where the objective functions for these tasks are typically built upon the data properties), in which the resultant encoder can be further fine-tuned with labeled data to support different downstream tasks. Early SSL works rely on designing handcrafted pretext tasks, such as predicting rotation angles [11, 13], solving jigsaw puzzles [22], or colorization [36]. Recently, the introduction of contrastive objectives to SSL algorithms [3, 4, 14, 16, 35] have brought a significant leap of performance, even providing superiority to some standard supervised learning baselines. Contrastive SSL tries to maximize the agreement between representations of positive samples (i.e. augmented views from the same source image). While some contrastive SSL approaches (such as SimCLR [4] and MoCov2 [16]) further leverage negative samples (i.e., augmented views from different images) to prevent the model collapse (i.e. learning trivial features) by utilizing large batch size and memory bank, some other works (e.g. SimSiam [6] and BYOL [14]) instead prove that the negative samples might not be necessary for contrastive SSL (e.g. the stop-gradient technique serve the same purpose to prevent model collapse). Though there exist other categories of SSL methods (e.g. clustering ones such as DeepCluster [2] and SWAV [3]), the contrastive ones still take the lead in the stream of SSL.

Masking in SSL (e.g. masking out a portion of input data sample followed by learning to recover the missing content) is firstly proved by the success of masked language modeling (e.g. BERT [7]) and later adapted to the vision data, thanks to the introduction of vision transformer backbones where the input images are firstly divided into patches then tokenized. For instance, MAE [15] as a seminal work proposes an autoencoder architecture where the transformer-based encoder turns the unmasked image patches into feature representations, which are further decoded back to the original image; while SimMIM [33] encodes the entire image, including the masked patches, and predicts the missing region with a lightweight one-layer head. Compared to the SSL methods for vision data which are based on the coupled masking operation and self-reconstruction loss but highly constrained to the transformer backbone, contrastive SSL becomes more friendly for adopting another important computer vision backbone, convolutional neural networks (also abbreviated as ConvNets), in which recently there comes some research works to investigate the plausibility of including masking operation as an augmentation method into contrastive SSL for ConvNets (also denoted as "siamese networks with ConvNets" in this paper).

MSCN [18] firstly discusses the issues of adopting masking operation in siamese networks with ConvNets (including the parasitic edges on the masked input, introduction of superficial solutions, distortion upon the balance between local and global features, and having fewer training signals) then proposes several designs to tackles these issues, such as adopting high-pass filter to alleviate the impact of parasitic edges or applying focal masks to balance short-range and long-range features. Basically, MSCN focuses more on the perspective of "how to mask". In comparison, our work not only provides more "how to mask" strategies (in addition to the one using a high-pass filter, two more based on strong blurring and filling mean value are proposed) but also explicitly includes saliency constraint (from the perspective of "where to mask") as well as extensions on learning mechanism (i.e. using masking to produce hard negative samples and manipulate variance across siamese branches); ADIOS [30] mainly tackles the issue of "where to mask", where instead of using random masking, it particularly adopts an occlusion module (UNet-based, acting as a segmentation model) which learns adversarially along with the feature encoder to determine the regions to be masked, hence the produced masks are semantically meaningful. As jointly training the feature encoder and occlusion module results in heavy computational cost for ADIOS, our proposed method strikes a better balance between having (partially) semantic masks (as being guided by saliency to separate foreground and background) and the computation efforts (as our saliency is estimated by a pretrained and frozen localization network). The visualization to highlight the difference among our proposed method, MSCN [18], and ADIOS [30] is provided in Figure 2.

3. Method

As motivated previously, we would like to include the masking operation as an extra augmentation method into contrastive self-supervised learning with ConvNets as backbone, where the saliency information is particularly leveraged to guide the masking. Our full model is shown in Figure 1 where in the following we will sequentially describe the saliency computation, our various saliency-guided masking strategies, the ways to construct positive and hard negative samples, as well as the learning scheme.

3.1. Saliency computation

The idea of saliency was firstly introduced to predict the eye-catching regions over an image, in which here we generalize such idea to localize the main objects in an input image (which are corresponding to "foreground" without loss of generality) while the rest is then treated as "background". To this end, in this work we adopt the Selective Convolutional Descriptor Aggregation (**SCDA** [32]) method to build our *localization network* f_ξ for producing saliency map M of a given image X, i.e. $M = f_\xi(X)$. The main reason to choose SCDA for our use stems from its simplicity of only requiring a pre-trained CNN model (typically for the task of classification) and demanding no further supervision to localize the main objects. With denoting the feature tensor obtained by the aforementioned pre-trained CNN model prior to its global average pooling layer as $S \in \mathbb{R}^{U \times V \times D}$, the aggregation (by summation) of S along the channel dimension results in the activation map $A \in \mathbb{R}^{U \times V}$. In addition to use the mean value $\bar{a}$ of A as the threshold on all the $U \times V$ elements in A to locate the positions of foreground objects (same as the original SCDA), we add another condition based on the standard deviation σ of A to have more flexible localization results which better fit our need to later guide the masking:

$$M(u,v) = \begin{cases} 1 & \text{if } A(u,v) \geq \bar{a} - 0.6 \cdot \sigma \\ 0 & \text{otherwise} \end{cases} \tag{1}$$

Take an input image of size 224×224 and use an ImageNet-pretrained CNN model based on ResNet-50 as an example,

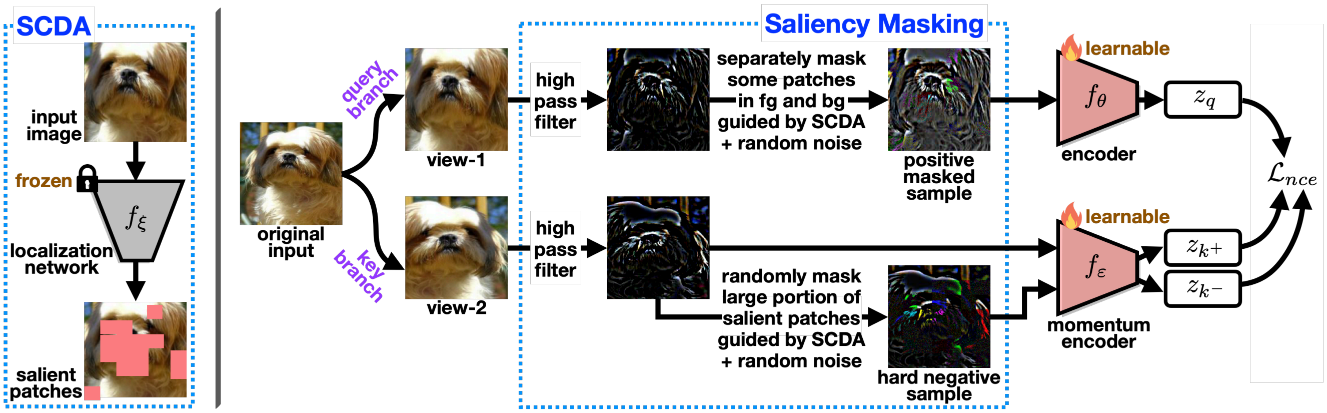

Figure 1. An overview for our proposed method of including saliency-guided masking augmentation into contrastive self-supervised learning, where the backbone of the feature extractor is ConvNets. Firstly, our localization network f_ξ produce the saleincy map which is built upon SCDA [32] (cf. Section 3.1), in which such saliency map helps to separate the foreground objects and background in an image. Given an input image, after conducting standard augmentations along the query and key branches (following the common practice of siamese network) to produce two views, our proposed saliency-guided masking strategies are adopted to produce positive and hard negative samples (please refer to Section 3.2 for more details, where in this figure we take the high-pass filtering strategy as an example). The constructed positive and (hard) negative samples are gone through the feature encoder to compute the contrastive objective function $\mathcal{L}_{nce}$ (please refer to our Section 3.3). Noting that here we base on the SSL framework of MoCov2 [5] to illustrate the computation flow, hence there exists an momentum encoder f_ε in addition to our main learning target, the feature encoder f_θ.

the resultant saliency map M is of size 7×7 and every its element is corresponding to a 32×32 patch in the original image, in which such image patches are the basic units for our performing masking operation later in constrative SSL.

3.2. Saliency-guided masking strategies

As naively masking out image patches in an original image will produce many parasitic edges (i.e. the boundaries between masked and unmasked patches/regions), the ConvNet-based feature encoder could be largely misled to focus on learning these unwanted edge features (since the convolutional kernels are typically good at capturing edges) thus causing problematic model training. In this work we propose to adopt three masking strategies for amending the aforementioned issue stemmed from parasitic edges:

- **High-pass filtering.** Such masking strategy is actually proposed by MSCN [18], where the high-pass filter is applied on the input image prior to the masking operation, in which the edges caused by masking in the filtered map is less visible. It is worth noting that, as the input for the feature encoder under such masking strategy is the high-pass-filtered images, in the downstream tasks the input data should follow the same form (i.e. needed to be firstly high-pass-filtered as well) to achieve better performance, in which this requirement would become a limitation for practical applications (i.e. users for the downstream tasks have to know how the encoder was trained in SSL stage).
- **Strong blurring.** The masking is performed firstly on the original input image, where the masked regions are not filled with the zero value but their own appearance processed by strong Gaussian blurring (i.e. each region to be masked is gone through a low-pass filter), leading to less obvious parasitic edges. Noting that within the patches/regions undergone such masking strategy, the image details are also lost while only the significant contours of objects are preserved. The Gaussian blurring kernel is of size 31×31 with variance set to 10 in our experiments unless otherwise stated.
- **Mean filling.** The masking is also performed on the original input image at first, then the masked regions are filled by the mean pixel value of that input image, such that the boundaries between the masked and unmasked regions becomes much less significant.

As previously discussed, if the masking operation used in these three masking strategies is completely random (i.e. the regions/patches to be masked are sampled randomly), there could exist the potential case where all the masked patches fall on either the foreground or the background objects thus leading to improper contrastiveness (e.g. forming a positive pair where one is fully background while the other still contains most of the foreground). To this end, we propose to use the saliency information (i.e. the saliency map M obtained by our localization network f_ξ stemmed from SCDA technique) to guide the masking operation used in all our three masking strategies. Basically, the saliency map M helps us to separate the foreground objects and the background, in which we perform random masking independently for foreground and background (i.e. distribute the masked patches more evenly to both foreground and background). In the following, we detail how we utilize saliency to create positive and (hard) negative samples for driving the contrastive self-supervised learning objective.

Assume an input image X is composed of N patches and γ denotes the ratio of N patches that are identified as foreground by SCDA (where $N = U \times V$, i.e. the total number of elements in saliency map M and there are $\gamma \cdot N$ patches be identified as foreground). Given a masking ratio α, as we would like to have the masking performed separately for both foreground and background, for a positive sample, the ratio of the number of masked patches between foreground and background is $\gamma : (1-\gamma)$, i.e. there are $\alpha \cdot \gamma \cdot N$ patches randomly chosen to be masked in the foreground (respectively $\alpha \cdot (1-\gamma) \cdot N$ randomly-masked patches in the background). Noting that in our experiments α is drawn from a uniform distribution $\mathcal{U}(0.05, 0.25)$ unless otherwise stated; When it comes to creating *hard negative samples*, the masking is only applied on the foreground patches (i.e. the main objects are mostly masked to remove the salient/important information of such input image). We achieve so by drawing $\beta \sim \mathcal{U}(0.4,\, 0.7)$ in our experiments and randomly masking $\beta \cdot \gamma \cdot N$ foreground patches.

Since the saliency-guided masking operation described above is applied upon the spatial dimension, we also name it as **spatial masking** (to make analogy to the terminology defined in MSCN [18], but please note that ours particularly has the guidance of saliency), and such saliency-guided spatial masking operation is adopted for all our three masking strategies. Moreover, as the masking strategy of high-pass filtering is inspired by MSCN [18], we also include/extend two other masking operations used in MSCN to our high-pass filtering strategy: **channel-wise masking** where our saliency-guided spatial masking operation is applied individually for each of the RGB channels, and **focal masking** where random cropping is performed (noting that such focal masking does not involve any saliency guidance). Specifically, for focal masking, the region outside of 200×200 (respectively region inside 130×130) is cropped and replaced by Guassian noise to produce positive samples (respectively hard negative samples) is our experiments. Furthermore, to be even more aligned with the original masking operation in MSCN [18], in our high-pass filtering strategy we will add random Gaussian noise to the masked sample regardless of which masking operation (i.e. spatial, channel-wise, and focal masking) is adopted (noting that for the strong blurring and mean filling masking strategies we do not apply such step). In Figure 3 we provide examples of positive and hard negative samples from all our three masking strategies.

3.3. Learning scheme

Given the saliency-guided masking strategies introduced above, here we summarize the overall learning scheme for our including masking augmentation into contrastive self-supervised learning framework, where the feature extractor is based on ConvNet-backbone. Following the common scenario of contrastive SSL, two views of an input image X are firstly produced by two different standard augmentations, where one view is denoted as the *key view* while the other is denoted as *query view*. Afterwards, we can apply saliency-guided masking operation (parameterized by γ and α) to the query view, where the resultant masked query view X_q together with the key view X_{k^+} form the positive pair; or, we can apply saliency-guided masking operation (now parameterized by γ and β) upon the key view, which results to be the hard negative sample X_{k^-} to the original key view X_{k^+}. Noting that the masked query view X_q together with the views of any other image different from X (denoted as $X_\neg$) naturally forms the negative pairs. With denoting the feature representation of X_q extracted by the feature encoder as z_q (analogously z_{k^+} for X_{k^+}, z_{k^-} for X_{k^-}, and $z_\neg$ for $X_\neg$), our contrastive objective $\mathcal{L}_{nce}$ is built to pull closer the features of positive pair while pushing away the feature of negative pair:

$$\mathcal{L}_{nce} = -\log \frac{\exp(z_q^\top z_{k^+}/\tau)}{\sum_{X_\neg} \exp(z_q^\top z_\neg/\tau) + \exp(\rho z_q^\top z_{k^-}/\tau)} \tag{2}$$

where τ is a temperature parameter and ρ is the penalty ratio for hard negative samples, in which our $\mathcal{L}_{nce}$ is stemmed from InfoNCE loss [23] and analogous to the one in [12].

It is worth noting that while constructing the positive pairs, the key view X_{k^+} does not undergo any saliency-guided masking operation. Such design is motivated from 1) empirical findings that the masked views show higher variance than the views produced by standard augmentations (in which we provide the corresponding study in Section 4.3), and 2) the statement made in [31] that having higher variance in the query branch than the key branch yields better results in the siamese network, where the benefit of such design to model training is demonstrated in our experiments provided in Table 6.

4. Experimental Results

We compare our proposed method with two state-of-the-art baselines including masking operations into the ConvNet-based SSL, i.e. MSCN [18] and ADIOS [30], as illustrated in Figure 2. Following ADIOS [30], we adopt the ImageNet-100 dataset [29] as the basis to conduct our experiments (i.e., being used to perform contrastive self-supervised learning for training the feature encoder), while we choose MoCov2 [5] and SimCLR [4] to be our experimental bed of contrastive SSL frameworks. Basically, ImageNet-100 dataset contains 100 ImageNet classes, and is composed of 1300 training images and 50 validation images for each class. We use ResNet-50 as our ConvNet-backbone for the feature encoder for both contrastive SSL frameworks. For MoCov2, we set the batch size to 128 and the learning rate to 0.015, and use SGD [28] as the optimizer; While for SimCLR, we set the batch size to 256

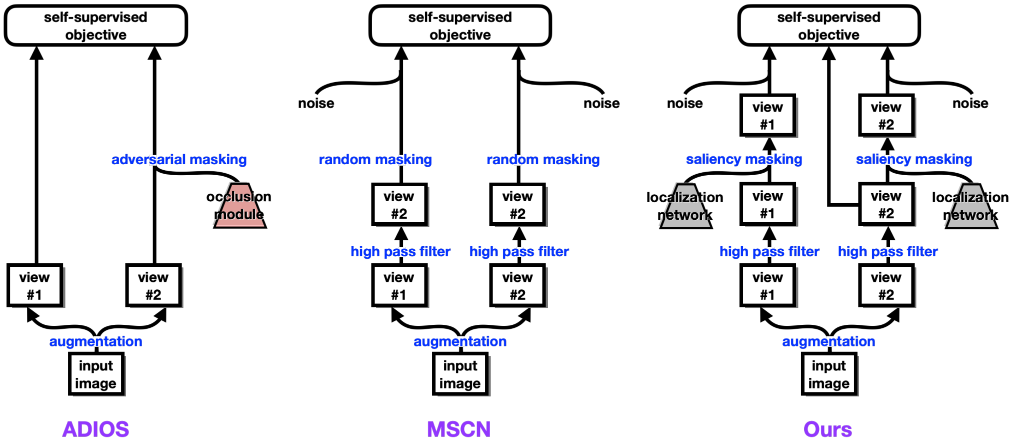

Figure 2. Comparison in terms of modelling among MSCN [18], ADIOS [30], and our proposed method. Please refer to the last paragraph of Section 2 for more detailed descriptions.

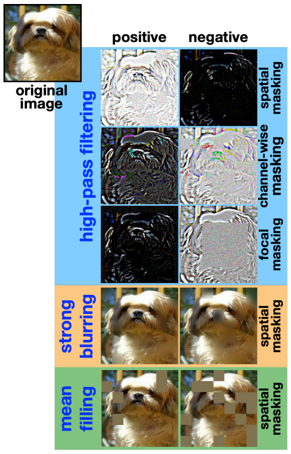

Figure 3. Three masking strategies (cf. Section 3.2).

Method	Linear Evaluation
MoCov2 [5]	68.22
+ MSCN [18]	70.28
+ ADIOS [30]	62.76
+ OURS (High-pass filtering)	**73.8**
+ OURS (Strong blurring)	72.50
+ OURS (Mean filling)	70.84
SimCLR [4]	69.77
+ MSCN [18]	77.18
+ ADIOS [30]	71.12
+ OURS (High-pass filtering)	**77.9**
+ OURS (Strong blurring)	77.78
+ OURS (Mean filling)	77.36

Table 1. Linear evaluation results on ImageNet-100 classification task, where MoCov2 or SimCLR are used as the contrastive SSL framework for pretraining the feature encoder. The best results are marked in bold.

and the learning rate to 0.3, and use LARS [34] as the optimizer. The contrastive self-supervised pretraining is run on a 4-GPU machine for 200 epochs, including 10 epochs of warm-up and using a cosine learning rate scheduler.

4.1. ImageNet-100 Classification

Once the feature encoders are pretrained via adopting our proposed method (with three different masking strategies) and baselines (i.e. MSCN and ADIOS) in the contrastive SSL frameworks (i.e. MoCov2 and SimCLR), we now evaluate their performances on various downstream tasks. Firstly, we experiment on the downstream task of classification based on ImageNet-100 dataset, where a linear classifier is supervisedly trained while the feature encoder is kept fixed/frozen. Please note again that, as described in Section 3.2, for the feature encoder pretrained by using the high-pass filtering masking strategy, its input in the downstream task should still be firstly gone through the high-pass filter (in which such requirement is also applied to MSCN). The evaluation results on ImageNet-100 classification are summarized in Table 1, where our proposed method (regardless of adopting any saliency-guided masking strategies in both SSL frameworks) consistently achieves superior performance comparing to MoCov2 baseline (i.e. no masking augmentation is applied), MoCov2+MSCN, and MoCov2+ADIOS, where the similar trend is also observable while using SimCLR as the contrastive SSL framework, thus verifying the contribution and the efficacy of our proposed saliency-guided masking methods. It is worth noting that, although our high-pass filtering masking strategies shares quite some common designs as MSCN, our explicit introduction of saliency guidance contributes to the resultant improvement of our proposed method, e.g. MoCov2+OURS (High-pass filtering) versus MoCov2+MSCN and SimCLR+OURS (High-pass filtering) versus SimCLR+MSCN in Table 1.

4.2. Transfer Learning

As one of the important goals of SSL is to obtain the feature encoder with better generalizability such that the encoder can be easily adapted to various tasks or datasets with little amount of labeled data, we thus further conduct experiments on different downstream tasks or datasets for better assessing the generality of the features learned by various methods. Here we take MoCov2 as the main experimental bed of SSL framework, and we adopt high-pass filtering masking strategy to present our proposed method for making comparison with the baselines (while the results of adopting strong blurring and mean filling masking strategies in our proposed method are provided in the Appendix).

Image classification on different datasets. Here we experiment on the classification downstream task based on two widely used benchmarks, i.e. Caltech-101 [10] and Flowers-102 [21], which are different from the one used for feature encoder pretraining (i.e. ImageNet-100). Again, we keep the pretrained feature encoder fixed and only train the

Method	VOC07+12 detection			COCO detection			COCO instance segmentation		
	AP_{all}	AP_{50}	AP_{75}	AP^{bb}_{all}	AP^{bb}_{50}	$AP^{bb}S_{75}$	AP^{mk}_{all}	AP^{mk}_{50}	AP^{mk}_{75}
MoCov2	50.27	76.68	54.76	38.52	57.62	41.67	33.75	54.70	35.86
+ MSCN	50.27	76.99	54.70	38.80	58.09	42.20	33.89	54.78	**36.36**
+ ADIOS	45.85	73.44	48.45	38.12	57.38	41.29	33.38	54.25	35.63
+ OURS (High-pass filtering)	**50.89**	**77.66**	**55.44**	**39.16**	**58.62**	**42.45**	**34.22**	**55.28**	36.30

Table 2. Transfer learning results on VOC07+12 and COCO detection tasks, and COCO instance segmentation task. Performances in terms of AP_{all}, AP_{50} and AP_{75} metrics are reported, and the best results are marked in bold.

Method	Caltech-101	Flowers-102
MoCov2	81.87	88.39
+ MSCN [18]	84.13	90.10
+ ADIOS [30]	79.83	88.39
+ OURS (High-pass filtering)	**84.91**	**90.95**

Table 3. Transfer learning results on Caltech-101 and Flowers-102 classification tasks.

Setting	Saliency	Mask non-salient	Top1
High-pass filtering	✗	✓	56.15
	✓	✓	**56.60**
	✓	✗	55.87
Strong blurring	✗	✓	55.33
	✓	✓	**56.39**
	✓	✗	54.37
Mean filling	✗	✓	55.93
	✓	✓	**55.94**
	✓	✗	55.59

Table 4. We conduct three different experiments on TinyImageNet classification task to justify the importance of saliency guidance.

linear classifier when learning the downstream task. The experimental results are reported in Table 3, where our proposed method outperforms both MSCN and ADIOS baselines on both Caltech-101 and Flowers-102 datasets.

Object detection & instance segmentation. Now we turn to different downstream tasks on object detection and instance segmentation, where the former is conducted on VOC07+12 [9] and COCO [20] datasets while the latter is conducted on the COCO dataset. With keeping the feature encoder fixed and only supervisedly training the detection or segmentation heads (where for VOC07+12 dataset we adopt the Faster-RCNN [27] model with a C4 backbone which finetuned for 24k iterations, while for COCO dataset we adopt Mask R-CNN [17] model with C4 backbone which is finetuned for 180K iterations, following the same experimental setting as in the original MoCov2 paper), the experimental results are summarized in Table 2. From the results we can again observe the consistent outperformance compared to both MSCN and ADIOS baselines.

The transfer learning results provided in Table 3 and Table 2 support that our method can effectively learn general-purpose features which can be transferred across different downstream tasks or datasets, providing a promising finding for future research in the field of self-supervised learning.

4.3. Ablation Study

Our ablation studies are carried out on TinyImageNet [19], a well-known subset of the ImageNet-1K dataset [29]. This dataset consists of 200 classes, comprising 500 training images and 50 validation images in each class. Compared to ImageNet-100, TinyImageNet contains more classes but fewer training images per class, which is even more challenging for the model to learn. As a result, we take it to better verify the contribution for each of our designs. We employ MoCov2 as our SSL framework with using ResNet-50 as the backbone for feature encoder, which is pretrained for 200 epochs. The evaluation is conducted by the downstream task of classification on TinyImageNet. If not mentioned, the positive masking ratio is set to 15% of the whole image, and the hard negative masking ratio is set to 40%-70% salient patches.

Impact of Saliency. Saliency is the soul of our work and has always been mentioned in this paper. We compare three settings: pure random masking (i.e. MSCN), masking with saliency constraint, and masking on salient patches only, and the results are reported in Table 4. Note that in this experiment, our hard negative samples are not applied. Our findings indicate that adding a saliency constraint can benefit our training, but only if we distribute the masked patches to the foreground and background patches. Surprisingly, masking totally on salient patches results in poor performance, possibly due to the reason that such processing causes the model to rely more heavily on the background, which can be detrimental to the overall performance.

Masking Controls Variance. Our approach involves masking only the query branch when constructing positive pairs. To investigate the effects of masking different branches on performance, we conduct a study consisting of three experiments, which are masking the key branch only, masking both branches, and masking the query branch only. The results of this study, as presented in Table 6, indicate that masking the query branch only performs the best, and masking the key branch only performs worse than the baseline MoCov2. Such results are aligned with [31], which suggests that maintaining a lower variance in the key branch than in the query branch during pretraining can be beneficial, and not the other way around. Moreover, we hypothesize that masking can also influence variance such that manipulating it through masking can lead to better results. To

Pretrained dataset	ImageNet-100	Caltech-101	Flowers-102	VOC07+12 det		
		Top1		AP_{all}	AP_{50}	AP_{75}
ImageNet-1K	73.80	84.91	90.95	50.89	77.66	55.44
COCO	73.78	85.68	90.83	50.22	77.41	54.28

Table 5. We compare the results of using localization networks pretrained on the ImageNet-1K and COCO dataset, where the localization network is to produce the saliency maps for guiding our masking operations. Similar performances produced by using these two localization networks demonstrate that our proposed method is insensitive to the selection of localization network for performing saliency computation, once it provides feasible localization results.

Setting	Mask branch	Top1
Baseline MoCov2	✗	56.00
High-pass filtering	key	52.25
	both	56.29
	query	**58.19**
Strong blurring	key	51.06
	both	56.83
	query	**58.28**
Mean filling	key	47.53
	both	56.86
	query	**58.34**

Table 6. Comparison of masking on different branches of SSL framework.

Augmentation	Variance (1e-3)
Standard	6.196
+ High-pass filtering masking	10.952
High-pass filtering w/o masking	8.67
+ Strong blurring masking	7.744
+ Mean filling masking	7.776

Table 7. Variance of the representations b/w standard augmentation and our saliency masking (refer to [31] for variance calculation, based on TinyImageNet validation set).

Setting	Positive mask	Negative mask	Top1
High-pass filtering	✓	✗	55.25
	✓	✓	**56.26**
Strong blurring	✓	✗	55.52
	✓	✓	**56.83**
Mean filling	✓	✗	55.18
	✓	✓	**56.21**

Table 8. Efficacy of our proposed hard negative samples.

further support our claim, we conduct an experiment comparing the variance of standard data augmentation with standard data augmentation combined with our saliency masking for all of our settings, where the results are presented in Table 7. We can observe that standard data augmentation combined with our saliency masking leads to a higher variance than standard data augmentation for all our settings.

Impact of Hard Negative Samples. We conduct a study to verify our designs of creating hard negative samples by masking large portion of salient patches (40%-70%) of the key view. Table 8 compares the results of being with or without our proposed hard negative samples, showing the benefit brought by our designs. By masking more salient patches, the remaining part of image has more background than foreground. We deem this view as negative one, though it comes from the same sample image as the positive view. The model might be confused if it is biased to the background when the similarity between the hard negative view and the query view is too high. In turn, the model with our hard negative samples will focus on learning more from the foreground thus boosting the performance.

Impact of Localization Networks Pretrained on Different Datasets. As we adopt ImageNet-pretained classfication model as the basis for SCDA to build our localization network (for producing saliency maps), there could exist potential concern if we take any advantage later in the SSL pretraining stage than other baselines. To resolve such potential concern, here we conduct a study to use another model, the ResNet-50 backbone from a Faster R-CNN detection model pretrained on the COCO dataset, as the basis for SCDA. With using our high-pass filtering masking strategy guided by the saliency maps respectively from different localization networks in MoCov2 SSL framework to training the feature encoders, the experimental results upon various downstream tasks and datasets (following the same settings as previous experiments) are summarized in Table 5. We are able to observe that both localization networks result in similar performances, thus verifying that our method is not sensitive to the selection of localization network once it is able to provide reasonable localization capability.

5. Conclusion

We propose a salient masking augmentation method for contrastive self-supervised learning with a ConvNet as its backbone. Compared to randomly masking patches of the input image, our salient masking provides more semantically meaningful masks while its efficacy is well verified in our ablation study. Besides masked positive samples, we further introduce a simple way to create hard negative samples according to three different masking strategies, which further improve the capability of training the feature encoder. The extensive experimental results demonstrate the effectiveness and superiority of our proposed method.

Acknowledgement. This work is supported by NSTC (National Science and Technology Council) 111-2628-E-A49-018-MY4 and 112-2221-E-A49-087-MY3, Higher Education Sprout Project of the National Yang Ming Chiao Tung University, as well as Ministry of Education, Taiwan.

References

[1] Hangbo Bao, Li Dong, and Furu Wei. Beit: Bert pre-training of image transformers. In *ICLR*, 2022. 1

[2] Mathilde Caron, Piotr Bojanowski, Armand Joulin, and Matthijs Douze. Deep clustering for unsupervised learning of visual features. In *ECCV*, 2018. 3

[3] Mathilde Caron, Ishan Misra, Julien Mairal, Priya Goyal, Piotr Bojanowski, and Armand Joulin. Unsupervised learning of visual features by contrasting cluster assignments. In *NeurIPS*, 2020. 2, 3

[4] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. In *ICML*, 2020. 2, 5, 6

[5] Xinlei Chen, Haoqi Fan, Ross Girshick, and Kaiming He. Improved baselines with momentum contrastive learning. *arXiv preprint arXiv:2003.04297*, 2020. 4, 5, 6

[6] Xinlei Chen and Kaiming He. Exploring simple siamese representation learning. In *CVPR*, 2021. 3

[7] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of Annual Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT)*, 2019. 1, 3

[8] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. *ArXiv:2010.11929*, 2020. 1

[9] Mark Everingham, Luc Van Gool, Christopher KI Williams, John Winn, and Andrew Zisserman. The pascal visual object classes (voc) challenge. *IJCV*, 2010. 7

[10] Li Fei-Fei, Rob Fergus, and Pietro Perona. Learning generative visual models from few training examples: An incremental bayesian approach tested on 101 object categories. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, 2004. 6

[11] Zeyu Feng, Chang Xu, and Dacheng Tao. Self-supervised representation learning by rotation feature decoupling. In *CVPR*, 2019. 2

[12] Songwei Ge, Shlok Mishra, Chun-Liang Li, Haohan Wang, and David Jacobs. Robust contrastive learning using negative samples with diminished semantics. In *NeurIPS*, 2021. 2, 5

[13] Spyros Gidaris, Praveer Singh, and Nikos Komodakis. Unsupervised representation learning by predicting image rotations. In *ICLR*, 2018. 2

[14] Jean-Bastien Grill, Florian Strub, Florent Altché, Corentin Tallec, Pierre Richemond, Elena Buchatskaya, Carl Doersch, Bernardo Avila Pires, Zhaohan Guo, Mohammad Gheshlaghi Azar, et al. Bootstrap your own latent-a new approach to self-supervised learning. In *NeurIPS*, 2020. 2, 3

[15] Kaiming He, Xinlei Chen, Saining Xie, Yanghao Li, Piotr Dollár, and Ross Girshick. Masked autoencoders are scalable vision learners. In *CVPR*, 2022. 1, 2, 3

[16] Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross Girshick. Momentum contrast for unsupervised visual representation learning. In *CVPR*, 2020. 2

[17] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask r-cnn. In *Proceedings of the IEEE international conference on computer vision*, pages 2961–2969, 2017. 7

[18] Li Jing, Jiachen Zhu, and Yann LeCun. Masked siamese convnets. *ArXiv:2206.07700*, 2022. 2, 3, 4, 5, 6, 7

[19] Ya Le and Xuan S. Yang. Tiny imagenet visual recognition challenge, 2015. 7

[20] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *Computer Vision–ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part V 13*, pages 740–755. Springer, 2014. 7

[21] Maria-Elena Nilsback and Andrew Zisserman. Automated flower classification over a large number of classes. In *Sixth Indian Conference on Computer Vision, Graphics & Image Processing (ICVGIP)*, 2008. 6

[22] Mehdi Noroozi and Paolo Favaro. Unsupervised learning of visual representations by solving jigsaw puzzles. In *ECCV*, 2016. 2

[23] Aaron van den Oord, Yazhe Li, and Oriol Vinyals. Representation learning with contrastive predictive coding. In *NeurIPS*, 2019. 5

[24] Deepak Pathak, Philipp Krahenbuhl, Jeff Donahue, Trevor Darrell, and Alexei A Efros. Context encoders: Feature learning by inpainting. In *CVPR*, 2016. 1

[25] Xiangyu Peng, Kai Wang, Zheng Zhu, Mang Wang, and Yang You. Crafting better contrastive views for siamese representation learning. In *CVPR*, 2022. 2

[26] Alec Radford, Karthik Narasimhan, Tim Salimans, Ilya Sutskever, et al. Improving language understanding by generative pre-training. *OpenAI*, 2018. 1

[27] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. *Advances in neural information processing systems*, 28, 2015. 7

[28] Sebastian Ruder. An overview of gradient descent optimization algorithms. *arXiv preprint arXiv:1609.04747*, 2016. 5

[29] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, Alexander C. Berg, and Li Fei-Fei. ImageNet Large Scale Visual Recognition Challenge. *IJCV*, 2015. 5, 7

[30] Yuge Shi, N Siddharth, Philip Torr, and Adam R Kosiorek. Adversarial masking for self-supervised learning. In *ICML*, 2022. 2, 3, 5, 6, 7

[31] Xiao Wang, Haoqi Fan, Yuandong Tian, Daisuke Kihara, and Xinlei Chen. On the importance of asymmetry for siamese representation learning. In *CVPR*, 2022. 2, 5, 7, 8

[32] Xiu-Shen Wei, Jian-Hao Luo, Jianxin Wu, and Zhi-Hua Zhou. Selective convolutional descriptor aggregation for fine-grained image retrieval. *IEEE TIP*, 2017. 3, 4

[33] Zhenda Xie, Zheng Zhang, Yue Cao, Yutong Lin, Jianmin Bao, Zhuliang Yao, Qi Dai, and Han Hu. Simmim: A simple

framework for masked image modeling. In *CVPR*, 2022. 1, 3

[34] Yang You, Igor Gitman, and Boris Ginsburg. Large batch training of convolutional networks. *arXiv preprint arXiv:1708.03888*, 2017. 6

[35] Jure Zbontar, Li Jing, Ishan Misra, Yann LeCun, and Stéphane Deny. Barlow twins: Self-supervised learning via redundancy reduction. In *ICML*, 2021. 2

[36] Richard Zhang, Phillip Isola, and Alexei A Efros. Colorful image colorization. In *ECCV*, 2016. 2

[37] Jinghao Zhou, Chen Wei, Huiyu Wang, Wei Shen, Cihang Xie, Alan Yuille, and Tao Kong. ibot: Image bert pre-training with online tokenizer. In *ICLR*, 2022. 1