Article

Multi-view Masked Autoencoder for General Image Representation

Seungbin Ji , Sangkwon Han , Jongtae Rhee *

Posted Date: 9 October 2023

doi: 10.20944/preprints202310.0524.v1

Keywords: Deep learning; image representation learning; self-supervised learning; masked image modeling; contrastive learning

Article

Multi-View Masked Autoencoder for General Image Representation

Seungbin Ji , Sangkwon Han and Jongtae Rhee *

Department of Industrial and Systems Engineering, Dongguk University, Seoul 04620, Korea; voiagerd@dgu.ac.kr (S.J.); hsk0314@dgu.ac.kr (S.H.)

* Correspondence: jtrhee@dongguk.edu

Abstract: Self-supervised learning is a method that learns general representation from unlabeled data. Masked image modeling (MIM), one of the generative self-supervised learning methods, has drawn attention showing state-of-the-art performance on various downstream tasks, though showing poor linear separability resulting from the token-level approach. In this paper, we propose a contrastive learning-based multi-view masked autoencoder for MIM, exploiting an image-level approach by learning common features from two different augmented views. We strengthen MIM by learning long-range global patterns from contrastive loss. Our framework adopts simple encoder-decoder architecture, learning rich and general representation by following a simple process: 1) two different views are generated from an input image with random masking and by contrastive loss, we can learn semantic distance of the representations generated by an encoder. By applying a high mask ratio, 80%, it works as strong augmentation and alleviates the representation collapse problem. 2) With reconstruction loss, decoder learns to reconstruct an original image from the masked image. We assess our framework by several experiments on benchmark datasets of image classification, object detection, and semantic segmentation. We achieve 84.3% fine-tuning accuracy on ImageNet-1K classification and 76.7% in linear probing, exceeding previous studies and show promising results on other downstream tasks. Experimental results demonstrate that our work can learn rich and general image representation by applying contrastive loss to masked image modeling.

Keywords: deep learning; image representation learning; self-supervised learning; masked image modeling; contrastive learning

1. Introduction

Deep learning, which has revolutionized over the past decade, has recently faced data-hungry problem due to the rapid growth of hardware and resources [1–3]. Self-supervised learning, which learns meaningful data representations from unlabeled data [4], has emerged as an alternative to supervised learning resulting from the inefficiency of labeling in terms of time and labor [5–7].

Masked autoencoding [8] is a method that learns representations by removing part of the input and predicting the masked part. Autoencoder [9,10] architecture is used for masked autoencoding, compressing high-dimensional data into latent representation with encoder and reconstruct the original data with decoder, as shown in Figure 1. It has been successful in NLP as a method of self-supervised pre-training. The approach of learning representation by reconstructing images from corrupted images is not new; the idea was already proposed before 2017 [11,12]. The idea was buried after the emergence of contrastive learning since it has shown promising results on downstream tasks [13–15]. Witnessing success of masked autoencoding in NLP fields [16–18], many works tried to apply masked autoencoding to vision, but lag behind due to the following reasons: 1) in vision, convolutional network architecture was dominant [19], where indicators like mask token [17] or positional embedding [20] are inapplicable. 2) With only a few neighboring pixels, missing parts of an image can be successfully predicted without deep understanding of an image [21]. However, when predicting a missing part/token, complex language understanding should be investigated. In other words, the masked autoencoding in vision field might not demand fully understanding of image which results in capturing less useful features. Due to these differences between the two modalities, masked autoencoding was limitedly applied in the vision field until Vision Transformer (ViT) [22].

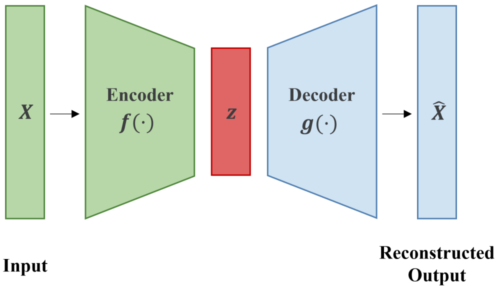

Figure 1. Overview of autoencoder architecture. Given input, encoder compresses the input into low-dimensional latent representation and reconstruct the original data with decoder. Autoencoder aims to make input X and reconstructed output $\hat{X}$ similar.

Motivated by the success of masked language modeling (MLM) in language understanding, masked image modeling (MIM), following the idea of MLM, learns rich and holistic representation by reconstructing masked original information (e.g., pixel, representation) from unmasked ones. MIM has gained much importance recently showing state-of-the-art performance [2,23,24] not only in ImageNet classification but alss in other downstream tasks like object detection and semantic segmentation.

Before MIM, contrastive learning (CL), which learns meaningful representation by using similarities and differences between image representations, was a dominant method in self-supervised learning [4]. By learning embedding space, in a way that contrasts each other so that positive samples are located close and negative samples to be far away, CL learns to discriminate instances using features of the entire image [25]. Contrary to CL, MIM does not learn instance discriminativeness since it only considers relationships between patches or pixels through the image reconstruction task [26]. Therefore, although MIM methods exceed the performance of CL methods in fine-tuning, they are shown to be less effective in linear separability.

In this work, we propose a simple yet effective framework, adopting multi-view autoencoder architecture, utilizing contrastive learning to MIM to overcome the gap between CL and MIM. CL performs better on linear probing, while MIM shows better performance in fine-tuning setting. We note that contrastive learning-based MIM method can learn common information from two different augmented views, away from the existing pixel-level approaches that learn local representation of images. We demonstrate that local representation considering instance discriminative information can be learnt by doing so.

In more detail, we adopt asymmetric encoder-decoder architecture using ViT [22] blocks. ViT makes the model to focus on important feature of an instance. We visualized maps of the attention of our pre-trained ViT encoder as shown in Figure 2, taking the average of ViT heads following [27]. CL is used to capture global information and learn discriminative representation by contrasting negative samples while pulling positive samples. By generating two augmented views by masking, with encoder, we compress them into latent representations, which are used for contrastive loss. While learning holistic information from contrastive loss, reconstruction loss helps the decoder to learn local representation by predicting patches from the masked image.

doi:10.20944/preprints202310.0524.v1

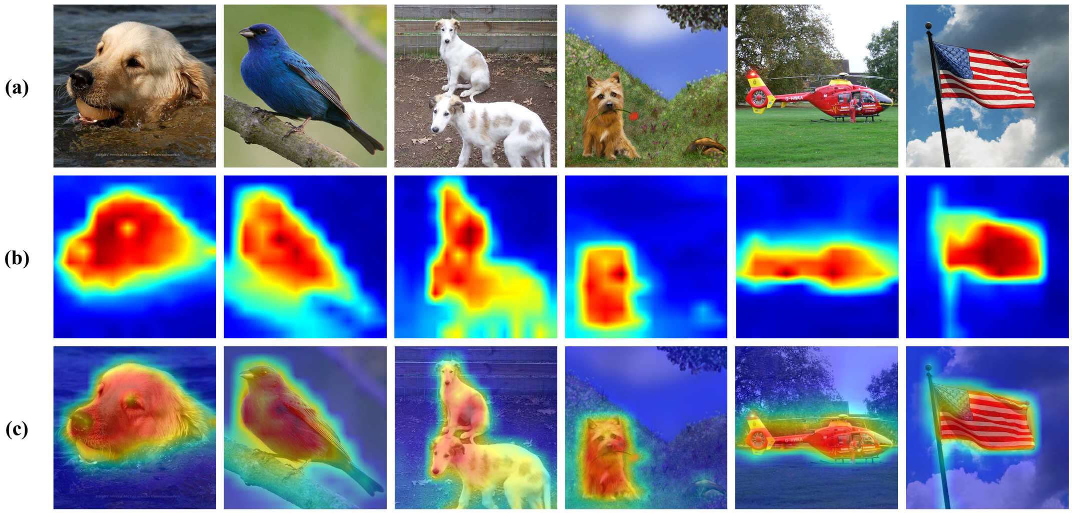

Figure 2. Visualization of attention heatmap using pre-trained ViT encoder of the proposed method. **(a)** is original images, **(b)** is heatmaps of each original image and **(c)** is heatmaps added to original images.

We conducted experiments to prove our work to be effective. Our work enables ViT encoder to exceed previous work, showing 84.3% ImageNet-1K classification top-1 accuracy. Though showing lower performance, but comparable, compared to other CL-based methods in linear probing, our work shows impressive performance gain compared to MIM methods achieving 76.7% accuracy. We also evaluate on transfer learning on object detection and segmentation. We record 51.3% AP^{box} and 45.6% AP^{mask} on COCO and 50.2% mIOU on ADE20K showing the best or second best performance compared to previous studies. Through ablation studies, we demonstrate that utilizing CL to MIM helps the model learn better representation.

Our contributions are summarized as follows:

- We propose a simple framework exploiting contrastive learning to MIM to learn rich and holistic representation. The model learns discriminative representation by contrasting two augmented views, while reconstructing original signals from the corrupted ones.
- A high masking ratio works as strong augmentation. Without additional augmentation like color distortion, blur, etc., our model shows better performance than previous CL-based methods by only using masking and random crop.
- Experimental results prove that our work is effective, outperforming previous MIM methods in ImageNet-1K classification, linear probing and other downstream tasks like object detection and instance segmentation.

The rest of this paper is structured as follows. Section 2 introduces related works. In Section 3, we give an overview and details of our framework. Then we show experimental results and analysis in Section 4. Finally, Section 5 concludes.

2. Related Work

2.1. Contrastive Learning

Contrastive learning [13–15,28–30] is a method of learning instance discriminative features by contrasting samples against each other to learn common features between data, which is categorized as discriminative self-supervised learning. CL has been a dominant self-supervised learning method [26] since it showed overwhelming performance over supervised learning. CL relies on negative samples and strong data augmentations to avoid the representation collapse problem, outputting constant given

doi:10.20944/preprints202310.0524.v1

different inputs. Previous studies have investigated the use of memory banks [15], and large batch size [13] for better informative negative samples. Recent works have shown that without discriminating between images, we can learn features by only using positive samples. BYOL [29] and SimSiam [31] use only positive samples in a different way; BYOL uses a momentum encoder while SimSiam uses a stop-gradient. Recent studies [14,32] exploit the use of ViT architecture that stands out compared to convolutional neural networks. DINO [14] discovers that ViT features contain explicit information about the semantic segmentation of an image and outperforming previous self-supervised methods.

2.2. Masked Language Modeling and Masked Image Modeling

Masked language modeling (MLM) [16–18] is one of the most used approaches for pre-training and shows promising results on various downstream tasks in NLP. BERT [17] makes the model learn context understanding of language by masking some parts, using special mask tokens, and predicting the missing parts.

In early works of Masked Image Modeling (MIM), denoising autoencoder [11,12] was introduced to restore from blurred, masked pixels to original clean pixels. MIM studies [2,33] was inspired by successful context understating of masking parts in MLM tasks have been introduced. In [33], sequences of pixels were used to predict unknown pixels. BEiT [2] and MAE [23] are a foundation model of MIM showing promising results on several downstream tasks. BEiT takes BERT-style pre-training by reconstructing visual tokens using pre-trained dicrete VAE [34] as a tokenizer while MAE predicts pixels directly using ViT [22]. Also, in [35] improved segmentation performance through pre-training to predict pixels from masked pixels. Recent works explore pixel or feature regression, only in a relatively small model. [36] proposes a model employing an enhancer network to either recover original image pixels or predict whether each visual token is replaced by a generator sample or not.

3. Method

3.1. Framework

The overall framework is shown in Figure 3, adopting autoencoder architecture. We propose contrastive learning to learn representation using multi-view of an image, using contrastive learning on latent representation of shared encoder. We generate multi-view of an image by simple augmentation with random masking. With encoder, we compress high-dimensional image data into low dimensional latent representation and reconstruct the original image given latent representation with decoder. In detail, firstly, with random masking, two different masked images are generated from an input image. Encoder, consisting of ViT layers, takes two masked images as input, compressing them into latent representations, which are used for contrastive learning. Afterwards, decoder reconstructs the original image from latent representations with mask tokens. The training process is specified below.

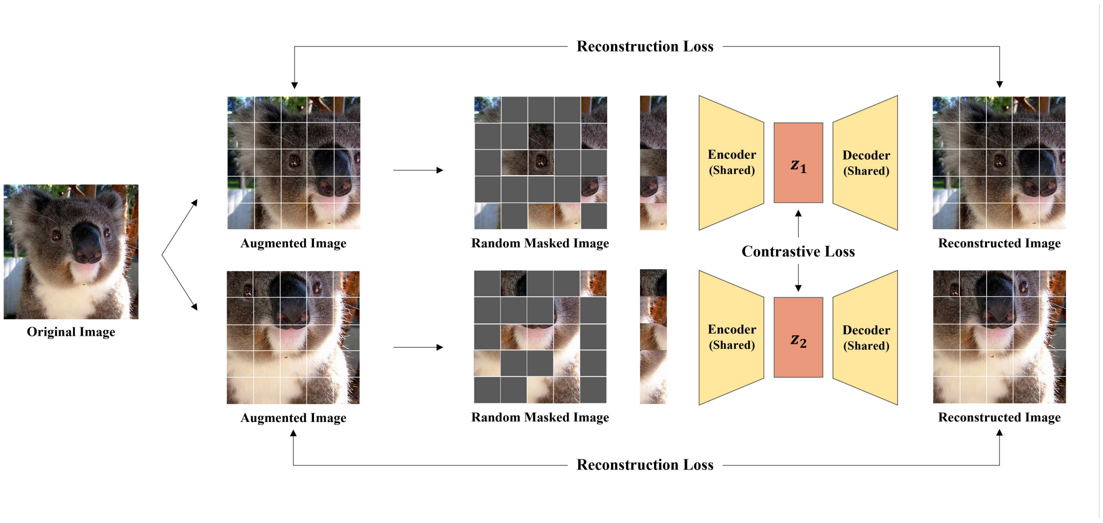

Figure 3. Overall architecture. Original image I_i is converted into randomly masked images x'_1 and x'_2 after augmentation process. The encoder compresses them into z_1 and z_2, used for contrastive learning. Given z_1 and z_2, the decoder predicts the masked parts, outputting reconstructed images y_1 and y_2.

3.1.1. Input and Target Views

We randomly sample N images in every iteration when pre-training. To create target views, denoted as x_i^+, we apply simple data augmentation, random resized crop, and horizontal flip. Also, we exploit applying augmentation two times, creating two different target views for effective use of contrastive learning as shown in Figure 4. Two different target views make the model see an input image from different point of view.

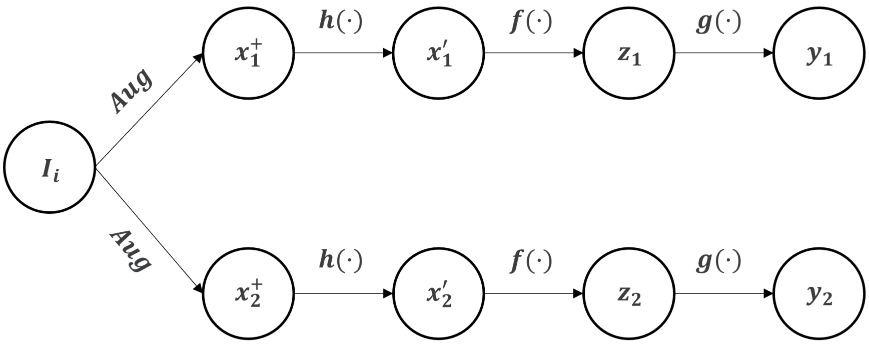

Figure 4. A framework utilizing contrastive learning to masked image modeling. We firstly generate two different target views x_1^+ and x_2^+ with simple augmentation. Given x'_1 and x'_2, generated by random masking operation $h(\cdot)$, our encoder $f(\cdot)$ converts patch sequence into latent representation z_1 and z_2. Finally, our decoder outputs y_1 and y_2, reconstructed images.

3.1.2. Patchify and Masking Strategy

Since we adopt ViT for the encoder, we patchify the target views into a non-overlapping 14×14 patch sequence. To retain spatial information about where the patch is located, we add positional embedding to them. For positional encoding, we used sine-cosine positional encoding. In addition, for augmented view, we apply random masking to the patches. Masking is simple; we generate numbers following a uniform distribution. Afterwards, we 'mask' a specific ratio of the total number of patches in the embedded patch sequence using the generated random numbers, i.e., random masking. The

doi:10.20944/preprints202310.0524.v1

randomly masked patch sequences can be formulated as $x'_i = h(x_i^+), i \in \{1, 2\}$, where h denotes patchifying and masking operation.

Conventionally, masked language models mask relatively low portion of tokens because more masking would result in insufficient context to learn good representation [37]. However, because image pixels are continuous contrary to discrete language tokens, higher masking ratio should be applied to eliminate redundancy in image. We choose certain masking ratio through experiments, see section 4.2.1.

3.1.3. Encoder

Our encoder $f(\cdot)$ adopts ViT architecture, specifically ViT base with patch size 16. Each masked image x'_1 and x'_2 can be decomposed into visible patches and masked patches. They can be formulated as follows:

$$x'_i \to x_i^v, x_i^m \tag{1}$$

$$z_i = f(x_i^v + x_{pos}) = ViT(x_i^v + x_{pos}) \tag{2}$$

where x_i^v, x_i^m, and x_{pos} are visible patches, masked patches, and positional embeddings, respectively. For encoder input, as represented in equation 2, only visible patches and positional encoding are passed through to generate latent representation denoted as z_1, and z_2, excluding masked patches in line with MAE which allows computing efficiency.

The encoder is learned to compress high dimensional vector retaining important information representing the given input data. We use latent representation, the output of the encoder, for contrastive learning by pulling the positive pair close and pushing the negative pairs away. Using two different target views, mentioned in Section 3.1.1, strengthens the encoder to better learn instance discriminativeness by using different point of view.

3.1.4. Decoder

To perform the reconstruction task, the decoder, $g(\cdot)$, reconstructs images from given input z_1, and z_2. Our decoder also adopts ViT but is lightweight compared to the encoder. Given z_1, and z_2 as input, mask tokens are added since our decoder computes over full patches. Also, we add positional embeddings to them. By doing so, mask tokens do know where they should be located. Following the setting of MAE, we also adopt an asymmetric encoder-decoder design, having a shallow depth of decoder. We ablated experiments on the depth of the decoder, see Table 5. As our goal is to learn image representation, not reconstructing corrupted images, the decoder is only used in pre-training.

3.2. Training Objectives

For the training objective, we use two objectives; reconstruction loss and contrastive loss. Both loss functions are specified below.

3.2.1. Reconstruction loss

We use reconstruction loss, mean squared error (MSE), as one of our training objectives, which is generally used in MIM. The model performs a pretext task to reconstruct the original image from corrupted (here we say masked) ones. Given y_1, and y_2, prediction from the model, reconstruction loss computes over patchified target image x_i^+, which is formulated as follows:

$$\mathcal{L}_r = \frac{1}{2N} \sum_{j=1}^{N} \sum_{i=1}^{2} (y_i - x_i^+)^2 \tag{3}$$

where N is batch size. We divide MSE over twice batch size because two different views are generated from one image. This loss helps the model to learn local representation of images since it uses neighboring patches to predict the masked ones.

3.2.2. Contrastive loss

For contrastive loss, we use NT-Xent (the normalized temperature-scaled cross entropy loss) proposed in [13]. This loss operates cosine similarity between given pairs, computing mutual information between them. In a minibatch of N samples, images augmented from the same image is regarded as a positive pair and the rest samples, $2(N-1)$ are treated as negative samples. Contrastive loss is defined as:

$$\ell(i,j) = -log \frac{exp(sim(z_i,z_j)/\tau}{\sum_{k=1}^{2N}(\mathbb{1}_{[k\neq i]}sim(z_i,z_k)/\tau)} \tag{4}$$

$$sim(i,j) = z_i^\top z_j/(\|z_i\|\|z_j\|) \tag{5}$$

$$\mathcal{L}_c = \frac{1}{2N}\sum_{k=1}^{N}\{\ell(2k-1,2k)+\ell(2k,2k-1)\} \tag{6}$$

where $\mathbb{1}_{[k\neq i]} \in \{0,1\}$ is an indicator representing 1 iff $k \neq i$ and τ denotes temperature constant. τ is set to 0.07, following the default setting of [13]. The denominator of $\ell(i,j)$ computes similarity over a positive pair and the final contrastive loss function, $\mathcal{L}_c$, is computed across all positive pairs. By doing so, different views from the same image, which we call positive samples, are pulled together while pushing away negative samples in embedding space.

The total loss is a weighted sum of reconstruction loss L_r and contrastive loss L_c as formulated as follows:

$$\mathcal{L} = \mathcal{L}_r + \lambda\mathcal{L}_c \tag{7}$$

where λ is a hyperparameter, deciding loss weight. Contrasting multi-view of an image makes the model learn general representation under high masking ratio, while the model performs reconstruction task with reconstruction loss.

4. Experiments

4.1. Implementation details

We pre-trained our model at 224×224 resolution on ImageNet-1K [38] training set without labels. ImageNet-1K is a benchmark dataset on image classification consisting of about 1.2M training images and 50K validation set with 1000 classes. It is commonly used for pre-training due to its high quality and diversity of instances. After pre-training, we conducted several experiments to evaluate the proposed method. We fine-tune our pre-trained model on ImageNet dataset and conduct experiments on linear probing for analyzing linear separability. To assess the transferability of the model, we used COCO [39] and ADE20K [40] benchmark datasets for object detection, instance segmentation and semantic segmentation. Implementation details are specified below.

4.1.1. Pre-training

Most of the settings follow MAE [23]. In detail, we apply random resized crop and random horizontal flip for augmentation. They are resized to be 224×224 so they can be divided into 16×16 patches. For the encoder, we use ViT-Base [22] with a 12-layer Transformer with 768 hidden size. We adopt the AdamW optimizer with β_1=0.9 and β_2=0.95 for optimization. The learning rate is set to 1.5e-4, with a warmup of 40 epochs and cosine learning rate decay. To initialize Transformer blocks, we used Xavier uniform initialization. We pre-trained the model for 1600 epochs with a batch size of 256. We set the hyperparmeters, mask ratio, loss weight and decoder depth through experimental results, as described in Section 4.2.1.

4.1.2. Fine-tuning

We conduct full fine-tuning on image classification, object detection, and semantic segmentation. Every experiment is trained on the training set and evaluated on the validation set of corresponding datasets.

Image Classification For image classification, we evaluate our model with top-1 accuracy on ImageNet validation set and trained for 100 epochs with a batch size of 512. Mixup [41] with a probability of 0.8 and RandAugment [42] is used. We use the vanilla ViT base for backbone architecture with a classifier for classification. Only encoder is used for fine-tuning initializing ViT with our pre-trained encoder weights.

Object Detection and Segmentation COCO [39] is a large-scale benchmark dataset used for object detection and segmentation, and we used COCO2017 dataset which consists of about 120k images with 80 common object classes. Mask-RCNN [43] framework is adapted, with FPN [44] backbone replaced with ViT, and initialized ViT with weights of our pre-trained model. The training settings follow [45]. To evaluate the model's performance, we use AP, a widely used metric for object detection and instance segmentation. AP^{box} and AP^{mask} are used to evaluate object detection and instance segmentation, respectively.

Semantic Segmentation ADE20K [40] is a benchmark dataset comprising 150 semantic categories with 20k images for the train set and 2k for validation. We used UperNet [46] for semantic segmentation on ADE20K dataset following the code of [2]. To evaluate semantic segmentation, we used mIOU for metric, which is the mean value of IOU.

4.1.3. Linear Probing

Linear probing follows a similar process as fine-tuning, but with a frozen backbone following the process described in [47–49]; we add a linear classifier on top while training. By doing so, we can evaluate the linear separability of the model. Different from fine-tuning, common regularization like color jittering, Mixup [41], or cutmix [50] is not used in linear probing following [32]. Since only linear classifier is activated, we trained the model for 100 epochs with a larger batch size of 1024.

4.2. Experimental Results

Experimental results on ImageNet classification, linear probing, object detection, and semantic segmentation are shown in Table 1, 2, 3, and 4. We compare our model to the previous CL [13–15] and MIM [23,35] methods using only ImageNet-1K for pre-training, except for BEiT [2], using additional data to train tokenizer. We report results of each model using ViT-Base/16 [22] for backbone and ResNet-50(4×) [19] for SimCLR.

Table 1. Top-1 accuracy on ImageNet-1K in fine-tuning setting. All models are pre-trained and fine-tuned on ImageNet-1K. Except for SimCLR, which uses CNN for backbone, we evaluate performance of models with ViT-B encoder. The best result is shown in bold and second best result is underlined.

Model	Approach	Training Epochs	Accuracy
SimCLR [13]	CL	1000	80.4
MoCo-v3 [32]	CL	300	83.2
DINO [14]	CL	300	82.8
CIM [36]	MIM	300	83.3
BEiT [2]	MIM	800	83.2
SimMIM [35]	MIM	800	83.8
CAE [24]	MIM	1600	83.9
MAE [23]	MIM	1600	83.6
Ours	MIM+CL	800	83.2
Ours	MIM+CL	1600	**84.3**

Table 2. Linear probing results on ImageNet-1K dataset. The best result is shown in bold and second best results are underlined.

Method	Approach	Pre-training Epochs	Accuracy
SimCLR [13]	CL	1000	76.5
MoCo-v3 [32]	CL	300	76.7
DINO [14]	CL	300	**78.2**
BEiT [2]	MIM	800	56.7
SimMIM [35]	MIM	800	56.7
CAE [24]	MIM	1600	71.4
MAE [23]	MIM	1600	68.0
Ours	MIM+CL	1600	76.7

As shown in Table 1, our model achieves 84.3% top-1 accuracy, which is 0.4% higher than previous best result [24], outperforming other CL, MIM-based methods. Table 2 shows linear probing results and our model records 76.7% accuracy. Though DINO shows higher performance in linear probing, our model shows comparable result compared to DINO. Especially, we achieve remarkable performance gain compared to MIM-based methods [2,23,24,35], which is 20%, 5.3%, 8.7% higher than previous best results. These results indicate that applying contrastive learning to MIM better captures rich and general features of an image and improves linear separability simultaneously. We also note that longer training improves performance. When pre-trained for 1600 epochs, there was 1.1% performance gain compared to when pre-trained for 800 epochs.

Table 3. Object detection and segmentation results on COCO dataset. The best result is shown in bold, second best result is underlined.

Method	**AP**^{box}	**AP**^{mask}
MoCo-v3 [32]	47.9	42.7
BeiT [2]	49.8	44.4
CAE [36]	50.0	44.0
SimMIM [35]	**52.3**	-
MAE [23]	50.3	44.9
Ours	51.3	**45.6**

Table 4. Semantic segmentation results on ADE20K dataset. The best result is shown in bold, second best results are underlined.

Method	mIOU
MoCo-v3 [32]	47.3
BeiT [2]	47.1
CAE [36]	<u>50.2</u>
SimMIM [35]	**52.8**
MAE [23]	48.1
Ours	<u>50.2</u>

To evaluate transfer learning performance, we conducted experiments on object detection and segmentation. Table 3 shows object detection and instance segmentation results on COCO dataset. Our model further improves the segmentation results achieving 51.3% AP^{box} and 45.6% AP^{mask}. As shown in Table 4, we achieve 50.2% mIOU on semantic segmentation showing second best performance compared to other models. Especially, we outperform MAE by 2.1% mIOU score. By these results, we demonstrate that our model can have better transferability through utilizing contrastive learning to MIM.

4.2.1. Architecture Analysis

To analyze the components of our architecture, we conducted experiments on masking ratio, loss weight, and decoder depth, evaluated on ImageNet-1K dataset. We pre-trained the model for 200 epochs and fine-tuned the model for 100 epochs. The default setting of our model in Section 4.1.1 is derived from these results. Experimental results on mask ratio, loss weight and decoder depth are shown in Table 5.

Table 5. Fine-tuning results on different masking ratio, loss weight and decoder depth. Best results are shown in bold.

Mask Ratio	Accuracy	Loss Weight	Accuracy	Decoder Depth	Accuracy
50%	79.22	0.1	78.10	1	74.89
75%	79.09	0.5	78.09	2	78.56
80%	**79.25**	1	78.09	4	**80.03**
90%	79.23	1.5	79.31	8	**80.03**
95%	78.30	2.0	**79.64**	12	79.11

Masking ratio Previous contrastive learning methods adopt strong augmentation, i.e., gaussian blur, and color distortion, due to the representation collapse problem. When the model learns the same representation losing input data diversity, resulting in constant output and performance decrement, it is called representation collapse. To avoid this, it is needed to have diverse negative samples, relying on data augmentation. Previous studies have investigated combinations of augmentation, showing performance gain depending on which augmentation is used. Besides, our work overcomes this representation collapse problem to some extent by simply masking a relatively high portion of image without additional augmentation. We conduct experiments on mask ratio, as shown in Table 5. Masking 80% of the input patch sequence showed the best performance, while extreme masking (90%) showed the lowest.

Loss weight As aforementioned, our total loss is a weighted sum of two training objectives. We conducted experiments to explore how λ, a hyperparameter of loss weight, affects model performance by changing the loss weight. Note that when λ is set to 0, the model is the same as the baseline, MAE. Results show that contrastive loss does affect model performance, in a good way. Interestingly, as loss weight increases, that is, the more the contrastive loss is contributed, the model's performance correspondingly increases. We can say that by adding contrastive loss, the encoder is trained to learn more general and holistic representation.

Decoder depth Since we adopted asymmetric encoder-decoder architecture, we had to figure out whether the model benefits from shallower decoder depth. It is clear that computational cost would be reduced due to fewer parameters; however, reconstruction task relies on decoder, requiring sufficient depth of decoder to reconstruct original signals from corrupted ones. We experimented on several depths, 1, 2, 4, 8, and 12. Our baseline, MAE, adopts a depth of 8 for decoder. As shown in Table 5, a deeper decoder does not benefit the model performance only contributes to more computation. Because the depth of 4 and 8 shows the same performance on fine-tuning and has no difference in reconstruction results as shown in Figure 5, we adopted a 4-layer decoder for computing efficiency.

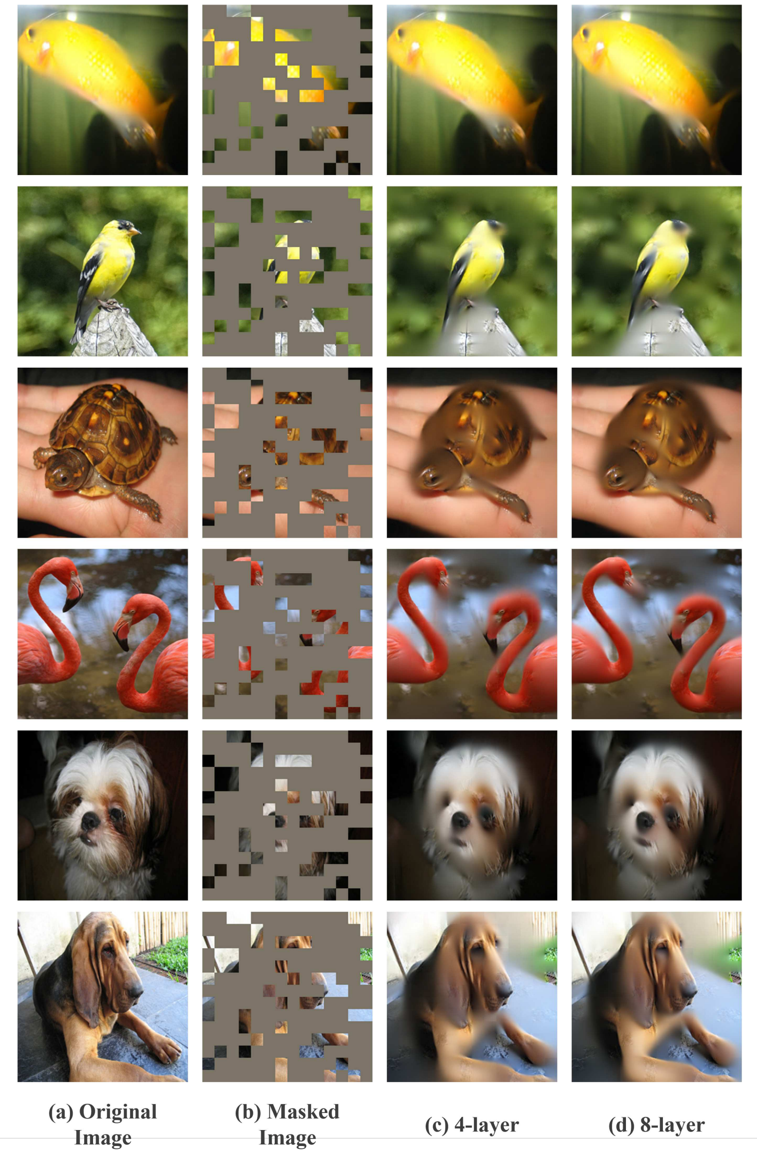

Figure 5. Visualization of reconstruction of different decoder depth. **(a)** is the original input image, **(b)** is the masked image, **(c)** and **(d)** is the predicted image with 4-layer and 8-layer, respectively.

4.3. Ablation Studies

We ablated studies on the main properties of our framework; two different target views and contrastive loss. ImageNet-1K top-1 accuracy is used for evaluation. Note that when two main properties are removed, the model is the same as the baseline, MAE. For a fair comparison, all models were trained at the same setting; fine-tuning after pre-training for 200 epochs with a batch size of 512.

Table 6 shows the results of ablation experiments. Among all methods, ours with two different target views and contrastive loss performs the best, showing 1.23%, 4.76% and 2.61% performance gain compared to other methods. When any of the component was removed, it is shown to be less effective or showing only marginal performance increment compared to the baseline. By these results, we deduce that each component benefits mutually in learning rich image representation.

Table 6. Ablation experiment results.

Methods	Accuracy
Ours	79.09
Ours w/o two different targets	77.86
Ours w/o contrastive loss	74.33
baseline [23]	76.48

5. Conclusion

In this paper, we introduce a simple framework applying contrastive learning to masked image modeling that enables the model to learn rich representations considering both global and local patterns. Masking a high portion of the entire image works as strong augmentation which overcomes the representation collapse problem of contrastive learning. In addition, we exploit an image-level approach by contrasting two different views, strengthening MIM to learn holistic representations. We conduct several experiments to prove the effectiveness, achieving promising results on various downstream tasks, image classification, object detection, and semantic segmentation. By these results, we demonstrate that utilizing contrastive learning to masked image modeling via multi-view autoencoder strengthen the model to learn rich representation considering both image and token-level features. Possible extensions may include pre-training with web-scale dataset and video representation learning.

Author Contributions: Conceptualization, S.J. and J.R.; methodology S.J.; software, S.J.; validation, S.J., S.H.; investigation, S.J.; writing—original draft preparation, S.J., S.H.; writing—review and editing, S.J., S.H.; visualization, S.J. and S.H.; supervision, J.R.; project administration, J.R.; funding acquisition, J.R. All authors have read and agreed to the published version of the manuscript.

Funding: This research was supported by a grant (21163MFDS502) from Ministry of Food and Drug Safety in 2023.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. LeCun, Y.; Bengio, Y.; Hinton, G. Deep learning. *nature* **2015**, *521*, 436–444.
2. Bao, H.; Dong, L.; Piao, S.; Wei, F. Beit: Bert pre-training of image transformers. *arXiv preprint arXiv:2106.08254* **2021**.
3. Liu, Y.; Sangineto, E.; Bi, W.; Sebe, N.; Lepri, B.; Nadai, M. Efficient training of visual transformers with small datasets. *Advances in Neural Information Processing Systems* **2021**, *34*, 23818–23830.
4. Jaiswal, A.; Babu, A.R.; Zadeh, M.Z.; Banerjee, D.; Makedon, F. A survey on contrastive self-supervised learning. *Technologies* **2020**, *9*, 2.
5. Liu, X.; Zhang, F.; Hou, Z.; Mian, L.; Wang, Z.; Zhang, J.; Tang, J. Self-supervised learning: Generative or contrastive. *IEEE transactions on knowledge and data engineering* **2021**, *35*, 857–876.
6. Hendrycks, D.; Mazeika, M.; Kadavath, S.; Song, D. Using self-supervised learning can improve model robustness and uncertainty. *Advances in neural information processing systems* **2019**, *32*.

7. Zhai, X.; Oliver, A.; Kolesnikov, A.; Beyer, L. S4l: Self-supervised semi-supervised learning. In Proceedings of the Proceedings of the IEEE/CVF international conference on computer vision, 2019, pp. 1476–1485.
8. Zhang, C.; Zhang, C.; Song, J.; Yi, J.S.K.; Zhang, K.; Kweon, I.S. A survey on masked autoencoder for self-supervised learning in vision and beyond. *arXiv preprint arXiv:2208.00173* **2022**.
9. Ng, A.; et al. Sparse autoencoder. *CS294A Lecture notes* **2011**, *72*, 1–19.
10. Bank, D.; Koenigstein, N.; Giryes, R. Autoencoders. *Machine Learning for Data Science Handbook: Data Mining and Knowledge Discovery Handbook* **2023**, pp. 353–374.
11. Vincent, P.; Larochelle, H.; Lajoie, I.; Bengio, Y.; Manzagol, P.A.; Bottou, L. Stacked denoising autoencoders: Learning useful representations in a deep network with a local denoising criterion. *Journal of machine learning research* **2010**, *11*.
12. Vincent, P.; Larochelle, H.; Bengio, Y.; Manzagol, P.A. Extracting and composing robust features with denoising autoencoders. In Proceedings of the Proceedings of the 25th international conference on Machine learning, 2008, pp. 1096–1103.
13. Chen, T.; Kornblith, S.; Norouzi, M.; Hinton, G. A simple framework for contrastive learning of visual representations. In Proceedings of the International conference on machine learning. PMLR, 2020, pp. 1597–1607.
14. Caron, M.; Touvron, H.; Misra, I.; Jégou, H.; Mairal, J.; Bojanowski, P.; Joulin, A. Emerging properties in self-supervised vision transformers. In Proceedings of the Proceedings of the IEEE/CVF international conference on computer vision, 2021, pp. 9650–9660.
15. He, K.; Fan, H.; Wu, Y.; Xie, S.; Girshick, R. Momentum contrast for unsupervised visual representation learning. In Proceedings of the Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, 2020, pp. 9729–9738.
16. Radford, A.; Narasimhan, K.; Salimans, T.; Sutskever, I.; et al. Improving language understanding by generative pre-training **2018**.
17. Devlin, J.; Chang, M.W.; Lee, K.; Toutanova, K. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805* **2018**.
18. Liu, Y.; Ott, M.; Goyal, N.; Du, J.; Joshi, M.; Chen, D.; Levy, O.; Lewis, M.; Zettlemoyer, L.; Stoyanov, V. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692* **2019**.
19. He, K.; Zhang, X.; Ren, S.; Sun, J. Deep residual learning for image recognition. In Proceedings of the Proceedings of the IEEE conference on computer vision and pattern recognition, 2016, pp. 770–778.
20. Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A.N.; Kaiser, Ł.; Polosukhin, I. Attention is all you need. *Advances in neural information processing systems* **2017**, *30*.
21. Battaglia, P.W.; Hamrick, J.B.; Bapst, V.; Sanchez-Gonzalez, A.; Zambaldi, V.; Malinowski, M.; Tacchetti, A.; Raposo, D.; Santoro, A.; Faulkner, R.; et al. Relational inductive biases, deep learning, and graph networks. *arXiv preprint arXiv:1806.01261* **2018**.
22. Dosovitskiy, A.; Beyer, L.; Kolesnikov, A.; Weissenborn, D.; Zhai, X.; Unterthiner, T.; Dehghani, M.; Minderer, M.; Heigold, G.; Gelly, S.; et al. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929* **2020**.
23. He, K.; Chen, X.; Xie, S.; Li, Y.; Dollár, P.; Girshick, R. Masked autoencoders are scalable vision learners. In Proceedings of the Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, 2022, pp. 16000–16009.
24. Chen, X.; Ding, M.; Wang, X.; Xin, Y.; Mo, S.; Wang, Y.; Han, S.; Luo, P.; Zeng, G.; Wang, J. Context autoencoder for self-supervised representation learning. *International Journal of Computer Vision* **2023**, pp. 1–16.
25. Caron, M.; Misra, I.; Mairal, J.; Goyal, P.; Bojanowski, P.; Joulin, A. Unsupervised learning of visual features by contrasting cluster assignments. *Advances in neural information processing systems* **2020**, *33*, 9912–9924.
26. Park, N.; Kim, W.; Heo, B.; Kim, T.; Yun, S. What Do Self-Supervised Vision Transformers Learn? *arXiv preprint arXiv:2305.00729* **2023**.
27. Abnar, S.; Zuidema, W. Quantifying attention flow in transformers. *arXiv preprint arXiv:2005.00928* **2020**.
28. Chen, X.; Fan, H.; Girshick, R.; He, K. Improved baselines with momentum contrastive learning. *arXiv preprint arXiv:2003.04297* **2020**.
29. Grill, J.B.; Strub, F.; Altché, F.; Tallec, C.; Richemond, P.; Buchatskaya, E.; Doersch, C.; Avila Pires, B.; Guo, Z.; Gheshlaghi Azar, M.; et al. Bootstrap your own latent-a new approach to self-supervised learning. *Advances in neural information processing systems* **2020**, *33*, 21271–21284.

30. Tian, Y.; Krishnan, D.; Isola, P. Contrastive multiview coding. In Proceedings of the Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XI 16. Springer, 2020, pp. 776–794.
31. Zhang, C.; Zhang, K.; Zhang, C.; Pham, T.X.; Yoo, C.D.; Kweon, I.S. How does simsiam avoid collapse without negative samples? a unified understanding with self-supervised contrastive learning. *arXiv preprint arXiv:2203.16262* **2022**.
32. Chen, X.; Xie, S.; He, K. An Empirical Study of Training Self-Supervised Vision Transformers. *arXiv e-prints* **2021**, p. arXiv:2104.02057, [arXiv:cs.CV/2104.02057]. https://doi.org/10.48550/arXiv.2104.02057.
33. Chen, M.; Radford, A.; Child, R.; Wu, J.; Jun, H.; Luan, D.; Sutskever, I. Generative pretraining from pixels. In Proceedings of the International conference on machine learning. PMLR, 2020, pp. 1691–1703.
34. Ramesh, A.; Pavlov, M.; Goh, G.; Gray, S.; Voss, C.; Radford, A.; Chen, M.; Sutskever, I. Zero-shot text-to-image generation. In Proceedings of the International Conference on Machine Learning. PMLR, 2021, pp. 8821–8831.
35. Xie, Z.; Zhang, Z.; Cao, Y.; Lin, Y.; Bao, J.; Yao, Z.; Dai, Q.; Hu, H. Simmim: A simple framework for masked image modeling. In Proceedings of the Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2022, pp. 9653–9663.
36. Fang, Y.; Dong, L.; Bao, H.; Wang, X.; Wei, F. Corrupted image modeling for self-supervised visual pre-training. *arXiv preprint arXiv:2202.03382* **2022**.
37. Wettig, A.; Gao, T.; Zhong, Z.; Chen, D. Should you mask 15% in masked language modeling? *arXiv preprint arXiv:2202.08005* **2022**.
38. Deng, J.; Dong, W.; Socher, R.; Li, L.J.; Li, K.; Fei-Fei, L. Imagenet: A large-scale hierarchical image database. In Proceedings of the 2009 IEEE conference on computer vision and pattern recognition. Ieee, 2009, pp. 248–255.
39. Lin, T.Y.; Maire, M.; Belongie, S.; Hays, J.; Perona, P.; Ramanan, D.; Dollár, P.; Zitnick, C.L. Microsoft coco: Common objects in context. In Proceedings of the Computer Vision–ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part V 13. Springer, 2014, pp. 740–755.
40. Zhou, B.; Zhao, H.; Puig, X.; Fidler, S.; Barriuso, A.; Torralba, A. Scene parsing through ade20k dataset. In Proceedings of the Proceedings of the IEEE conference on computer vision and pattern recognition, 2017, pp. 633–641.
41. Zhang, H.; Cisse, M.; Dauphin, Y.N.; Lopez-Paz, D. mixup: Beyond empirical risk minimization. *arXiv preprint arXiv:1710.09412* **2017**.
42. Cubuk, E.D.; Zoph, B.; Shlens, J.; Le, Q.V. Randaugment: Practical automated data augmentation with a reduced search space. In Proceedings of the Proceedings of the IEEE/CVF conference on computer vision and pattern recognition workshops, 2020, pp. 702–703.
43. He, K.; Gkioxari, G.; Dollár, P.; Girshick, R. Mask R-CNN. In Proceedings of the 2017 IEEE International Conference on Computer Vision (ICCV), 2017, pp. 2980–2988. https://doi.org/10.1109/ICCV.2017.322.
44. Lin, T.Y.; Dollár, P.; Girshick, R.; He, K.; Hariharan, B.; Belongie, S. Feature Pyramid Networks for Object Detection. In Proceedings of the 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2017, pp. 936–944. https://doi.org/10.1109/CVPR.2017.106.
45. Li, Y.; Xie, S.; Chen, X.; Dollár, P.; He, K.; Girshick, R.B. Benchmarking Detection Transfer Learning with Vision Transformers. *ArXiv* **2021**, *abs/2111.11429*.
46. Xiao, T.; Liu, Y.; Zhou, B.; Jiang, Y.; Sun, J. Unified perceptual parsing for scene understanding. In Proceedings of the Proceedings of the European conference on computer vision (ECCV), 2018, pp. 418–434.
47. Kornblith, S.; Shlens, J.; Le, Q.V. Do better imagenet models transfer better? In Proceedings of the Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, 2019, pp. 2661–2671.
48. Oord, A.v.d.; Li, Y.; Vinyals, O. Representation learning with contrastive predictive coding. *arXiv preprint arXiv:1807.03748* **2018**.
49. Kolesnikov, A.; Zhai, X.; Beyer, L. Revisiting self-supervised visual representation learning. In Proceedings of the Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, 2019, pp. 1920–1929.
50. Yun, S.; Han, D.; Oh, S.J.; Chun, S.; Choe, J.; Yoo, Y. Cutmix: Regularization strategy to train strong classifiers with localizable features. In Proceedings of the Proceedings of the IEEE/CVF international conference on computer vision, 2019, pp. 6023–6032.

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.