MEJIGCLU: MORE EFFECTIVE JIGSAW CLUSTERING FOR UNSUPERVISED VISUAL REPRESENTATION LEARNING

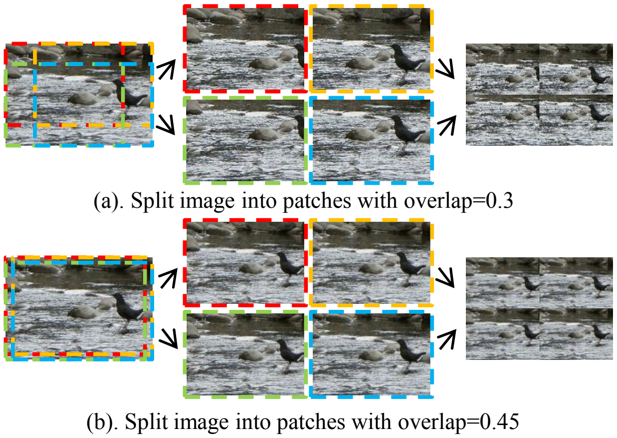
Yongsheng Zhang[⋆†] *Qing Liu*[⋆†] *Yang Zhao*[⋆] *Yixiong Liang*[⋆‡]

[⋆]School of Computer Science, Central South University, Changsha 410083, P.R. China

ABSTRACT

Unsupervised visual representation learning aims to learn general features from unlabelled data. Early methods design intra-image pretext tasks as learning targets and can be achieved with low computational overhead but unsatisfactory performance. Recent methods introduce contrastive learning and achieve surprising performance, but multiple views of training data are required in one batch, resulting in high computational overhead. To achieve competitive results to contrastive learning with low computational overhead, we propose a new unsupervised representation learning method with jigsaw clustering and classification as pretext tasks motivate the network to learn discriminative feature. To increase the data diversity, we propose to partition each training image into patches with random overlap, then randomly permute and stitch them into new training batch. Comparing with SOTAs, our method achieves state-of-the-art performance on both image classification/semi-classification on ImageNet and object detection on COCO.

Index Terms— Unsupervised learning, contrastive learning, jigsaw clustering, classification



(a). Split image into patches with overlap=0.3

(b). Split image into patches with overlap=0.45

Fig. 1. Image splitting with different overlapping rate. When a small object is not located in image centre, splitting image with a small overlap of 0.3 in (a) results in large diversity of patches, which makes the CNN learning difficulty. In (b), increasing the overlap to 0.45. Splitting image with different overlap, the diversity of patches decreases, which facilitates for CNN learning with our proposed pretext tasks: patch classification task and clustering task.

1. INTRODUCTION

Recently, unsupervised learning has made great achievements in computer vision, even surpassing supervised learning on some downstream tasks. Its goal is to learn general deep features with unlabelled data for downstream tasks via motivating CNNs to learn pretext tasks. Unsupervised learning greatly reduces the cost of manual labelling and is important for making full use of large-scale unlabelled data in reality.

Most existing unsupervised methods design either intra-image tasks such as colourization [1] and jigsaw puzzle [2] to motivate CNNs learn invariant information inside the images or inter-image tasks such as MoCo [3], SimCLR [4] to motivate CNNs learn powerful inter-image discriminative information. For intra-image task-based unsupervised methods, in each iteration, only the training batch itself is fed into the CNNs. Thereby computational cost is low but the learning ability of CNNs is also limited. For inter-image task-based methods, in each iteration, both the training batch and their augmentation are simultaneously fed into the CNNs to motivate CNNs learn robust features to transform performed in augmentation. However, this significantly increases the requirement on computation resources.

To achieve good performance with small training resources, JigClu [5] is proposed to combine the advantages of jigsaw puzzle [2] and contrastive learning. In detail, images in one batch are first split into patches uniformly with a fixed overlap of 0.3, then patches are randomly permuted and stitched to form a new batch to train CNNs to predict which image each patch belongs to as well as the location in original image. For images containing objects in proper size in central, the overlap ensures that each patch is dominated by objects rather than the diverse background, which further facilitates for learning. However, when small objects are far away from the image centre, splitting image with fix overlap of 0.3 results in large diversity over patches, as illus-

[†]Equal contribution.

[‡]Corresponding author: yxliang@csu.edu.cn

trated in Fig. 1(a), with which learning to perform pretext tasks particularly the patch location task is still a challenge. To relax the difficulty of pretext tasks, we propose a more effective jigsaw clustering unsupervised method, termed as MEJigClu, as illustrated in Fig. 2. It degrades the location task to classification task to enforce CNNs learn to predict which patches belong to the same original image. Instead of reducing the patch diversity to make the patch location be easier, we argue that our proposed pretext task requires large diversity patches to motivate CNNs learn robust features. To this end, we propose to augment training data via splitting training images with random overlap. As shown in Fig. 1(a) and (b), with random overlap, the patch diversity increases.

We validate the effectiveness of our proposed MEJigClu on linear evaluation task as well as tasks of semi-supervised learning and transfer learning on object detection. Comparing with SOTAs, our method achieves best performance. Particularly, comparing with the strong baseline JigClu [5], our method improves the top-1 accuracy of ImageNet by 0.4% on linear evaluation; on semi-supervised learning, the accuracy of top-1 and top-5 improves by 0.5% and 0.8%, respectively, on the 1% labeled dataset; being transferred object detection on COCO, our method improves the AP by 0.7%.

2. RELATED WORK

Vanilla unsupervised learning. The vanilla unsupervised learning relies heavily on hand-designed pretext tasks to guide the training of feature extractors. They almost leverage pretext tasks of either data prediction or transformation prediction. For the former, Colorization [1] guides the learning of network by predicting the *a*- and *b*-bands of an image based on the *L*-band, and Image Inpainting [6] takes image missing part prediction as the pretext task. In Cross-Channel Prediction [7], the mutual prediction of the greyscale and colour channel, and the mutual prediction of the RGB channel and the HHA channel are proposed as pretext tasks. For the latter, Context Prediciton [8] predicts the relative spatial position relationship between patches segmented in advance from the image. Jigsaw [2] forces the network to solve the jigsaw puzzle. Rotation prediction [9] drives CNNs to learn more robust features by predicting the rotation angle.

Contrastive learning. Contrastive learning aims to close the distance between positive sample pairs and push away the distance between negative sample pairs. To create positive sample pairs, MoCo [3] and SimCLR [4] use a serials of data augmentation strategies. To ensure the learning performance, a large number of negative samples are generated. BYOL [10] and SimSiam [11] use asymmetric prediction head and predictor head respectively and discard negative samples completely. SCLR [12], among others, designs more appropriate comparison learning strategies for the characteristics of different downstream tasks.

JigClu. JigClu [5] is the first single-batch method to combine the advantages of both single-batch methods and dual-batch methods. It solves the jigsaw puzzles problem of performing randomly disrupted patches within a batch by adding clustering and localization branches at the end of backbone network, thus using both intra- and inter-images information to help the learn of feature extractor.

3. METHOD

Fig. 2 illustrates our method overview. We train the feature extractor via two task branches, i.e., patch clustering and classification branch. To increase the diversity of patches, we propose a new image splitting and stitching strategy.

Method Overview. The overview of our method is shown in Fig. 2. It consists of a random image splitting and stitching module to generate montage images, a backbone network as feature extractor, a decouple module, a clustering branch and our proposed classification branch to motivate the learning of backbone network. First, n montage images are constructed from $n \times m \times m$ image patches by our proposed random image splitting and stitching module. Then a batch of montage images are fed into a backbone network to obtain the feature maps. Thereafter, same to JigClu [5], the decouple module is employed to decouple the entire feature maps into patches for each input patch. Finally, same to JigClu [5], a multilayer perceptron (MLP_1) is used to encode the $n \times m \times m$ feature patches for the clustering task. Simultaneously, we use an another multilayer perceptron (MLP_2) to encode the $n \times m \times m$ feature patches into $n \times m \times m$ 1-dimensional vectors for the classification task. Here the parameters in MLP_1 and MLP_2 are not shared and we follow JigClu [5] and set $m = 2$.

Image splitting with random overlap and stitching module. This module is performed on images in each training batch to generate montage images for training of feature extractor. First, we randomly select n images from the training set and form a batch $\mathbf{X} = \{x_1, x_2, \dots, x_n\}$. Then we crop images in $\mathbf{X}$ into $m \times m$ patches with an overlap. Unlike JigClu [5], which crops two adjacent patches with a fixed overlap of 0.3, the overlap of our method is randomly set varying from 0.15 and 0.45. With those image patches, we permute them randomly and stitch them back to n montage images. In this way, we are able to generate image patches with large diversities and stitched them into montage images forming new training batches. In this way, via the two pretext tasks, the CNNs are able to learn more robust features.

Classification branch. This branch aims to predict the category of each of the $m \times m$ patches in a montage image that fuses different image patches. Patches from the same original image are assigned with same category label. Unlike the localization branch in JigClu [5] which focuses on the location information of each patch in the original image, the classification branch is more concerned with the attribution of each patch to the original image such that the network is able to learn the essential information from the local patches. In ad-

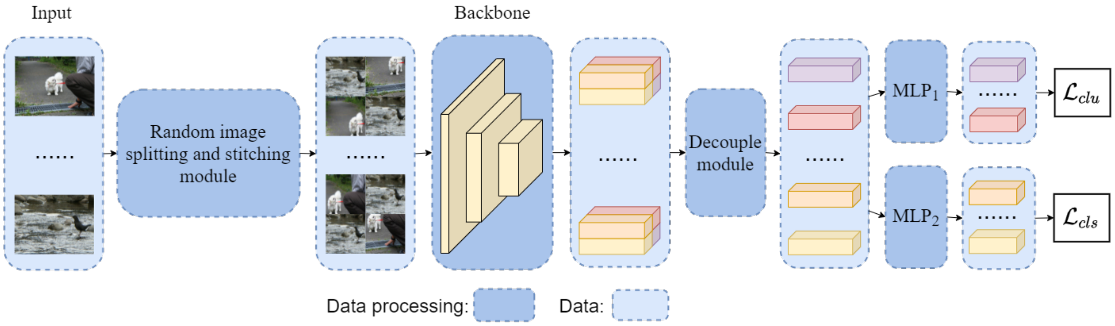


Fig. 2. The overview of our method.

dition, both the classification and clustering branch assign a label to each patch, but their goals are different. The classification branch more explicitly instructs at a high level of semantics that each patch should have similarity to its patches from the same image while the clustering branch only makes the features of patches from the same image similar in terms of feature dimension.

Loss Function. For classification task, we assign each patch a ground truth category and patches from the same original image in each training batch are assigned with the same category. The category number is determined by batch size. The classification branch predicts the category of each patch and cross-entropy loss is adopted:

$$\mathcal{L}_{cls} = CrossEntropy(L, L_c) \tag{1}$$

where L and L_c are the predicted label and ground truth for each patch respectively.

For clustering branch, we follow JigClu [5] and use cosine similarity to calculate the distance between feature vectors. For each pair of patches, the loss function is:

$$\ell_{i,j} = -\log \frac{\exp(\cos(\mathbf{z}_i, \mathbf{z}_j)/\tau)}{\sum_{k=1}^{nmm} \mathbb{1}_{k \neq i} \exp(\cos(\mathbf{z}_i, \mathbf{z}_k)/\tau)} \tag{2}$$

The loss function of clustering branch is the summation of losses of all patch pairs belonging to the same class:

$$\mathcal{L}_{clu} = \frac{1}{nmm} \sum_i \left(\frac{1}{mm-1} \sum_{j \in C_i} \ell_{i,j}\right) . \tag{3}$$

The ultimate optimization goal of our approach is:

$$\mathcal{L} = \alpha \mathcal{L}_{cls} + \beta \mathcal{L}_{clu} \ , \tag{4}$$

where α and β are hyper-parameters to balance the classification task and the clustering task. In our experiments, we set $\alpha = \beta = 1$.

4. EXPERIMENT

We validate our MEJigClu on three tasks, i.e., linear evaluation, semi-supervised learning and transfer learning.

Training details. We use the same settings as JigClu [5] to train the feature extractor with the training images of ImageNet-1K.

Results on linear evaluation task. We first train the ResNet-50 model with the training set of ImageNet-1K by unsupervised training, then we fix its parameters and stitch a linear classifier and only train the linear classifier by supervised learning with training set of ImageNet-1K. We validate the well-trained classification model on validation set of ImageNet-1K and report the results of ours and the SOTAs in Table 1. Our method improves the top-1 accuracy on ImageNet-1K by 0.4% over JigClu [5]. The possible reason is that the classification branch in our method is able to utilise more explicit supervised information and match with the clustering branch better, which further improves the learning of inter-image and intra-image information. Comparing with dual-batch methods, our method achieves competitive performance with only half of its training resources.

Results on semi-supervised learning task. We use using training data of ImageNet-1K with 1%, 10% labels respectively to fine-tune the classifier attached on top of the ResNet-50 which is well-trained via unsupervised training methods. The results by ours and eight SOTAs are listed in Table 2.

To make a fair comparison, the results by MoCov2 [21] here are provided by the model trained 200 epochs. As we can see from Table 2, our method achieves the first and second best on the 1%- and 10% labelled datasets respectively. Particularly, on the 1%-labelled dataset, our method outperforms JigClu [5] by 0.5% and 0.8% in top-1 and top-5 accuracy, respectively. Obviously, our method has better generalization when there are fewer labels. On 10%-labelled dataset, our method is inferior to UDA [25], which is specifically designed for semi-supervised learning. The possible reason is that UDA [25] adopts stronger data augmentation.

Table 1. Linear evaluation results of ResNet-50 models on the ImageNet-1K.

Method	#of Batch in Training	Accuracy
Supervised	single-batch	77.2
Colorization [1]	single-batch	39.6
JigPuz [2]	single-batch	45.7
DeepCulster [13]	single-batch	48.4
NPID [14]	single-batch	54.0
BigBiGan [15]	single-batch	56.6
LA [16]	single-batch	58.8
SeLa [17]	single-batch	61.5
CPC v2 [18]	single-batch	63.8
JigClu [5]	single-batch	66.4
MEJigClu(Ours)	single-batch	**66.8**
MoCo [3]	dual-batches	60.6
PIRL [19]	dual-batches	63.6
SimCLR [4]	dual-batches	64.3
PCL [20]	dual-batches	65.9
MoCo v2 [21]	dual-batches	67.7

Table 2. Semi-supervised classification results of ResNet-50 models on the ImageNet-1k validation set.

Method	Label fraction			
	1%		10%	
	Top-1	Top-5	Top-1	Top-5
Supervised	25.4	48.4	56.4	80.4
Methods using label-propagation:				
Pseudo-label [22]	-	51.6	-	82.4
Entropy-Min [23]	-	47.0	-	83.4
S4L-Rotation [24]	-	53.4	-	83.8
UDA* [25]	-	68.8	-	88.5
Methods using unsupervised learning:				
NPID [14]	-	39.2	-	77.4
PIRL [19]	-	57.2	-	83.8
MoCo v2 [21]	34.5	62.2	61.1	83.9
JigClu [5]	40.7	68.9	63.0	85.2
MEJigClu(Ours)	**41.2**	**69.7**	**63.2**	**85.3**

Table 3. Results of Faster-RCNN R50-FPN models on COCO validation set.

Method	AP	AP_{50}	AP_{75}	AP_s	AP_m	AP_l
MoCo v2 [21]	38.9	58.8	42.5	23.3	41.8	50.0
JigClu [5]	39.3	59.4	42.5	23.6	42.5	49.7
MEJigClu(Ours)	**40.0**	**60.2**	**43.9**	**24.4**	**43.3**	**50.6**

Table 4. Results on COCO validation set by fine-tuning Faster-RCNN-R50 models with different pre-trained models.

$\mathcal{L}_{loc}$	$\mathcal{L}_{clu}$	$\mathcal{L}_{cls}$	overlap	AP	AP_{50}	AP_{75}
✓	✓		0.3	39.3	59.4	42.5
	✓	✓	0.3	39.6	59.8	43.2
	✓	✓	random	**40.0**	**60.2**	**43.9**

Results on transfer learning task. We conduct object detection experiments on COCO. Same to JigClu [5], we use Faster-RCNN R-50-FPN architecture as object detector. We initialize the backbone with pre-trained models by unsupervised methods and fine-tune model parameters with COCO training set. Results by MoCo v2 [21], JigClu [5] and our MEJigClu are listed in Table 3. Our MEJigClu outperforms MoCo v2 [21], JigClu [5] by 1.1% and 0.7% in AP respectively.

Ablation study. We validate the effectiveness of our classification pretext task and image splitting with random overlap and stitching module on COCO validation set. Performances by three models are provided in Table 4. We can find from Table 4 that the AP is improved by 0.3% when replacing localization branch with our classification branch. Splitting images in training batch with random overlap further improve 0.4% . Comparing with JigClu [5], our method finally improves AP by 0.7%. In addition, in order to avoid the problem of label collision caused by the same image in two different batches, we can select one image from each of the 256 independent subsets to form the input of the network.

5. CONCLUSION

In this paper, we propose an effective unsupervised representation learning method, which motivates CNNs learn robust and powerful feature extractor via two pretext tasks, i.e. clustering and classification task. To increase patch diversity for robust feature learning, a novel image splitting with random overlap and stitching module is proposed. Thereby, our method enables the network to learn more discriminative instance-level information from diverse images, which is significantly helpful for image classification and object detection. Extensive experiments are conducted and results show that our method achieves SOTA performance.

6. ACKNOWLEDGMENTS

This work is supported in part by the High Performance Computing Center of Central South University. Qing Liu is partially supported by the National Natural Science Foundation of China (NSFC No. 62006249), Changsha Municipal Natural Science Foundation (kq2014135) and Hunan Provincial Natural Science Foundation of China (2021JJ40788). Yixiong Liang is partially supported by the Hunan Provincial Science and Technology Innovation Leading Plan (No.2020GK2019).

7. REFERENCES

[1] Richard Zhang, Phillip Isola, and Alexei A Efros, "Colorful image colorization," in *ECCV*. Springer, 2016, pp. 649–666. 1, 2, 1

[2] Mehdi Noroozi and Paolo Favaro, "Unsupervised learning of visual representations by solving jigsaw puzzles," in *ECCV*. Springer, 2016, pp. 69–84. 1, 2, 1

[3] Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross Girshick, "Momentum contrast for unsupervised visual representation learning," in *CVPR*, 2020, pp. 9729–9738. 1, 2, 1

[4] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton, "A simple framework for contrastive learning of visual representations," in *ICML*, 2020, pp. 1597–1607. 1, 2, 1

[5] Pengguang Chen, Shu Liu, and Jiaya Jia, "Jigsaw clustering for unsupervised visual representation learning," in *CVPR*, 2021, pp. 11526–11535. 1, 2, 3, 3, 4, 4, 1, 4, 2, 3

[6] Deepak Pathak, Philipp Krahenbuhl, Jeff Donahue, Trevor Darrell, and Alexei A Efros, "Context encoders: Feature learning by inpainting," in *CVPR*, 2016, pp. 2536–2544. 2

[7] Richard Zhang, Phillip Isola, and Alexei A Efros, "Split-brain autoencoders: Unsupervised learning by cross-channel prediction," in *CVPR*, 2017, pp. 1058–1067. 2

[8] Carl Doersch, Abhinav Gupta, and Alexei A Efros, "Unsupervised visual representation learning by context prediction," in *ICCV*, 2015, pp. 1422–1430. 2

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

[10] Jean-Bastien Grill, Florian Strub, Florent Altché, et al., "Bootstrap your own latent: A new approach to self-supervised learning," in *NeurIPS*, 2020. 2

[11] Xinlei Chen and Kaiming He, "Exploring simple siamese representation learning," in *CVPR*, 2021, pp. 15750–15758. 2

[12] Byungseok Roh, Wuhyun Shin, Ildoo Kim, and Sungwoong Kim, "Spatially consistent representation learning," in *CVPR*, 2021, pp. 1144–1153. 2

[13] Mathilde Caron, Piotr Bojanowski, Armand Joulin, and Matthijs Douze, "Deep clustering for unsupervised learning of visual features," in *ECCV*, 2018, pp. 132–149. 1

[14] Zhirong Wu, Yuanjun Xiong, Stella X Yu, and Dahua Lin, "Unsupervised feature learning via non-parametric instance discrimination," in *CVPR*, 2018, pp. 3733–3742. 1, 2

[15] Jeff Donahue and Karen Simonyan, "Large scale adversarial representation learning," *NeurIPS*, vol. 32, pp. 10542–10552, 2019. 1

[16] Chengxu Zhuang, Alex Lin Zhai, and Daniel Yamins, "Local aggregation for unsupervised learning of visual embeddings," in *ICCV*, 2019, pp. 6002–6012. 1

[17] Yuki M. Asano, Christian Rupprecht, and Andrea Vedaldi, "Self-labelling via simultaneous clustering and representation learning," in *ICLR*, 2020. 1

[18] Olivier Henaff, "Data-efficient image recognition with contrastive predictive coding," in *ICML*, 2020, pp. 4182–4192. 1

[19] Ishan Misra and Laurens van der Maaten, "Self-supervised learning of pretext-invariant representations," in *CVPR*, 2020, pp. 6707–6717. 1, 2

[20] Junnan Li, Pan Zhou, Caiming Xiong, and Steven C.H. Hoi, "Prototypical contrastive learning of unsupervised representations," in *ICLR*, 2021. 1

[21] Xinlei Chen, Haoqi Fan, Ross Girshick, and Kaiming He, "Improved baselines with momentum contrastive learning," *arXiv preprint arXiv:2003.04297*, 2020. 4, 1, 2, 3

[22] Dong-Hyun Lee et al., "Pseudo-label: The simple and efficient semi-supervised learning method for deep neural networks," in *Workshop on challenges in representation learning, ICML*, 2013, vol. 3, p. 896. 2

[23] Yves Grandvalet, Yoshua Bengio, et al., "Semi-supervised learning by entropy minimization.," *CAP*, vol. 367, pp. 281–296, 2005. 2

[24] Xiaohua Zhai, Avital Oliver, Alexander Kolesnikov, and Lucas Beyer, "S4l: Self-supervised semi-supervised learning," in *ICCV*, 2019, pp. 1476–1485. 2

[25] Qizhe Xie, Zihang Dai, Eduard Hovy, Thang Luong, and Quoc Le, "Unsupervised data augmentation for consistency training," *NeurIPS*, vol. 33, 2020. 4, 2