2022 IEEE International Conference on Image Processing (ICIP) | 978-1-6654-9620-9/22/$31.00 ©2022 IEEE | DOI: 10.1109/ICIP46576.2022.9897945

SELF-SUPERVISED LEARNING FOR TEXTURE CLASSIFICATION USING LIMITED LABELED DATA

Sahana M. Prabhu, Jitendra Y. Katta, Amit A. Kale

Bosch Corporate Research, Bosch Global Software Technologies,
Bangalore, India

sahana.muraleedharaprabhu@in.bosch.com

ABSTRACT

Fine-grained texture classification differentiates between similar materials. When there are large unlabeled datasets available, representation learning is useful to distinguish between classes. In this paper, we show that harnessing contrastive self-supervised learning (SSL) for visual representations leads to performance gains for fine-grained texture classification. We demonstrate that, in the absence of sufficient labeled training data, SSL pre-training provides better representation for classification, when compared to supervised methods. We propose a novel pretext task, part-to-whole, in which we use the property of textures that a randomly cropped patch is similar in structure to the whole image. We also propose the usage of representations that are tapped from multiple layers of a convolutional neural network (CNN) and show the effectiveness of combining high-level and low-level features in improving discriminability. We present extensive experiments on the ground-terrain outdoor scenes (GTOS) dataset and show that multi-layer global average pooling (multi-GAP) representations from EfficientNet-B4 model trained using part-to-whole pretext task, beats the current state-of-the-art (SOTA) methods on single-view material classification in limited labeled data settings.

Index Terms— Texture Classification, Fine-grained, Deep Learning, Convolutional Neural Network, Self-supervised learning

1. INTRODUCTION

Texture classification is used to distinguish between multiple classes of textures, and is a classic problem in image processing. It is useful for a variety of real-world applications, such as industrial inspection, material selection, counterfeit detection, satellite imagery and microscopic image classification. Fine-grained classification of texture depends more on subtle local differences rather than global object characteristics. Rather than using pre-trained deep learning models such as ImageNet [1], which has object classes, we need to model subtle differences between materials. Texture images are characterized by patterns of local spatial distribution, and this information can be exploited to deal with this problem more efficiently [2].

In situations where large amount of labeled data is available, fine-tuning of ImageNet pretrained models [1] which provide good mid-level texture features work well. Recently self-supervised learning methods, especially a Simple framework for contrastive learning of visual representations (SimCLR) [3], have been proposed which are capable of representation learning for limited training data. Previous works in literature have not particularly investigated its usefulness for texture classification [4], which is a gap we attempt to fill. The pretext task in SSL influences learning of an intermediate representation that is beneficial for the end task. We propose changes to the SimCLR framework in the form of a pre-text task that is suitable for texture modeling. Each image undergoes two independent augmentations and passed through a base encoder network and a projection head for training. The training process uses contrastive loss function to maximize agreement between the two augmentations. After training is completed, feature representations for downstream task, which is texture classification, are obtained from the encoder.

Different layers of a CNN capture various image semantics, using both low-level and high-level features. CNNs trained for image classification tasks utilize features from final or higher layers that capture semantics suitable for category-level classification. Local characteristics of textures are preserved at lower layers of a CNN [5]. We devise a novel set of features that are capable of capturing low-level as well as high-level concepts in images, particularly suitable for representing textures. These representations are tapped from multiple convolutional layers of a CNN and concatenated. We show that such an approach is more powerful for texture classification. The workflow includes training a model on SSL pretext task with unlabeled images, from which features are extracted (EfficientNet-B4 [6] has the best accuracy), and logistic regression (LR) is used as a classifier.

The contributions of this paper for fine-grained texture classification are as follows:
i) We introduce a novel pretext task, part-to-whole, which exploits the similarity between local patch and global pat-

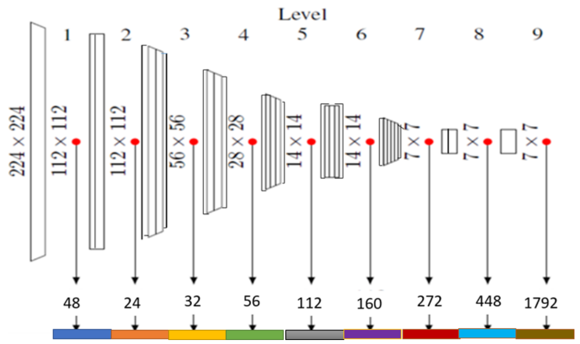

Fig. 1. Formation of Multi-GAP features by concatenating GAP features from convolution layers of EfficientNet-B4.

tern of texture. This hypothesis leads to gain in performance especially for limited labeled training data. We explore the effectiveness of jigsaw [7] for texture recognition and compare it with our proposed part-to-whole scheme. SimCLR training [3] on full training set without labels is used for feature extraction and then a classifier is trained with 50% of labels, and compared with state-of-the-art (SOTA) [8] [9].
ii) We show the benefits of combining multi-layer features for texture representation. Specifically we show the effectiveness of Multi-layer Global Average pooling (multi-GAP) for this downstream task.
Various architectures are compared: VGG-16, ResNet-18, ResNet-50, DenseNet and EfficientNet, for their performance in classification of the GTOS dataset.

2. RELATED WORK

Surveys of texture classification methods were presented in [2] and [4]. An approach for material classification using differential images from the GTOS dataset [10], called Differential Angular Imaging Network (DAIN), extracts the characteristics of materials encoded in the angular and spatial gradients [10]. Deep Encoding Pooling Network (DEP) for classification of GTOS dataset was proposed in [11], which learns a parametric distribution in feature space in a fully supervised manner. Since texture analysis requires features that describe the local spatial distribution, histogram layer features which were estimated during backpropagation were proposed in [8]. Material recognition using texture-encoded angular network (TEAN) that combined deep encoding pooling of RGB information and differential angular images for GTOS was presented in [9]. We compare our proposed method with two SOTA papers, viz., [8] and [9]. Texture representation using SSL in particular is significant as only fine-grained objects categories have been explored using SSL so far [12]. Self-supervised approach was shown to improve the performance of zero-shot learning for the case of fine-grained classification of similar objects in [13]. Pretext task called image enhanced rotation prediction (IE-Rot) for SSL was proposed in [14].

3. METHODOLOGY

The workflow of the proposed method is described in this section. The dataset without labels is input to the SSL framework. Part-to-whole pretext task with contrastive loss is used for SSL training. Multi-GAP features extracted from SSL model are used to train a logistic regression (LR) classifier with 50% of the labeled data, and tested on an unseen set.

3.1. Dataset

Ground terrain outdoor scenes (GTOS) is a 40-class texture dataset of above 30,000 images, which is split into 5-fold training and testing sets [10] and captured under 18 viewing angles and different illumination outdoor conditions. The classes are fine-grained as some of them look similar. There is also intra-class variability, since samples for same class have different colour. Although other two datasets, viz. MINC-2500 and DTD are also presented in [8], here we have considered only GTOS, since MINC-2500 is not a texture dataset, and DTD has very less data.

3.2. Multi-GAP features

In situations where large amount of labeled data is available, fine-tuning of ImageNet pretrained models [1] which provide good mid-level texture features work well. In most object classification tasks, the last convolutional layer of the CNN is transformed into a fully connected layer, and then used as a feature. While this is sufficient for high level semantic content viz. objects, we find that it does not contain sufficient information from lower layers of the network such as edges, patterns which may be beneficial for texture modeling. Multi-layer features are tapped at every convolutional block and concatenated to represent fine-grained characteristics using two ways: (i) multi-layer maximal activation of convolutions (multi-MAC) and (ii) multi-layer global average pooling (multi-GAP). MAC representations [15] are formed by taking maximum local response of each convolutional filter in a given layer. It encodes the highlights of objects or parts in images that contribute to image similarity. GAP is the average of each particular filter response in a specific layer, and captures continuous uniform patterns which is characteristic of textures. Multi-GAP is formed by concatenating GAP features at multiple convolutional layers which is representative of both low-level and high-level texture information. Multi-GAP for EfficientNet-B4 is shown in Figure 1, where 9 convolutional blocks are tapped and the dimensions of layers, resolution and channels are presented. For instance, convolutional block 5 has resolution 14x14, with number of channels 115 and 5 layers.

Each input image I is encoded in each convolutional layer by the filter responses to that image. A layer with N_p filters yields N_p feature maps, each of size M_p, where M_p is the height times the width of the feature map. The filter outputs in a layer p are stored in matrix:

$$F^p \in R^{N_p M_p} \tag{1}$$

where F^p_{ij} is the activation of i^{th} filter at position j in layer p. For each filter i, in layer p, the GAP feature at a particular layer p is computed as

$$G^p = [g^p_1 ... g^p_i ... g^p_{N_p}]^T, \text{where} \quad g^p_i = \frac{1}{M_p} \sum_{j=1}^{M_p} F^p_{ij} \tag{2}$$

Similarly, GAP features from multiple convolutional layers are concatenated together to form multi-GAP.

We consider several CNN architectures: (i) VGG-16: baseline pre-trained model used for image classification [16]; (ii) DenseNet: it ensures maximum information flow through feature reuse connections that need fewer parameters [17]; (iii) Residual network (ResNet): uses residual skip connections to overcome the vanishing gradient problem [18]; (iv) EfficientNet: lighter network (version B4 is chosen for optimum performance) which efficiently uses compound scaling approach for height, width and depth [6].

3.3. Pretext tasks

A texture consists of continuous uniform patterns which has similarities between its parts and the whole image. We use this property to propose a new pretext task, part-to-whole, for SSL which exploits the similarity between local and global patterns between patch and the whole image.
Part-to-whole: This scheme is shown in Figure 2. For the first image input, we double the size of image I (from 240x240 for $I(x, y)$ of the original size to $I(2x, 2y)$ of size 480x480) and then randomly crop 96x96 patch, and apply all the default SimCLR [3] augmentations. For the second input, we simply resize the original image of 240x240 to 96x96, and apply augmentations. The two inputs are:

$$I(2x - c : 2x + c, 2y - c : 2y + c) \quad \text{and} \quad I(rx, ry) \tag{3}$$

where the scaling parameter is r=1/2.5 and cropping parameter is c=48.

We also compare our pretext task with the jigsaw pretext task from [7]. Jigsaw pretext task [7] uses randomly rearranged patches of the texture image which retains similar structure. For the first input, we resize to 96x96, and apply all the default SimCLR [3] augmentations. For the second input, resize to 126x126, and independently apply all the augmentations, then divide the image into a grid as shown in Figure 3, of 42x42 each and then discard 5 pixels at the borders to get a center crop of 32x32 from each box of the grid (to break continuity of the image and prevent the model from finding shortcuts during training). These nine patches are randomly shuffled and rearranged into a 96x96 grid.

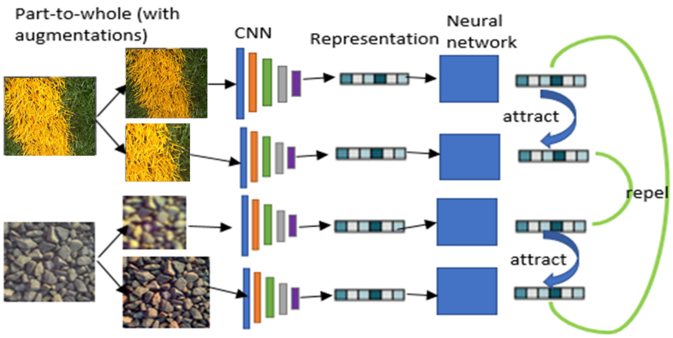

Fig. 2. Part-to-whole SimCLR (with augmentations)

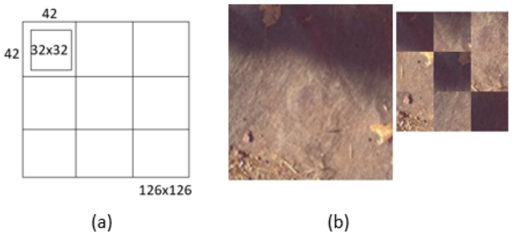

Fig. 3. (a) Jigsaw (b) Image and its jigsaw rearrangement.

3.4. SimCLR

SimCLR is a simple framework for contrastive learning of visual representations [3]. It is an aligning approach which learns representations using objective functions similar to those used for supervised learning, but trains networks to perform tasks where both the inputs and labels are derived from an unlabeled dataset. It maximizes the agreement between differently augmented images of the same sample using contrastive loss. Learnable non-linear transformation and representation learning with contrastive cross-entropy loss with normalized embeddings are advantages of SimCLR [3] for fine-grained classification.

We have used the architectures mentioned in Section 3.1 for SimCLR (using pre-training with ImageNet) for two pretext tasks from Section 3.2. All the default data augmentations in [3] including random crop (with resize and flip), color jitter distortion, and Gaussian blur are used for each image pair separately. Two separate data augmentation operators are sampled from the same family of augmentations and applied to each data example to obtain two correlated views. For the part-to-whole pretext task, one view is a random crop of the original image and then the data augmentations are applied separately, and the pipeline of SimCLR is detailed in Figure

2. For the jigsaw pretext task, one view is the original image and the other view is the jigsaw re-arranged image as shown in Figure 3. A base encoder network and a projection head are trained to maximize agreement using a contrastive loss.

Normalized temperature-scaled cross entropy loss (NT-Xent) is the type of contrastive loss used in SimCLR. Let the distance metric $sim(\mathbf{u}, \mathbf{v}) = \mathbf{u}^T\mathbf{v}/||\mathbf{u}||||\mathbf{v}||$ denote the dot product or cosine similarity. Contrastive loss is defined for positive pair: image x_i and its patch x_j, where the contrastive prediction task aims to identify x_j in x_k for a given x_i.

$$l_{i,j} = -\log \frac{\exp(sim(x_i, x_j)/\tau)}{\Sigma_{k=1}^{2N} 1_{[k \neq i]} \exp(sim(x_i, x_k)/\tau)} \qquad (4)$$

Each batch has N positive examples and 2N pairs of augmentations; 2(N-1) negative pairs per each positive pair, and τ is temperature parameter.

4. EXPERIMENTAL RESULTS

Quantitative results of the methods compared for GTOS dataset are presented in Table 1. There are 5-fold training and testing sets in GTOS, and we have shown the mean accuracy (with standard deviation) across five test sets, with logistic regression (LR) as the classifier. SimCLR training is useful for cases where there is considerable (possibly unlabeled) training dataset, which is of similar distribution as the test set. Various architctures which are pre-trained on ImageNet [1] are compared (specified as "pre" in Table 1), viz., VGG-16, DenseNet, EfficientNet-B4, ResNet-18 and ResNet-50. We have compared multi-MAC and multi-GAP for VGG-16 and EfficientNet-B4, respectively. In ResNet, features are combined through summation before they are passed into a layer; hence it is more difficult to lend itself to multi-MAC and multi-GAP features, and therefore is left as future work. EfficientNet-B4 ("EffNet" in Table 1) turns out to be the most suitable architecture, and multi-GAP features have an advantage as the average pooling of intermediate layers' features more accurately represent the local fine-grained texture.

We note that some of the texture classes in GTOS are more homogeneous, viz., we manually inspected each of the classes and selected 17 classes wherein the training samples were more homogeneous. The accuracy of our approach for the chosen 17 classes, using part-to-whole SimCLR without any other default augmentations is 88%, which indicates it is highly effective for homogeneous textures in particular. Other classes are composite textures and shapes.

SimCLR with part-to-whole pretext task and Multi-GAP features yields the best results for the LR classifier trained on 50% and 100% labeled data. For all SimCLR training schemes, the hyperparameters are: batch size=256, number of epochs 300, learning rate = 0.04, temperature = 0.1. We compare with two SOTA papers: (i) histogram-based features for texture classification in [8] and (ii) Deep texture-encoded angular network (Deep-TEAN) [19]. For 100% labeled GTOS training, our proposed method, part-to-whole SimCLR multi-GAP EfficientNet-B4 is similar in accuracy to Deep-TEAN, and outperforms Histogram-based features; whereas for 50% GTOS training, our method outperforms both Deep-TEAN and Histogram-based features.

Table 1. Accuracy for GTOS: mean and standard deviation across 5 test sets. 50% and 100% indicate the amount of labeled data used for training LR classifier. "EffNet" is EfficientNet-B4 and "pre" is model pre-trained on ImageNet.

Method	Train	Accuracy
VGG-16	pre	73 ± 1.7
DenseNet	pre	73.57 ± 2.3
EffNet	pre	74.79 ± 3.2
VGG-16 Multi-MAC	pre	77.32 ± 3.9
EffNet Multi-GAP	pre	80.76 ± 2.1
EffNet Multi-MAC	pre	79.36 ± 1.8
MobileNet-V2 [9]	100%	80.4 ± 3.2
SimCLR Resnet-18	100%	77.16 ± 2.8
SimCLR Resnet-50	100%	79.78 ± 2.5
SimCLR VGG-16 Multi-MAC	100%	77.84 ± 3.4
SimCLR EffNet	100%	79.86 ± 1.8
SimCLR EffNet Multi-GAP	100%	82.7 ± 1.6
Jigsaw SimCLR EffNet Multi-GAP	100%	83.1 ± 1.4
Histogram-based features [8]	100%	82.5 ± 1.7
Deep-TEAN [9]	100%	84.7 ± 1.7
Part-to-whole SimCLR EffNet Multi-GAP	100%	$\mathbf{84.5 \pm 1.4}$
SimCLR EffNet Multi-GAP	50%	80.72 ± 2.5
Jigsaw SimCLR EffNet	50%	82.3 ± 2.1
Histogram-based features (from [8])	50%	81.08 ± 2.3
Deep-TEAN (from [9])	50%	82.1 ± 1.9
Part-to-whole SimCLR EffNet Multi-GAP	50%	$\mathbf{83.9 \pm 1.8}$

5. DISCUSSION AND CONCLUSIONS

We have presented experiments on GTOS dataset and compared multi-GAP and multi-MAC features from various deep architectures. The proposed part-to-whole pretext task for SSL using SimCLR with EfficientNet-B4 multi-GAP provides gains for fine-grained classification, especially when limited labeled data is available. By comparing a random crop part with whole image, the global to local view is used for sampling contrastive prediciton tasks. Multi-GAP includes first few layers which represent low-level properties that are useful for texture cues, while later layers have high-level shape cues. The purpose of using only 50% downstream training labels is to show that SSL is beneficial in harnessing unlabeled data, such that we can train a classifier with less number of labels. Representational learning distinguishes between fine-grained textures, which do not have shape cues to aid classification. For future work, we shall explore suitable pretext tasks for other applications.

6. REFERENCES

[1] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei, "Imagenet: A large-scale hierarchical image database," in *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, 2009, pp. 248–255.

[2] Paulo Cavalin and Luiz S Oliveira, "A review of texture classification methods and databases," in *SIBGRAPI Conf. Graphics, Patterns and Images Tutorials (SIBGRAPI-T)*. IEEE, 2017, pp. 1–8.

[3] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton, "A simple framework for contrastive learning of visual representations," *arXiv preprint arXiv:2002.05709*, 2020.

[4] Li Liu, Jie Chen, Paul Fieguth, Guoying Zhao, Rama Chellappa, and Matti Pietikäinen, "From bow to cnn: Two decades of texture representation for texture classification," *International Journal of Computer Vision*, vol. 127, no. 1, pp. 74–109, 2019.

[5] Joe Yue-Hei Ng, Fan Yang, and Larry S Davis, "Exploiting local features from deep networks for image retrieval," in *Proc. IEEE Conf. Computer Vision and Pattern Recognition workshops*, 2015, pp. 53–61.

[6] Mingxing Tan and Quoc Le, "Efficientnet: Rethinking model scaling for convolutional neural networks," in *International Conference on Machine Learning*, 2019, pp. 6105–6114.

[7] Mehdi Noroozi and Paolo Favaro, "Unsupervised learning of visual representations by solving jigsaw puzzles," in *European conference on computer vision*. Springer, 2016, pp. 69–84.

[8] Joshua Peeples, Weihuang Xu, and Alina Zare, "Histogram layers for texture analysis," *IEEE Transactions on Artificial Intelligence*, 2021.

[9] Jia Xue, Hang Zhang, Ko Nishino, and Kristin Dana, "Differential viewpoints for ground terrain material recognition," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2020.

[10] Jia Xue, Hang Zhang, Kristin Dana, and Ko Nishino, "Differential angular imaging for material recognition," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 764–773.

[11] Jia Xue, Hang Zhang, and Kristin Dana, "Deep texture manifold for ground terrain recognition," in *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, 2018, pp. 558–567.

[12] Xiu-Shen Wei, Yi-Zhe Song, Oisin Mac Aodha, Jianxin Wu, Yuxin Peng, Jinhui Tang, Jian Yang, and Serge Belongie, "Fine-grained image analysis with deep learning: A survey," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2021.

[13] Jiamin Wu, Tianzhu Zhang, Zheng-Jun Zha, Jiebo Luo, Yongdong Zhang, and Feng Wu, "Self-supervised domain-aware generative network for generalized zero-shot learning," in *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, 2020, pp. 12767–12776.

[14] Shinya Yamaguchi, Sekitoshi Kanai, Tetsuya Shioda, and Shoichiro Takeda, "Image enhanced rotation prediction for self-supervised learning," in *IEEE International Conference on Image Processing (ICIP)*, 2021, pp. 489–493.

[15] Giorgos Tolias, Ronan Sicre, and Hervé Jégou, "Particular object retrieval with integral max-pooling of cnn activations," in *International Conference on Learning Representations*, 2016, pp. 1–12.

[16] Karen Simonyan and Andrew Zisserman, "Very deep convolutional networks for large-scale image recognition," *arXiv preprint arXiv:1409.1556*, 2014.

[17] Gao Huang, Zhuang Liu, Laurens Van Der Maaten, and Kilian Q Weinberger, "Densely connected convolutional networks," in *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, 2017, pp. 4700–4708.

[18] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, 2016, pp. 770–778.

[19] Hang Zhang, Jia Xue, and Kristin Dana, "Deep ten: Texture encoding network," in *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, 2017, pp. 708–717.