Neurocomputing 338 (2019) 191–206

Contents lists available at ScienceDirect

Neurocomputing

journal homepage: www.elsevier.com/locate/neucom

A novel scene classification model combining ResNet based transfer learning and data augmentation with a filter

Shaopeng Liu, Guohui Tian*, Yuan Xu

School of Control Science and Engineering, Shandong University, Jinan, China

ARTICLE INFO

Article history:
Received 11 September 2018
Revised 7 December 2018
Accepted 30 January 2019
Available online 7 February 2019

Communicated by Dr Li Sheng

Keywords:
Scene classification
Transfer learning
ResNet
Data augmentation
CNN

ABSTRACT

Scene classification is a significant aspect of computer vision. Convolutional neural networks (CNNs), a development of deep learning, are a well-understood tool for image classification. But training CNNs requires large-scale datasets. Transfer learning addresses this problem and produces a solution for small-scale datasets. Because scene image classification is more complex than common image classification. We propose a novel ResNet based transfer learning model utilizing multi-layer feature fusion, taking full advantage of interlayer discriminating features and fusing them for classification by softmax regression. In addition, a novel data augmentation method with a filter useful for small-scale datasets is presented. New image patches are generated by sliding block cropping of a raw image, which are then filtered to insure that the new images sufficiently represent the original categorization. Our new ResNet based transfer learning model with enhanced data augmentation is evaluated on six benchmark scene datasets (LF, OT, FP, LS, MIT67, SUN397). Extensive experimental results show that on the six datasets our method obtains better accuracy than other state-of-the-art models.

1. Introduction

Scene classification, an important aspect of computer vision, categorizes scene images into a discrete set of semantic classes by images content, which is crucial to image browsing, retrieval, understanding and so on [1]. Various approaches to scene classification have been proposed. A multi-spectral scale-invariant feature transform (SIFT) method for scene recognition was presented, combining SIFT with a classifier [2]. Lazebnik proposed spatial pyramid matching (SPM) for scene classification based on bag-of-feature (BOF) model [3]. Besides the methods mentioned above utilizing handcrafted features, the deep convolutional neural network (CNN) approach has made significant progress especially in the domain of image classification. With the appearance of large-scale category-level datasets such as ImageNet [4], CNNs became able to provide a high performance tool for scene recognition. In 2012, the AlexNet model got the first prize in the ImageNet competition [5]. In the subsequent years of the ILSVRC competition a series of CNN models were created, VGG [6], ResNet [7], GoogleNet [8] and Inception [9], constantly boosting the accuracy of image classification. However CNNs have some shortcomings: (1) CNNs need large-scale datasets for training and some scene datasets contain insufficient data. (2) CNNs, especially deep CNNs, contain millions of parameters, which require to be trained by powerful GPUs to expedite the training process. The training is therefore time-consuming and expensive. (3) Using deep CNN models to train small-scale datasets can easily result in overfitting even with the use of preventive techniques such as “dropout [10]”.

Transfer learning offers a solution to the problems of training deep CNNs on small-scale datasets. Transfer learning improves a learner in one domain by transferring information from a related domain [11]. The common transfer learning approach with deep CNNs is to leverage a pre-trained CNN model and change the classifier layer to fine-tune the weight parameters on the target dataset. This technique can accelerate training and improve the accuracy, but it has weaknesses. The standard transfer method only takes activations of the fully connected (FC) layer as the image representation. This loses the object description detail contained in the features of the convolutional layers. The relationship between objects and scenes is especially relevant and information on objects can promote scene recognition [12]. In a CNN structure the feature data varies by layer. Compared with low-layer features, high-layer features have more semantic information and less detailed information. On the contrary, low-layer features contain more detailed information but suffer from the problem of background clutter and semantic ambiguity [13]. Therefore the features from different layers should be utilized when exploring CNNs scene classification potential.

* Corresponding author.
E-mail address: g.h.tian@sdu.edu.cn (G. Tian).

https://doi.org/10.1016/j.neucom.2019.01.090
0925-2312/

Table 1
Different types of transfer learning.

Type	Method	Source data	Target data	Application
Instance-based	CP-MDA [23]	Labeled	Limited labels	Signal processing
	2SW-MDA [23]	Labeled	Unlabeled	Signal processing
Asymmetric feature-based	JDA [24]	Labeled	Unlabeled	Image classification
Symmetric feature-based	TCA [25]	Labeled	Unlabeled	Text classification
Relational-based	RAP [26]	Labeled	Unlabeled	Sentiment lexicon extraction
Parameter-based	MMKT [27]	Labeled	Limited labels	Image classification

Data augmentation is another way to improve the training effect. Image transformation methods are used to created artificial data based on the raw datasets. Rescaling, rotation, flipping and caffe embedded cropping for image patches can be applied to full-size images to add to the dataset [14]. Montserrat employed color changes and blended object images into background images in random positions to generate synthetic data [15]. Although image transformation based data augmentation can increase accuracy, these methods are most effective on simple datasets such as single object or face. When it comes to complicated scene images, which contain multiple objects and complicated backgrounds, improvements by this technique are not very obvious, which is proved in this paper.

This paper presents 2 solutions to these problems. On the one hand, a transfer learning model is proposed based on a pre-trained 18-layer ResNet with feature fusion. Global average pooling (GAP) is utilized to extract features from the intermediate layers and the features are fused linearly and classified by softmax regression. On the other hand, we focus on a novel data augmentation method based on image patches generated by a sliding block and filtered with a pre-trained 152-layer ResNet. With these two improvements the accuracy of scene recognition is increased and to some degree overfitting is weakened. The shortcoming is that the transfer learning model is more structurally complicated than the ordinary ResNet. Further, our data augmentation method has one more step than the usual methods, data selection, but improves results.

The main contributions of this work include the following:

(1) A novel transfer learning model based on pre-trained ResNet is proposed with multi-layer feature fusion for scene classification.
(2) An effective data augmentation method of image patch creation and data filtration is applied on scene datasets to generate sufficient and suitable data for training.
(3) A solution combining the transfer learning model and the data augmentation method is tested on six benchmark scene datasets, (LF [16], OT [17], FP [18], LS [19], MIT67 [20], SUN397 [21]), yielding state-of-the-art results.

The rest of this paper is organized as follows: Section 2 reviews the related studies of transfer learning, CNN models on image classification, scene classification and data augmentation. In Section 3 our transfer model and data augmentation method is described. These are extensively evaluated in Section 4. Section 5 summarizes the evaluations and explains their significance.

2. Related works

2.1. Transfer learning

Transfer learning is defined to be a process which, given a source domain D_S with a corresponding source task T_S and a target domain D_T with a corresponding task T_T, boosts the target predictive function $f_T(\cdot)$ by leveraging the correlative information from D_S and T_S, where $D_S \neq T_S$ or $T_S \neq T_T$ [22]. Different types of transfer learning are listed in Table 1. Chattopadhyay proposed two separate solutions, the conditional probability based multi-source domain adaptation (CP-MDA) approach and the two stage weighting framework for multi-source domain adaptation (2SW-MDA). Both use multiple labeled source domains and are examples of instance-based transfer learning [23]. Another transfer learning approach is based on features, both asymmetric and and symmetric features. Long's paper put forward the Joint Distribution Adaptation (JDA) method, a novel transfer learning approach based on asymmetric feature transfer learning which aimed to jointly adapt both the marginal distribution and conditional distribution with principal component analysis (PCA). It constructs new feature representations which are effective and robust in yielding substantial distribution differences [24]. The study of symmetric feature transfer learning by Pan proposed finding suitable representations through a new learning method, transfer component analysis (TCA), which performs domain adaptation with no need for labeled target data [25]. The research on relational-based transfer learning in a paper by Li presents a domain adaptation framework for semantic and topic lexicon co-extraction in a domain of interest which has no labeled data but where there is lots of labeled data in a related domain [26]. A relational model is built between the source and target domains by learning the syntax and grammar of the source. For parameter-based transfer learning a SVM-based model adaptation algorithm is proposed by Tommasi et al. [27] which selects and weights appropriate prior knowledge of different categories.

The research mentioned above provides a brief overview of transfer learning, while not all of them are employed in image field. There is also some research which employs transfer learning in image analysis. Shin made use of fine-tuned (supervised) CNN models pre-trained from a natural image dataset of medical images, solving computer-aided detection problems in medicine [28]. A paper by Lei leveraged a cross-modal transfer learning strategy to fine-tune a residual network of 50 layers to classify HEp-2 cell images [29]. Han applied an RGB-based deep neural network structure to a depth view and fused the deep representations of both views automatically for salient detection [30]. Transfer learning can be utilized on not only medical images or salient detection but also scene classification. In this paper we focus on the parameter-transfer of pre-trained model and then redesign the structure of the transfer model based on the parameter-transfer for scene classification.

2.2. CNN models on image classification

In 1989 Yann designed a multi-layer neural network based on back propagation algorithm (BP) to train hand-written numeral recognition, first proposing the conception of CNN [31]. In 1998 a paper by Yann presented the first formal CNN model, LeNet-5, expounding three important thoughts: local receptive fields, shared weights and sub-sampling [32]. However, CNN did not develop rapidly on account of dataset scale and the hardware computing power limitations of the time. After that, with the continued development of computer hardware, especially GPU acceleration techniques and the appearance of large-scale datasets

such as ImageNet [5] and PASCAL VOC [33], CNN has entered a period of rapid development. A series of excellent CNN models has emerged significant progress in the image classification field. AlexNet by Krizhevsky obtained top-1 in the ImageNet LSVRC-2010 contest, which replaced the traditional activation function *tanh* with the non-linear activation function *Relu* and trained data on multiple GPUs simultaneously [5]. With an increase in convolutional network depth and an architecture of very small (3×3) convolution filters the VGG model secured first and the second places in the localisation and classification tracks, respectively, in the ImageNet Challenge 2014 [6].

Network depth is of crucial importance in image classification, although stacking more conventional layers to increase depth can easily result in the notorious problem of vanishing gradients. To address this problem ResNet presented a residual block with a shortcut connection, which solved the problem of vanishing gradients and accelerated training, obtaining first place at ILSVRC2015 [7]. A ResNet model with a depth of up to 152-layers, although 8 times deeper than a 19-layer VGG model, has lower complexity and better accuracy. For this reason we use the ResNet model as this paper's basal network.

2.3. Scene classification

Scene classification is an indispensable but challenging technique in computer vision. Various methods for scene classification have been proposed which utilize different level features in scene classification. A paper by Zhou presented an approach for scene classification based on a visual descriptor called GBPWHGO, which retained the advantages of both low-level feature retention and semantic modeling strategies, while avoiding the weaknesses of these two strategies [34]. The research, based on topic features, which were defined as the thematic representation of images constructed by the latent variables of latent dirichlet allocation (LDA), addressed image scene classification [35]. Combining different level features is a promising approach. Feature combination based on "bag of features" approach could improve the accuracy of scene classification. Feature combination could extract more effective information describing characteristics of scene images independent of the influence of illumination, rotation and scale [36]. Therefore feature extraction is a key element in scene recognition. The studies mentioned above are mainly based on handcrafted features, which have limitation on image representation. The development of CNN has made features based on convolution popular with powerful representation capacity. A convolutional feature extraction method, Centered Convolutional Restricted Boltzmann Machines (CCRBM), employable on large-sized scene images, introduced centered factors into the learning strategy to reduce the sources of instabilities [37]. Demonstrating the success of deep network in image classification, a paper by Qi leveraged a pre-trained CNN as a feature extractor to extract mid-level features of scene images based on transfer learning [38] but without factor of high-level features. For convolutional feature fusion, a model named "GoogLeNet based multi-stage feature fusion (G-MS2F)" was designed for scene recognition. It fused the three stages features of the GoogLeNet model [39] while the ResNet model was batter than GoogLeNet.

Comparatively speaking, the convolutional features have a more discriminative representation of scene images than do the handcrafted features. The convolutional features are learning-based features containing abundant semantic information, which are more robust and better accommodate on scene classification. Meanwhile, the low-level convolutional features with descriptive detail can not be ignored. Hence, based on the ResNet model, we take full advantage of the convolutional features at different levels of the network and fuse them for scene classification.

2.4. Data augmentation

Data augmentation is useful for expanding available datasets, particularly small-scale datasets. When classifying images, effective data augmentation can further improve accuracy. Image processing is a simple and rapid way to augment data. Work by Xue applied flipping, rescaling and rotation of images and used caffe embedded cropping to produce image patches to increase dataset size [14]. In [15], synthetic data were generated by image transformations such as color changes, rotations, and blending into background images. Hu presented a well-directed cropping scheme merging three cropping methods and chose the cropping scheme for each based on a weighted-selection approach [40]. All these data augmentation techniques by image transformations form the basis of methods which generate new images from pre-existing raw image data. Beside, some researches leverage the collection of image data from the Internet. The work by Fergus obtained images to argument through Google searching the category name, but without any methods to guarantee search result accuracy [41]. To address this problem Han designed a search result filtering method which augments images retrieved from the web to obtain high-quality images. The weakness of these methods based on the Internet data are a little complicated and time-consuming.

The cited research shows that data augmentation can improve classification results. When it comes to scene images with multiple objects and complicated backgrounds, transformation can change scene composition. With the advent of Generative Adversarial Networks (GANs) [42], artificial data could be generated from the raw data. GANs may generate artificial images which are too small, sizes such as 28×28 or 64×64 pixels, which are too small to contains scenes. Therefore, we utilize a sliding window to generate image patches from the raw image and design a data filter to select suitable image data.

3. Proposed method

The proposed method is comprised of two parts. Part one focuses on constructing a novel transfer learning model with ResNet and features fusion. Part two focuses on dataset augmentation and data filtering. Details of the two parts are presented and described in the following subsections.

3.1. The transfer learning model

3.1.1. Obtaining a pre-trained ResNet

In CNNs, images from different datasets share similar low-level features after the convolution process. It is uneconomical to train from scratch, especially on a small-scale dataset. The training strategy for a new dataset utilizes parameters transferred from pre-trained models, fine-tuned on the basis of the new dataset. So we need pre-trained models to begin our task.

To develop deep learning there are plenty of public toolkits such as Theano [43], Caffe [44], MXNet [45] and so on. These toolkits make it easier for people to research deep learning and accelerate the development of artificial intelligence. MXNet, one of the most popular toolkits, is utilized in this paper. MXNet is a multi-language deep learning (DL) library to ease the development of DL algorithms. Embedded in the host language it blends declarative symbolic expression with imperative tensor computation and offers auto differentiation to derive gradients. MXNet is computation and memory efficient and runs on heterogeneous systems, ranging from mobile devices to distributed GPU clusters [45]. Because training a deep CNN on a large scale dataset is time consuming and can require huge computational resources we adopt pre-trained models from MXNet.

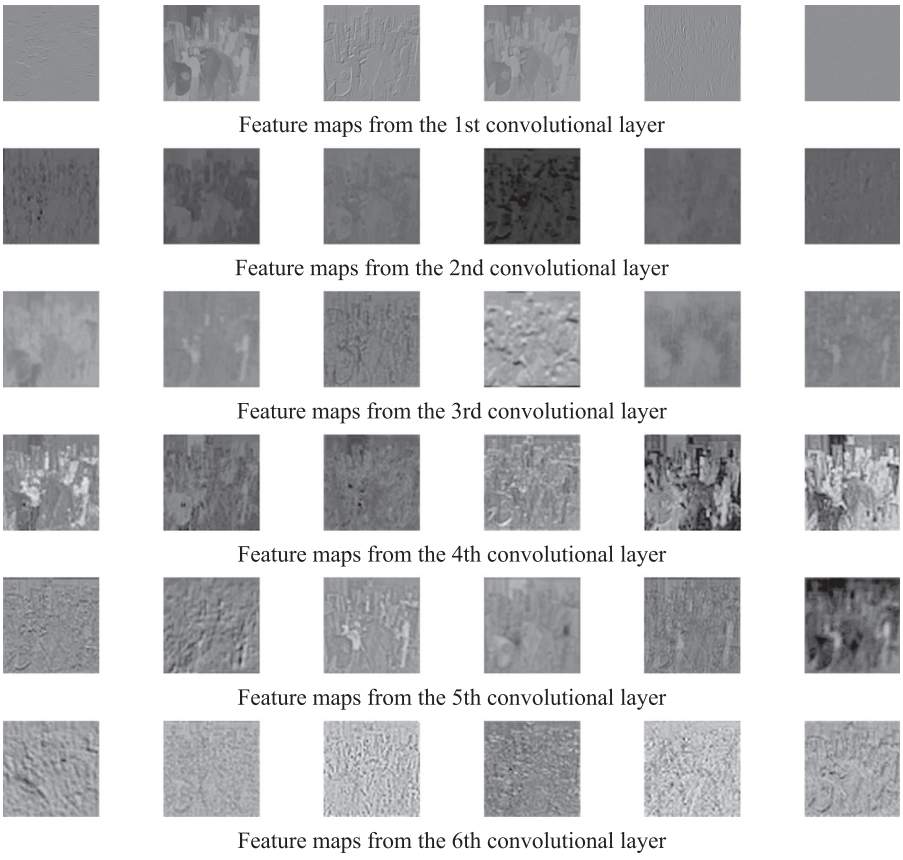

Fig. 1. Feature maps from different ResNet layers.

MXNet provides multiple pre-trained models on various datasets.[1] A 18-layer ResNet trained on ImageNet 1K dataset (about 1.2 M images with 1000 classes [5]) with a top-5 accuracy of 88.66% is employed as the basis of our transfer learning model. Meanwhile, a 152-layer ResNet trained on the dataset combining the ImageNet 11 K dataset with the Place 365 challenge dataset [46] is leveraged to extract image features for our data filter. The former dataset contains 11,221 classes with 11,797,630 images for training and the latter dataset contains 365 classes with 8 millions images. As a result, the combined dataset contains around 20 million images.

Different depth ResNets are selected for different purposes. We use a relatively shallow ResNet (18-layer) as training model since most of scene datasets are small scale, which can easily lead to overfitting if training model is too deep. However, a pre-trained deeper (152-layer) ResNet based on large datasets has capacity to excavate more abundant deep information, which is more suitable for feature extraction.

3.1.2. Construction of the transfer learning model

A pre-trained CNN can be seen as a powerful feature extractor. The features from different layers contain different visual characteristics. Low-layer features contain more descriptive detail but suffer from the problem of background clutter and semantic ambiguity. Contrarily, high-layer features emphasize semantic information but contain less information on detail. With increasing layer depth the generated features are more abstract and more varied, which gives the network an enhanced capacity for learning. Therefore, deep CNNs have enabled breakthroughs in computer vision, including image classification and object detection. However, CNNs cannot get satisfying scene classification results because scene images include multiple objects in complicated spatial relationships in comparison with single object images. And in deep CNN layers detailed features from low-layer are weakened, which makes a CNN no longer discriminative when performing scene classification. Hence it is essential that scene image classification take full advantage of features data in inter-layer of deep CNNs. Feature data from a pre-trained 18-layer ResNet is employed for this purpose. The feature maps extracted from disparate layers in the ResNet are shown in Fig. 1.

Residual blocks with shortcut connections are utilized in the ResNet, gaining higher accuracy by considerably increasing depth without overfitting [7]. There are 8 residual blocks in a 18-layer ResNet. A residual block consists of 2 kinds of convolutional kernels with small sizes 1×1 and 3×3, which contributes to reducing parameter capacity and heightening the expressive ability of the network.

To utilize the excellent performance of ResNet, a novel structure based on ResNet is built with parameters transferred from the pre-trained model. Our method is distinguished from the conventional transfer method which only changes fully-connected layers. We capture the intermediate layer features independently and combine them using a fusion strategy, improving their discriminative property on scenes images. Our transfer learning model is demonstrated in Fig. 2.

The pre-trained 18-layer ResNet contains 8 residual blocks. The feature maps created from the adjacent residual block have certain similarities. It is not wise to extract features from every block, which can increase both the network complexity and computation time. Therefore, we extract features between residual blocks and perform feature extraction only 5 times. With the augmen-

[1] http://data.mxnet.io/models/.

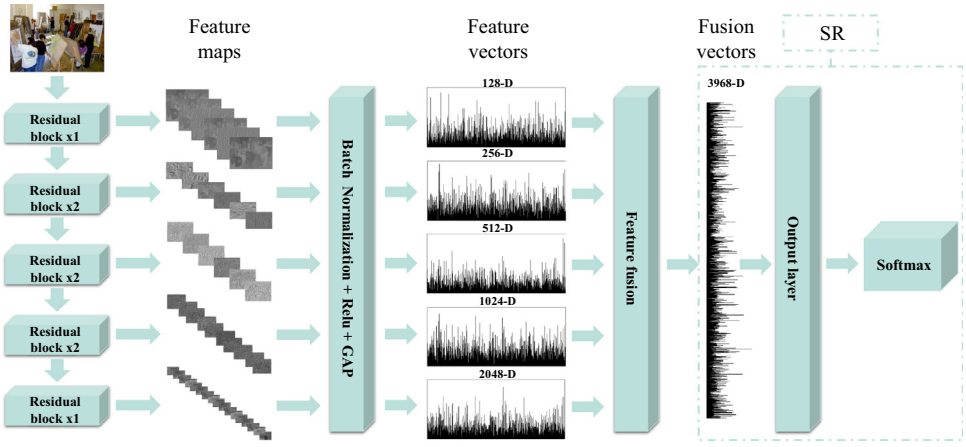

Fig. 2. The proposed transfer learning model based on ResNet and features fusion.

tation of layer depth the dimensionality of the feature map is heightened. If we convolute and pool the feature maps sequentially the dimensionality increases continuously. Inspired by NIN [47], we utilize global average pooling (GAP) to reduce the dimensionality of the feature maps. Before GAP, the feature maps are preprocessed by batch normalization (BN) [48] and an activation function (Relu). For a feature map with d-dimensional $X = (x_{(1)}, x_{(2)}, \ldots, x_{(d)})$, each dimension is normalized by

$$\hat{x}_{(k)} = \frac{x_{(k)} - E[x_{(k)}]}{\sqrt{Var[x_{(k)}]}}. \tag{1}$$

The output $y_{(k)}$ written as:

$$y_{(k)} = \gamma_{(k)}\hat{x}_{(k)} + \beta_{(k)}, \tag{2}$$

introduces a pair of parameters $\gamma_{(k)}$, $\beta_{(k)}$, which can recover the original activations:

$$\gamma_{(k)} = \sqrt{Var[x_{(k)}]}, \beta_k = E[x_{(k)}]. \tag{3}$$

Rectified Linear Units (Relu):

$$\Phi(y_{(k)}) = max(0, y_{(k)}). \tag{4}$$

is leveraged as activation function, adding some nonlinearities to the neural network. After BN and Relu, the dimensionality of the feature maps are decreased by the application of GAP, generating 5 types of feature vectors with 128, 256, 512, 1024, 2048 dimensions, respectively. All feature vectors are linearly integrated to form a 3968-dimension fusion vector $D = (d_{(1)}, d_{(2)}, \ldots, d_{(3968)})$. Each element $d_{(i)} \in D$ does not have equivalent conclusiveness in the final classification. Indeed, an attention mechanism is introduced and every element $d_{(i)} \in D$ is equipped with a weight and a bias adjust for the desired result. Utilized as a multi-layer perceptron (MLP), the 3968-dimension fusion vector becomes the input layer, connecting directly to a k-dimension (k representing the number of categories) output layer, dispensing with hidden layers, which is then followed by a softmax layer. This construction transforms the problem into one of softmax regression (SR) classification. The hypothesis function of SR is formulated as:

$$h_\theta(D^{(i)}) = \begin{bmatrix} p(y^{(i)} = 1|D^{(i)};\theta) \\ p(y^{(i)} = 2|D^{(i)};\theta) \\ \vdots \\ p(y^{(i)} = k|D^{(i)};\theta) \end{bmatrix} = \frac{1}{\sum_{j=1}^{k} e^{\theta_j^T D^{(i)}}} \begin{bmatrix} e^{\theta_1^T D^{(i)}} \\ e^{\theta_2^T D^{(i)}} \\ \vdots \\ e^{\theta_k^T D^{(i)}} \end{bmatrix} \tag{5}$$

where k and θ represent the category and parameters to be trained severally. Using the matrix θ to express $\theta_1, \theta_2, \ldots, \theta_k$:

Table 2
The parameter specification of Nadam (Algorithm 1).

Parameter	Specification
g	Gradient
μ	Decay constant
m	Momentum vector
n	Norm vector
v	Decaying mean

$$\theta = \begin{bmatrix} \theta_1^T \\ \theta_2^T \\ \vdots \\ \theta_k^T \end{bmatrix}. \tag{6}$$

The cost function of SR is written as

$$J(\theta) = -\sum_{i=1}^{m}\sum_{j=1}^{k} sign(y^{(i)} = j) log \frac{e^{\theta_j^T D^{(i)}}}{\sum_{j=1}^{k} e^{\theta_j^T D^{(i)}}} + \frac{\lambda}{2}\sum_{i=1}^{k}\sum_{j=1}^{n}\theta_{ij}^2 \tag{7}$$

where *sign*(*expression is true*) is 1, λ is the parameter of weight decay, and n is the dimension of D. In the end, *Nadam* [49] is utilized to optimize the cost function $J(\theta)$ in Algorithm 1. The parameter

Algorithm 1 Nadam

$g_t \leftarrow \nabla_{\theta_{t-1}} J(\theta_{t-1})$
$\hat{g} \leftarrow \frac{g_t}{1-\prod_{i=1}^{t}\mu_i}$
$m_t \leftarrow \mu m_{t-1} + (1-\mu) g_t$
$\hat{m}_t \leftarrow \frac{m_t}{1-\prod_{i=1}^{t+1}\mu_i}$
$n_t \leftarrow v n_{t-1} + (1-v) {g_t}^2$
$\hat{n}_t \leftarrow \frac{n_t}{1-v^t}$
$\bar{m}_t \leftarrow (1-\mu_t)\hat{g}_t + \mu_{t+1}\hat{m}_t$
$\theta_t \leftarrow \theta_{t-1} - \eta \frac{\bar{m}_t}{\sqrt{\hat{n}_t + \varepsilon}}$

specification of Algorithm 1 is listed in Table 2.

3.2. Data augmentation

3.2.1. Data generation

Data scale has a great influence on the accuracy of a network model. Data augmentation is crucial to ameliorating loss in classification accuracy. Traditional augmentation techniques can improve

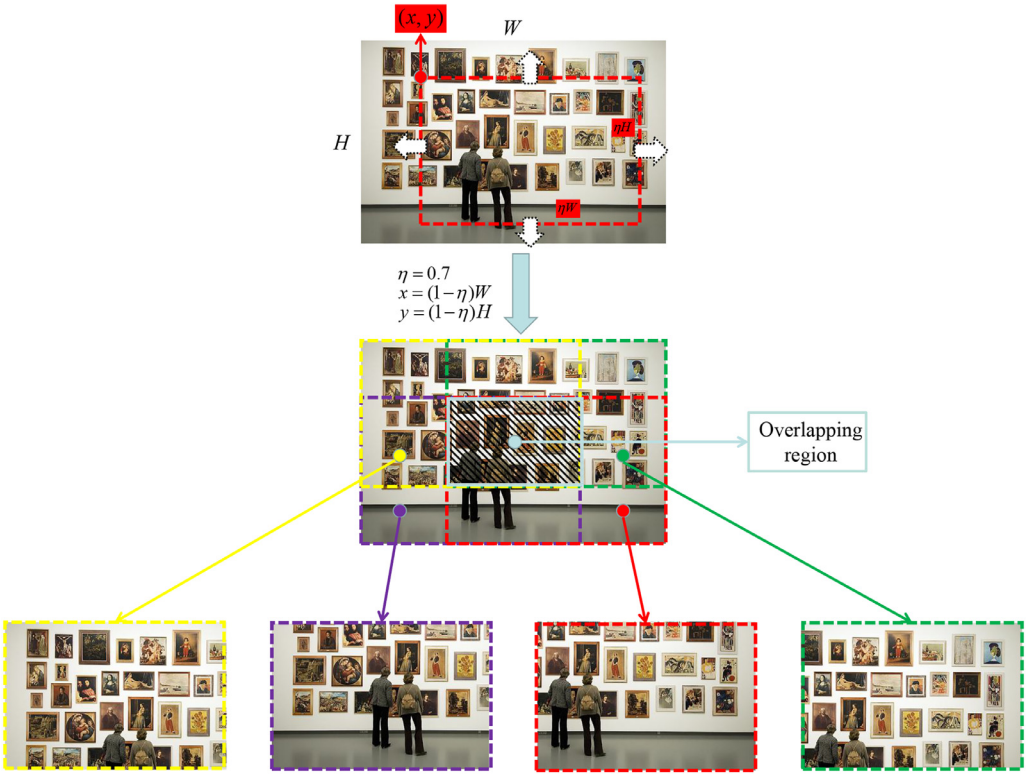

Fig. 3. The workflow of the proposed data augmentation.

the task of simple object recognition but can not satisfy the requirements of complicated scene image analysis. The traditional augmentation methods destroy content information or object spatial information in the image to varying degrees. To cope with this problem we present a novel method that selects a set of regions in the raw image for data augmentation, selections representative of the image's category.

The workflow of the proposed data augmentation method is demonstrated in Fig. 3. A sliding block (the red imaginary line region in Fig. 3) is put on the original image. The width and height of the original image are W and H, respectively. The size of the sliding block is set to ηW and ηH, which is in proportion to the original image for data uniformity. The value of η must be set. If η is too big, close 1, the sliding block region is quite similar to the raw image and there is no discrimination. If η is too small, less than 0.5, the sliding block region can easily be a region of the raw image not representative of the category, severing the region from the semantics of the total image. Therefore η is set to 0.7.

(x, y) is the left vertex coordinate of the sliding block, fixing the sliding block's location within the image. The vertex coordinate (x, y) is set to 4 constant values to minimize the region overlapped: $(0,\ 0)$, $((1-\eta)W, 0)$, $(0, (1-\eta)H)$, and $((1-\eta)W, (1-\eta)H)$. In this way the region overlapped by the 4 image patches is only the central region, with the size of $0.4W \times 0.4H$. Ultimately, 4 new images are generated from an original image – realizing data augmentation.

3.2.2. The data filter

It is reasonable to base the creation of most new images on the proposed data augmentation scheme. However there are some special cases. After data augmentation a few of the generated image patches may no longer be in the original image's category. Some examples are exhibited in Fig. 4. The category of image *a* in Fig. 4 is abbey. Since the most representative region is located in the upper right of the original map, the generated image patch from the bottom left region belongs to new class riverside rather than abbey. Image patches change category as well in images *b* and *c*. In these cases missing content in the incorrect image patches would disturb the network training. Hence a data filter is designed to make the final selection of all new images.

Deciding whether a new image patch falls in the right category is the key. To solve this problem we utilize the image similarity to measure the degree of category relationship between the image patch and the original image. Traditional methods such as HOG [50], SIFT [51], utilize manual features, and these methods are based on low-level features rather than high-level semantic features. With scene images the images from the same class share varying low-level features. In consideration of this condition a pre-trained 152-layer ResNet, named feature network (FN), is employed to extract semantic features for image similarity calculation. The fully connected layer of FN is deliberately removed. After removal the output of FN is a 2048-dimension vector rich in semantic information. The cosine similarity function $s(v_1, v_2)$:

$$s(v_1, v_2) = \cos\theta = \frac{v_1 \cdot v_2}{\| v_1 \| \| v_2 \|} \tag{8}$$

is used as a metric for the similarity between 2 images, where v_1 and v_2 are the feature vectors of the 2 images respectively. More similar images have $s(v_1, v_2)$ closer to 1 whereas with less similar $s(v_1, v_2)$ is closer to -1.

The data filtering method is defined as follows. An image is fed into FN to generate the corresponding vector V_i. For jth category, the average vector V_{avg} is calculated by

$$V_{avg}^{(j)} = \frac{\sum_{i=1}^{N} V_i}{N}, \tag{9}$$

where N is the image number of the category. Then the average similarity of jth category is written as:

$$\eta_{avg}^{(j)} = \frac{\sum_{i=1}^{N} s\left(V_i, V_{avg}^{(j)}\right)}{N}. \tag{10}$$

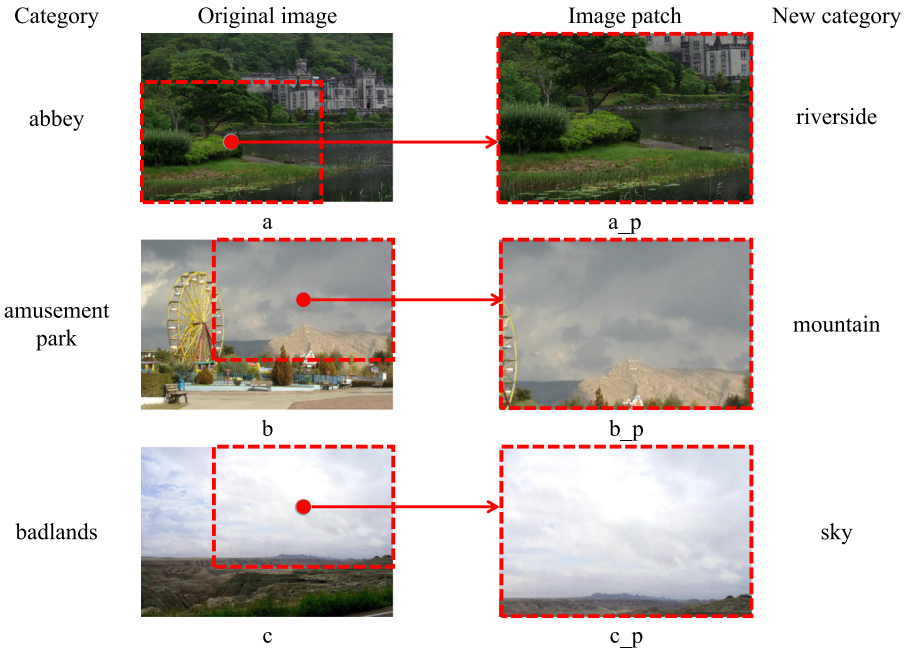

Fig. 4. Some examples of incorrect image patches generated from data augmentation.

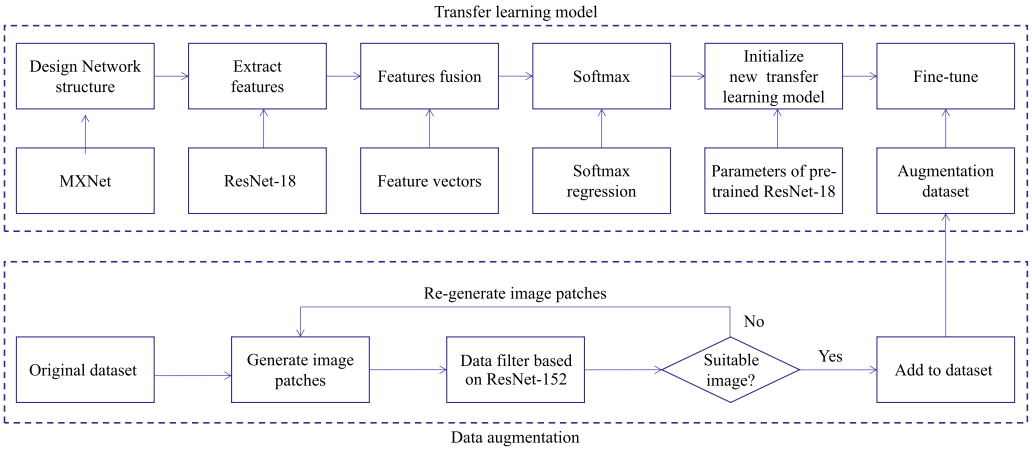

Fig. 5. Workflow of the proposed method includes two parts. Part one focuses on constructing a transfer learning model with a 18-layer ResNet and features fusion. Part two focuses on dataset augmentation and data filtering using a 152-layer ResNet. Finally, the transfer learning model is fine-tuned on the augmentation dataset.

After that the function formulated as:

$$\beta = s\left(V_{new}^{(j)}, V_{avg}^{(j)}\right) - \eta_{avg}^{(j)}, \tag{11}$$

where $V_{new}^{(j)}$ denotes the vector of new generated image, is used to decide whether the new generated image truly belongs in the jth category, or not. A new image is added into the training dataset only when its differentiate coefficient β is greater than 0. Only the suitable images from the preliminary data augmentation are kept to train the model.

To sum up, the proposed data augmentation method contains the data generation using sliding block and the data filter utilizing a ResNet-152. The data generation is similar to common augmentation techniques, which is based on image processing methods. The crucial step is the data filter that traditional data augmentation techniques do not have. By mean of transfer learning the data filter employs deep CNN features from a 152-layer ResNet to delete unsuitable data in the raw augmentation data since the unsuitable data can disturb training and prevent the model from satisfying results. Better scene classification results can be obtained by combining the data generation with the data filter, which will be proved in Section 4.

3.3. Overview of the proposed method

With data augmentation a larger dataset is obtained to fine-tune the proposed transfer learning model. A workflow shown in Fig. 5 is used to make a summary of the overall method of this paper. There are two parts in the overall method. A ResNet based transfer learning model with features fusion is built by using deep learning toolkit MXNet in part one. Parameters of the transfer learning model are initialized by a pre-trained 18-layer ResNet. Part two focuses on data augmentation. Original dataset is augmented by the proposed data augmentation method to generate new image patches. And a filter based on a 152-layer ResNet is designed to delete unsuitable generated data. Finally, the transfer

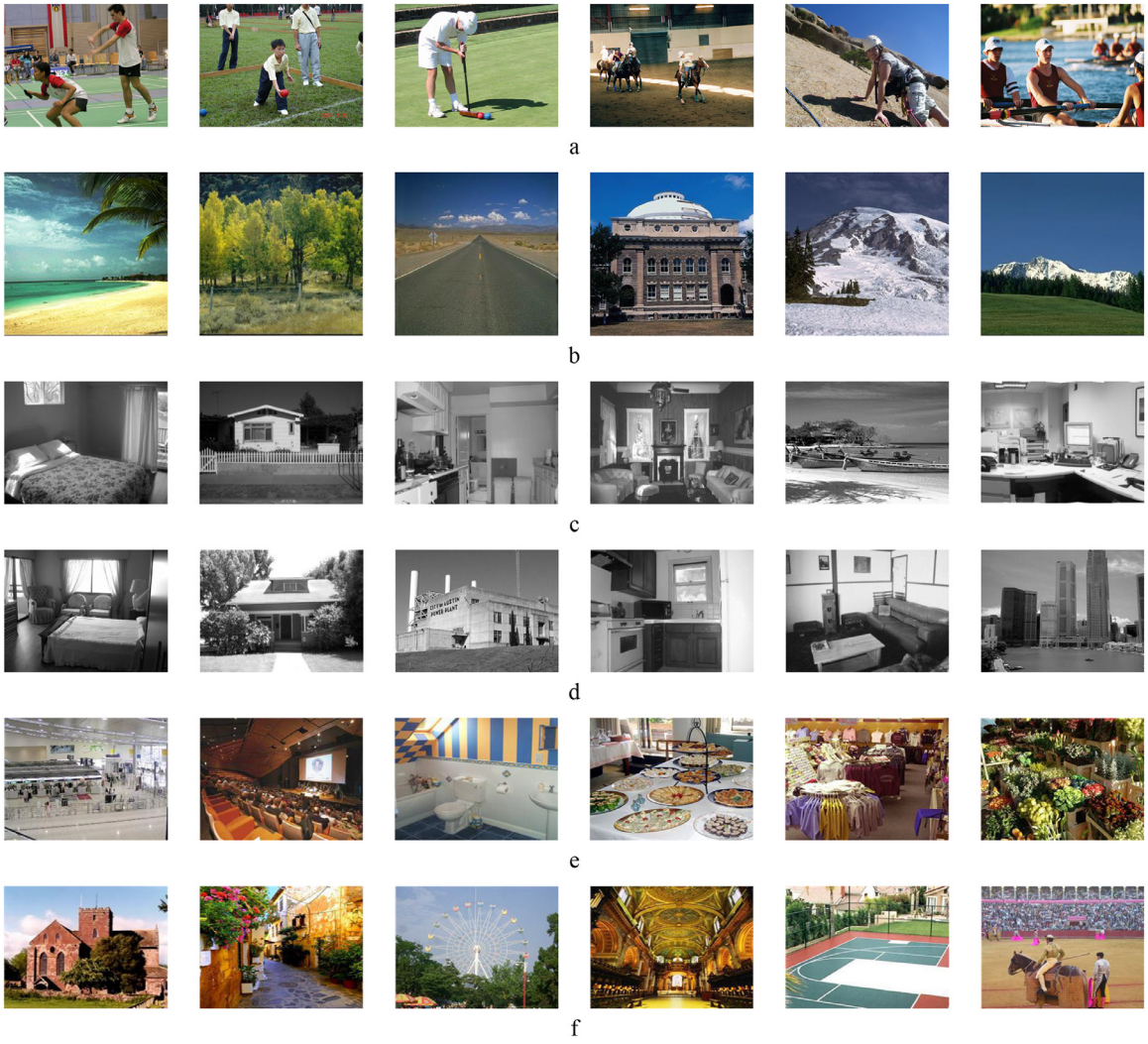

Fig. 6. Examples from the LF(a), OT(b), FP(c), LS(d), MIT67(e), SUN397(f) datasets.

Table 3
A comparison of the six benchmark datasets.

Dataset	Total images	Num. of classes	Size of classes
LF	1579	8	137–250
OT	2688	8	292–410
FP	3859	13	210–410
LS	4485	15	210–410
MIT67	15,620	67	101–734
SUN397	130,519	397	100–2361

learning model is fine-tuned on the augmentation dataset in order to obtain better results.

4. Experiments

In this section extensive experiments are implemented to evaluate our proposed method. The hypotheses tested in the experiments are as follows: (1) Our transfer learning model obtains better accuracy than commonly used transfer learning models on diverse scene datasets. (2) Our data augmentation method further improves the accuracy of scene classification. (3) Our data augmentation method is more efficient for scene classification than traditional data augmentation. (4) The data filter is a significant part of our proposed data augmentation method.

4.1. Raw datasets

The presented method is validated on the following six benchmark datasets:

(1) 8-category sports events (LF) [16]. The LF dataset contains 1579 RGB images with the 8 sports: badminton, bocce, croquet, polo, rock climbing, rowing, sailing and snow boarding. The number of images in each category is 137 to 250.
(2) 8-category outdoor scenes (OT) [17]. The OT dataset includes 2688 images in total, in 8 categories: coasts, forests, highways, inner city scenes, mountains, open country, streets and tall buildings. The number in each category varies from 292 to 410. All images in OT dataset have the same size, 256×256 pixels.
(3) 13-category natural scenes (FP) [18]. In total the FP dataset contains 3,859 grayscale images in 13 categories. These numbers include 1,172 images of offices, living rooms, suburbs, kitchens and bedrooms based on the OT dataset. The size of each image is about 250×300 pixels.
(4) 15-category scenes (LS) [19]. The total number of images in this dataset is 4,485, 3,859 of which are from FP dataset. In addition to the FP categories there are the new categories of industrial and store. The LS dataset contains from 200 to

400 grayscale images per category with an approximate size of 300×250 pixels.

(5) 67-category scenes (MIT67) [20]. This database contains 67 indoor categories and a total of 15,620 images. The number of images varies across categories but there are at least 100 images per category. Each image has a minimum resolution of 200 pixels on the smallest axis.

(6) 397-category scenes (SUN397) [21]. SUN397 includes 397 well-sampled categories with 130,519 images. This dataset contains from 100 to 2361 images per category.

The LF, OT, FP and LS dataset contain a relatively small number of categories. The MIT67 and SUN397 dataset have more. A comparison of the six datasets is listed in Table 3. Some examples from the six datasets are shown in Fig. 6.

4.2. Experimental settings

The experiments are implemented using MXNet v1.1.0 and designed based on 10-fold cross-validation. All datasets are divided into 10 parts, 8 parts of which are trained and 1 parts of which is as validation set and the rest of which is tested in turn. The mean accuracy of the 10 times testing is recorded as the final result. All images are resized to 224×224 pixels. The hyper-parameters for training are set as follows. The learning rate (η) is initialized to 0.0001 and changed according to the value of the error function. If the error rate decreases in comparison with the prior error rate, η is increased by 5%. Otherwise, η is decreased by 50%. Nadam is utilized as the optimizer and its parameter settings are *beta*1(0.9), *beta*2(0.999), *epsilon*(1^{-8}), *schedule_decay*(0.004). The value of min_batch is 128.

The experiments are conducted using Ubuntu16.04.4 running on Intel hardware with an Intel i9 7900X CPU, 32 GB memory and a 1T SSD hard disk. Two NVIDIA TITAN XP graphics cards are employed to accelerate the training.

4.3. Training on raw datasets

This section describes the training on raw datasets. The six raw datasets are trained from scratch (TFS); fine-tuned on both a pre-trained 18-layer ResNet (FTOR) and fine-tuned on the proposed transfer learning model (FTOTLM). The fine-tuning strategies for the pre-trained ResNet and the proposed model involve fine-tuning all the weights of the 18 layers to exploit the potentialities and

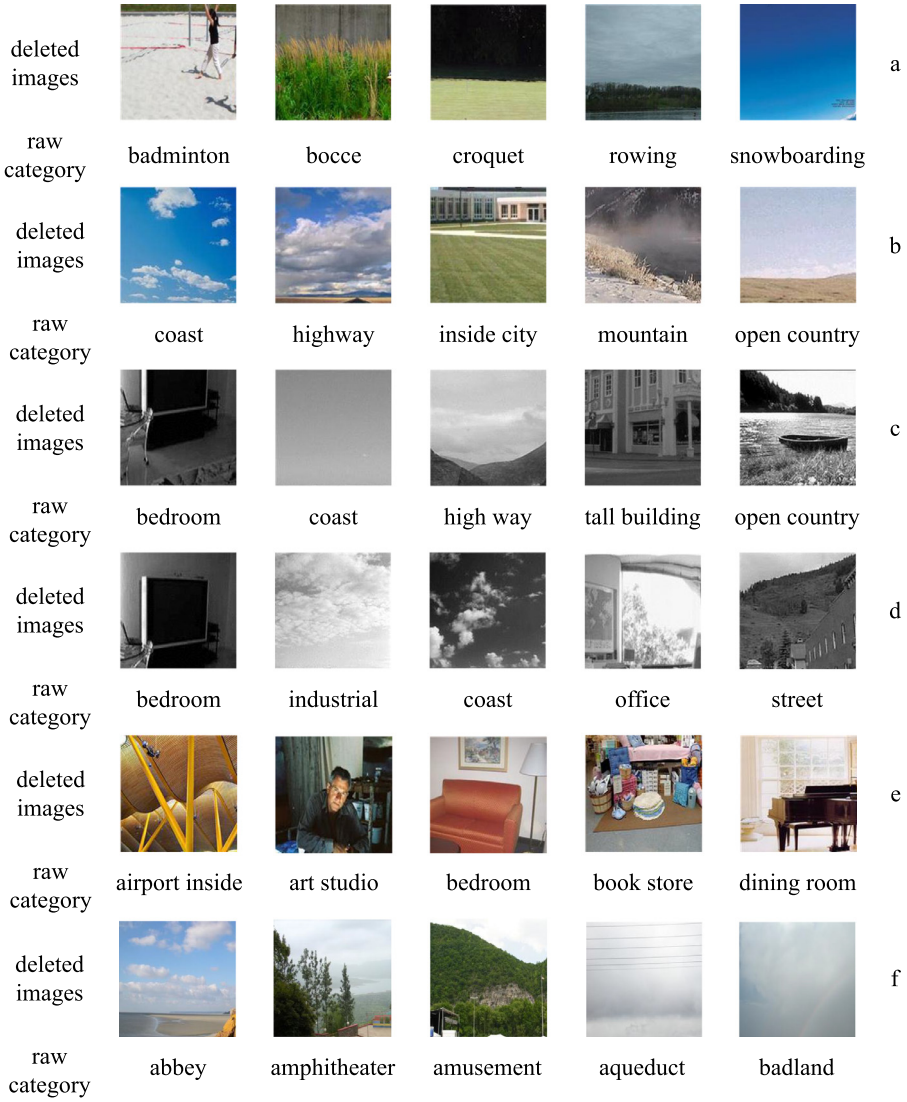

Fig. 7. Examples of deleted images from the LF(a), OT(b), FP(c), LS(d), MIT67(e), SUN397(f) datasets. The content of the images are quite different from the corresponding category labels.

Table 4
A quantitative data analysis of the six augmented datasets.

Dataset	Raw	Augmented	Increased scope (%)	Average increase(%)
LF	1579	7637	455–500	484
OT	2688	12,646	440–492	470
FP	3859	18,662	463–499	483
LS	4485	21,362	410–499	476
MIT67	15,620	74,149	395–498	475
SUN397	130,519	516,620	400–500	396

Table 5
Settings used for the traditional data augmentation methods.

Method	Setting
Rotation	Random angle in [0° - 180°]
Rotation	Random angle in [180° - 360°]
Flip	Left to right
Translation	Random shift in [0, 20] pixels

enhance the adaptability of the network to each dataset. The network structure should be changed to accommodate each dataset because the different datasets have a different set of categories. The last layer of the pre-trained network is replaced with a new softmax layer which is relevant to the category number. For example, when the training set is LF dataset, which contains 8 classes, the new softmax layer should have 8 categories.

4.4. Training on augmented datasets

4.4.1. Dataset augmentation

This section describes how the six original datasets are extended by our data augmentation method. Firstly, a single raw image is transformed into 4 image patches by a sliding block. The newly created images and the raw images form the preliminary augmentation datasets, which is 5 times larger than the raw datasets. Then these preliminary augmented datasets are filtered, as proposed, to remove improper images. A sample of deleted images from the six datasets are shown in Fig. 7. After that the final augmented datasets are generated and used for training.

The results of dataset augmentation, comparisons between the raw datasets and the augmented datasets, are shown as a set of graphs. The small-scale class datasets (LF, OT, FP and LS) are displayed in Fig. 8 showing the total number of images in each category, before and after augmentation. The remaining datasets (MIT67 and SUN397) shown in Fig. 9, without category labels due to the limitations print. To facilitate analysis, quantitative data is listed in Table 4. This tables shows that augmentation increased the average size of most dataset's categories by a factor of 4.70 or more, the exception being the SUN397 dataset which increased by only a factor of 3.96.

4.4.2. Fine-tuning on augmented datasets

Our proposed data augmentation method is applied to generate new training data for every dataset. Then the augmented datasets are trained on the pre-trained ResNet and using our transfer learning model and evaluated for fitness; the results are compared with the training on the raw datasets. By contrast, the traditional data augmentation methods (shown in Fig. 10), comprising rotation, flip and translation, are utilized to generate the same number of images. The settings for each of these methods are listed in Table 5. The comparative results are shown in Fig. 11. An overview of the difference between the traditional and our data augmentation methods, as a sample graph of success rates on the OT dataset per training epoch, of the various training methods, is provided in Fig. 12.

Table 6
Comparison of three learning strategies on the six datasets: trained from scratch (TFS), fine-tuning on ResNet-18 (FTOR) and fine-tuning on our transfer learning model (FTOTLM). In this table, Ds, Da, Str, Acc stands for Dataset, Data augmentation, Strategies, and Accuracy, respectively.

Ds	Da	Str	Acc(%)	Ds	Da	Str	Acc(%)
LF	No	TFS	76.43	OT	No	TFS	85.57
		FTOR	95.24			FTOR	95.12
		FTOTLM	96.48			FTOTLM	96.38
	Yes	TFS	85.67		Yes	TFS	94.68
		FTOR	99.11			FTOR	98.28
		FTOTLM	99.87			FTOTLM	99.45
FP	No	TFS	78.23	LS	No	TFS	75.92
		FTOR	93.15			FTOR	92.31
		FTOTLM	94.56			FTOTLM	94.01
	Yes	TFS	91.52		Yes	TFS	90.35
		FTOR	96.32			FTOR	96.11
		FTOTLM	97.45			FTOTLM	97.38
MIT67	No	TFS	41.11	SUN397	No	TFS	51.21
		FTOR	73.12			FTOR	62.32
		FTOTLM	74.63			FTOTLM	65.46
	Yes	TFS	79.47		Yes	TFS	70.13
		FTOR	91.09			FTOR	82.54
		FTOTLM	94.05			FTOTLM	85.21

Table 7
LF dataset: FTOTLM (fine-tuning on our transfer learning model) v.s. other state-of-the-arts.

Method	Accuracy (%)
LDA [35]	71
Full model in [16]	73.4
GIST [52]	74.58 ± 1.39
LBP(sPACT) [53]	78.50 ± 0.99
HOG [50]	80.8
Model in [54]	82.96 ± 1.51
GOC [55]	83.07 ± 0.70
FTOTLM without data augmentation	96.48
FTOTLM with data augmentation	99.87

In addition, the preliminarily augmented datasets, those not filtered by the data filter, are tested and their efficacy compared with that of the final augmented datasets. Both the ResNet-18 and our transfer learning model are fine-tuned to the preliminarily augmented datasets before training. As an example, the preliminarily augmented MIT67 dataset contains 78,100 images, while the filtered dataset contains 74,149. The loss data are unsuitable images. The MIT67 training is shown in Fig. 13. For other datasets, the training results and comparison are shown in Fig. 14.

4.5. Results and analysis

In these experiments three strategies (trained from scratch, fine-tuning on ResNet-18 and fine-tuning on our transfer learning model) are employed to train on the six datasets, with and without the proposed data augmentation method. The training results are listed in Table 6. To assess the methods this paper proposes, our methods and the state-of-the-art methods are compared on the six datasets. The results are given in Tables 7–12, respectively.

Comparing classification results on the six datasets, we arrive at the following conclusions:

(1) Table 6 shows training from scratch on the six datasets obtains the lowest classification accuracy. ResNet, because of its extensive parameters, can not be trained adequately given the restricted data, especially when using the small-scale datasets.

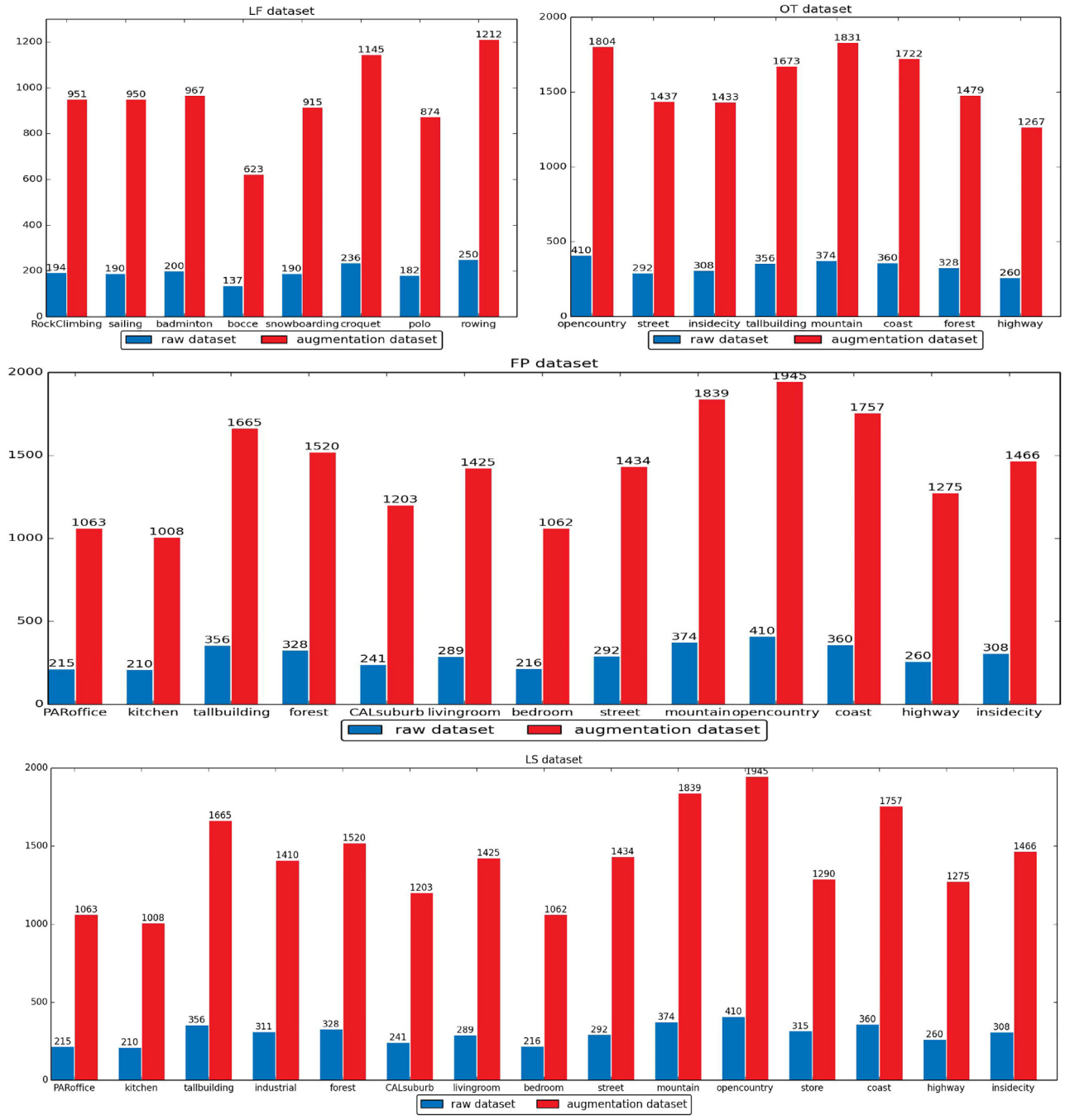

Fig. 8. Comparison of the raw and the augmented LF, OT, FP and LS datasets.

(2) Transferring parameters before training, from a pre-trained 18-ResNet to an un-trained ResNet, improves accuracy to a degree. For datasets of small-scale (8–15 classes) (LF, OT, FP, LS), accuracies are over 90%. For MIT67 and SUN397, this method yields an improvement of 32.01% and 11.11%, respectively, over training from scratch.

(3) Fine-tuning our transfer learning model on the six datasets increases classification accuracy; with an improvement of more than 1.2% on LF, OT, FP, about 1.7% on LS, and about 1.5% on MIT67 and 3.1% on SUN397, over the fine-tuning raw ResNet. The proposed transfer model produces better results than the commonly used fine-tuned ResNet model.

(4) The results shown in Fig. 11 demonstrates that when training with FTOR or FTORM on the six datasets our data augmentation technique with or without data filter produces higher classification accuracies than traditional data augmentation. Even without data filter based on pre-trained 152-layer ResNet our data augmentation method can also obtain better results in the six datasets. The example of the OT dataset (shown in Fig. 12) demonstrates that the training process is smoother and the rate of convergence is faster with our data augmentation compared to traditional data augmentation.

(5) The data filter plays an important role in the data augmentation method we present. Fig. 13 shows the training processes are more stable with the data filter and that

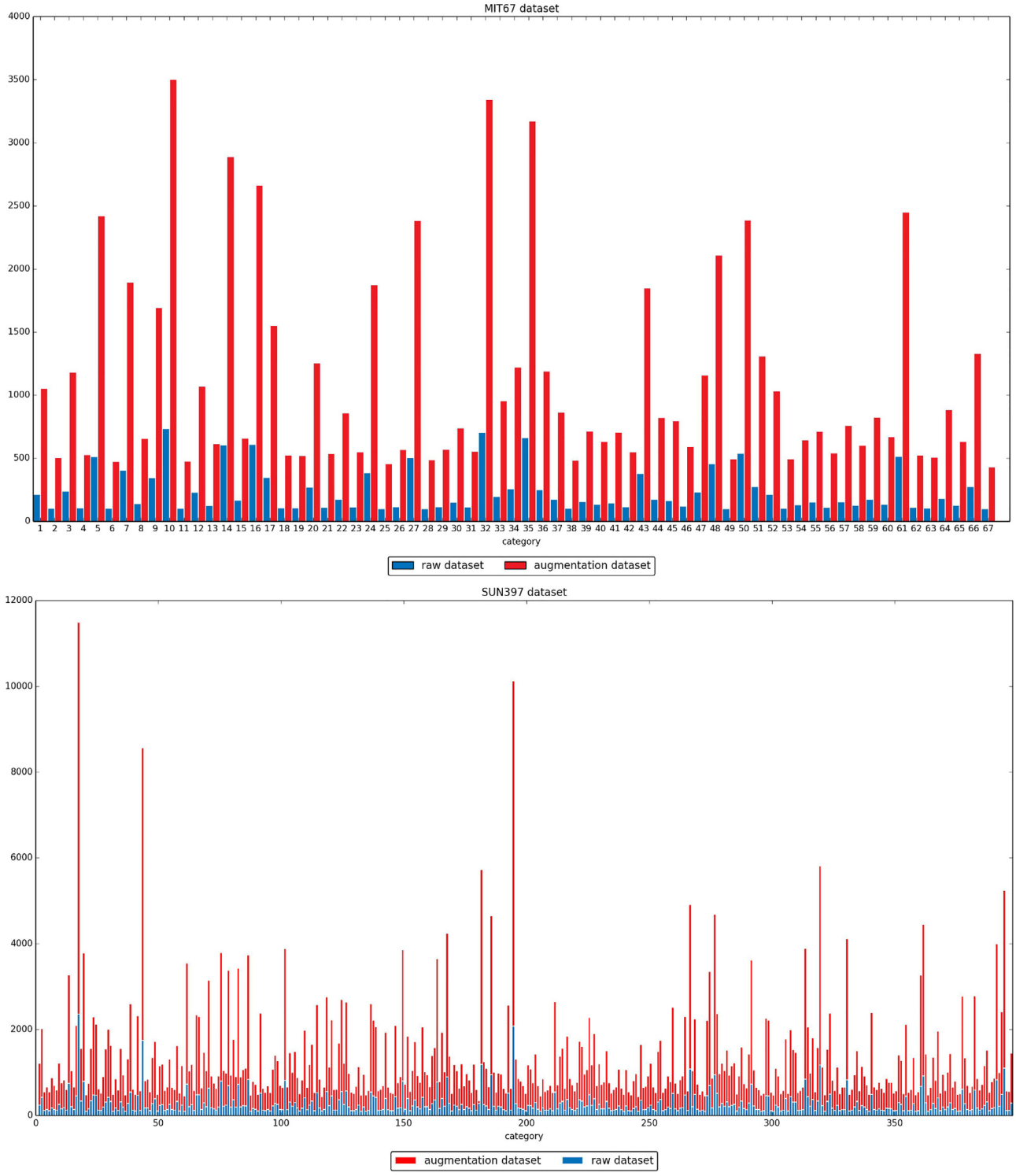

Fig. 9. Comparison of the raw and augmented MIT67 and SUN397 datasets.

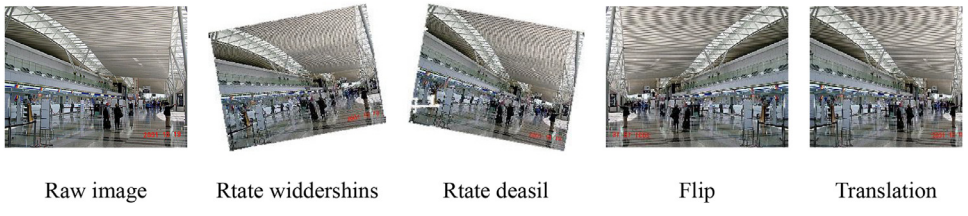

Fig. 10. Four transformation methods of traditional data augmentation.

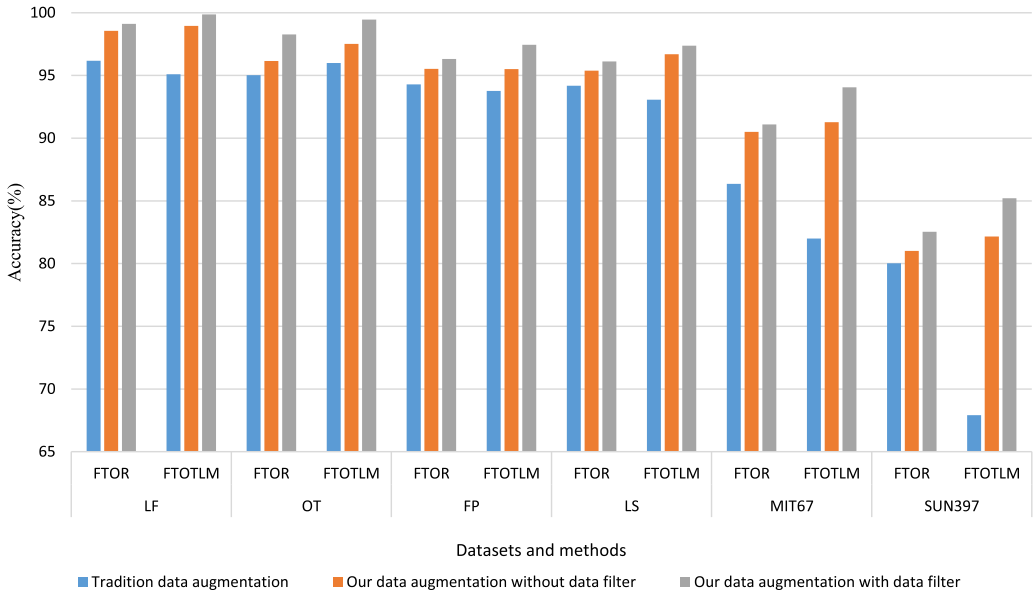

Fig. 11. Application of the traditional and our data augmentation methods without or with data filter to the six datasets.

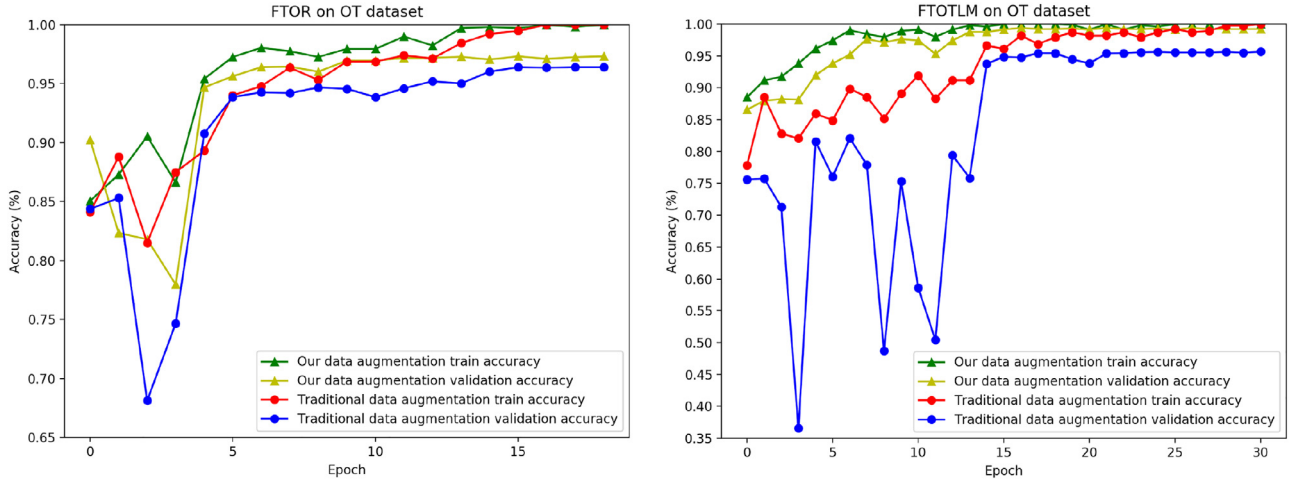

Fig. 12. Training results, by FTOR (Fig.a) and FTOTLM (Fig.b), on the OT dataset, augmented by traditional methods and our data augmentation method.

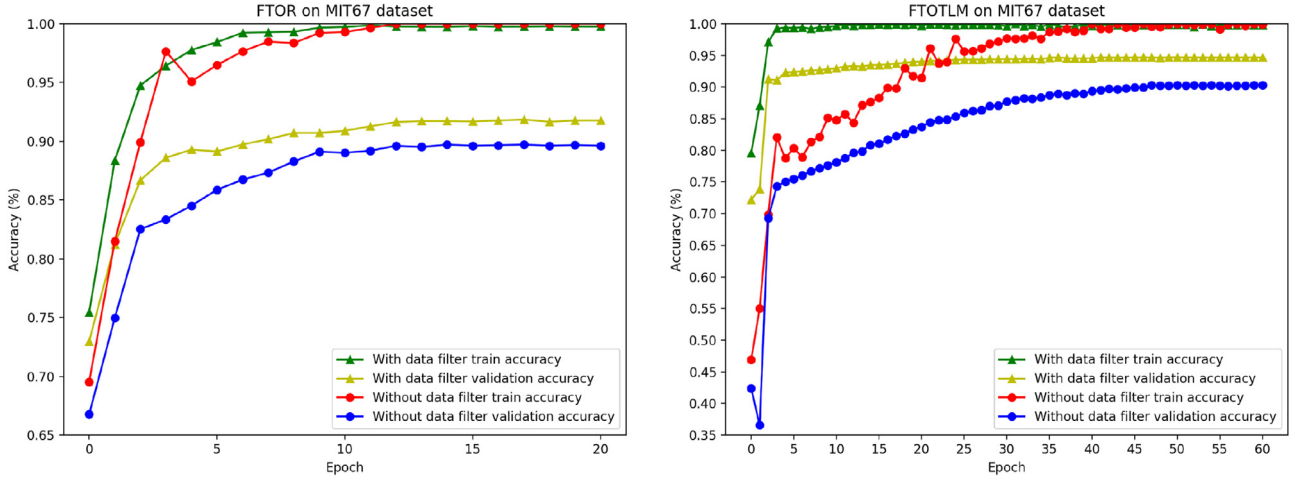

Fig. 13. Efficacy of training on the MIT67 dataset with and without data filter, by FTOR (Fig.a) and FTOTLM (Fig.b).

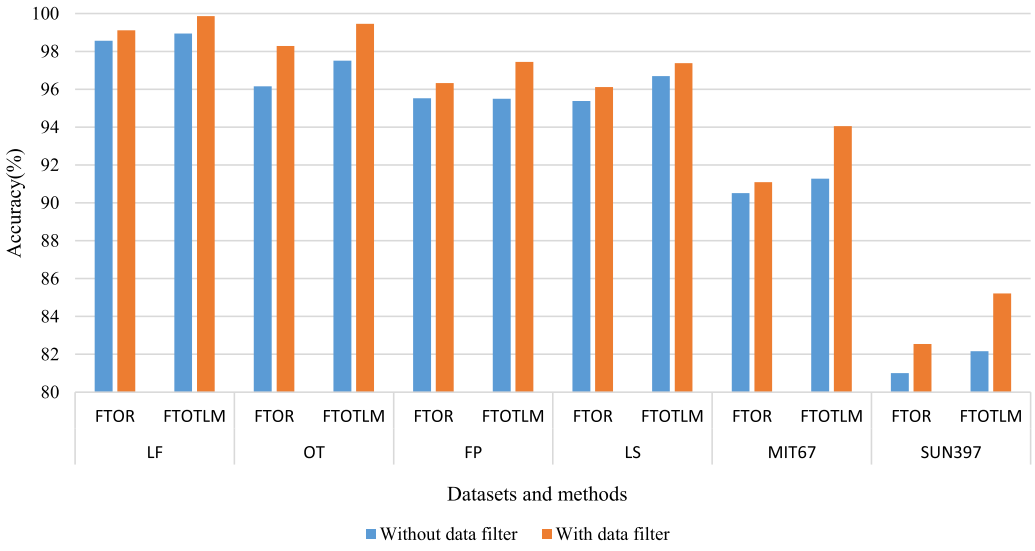

Fig. 14. Efficacy of training on the six datasets by our data augmentation method, with and without data filter.

Table 8
OT dataset: FTOTLM (fine-tuning on our transfer learning model) v.s. other state-of-the-arts.

Method	Accuracy (%)
HRSE [17]	83.7
CENTRIST [56]	86.22 ± 1.02
Model in [54]	88.10 ± 0.9
GBPWHGO [34]	88.4
Combined feature [36]	89.9
SP-pLSA [57]	94.8
FTOTLM without data augmentation	96.38
FTOTLM with data augmentation	99.45

Table 9
FP dataset: FTOTLM (fine-tuning on our transfer learning model) v.s. other state-of-the-arts.

Method	Accuracy (%)
GBPWHGO [34]	87.0
Combined feature [36]	87.8
Model in [18]	65.2
SP-pLSA [57]	85.9
Model in [58]	84.0
GMM [59]	84.1
FTOTLM without data augmentation	94.56
FTOTLM with data augmentation	97.45

Table 10
LS dataset: FTOTLM (fine-tuning on our transfer learning model) v.s. other state-of-the-arts.

Method	Accuracy (%)
CMN [60]	77.2
SPMSM [61]	82.5
EMFS [62]	85.7
SR-LSR [63]	85.7
Sun et al. [64]	86.5
Object-to-Class kernels [65]	88.8
G-MS2F [39]	92.90
FTOTLM without data augmentation	94.01
FTOTLM with data augmentation	97.38

Table 11
MIT67 dataset: FTOTLM (fine-tuning on our transfer leraning model) v.s. other state-of-the-arts.

Method	Accuracy (%)
CCRBM [37]	41.2
Sun et al. [64]	46.4
PlaceNet [66]	68.24
DSFL+CNN [67]	76.23
G-MS2F [39]	79.63
Bai et al. [1]	80.75
CFV(VGG-19) [68]	81.00
CS(VGG-19) [69]	82.24
VSAD [70]	86.20
SDO [71]	86.76
FTOTLM without data augmentation	74.63
FTOTLM with data augmentation	94.05

Table 12
Sun397 dataset: FTOTLM (fine-tuning on our transfer leraning model) v.s. other state-of-the-arts.

Method	Accuracy (%)
DeCAF [72]	40.94
CCRBM [37]	48.3
MOP-CNN [73]	51.98
Places-CNN fc [66]	56.20
MetaObject-CNN [74]	58.11
Bai et al. [1]	59.53
G-MS2F [39]	64.06
CS(VGG-19) [69]	64.53
FTOTLM without data augmentation	65.46
FTOTLM with data augmentation	85.21

improper generated images can disturb the training. In addition, the data filter accelerates training convergence speed and boosts accuracy. Fig. 14 shows using the data filter on the six datasets increases accuracy. As the number of categories increases, the improvement is more evident. For example, with the FTOTLM strategy the data filter improves accuracy by less than 1% on LF dataset but improvement is more than 3% on SUN397. Whether utilizing the FTOR or the FTOTLM method, accuracies decline without the data filter. This shows the data filter has a significant effect on training results.

(6) The comparisons shown in Tables 7–12, demonstrate that on the six benchmark datasets our transfer learning model combined with our proposed data augmentation method can outperform other state-of-the-art models.

5. Conclusions

We propose a novel method of improving the accuracy of scene classification, employing transfer learning and data augmentation. Based on a pre-trained ResNet, we build a modified transfer learning model by fusing multi-level features from selected layers. Different layers contain different visual characteristics, we utilise this to improve scene classification accuracy. In addition, an effective data augmentation method is described which generates and filters image patches based on raw images. Applying this filter scene datasets create sufficient and suitable training data, which increases classification accuracy; especially when the dataset is limited. The proposed techniques are evaluated in extensive experiments on six scene benchmark datasets, outperforming existing state-of-the-art models.

Acknowledgments

This work is supported by the National Natural Science Foundation of China (U1813215 and 61773239) and the Taishan Scholars Program of Shandong Province. We would like to express our heartfelt gratitude to Karl O. Pinc and Jiaai Wang, who provided some advice on academic writing and help.

References

[1] S. Bai, H. Tang, Categorizing scenes by exploring scene part information without constructing explicit models, Neurocomputing 281 (2018) 160–168.
[2] M. Brown, S. Susstrunk, Multi-spectral sift for scene category recognition, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2011, pp. 177–184.
[3] S. Lazebnik, C. Schmid, J. Ponce, Beyond bags of features: spatial pyramid matching for recognizing natural scene categories, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2006, pp. 2169–2178.
[4] J. Deng, W. Dong, R. Socher, L.J. Li, K. Li, F.F. Li, Imagenet: a large-scale hierarchical image database, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2009, pp. 248–255.
[5] A. Krizhevsky, I. Sutskever, G. E.Hinton, Imagenet classification with deep convolutional neural networks, in: Proceedings of the International Conference on Neural Information Processing Systems (ICONIP), 2012, pp. 1097–1105.
[6] K. Simonyan, A. Zisserman, Very deep convolutional networks for large-scale image recognition, arXiv:1409.1556 (2014).
[7] K. He, X. Zhang, S. Ren, J. Sun, Deep residual learning for image recognition, in: Proceedings of the Computer Vision and Pattern Recognition (CVPR), 2016, pp. 770–778.
[8] C. Szegedy, W. Liu, Y. Jia, et al., Going deeper with convolutions, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2015, pp. 1–9.
[9] C. Szegedy, V. Vanhoucke, S. Ioffe, et al., Rethinking the inception architecture for computer vision, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2016, pp. 2818–2826.
[10] N. Srivastava, G. Hinton, A. Krizhevsky, et al., Dropout: a simple way to prevent neural networks from overfitting, J. Mach. Learn. Res. 15 (1) (2014) 1929–1958.
[11] K. Weiss, T.M. Khoshgoftaar, D.D. Wang, A survey of transfer learning, J. Big Data 3 (1) (2016) 9.
[12] L. Herranz, S. Jiang, X. Li, Scene recognition with CNNs: Objects, scales and dataset bias, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2016, pp. 571–579.
[13] W. Yu, K. Yang, H. Yao, X. Sun, P. Xu, Exploiting the complementary strengths of multi-layer CNN features for image retrieval, Neurocomputing 237 (2016) 235–241.
[14] D.X. Xue, R. Zhang, H. Feng, Y.L. Wang, CNN-SVM for microvascular morphological type recognition with data augmentation, J. Med. Biol. Eng. 36 (6) (2016) 755–764.
[15] D.M. Montserrat, Q. Lin, J. Allebach, E.J. Delp, Training object detection and recognition CNN models using data augmentation, Electron. Imaging 2017 (10) (2017) 27–36.
[16] L.J. Li, F.F. Li, What, where and who? classifying events by scene and object recognition, in: Proceedings of the IEEE International Conference on Computer Vision, 2007, pp. 1–8.
[17] O. Aude, T. Antonio, Modeling the shape of the scene: a holistic representation of the spatial envelope, Int. J. Comput. Vis. 42 (3) (2001) 145–175.
[18] F.F. Li, P. Perona, A Bayesian hierarchical model for learning natural scene categories, in: Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2005, pp. 524–531.
[19] S. Lazebnik, C. Schmid, J. Ponce, Beyond bags of features:spatial pyramid matching for recognizing natural scene categories, in: Proceedings of the IEEE Computer Vision and Pattern Recognition, 2006, pp. 2169–2178.
[20] A. Quattoni, A. Torralba, Recognizing indoor scenes, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2001, pp. 413–420.
[21] J. Xiao, J. Hays, K.A. Ehinger, A. Oliva, A. Torralba, Sun database: large-scale scene recognition from abbey to zoo, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2010, pp. 3485–3492.
[22] S.J. Pan, Q. Yang, A survey on transfer learning, IEEE Trans. Knowl. Data Eng. 22 (10) (2010) 1345–1359.
[23] R. Chattopadhyay, J. Ye, S. Panchanathan, W. Fan, I. Davidson, Multi-source domain adaptation and its application to early detection of fatigue, in: Proceedings of the ACM Sigkdd International Conference on Knowledge Discovery & Data Mining, 2011, pp. 717–725.
[24] M. Long, J. Wang, G. Ding, J. Sun, P.S. Yu, Transfer feature learning with joint distribution adaptation, in: Proceedings of the IEEE International Conference on Computer Vision, 2014, pp. 2200–2207.
[25] S.J. Pan, I.W. Tsang, J.T. Kwok, Q. Yang, Domain adaptation via transfer component analysis, IEEE Trans. Neural Netw. 22 (2) (2011) 199.
[26] F. Li, S.J. Pan, O. Jin, Q. Yang, X. Zhu, Cross-domain co-extraction of sentiment and topic lexicons, in: Proceedings of the Meeting of the Association for Computational Linguistics: Long Papers, 2012, pp. 410–419.
[27] T. Tommasi, F. Orabona, B. Caputo, Learning categories from few examples with multi model knowledge transfer, IEEE Trans. Pattern Anal. Mach. Intell. 36 (5) (2014) 928–941.
[28] S. Hoochang, H.R. Roth, M. Gao, et al., Deep convolutional neural networks for computer-aided detection: CNN architectures, dataset characteristics and transfer learning, IEEE Trans. Med. Imaging 35 (5) (2016) 1285.
[29] H. Lei, T. Han, F. Zhou, et al., A deeply supervised residual network for HEp-2 cell classification via cross-modal transfer learning, Pattern Recognit. 79 (2018) 290–302.
[30] J. Han, H. Chen, N. Liu, C. Yan, X. Li, CNNs-based RGB-D saliency detection via cross-view transfer and multiview fusion, IEEE Trans. Cybern. PP (99) (2017). 1–13
[31] Y. Lecun, B. Boser, J.S. Denker, et al., Backpropagation applied to handwritten zip code recognition, Neural Comput. 1 (4) (2014) 541–551.
[32] Y. Lecun, L. Bottou, Y. Bengio, P. Haffner, Gradient-based learning applied to document recognition, Proc. IEEE 86 (11) (1998) 2278–2324.
[33] M. Everingham, L.V. Gool, C.K.I. Williams, J. Winn, A. Zisserman, The pascal visual object classes (VOC) challenge, Int. J. Comput. Vis. 88 (2) (2010) 303–338.
[34] L. Zhou, Z. Zhou, D. Hu, Scene classification using multi-resolution low-level feature combination, Neurocomputing 122 (2013) 284–297.
[35] M. Zang, D. Wen, K. Wang, T. Liu, W. Song, A novel topic feature for image scene classification, Neurocomputing 148 (2015) 467–476.
[36] L. Yuan, F. Chen, L. Zhou, D. Hu, Improve scene classification by using feature and kernel combination, Neurocomputing 170 (2015) 213–220.
[37] J. Gao, J. Yang, G. Wang, M. Li, A novel feature extraction method for scene recognition based on centered convolutional restricted Boltzmann machines, Neurocomputing 214 (2015) 708–717.
[38] X. Qi, C.G. Li, G. Zhao, X. Hong, Dynamic texture and scene classification by transferring deep image features, Neurocomputing 171 (2015) 1230–1241.
[39] P. Tang, H. Wang, S. Kwong, G-MS2F: GoogLeNet based multi-stage feature fusion of deep CNN for scene recognition, Neurocomputing 225 (2017) 188–197.
[40] B. Hu, J.H. Lai, C.C. Guo, Location-aware fine-grained vehicle type recognition using multi-task deep networks, Neurocomputing 243 (2017) 60–68.
[41] R. Fergus, F.F. Li, P. Perona, A. Zisserman, Learning object categories from internet image searches, Proc. IEEE 98 (8) (2010) 1453–1466.
[42] I.J. Goodfellow, J. Pouget-Abadie, M. Mirza, et al., Generative adversarial nets, in: Proceedings of the International Conference on Neural Information Processing Systems, 2014, pp. 2672–2680.
[43] T. Team, R. Al-Rfou, G. Alain, et al., Theano: a python framework for fast computation of mathematical expressions, arXiv:1605.02688 (2016).
[44] Y. Jia, E. Shelhamer, J. Donahue, et al., Caffe: Convolutional architecture for fast feature embedding, in: Proceedings of the ACM International Conference on Multimedia, 2014, pp. 675–678.
[45] T. Chen, M. Li, Y. Li, et al., Mxnet: a flexible and efficient machine learning library for heterogeneous distributed systems, arXiv:1512.01274 (2015).
[46] B. Zhou, A. Lapedriza, A. Khosla, et al., Places: a 10 million image database for scene recognition, IEEE Trans. Pattern Anal. Mach. Intell. 99 (2017) 1–10.
[47] M. Lin, Q. Chen, S. Yan, Network in network, arXiv:1312.4400 (2013).
[48] S. Ioffe, C. Szegedy, Batch normalization: accelerating deep network training by reducing internal covariate shift, arXiv:1502.03167 (2015).
[49] T. Dozat, Incorporating Nesterov momentum into Adam, 2015, ([online] Available: http://cs229.stanford.edu/proj2015/054_report.pdf).
[50] Dalal, Navneet, B. Triggs, Histograms of oriented gradients for human detection, in: Proceedings of the Computer Vision and Pattern Recognition, 2005, pp. 886–893.
[51] D. Lowe, Distinctive image features from scale-invariant keypoints, Int. J. Comput. Vis. 60 (2) (2004) 91–110.
[52] C. Siagian, L. Itti, Rapid biologically-inspired scene classification using features shared with visual attention., IEEE Trans. Pattern Anal. Mach. Intell. 29 (2) (2007) 300–312.
[53] J. Wu, J.M. Rehg, Where am I: place instance and category recognition using spatial pact, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2008, pp. 1–8.
[54] X. Meng, Z. Wang, L. Wu, Building global image features for scene recognition, Pattern Recognit. 45 (1) (2012) 373–380.

[55] C. Gao, N. Sang, R. Huang, Spatial multi-scale gradient orientation consistency for place instance and scene category recognition, Inf. Sci. 372 (2016) 84–97.
[56] J. Wu, J.M. Rehg, Centrist: a visual descriptor for scene categorization, IEEE Trans. Pattern Anal. Mach. Intell. 33 (8) (2011) 1489–1501.
[57] A. Bosch, A. Zisserman, X. Muoz, Scene classification using a hybrid generative/discriminative approach, IEEE Trans. Pattern Anal. Mach. Intell. 30 (4) (2008) 712–727.
[58] S. Battiato, G.M. Farinella, G. Gallo, D. Rav, Spatial hierarchy of textons distributions for scene classification, Adv. Multimed. Model. 5371 (2009) 333–343.
[59] X. Zhou, X. Zhuang, H. Tang, M. Hasegawa-Johnson, T.S. Huang, Novel Gaussianized vector representation for improved natural scene categorization, Pattern Recognit. Lett. 31 (8) (2010) 702–708.
[60] N. Rasiwasia, N. Vasconcelos, Holistic context models for visual recognition, IEEE Trans. Pattern Anal. Mach. Intell. 34 (5) (2012) 902–917.
[61] R. Kwitt, N. Vasconcelos, N. Rasiwasia, Scene recognition on the semantic manifold, in: Proceedings of the European Conference on Computer Vision, 2012, pp. 359–372.
[62] H.O. Song, R. Girshick, S. Zickler, C. Geyer, Generalized sparselet models for real-time multiclass object recognition, IEEE Trans. Pattern Anal. Mach. Intell. 37 (5) (2015) 1001–1012.
[63] L.J. Li, H. Su, Y. Lim, F.F. Li, Object bank: an object-level image representation for high-level visual recognition, Int. J. Comput. Vis. 107 (1) (2014) 20–39.
[64] X. Sun, Z. Liu, Y. Hu, L. Zhang, R. Zimmermann, Perceptual multi-channel visual feature fusion for scene categorization, Inf. Sci. 429 (2018) 37–48.
[65] L. Zhang, X. Zhen, L. Shao, Learning object-to-class kernels for scene classification, IEEE Trans. Image Process. Publ. IEEE Signal Process. Soc. 23 (8) (2014) 3241–3253.
[66] B. Zhou, A. Lapedriza, J. Xiao, A. Torralba, A. Oliva, Learning deep features for scene recognition using places database, in: Proceedings of the International Conference on Neural Information Processing Systems, 2014, pp. 487–495.
[67] Z. Zuo, G. Wang, B. Shuai, L. Zhao, Q. Yang, X. Jiang, Learning discriminative and shareable features for scene classification, in: Proceedings of the European Conference on Computer Vision, 2014, pp. 552–568.
[68] M. Cimpoi, S. Maji, A. Vedaldi, Deep filter banks for texture recognition and segmentation, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2015, pp. 3828–3836.
[69] G.S. Xie, X.Y. Zhang, S. Yan, C.L. Liu, Hybrid CNN and dictionary-based models for scene recognition and domain adaptation, IEEE Trans. Circuits Syst. Video Technol. 27 (6) (2017) 1263–1274.
[70] Z. Wang, L. Wang, Y. Wang, B. Zhang, Y. Qiao, Weakly supervised PatchNets: describing and aggregating local patches for scene recognition, IEEE Trans. Image Process. 26 (4) (2017) 2028–2041.
[71] X. Cheng, J. Lu, J. Feng, B. Yuan, J. Zhou, Scene recognition with objectness, Pattern Recognit. 74 (2017) 474–487.
[72] J. Donahue, Y. Jia, O. Vinyals, J. Hoffman, N. Zhang, E. Tzeng, T. Darrell, Decaf: a deep convolutional activation feature for generic visual recognition, in: Proceedings of the International Conference on Machine Learning, 2014, pp. 647–655.
[73] Y. Gong, L. Wang, R. Guo, S. Lazebnik, Multi-scale orderless pooling of deep convolutional activation features, in: Proceedings of the European Conference on Computer Vision, 2014, pp. 392–407.
[74] R. Wu, B. Wang, W. Wang, Y. Yu, Harvesting discriminative meta objects with deep CNN features for scene classification, in: Proceedings of the IEEE International Conference on Computer Vision, 2015, pp. 1287–1295.

Shaopeng Liu is currently pursuing M.S. degree in School of Control Science and Engineering of Shandong University, Jinan, China. He obtained his B.S. degree from Qingdao University of science and technology in 2016. His research interests include service robot, robot cognition, deep learning and computer vision.

Guohui Tian is a Professor in the School of Control Science and Engineering, Shandong University, Jinan, China. And also he is the Vice Director of the Intelligence Robot Specialized Committee of Chinese Association for Artificial Intelligence, the Vice Director of the Intelligent Manufacturing System Specialized Committee of Chinese Association for Automation, and the member of the IEEE Robotics and Automation Society. His research interests include Service robot, Intelligent space, Cloud Robotics, Brain-Inspired Intelligent Robotics, et al.

Yuan Xu is currently a Lecturer of School of Electrical Engineering with the University of Jinan, Jinan, China. He received the B.S. degree in Automation from Shandong Polytechnic University in 2007, and the M.S. degree in Detection Technology and Automation Device from Shandong Polytechnic University in 2010. He received the Ph.D. degree in Instrument Science from Southeast University in 2014.