

Image classification based on convolutional neural networks with cross-level strategy

Yu Liu¹ · Baocai Yin¹ · Jun Yu¹ · Zengfu Wang^{1,2}

Received: 10 November 2015 / Revised: 23 March 2016 / Accepted: 14 April 2016 /
Published online: 1 August 2016
© Springer Science+Business Media New York 2016

Abstract In the past few years, convolutional neural networks (CNNs) have exhibited great potential in the field of image classification. In this paper, we present a novel strategy named cross-level to improve the existing networks' architecture in which different levels of feature representation in a network are merely connected in series. The basic idea of cross-level is to establish a convolutional layer between two nonadjacent levels, aiming to extract more sufficient features with multiple scales at each feature representation level. The proposed cross-level strategy can be naturally integrated into an existing network without any change on its original architecture, which makes it very practical and convenient. Four popular convolutional networks for image classification are employed to illustrate its implementation in detail. Experimental results on the dataset adopted by the ImageNet Large-Scale Visual Recognition Challenge (ILSVRC) verify the effectiveness of the cross-level strategy on image classification. Furthermore, a new convolutional network with cross-level architecture is presented to demonstrate the potential of the proposed strategy in future network design.

Keywords Convolutional neural networks (CNNs) · Image classification · Network architecture · Feature representation · Deep learning

✉ Zengfu Wang
zfwang@ustc.edu.cn

¹ Department of Automation, University of Science and Technology of China, Hefei 230027, People's Republic of China

² Institute of Intelligent Machines, Chinese Academy of Sciences, Hefei 230031, People's Republic of China

1 Introduction

As an important issue in the field of computer vision, image classification has achieved great progress in the past decade, which is primarily driven by the ever-increasing demand of image retrieval technique on the internet. Many worldwide competitions on image classification have been carried out, such as the Pattern Analysis, Statistical Modelling and Computational Learning, Visual Object Classes (PASCAL VOC) Challenge from 2005 to 2012 and ImageNet Large-Scale Visual Recognition Challenge (ILSVRC) since 2010. In recent years, a variety of image classification methods have been proposed in the literature. Conventional image classification methods are usually based on manually designed feature descriptors, such as Scale-Invariant Feature Transform (SIFT) [26] and Histogram of Oriented Gradients (HOG) [6]. Generally, these methods consist of three main steps. First, extract local feature descriptors like SIFT and HOG. Then, Encode the extracted features with some linear or non-linear transformations. This step is usually known as feature selection, which contains some sub-steps such as dictionary learning, feature coding and spatial pooling. Some popular approaches used for feature selection include Bag of Words (BoW) [27, 30], Spatial Pyramid Matching (SPM) [17], Locality-constrained Linear Coding (LLC) [34], Sparse coding Spatial Pyramid Matching (ScSPM) [35], Nearest Neighbor Basis Vectors Spatial Pyramid Matching (NNBVSPM) [25], etc. Finally, employ some classifiers like Support Vector Machine (SVM) [5] and Adaptive Boosting (AdaBoost) [9] to classify input images. This category of methods can work well when the scale of dataset is not very large, such as Scene-15 [17] and Caltech-101 [22]. However, these methods usually obtain unsatisfactory performance when the dataset (e.g., ImageNet [14]) has a large number of categories and each category contains too many images varying greatly in terms of camera viewpoint, object pose, illumination and occlusion. The performance of visual object recognition has achieved a dramatic improvement since convolutional neural networks (CNNs) [18–20] were first introduced into image classification by Krizhevsky et al. [16] in 2012. In the last three years, various CNN-based classification approaches have been presented [10, 23, 33, 36], and the latest method [11] can even surpass the human-level performance.

As one of the most representative deep learning models, CNN is designed for hierarchical data/feature representation mechanism from lower level to higher level. Specifically, CNN is a trainable multi-stage architecture and each stage consists of a certain number of *feature maps*. The feature maps at each stage indicate a level of feature representation. The feature maps at a certain stage are obtained from the maps at its previous stage through several operations such as linear convolution, non-linear activation and spatial pooling. In this paper, to make the following descriptions clearer, we use the term *layer* to specially denote a certain operation between two adjacent levels of feature maps, and the term *level* to indicate the data representation stage which is characterized by a set of feature maps. A typical CNN architecture for visual recognition is shown in Fig. 1. Different from traditional three-phase (feature extraction, feature selection and classification) recognition workflow, it can be seen that the CNN-based framework is an end-to-end system since the input is a three-channel color image and the output is a label vector that can be directly used for classification task. In Fig. 1, there are three convolutional layers, two max-pooling layers and two fully-connected layers between the two ends. The convolutional layers here include 3D linear convolution and pointwise non-linear activation such as $\tanh(x)$ and $\max(0, x)$. The non-linear activation layer using the latter one is known as Rectified linear units (ReLU). In this work, since a convolutional layer in a CNN is usually followed by a non-linear layer like ReLU, the non-linear layer will not be explicitly mentioned later. The max-pooling

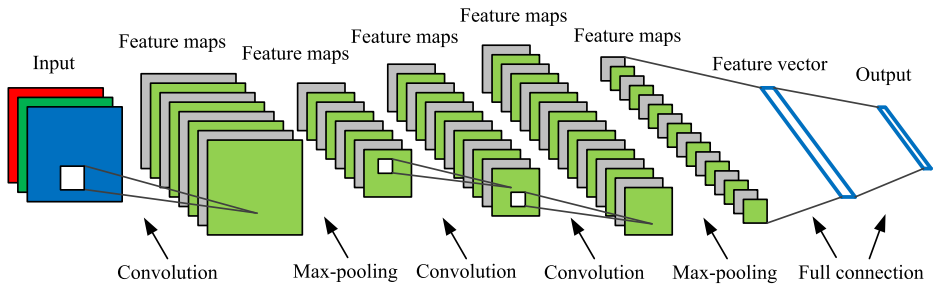


Fig. 1 A typical Convolutional neural network architecture visual recognition

layer aims to reduce data dimension by sub-sampling. The fully-connected layers usually exist at the output end, which can be viewed as the classification part in the whole system.

Historically, convolutional network was first applied to visual object recognition by LeCun et al. [18], in which the problem of handwritten digit recognition was well tackled by a network containing two convolutional layers and two fully-connected layers. However, this method did not obtain enough attentions in generalized visual recognition for a long time, until the rise of deep learning theory [1, 12] as well as the huge improvement on the computation capacity of hardware. Starting with the AlexNet [16], many representative CNN architectures such as Network in Network (NIN) [23], VGG-Net [29] and GoogLeNet [33] have been proposed in the literature. The existing CNNs share similar architectures, namely, convolutional layers for feature extraction and spatial pooling layers like max-pooling for dimension reduction. Different levels of representation in a network are merely connected in series. In other words, each layer only locates between two adjacent levels, and there is no layer or direct connection between two nonadjacent levels. Figure 2a shows the core structure of existing CNNs. However, the connection mechanism of visual neurons is generally believed to be very complex from the perspective of visual neuroscience [7, 31].

In this paper, we mainly argue that the existing serial connection approach can be improved by adding a direct connection between two nonadjacent levels. Specifically, a convolutional layer is established between two nonadjacent levels to realize this idea. This strategy is logically named *cross-level*, and it can be naturally integrated into an existing convolutional network without any change on its original architecture. The illustration of cross-level strategy is shown in Fig. 2b. The primary motivation of this strategy is to extract more sufficient features with multiple scales at each feature representation level to pursue a better performance on image classification. Therefore, this work can be grouped into the

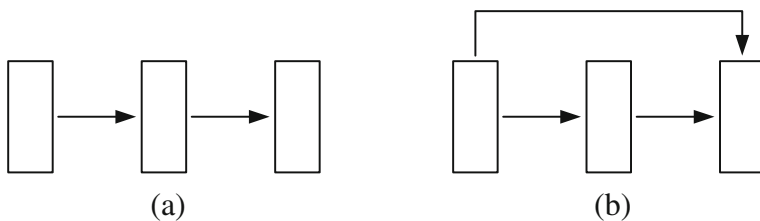


Fig. 2 Comparison of (a) Conventional structure of CNN and (b) the improved structure with cross-level strategy. Note that a *block* denotes a level of representation and an *arrow* denotes some operational layers between two levels

strategy-based contributions in the field of deep learning, which aim at providing universal approaches to enhance the performance of existing networks by modifying their architectures. Some strategy-based works include dropout [13], deeply-supervised nets (DSN) [21], etc. A preliminary version of this work appeared in our conference paper [24], where some backgrounds are not well introduced and the experiments conducted are quiet limited. This article will provide a more exhaustive presentation of this work. In particular, more experimental results are exhibited to further verify the effectiveness of the proposed cross-level strategy.

The rest of this paper is organized as follows. In Section 2, some basic knowledge about CNN and four popular CNN architectures for image classification are reviewed. The implementation details of the cross-level strategy are presented in Section 3. The experimental results are given in Section 4. Finally, Section 5 concludes the paper and puts forward some future work.

2 Related work

2.1 Convolutional neural network

In [19], LeCun pointed out that three architectural ideas of convolutional neural networks are: *local receptive fields*, *shared weights* and *sub-sampling*. The idea of local receptive fields means connecting units to local regions on the input, i.e., local convolutional operation is required. Moreover, the weights of a convolutional kernel is spatially invariant, which means that the feature maps is convoluted by one kernel. These two ideas significantly reduce the number of free parameters in the network, ensuring that a deep network is trainable. The sub-sampling operation is now known as pooling, which is mainly used for dimension reduction. Therefore, in a CNN architecture, convolution and pooling are two basic operations. Let x^i and y^j denote the i -th input feature map and j -th output feature map of a convolutional layer, respectively. The convolution operation is formulated as

$$y^j = \max(0, b^j + \sum_i k^{ij} * x^i), \quad (1)$$

where k^{ij} is the convolutional kernel between x^i and y^j , and b^j is the bias of y^j . The symbol $*$ denotes convolution operation. The ReLU nonlinearity is used here. Actually, supposing that the numbers of input and out feature maps are M and N , there are N 3D kernel of size $d \times d \times M$ used within this convolutional layer, where $d \times d$ is the kernel's spatial size. The max-pooling operation is expressed as

$$y_{r,c}^i = \max_{0 \leq m, n < s} \{x_{r-s+m, c-s+n}^i\}, \quad (2)$$

where $y_{r,c}^i$ is the neuron (r, c) in the i -th output map of a max-pooling layer. It is obtained by choosing the maximal value over an $s \times s$ non-overlapping local region in the i -th input map x^i .

2.2 CNNs for image classification

In this subsection, we briefly review four representative convolutional neural networks presented for image classification in the last three years, which are the AlexNet [16], Network-in-Network (NIN) [23], VGG-Net [29] and GoogLeNet [33].

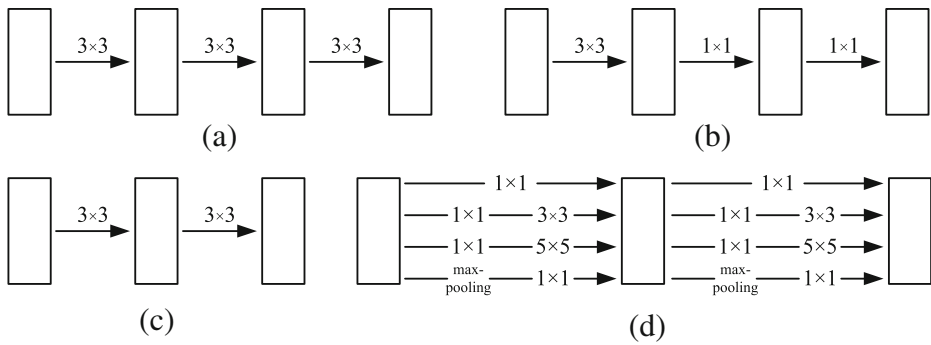


Fig. 3 Core structures of four popular CNNs. **a** AlexNet. **b** NIN. **c** VGG. **d** GoogLeNet

The AlexNet [16] proposed in 2012 can be viewed as a milestone in the field of image classification. It is the first time that CNN was employed for generalized image classification. The classification method based on AlexNet is the winner of ILSVRC 2012 with a significant breakthrough with respect to the previous approaches. The AlexNet reported in [16] contains five convolutional layers and three fully-connected layers. There is a local response normalization (LRN) layer that follows the first as well as the second convolutional layer. There are three max-pooling layers in AlexNet. The first two follow the two LRN layers, respectively. The last max-pooling layer follows the fifth convolutional layer. The core structure of AlexNet locates between the second and third max-pooling layers, which contains three convolutional layers each with 3×3 convolution kernel. Four levels of feature maps of spatial size 13×13 are connected by these three convolutional layers. The authors reported in [16] that the removal of any of these layers leads to a loss of about 2 % in terms of top-1 performance. The core structure of AlexNet is shown in Fig. 3a.

Lin et al. [23] proposed NIN to obtain a better representation of local patches by adding a multi-layer perceptron after a convolutional layer. In their method, they use a three-layer perceptron, and it is essentially equivalent to add two 1×1 convolutional layers after a 3×3 or 5×5 convolutional layer. Thus, the core structure or unit of NIN has three convolutional layers in series, as shown in Fig. 3b. The network applied in [23] has four such units and there is a max-pooling layer between every two units. Furthermore, after the last three-layer convolution unit, instead of employing traditional fully-connected layers, the authors generate one feature map for each class and use the global average pooling scheme to obtain the resulting vector, which can reduce the number of parameters to a great extent and prevent overfitting for neural networks.

Simonyan and Zisserman [29] from the Visual Geometry Group at University of Oxford proposed several deep convolutional networks ranging from 11 to 19 weight layers. They named their networks VGG based on their research group's name. Each proposed VGG network has five max-pooling layers. Between two adjacent max-pooling layers, these networks usually contain two or three 3×3 convolutional layers, which constructs a core unit of the VGG-Net. For example, the VGG-11 net contains three such units which consists of two 3×3 convolutional layers. Figure 3c shows the core unit of the VGG-Net. The number of feature maps gradually increases with the increasing of feature representation level. After the last max-pooling layer, all the VGG networks contain three fully-connected layers with 4096, 4096 and 1000 neurons, respectively. Thus, the model size (number of parameters) of VGG-NET is quiet large even for the VGG-11 network.

GoogLeNet, a 22-layer deep convolutional network proposed by Szegedy et al. [33], is the winner of ILSVRC 2014 classification competition. Since increasing the depth of a network directly needs a sharp increasing requirement of computational resources and tends to cause severe overfitting, the GoogLeNet is designed to make a balance between the network size and computational budget. The core structure adopted in GoogLeNet is called *Inception*. Figure 2c shows two serial Inceptions. In each Inception, the feature maps at the output level are obtained from four branches, namely, a 1×1 convolutional layer, a 3×3 convolutional layer with a 1×1 layer for parameter reduction, a 5×5 convolutional layer with a 1×1 layer for parameter reduction, and a max-pooling layer followed by a 1×1 layer to limit the number of output feature maps for parameter reduction at the next level. It is worthwhile to note that the intermediate feature maps generated in the last three branches do not construct a level of representation since those three 1×1 layers are essentially designed for parameter reduction. Therefore, there are only three levels of representation in Fig. 3d. In GoogLeNet, there are totally nine Inceptions which are separated into three parts. The first part and last part both have two Inceptions just like the illustration given in Fig. 3d. The middle part has five Inceptions in series. Moreover, there is no max-pooling layer within each of the three parts, so all the feature maps within each part have the same spatial size. In GoogLeNet, there exists a max-pooling layer between every two parts for dimension reduction of feature maps.

3 Cross-level

In this section, we mainly describe the implementation details of the cross-level strategy via the above four convolutional networks, namely, the AlexNet [16], Network-in-Network (NIN) [23], VGG-Net [29] and GoogLeNet [33]. Figure 4 shows the improved structure of each network after applying the cross-level strategy. The basic idea of cross-level is to

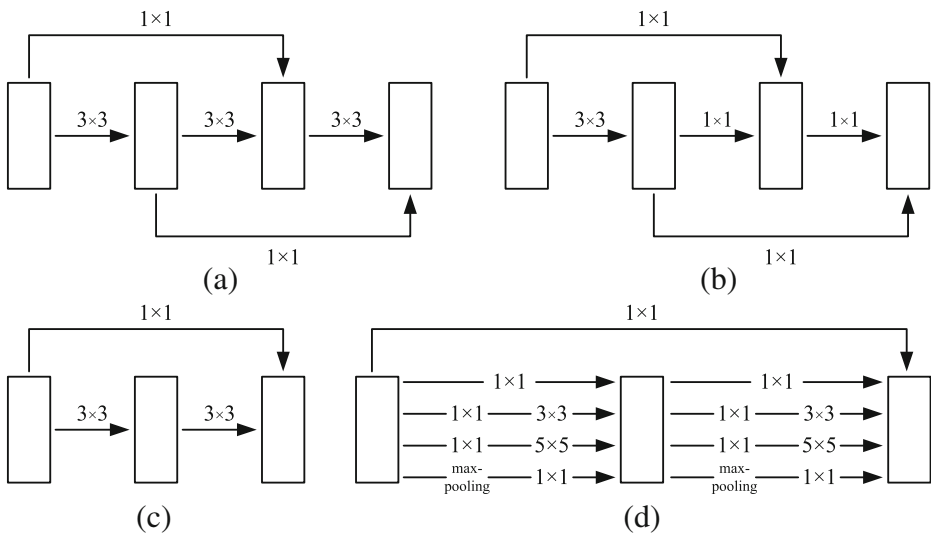


Fig. 4 The improved structure of four networks after applying the cross-level strategy. **a** AlexNet. **b** NIN. **c** VGG. **d** GoogLeNet

establish a convolutional layer between two nonadjacent levels. Naturally, the added convolutional layer can be called *cross layer*. Thus, the feature maps at the output level come from two aspects: the layers in the original structure and the cross layer. In our approach, considering the cost of computational resource, the size of convolution kernel in each cross layer is fixed to 1×1 , and the number of feature maps generated by a cross layer is universally set as half number of the original maps at that level.

As shown in Fig. 4a, for the AlexNet, two 1×1 convolutional layers are established from the first and second levels to the third and fourth levels, respectively. Notice that the core structure shown in Fig. 3a appears only once in the AlexNet, all the other parts of the network are not changed. The situation of NIN is similar to that of AlexNet, as shown in Fig. 4b. The only difference is that there are several core structures/units (see Fig. 3b) in the NIN architecture. For each unit except the first and last one, two 1×1 convolutional layers are added on the original structure. Thus, when there are four units [23], only four 1×1 layers are created on the second and third units, while the other parts in NIN remain unchanged. The improved structure of VGG-Net is shown in Fig. 4c, which is similar to the situation of AlexNet shown in Fig. 4a. In particular, when there are three 3×3 convolutional layers in each unit such as the VGG-16 network, the situation is exactly the same as the AlexNet. For the VGG-11 network, we add two 1×1 cross layers for its third and fourth units. Finally, Fig. 4d shows the modified structure of GoogLeNet with cross-level strategy, which connects the input level of the former Inception and the output level of the latter one with a 1×1 convolutional layer. As mentioned before, the GoogLeNet also contains a structure of five consecutive Inceptions. The cross-level strategy deals with this situation just using the same approach in AlexNet (see Fig. 4a) and NIN (see Fig. 4b). Accordingly, there are totally six 1×1 convolutional layers added on the original GoogLeNet after applying the cross-level strategy.

From the above four examples, it can be seen that the presented cross-level strategy can be easily applied to an existing convolutional network without changing its original architecture, and the depth of the modified network also remains the same. The only requirement is that all the feature maps within the two levels connected by a cross layer should have the same spatial size. That is to say, there must be no inside spatial pooling layers (e.g., max-pooling) with a stride larger than one.

It is worthwhile to notice that some existing CNN architectures have partly applied some strategies which are similar to the proposed cross-level strategy. Fan et al. [8] introduced a convolutional network with multiple paths for human tracking. In their method, the network between the first convolutional and the output layer is split into two branches, namely, global branch and local branch. The global branch is the same as traditional CNN architecture, which consists of several convolutional layers and pooling layers. The purpose of global branch is to enlarge the receptive field to address global structures. The local branch only has a convolutional layer, which aims to extract more details about local structures. Sermanet and LeCun [28] employed a similar multi-scale CNN architecture for traffic sign recognition. In [32], Sun et al. proposed a face verification method based on a convolutional network, in which the last hidden layer is connected with both the third and fourth convolutional layers. The main purpose of this design is to avoid the loss of useful information, since the fourth layer contains too few neurons. The networks used in the publications referred above are generally known as multi-scale CNNs. Although these networks have bypassing connections, there exist clear difference between them and the networks applying the proposed cross-level strategy. In the above multi-scale CNNs, bypassing connections are only linked to the output layer. Moreover, the main motivation using multi-scale CNNs is for specific object recognition such as human and face, in which features with different scales

are all required in the output layer. However, the target of the proposed cross-level strategy is generalized object classification [16, 23, 33], and the basic motivation of this strategy is to extract more features with different scales at each feature representation level, not just the output one. Therefore, the usage of the proposed cross-level strategy for convolutional network design is more flexible.

4 Experiments

4.1 Comparison with conventional CNNs

In this subsection, we compare the performance of conventional serial CNNs and their improved versions after applying the proposed cross-level strategy on image classification. The AlexNet [16], Network-in-Network (NIN) [23], VGG-11 [29] and GoogLeNet [33] are employed to verify the effectiveness of the cross-level strategy. In this work, we use the dataset adopted by ILSVRC (used for ILSVRC classification challenges from 2012 to 2014). As a subset of ImageNet dataset, it contains 1000 categories and each category has about 1300 images for training and 50 images for validation. Totally, there are about 1.28 million training images and 50000 validation images.

The experimental setup is similar to the approach reported in [16]. All the images are first down-sampled to a fixed spatial resolution of 256×256 and the mean intensity over the training set from each pixel is subtracted. All the models are learned using stochastic gradient descent algorithm. All the experiments are conducted on Caffe [2, 15], which is a popular deep learning framework created by Jia et al. The implementation files of AlexNet, NIN and GoogLeNet are publicly available on Caffe model zoo website [3]. Specifically, the “BVLC AlexNet” and “BVLC GoogLeNet” which have been integrated into Caffe-master toolbox are used for training AlexNet and GoogLeNet, respectively. The “Network in Network model” available on website [4] (derived from [3]) is adopted to train NIN. The VGG-11 network is implemented strictly based on the configuration provided in [29]. In our experiments, all the parameters used for training are set as default values reported in the configuration and solver files. The cross-level strategy is applied to these four networks by modifying the corresponding network configuration files. For a fair comparison, all the parameters with respect to model training remain the same with the original networks. For simplicity, the modified versions of these four networks are named AlexNet-Cross, NIN-Cross, VGG-11-Cross and GoogLeNet-Cross, respectively.

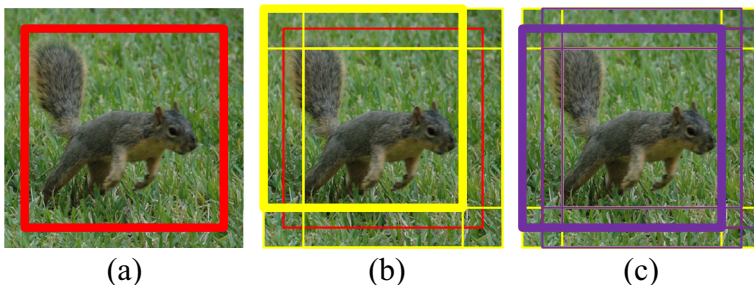


Fig. 5 Three cropping approaches for prediction. **a** 1-crop. **b** 10-crop, the upper-left crop is exhibited in *bold*. **c** 18-crop, the left-center crop is exhibited in *bold*

Table 1 The top-1 and top-5 accuracy rates of eight learned CNN models: 1-crop prediction

Model	Top-1	Top-5
AlexNet	56.48 %	79.56 %
AlexNet-Cross	57.37 %	80.52 %
NIN	59.42 %	81.60 %
NIN-Cross	60.56 %	82.61 %
VGG-11	64.79 %	86.12 %
VGG-11-Cross	65.04 %	86.39 %
GoogLeNet	68.93 %	88.90 %
GoogLeNet-Cross	70.28 %	90.08 %

The classification performance of each learned CNN model is evaluated with two commonly used measures, namely, the top-1 and top-5 accuracy rates. These two measures are both calculated using the validation image set since the images in the testing image set used in the final ILSVRC competition are with held-out class labels. The top-1 accuracy rate is the ratio of images whose ground truth category is exactly the prediction category with maximum probability, while the top-5 accuracy rate indicates the ratio of images whose ground-truth category is within the top-5 prediction categories (sorted by the probabilities).

For each test image, as shown in Fig. 5, we employ three cropping approaches which are 1-crop, 10-crop and 18-crop to obtain the prediction score. As shown in Fig. 5a, the 1-crop prediction approach only extracts the center patch of appropriate size such as 224×224 according to the spatial size of the network's input data layer for prediction. The 10-crop prediction approach [16] is mostly used in the literature, as shown in Fig. 5b, in addition to the center crop, four corner crops are also extracted for prediction. Moreover, the horizontal reflections of these five crops are also used for prediction, so there are totally 10 crops involved. The 18-crop prediction approach is similar to the 10-crop prediction one, as shown in Fig. 5c, another four crops locating at left-center, right-center, up-center and bottom-center as well as their flipped versions are added into the prediction set. For the latter two multi-crop prediction approaches, one prediction result (a 1000-dimensional vector) is obtained from one crop by the network's softmax layer, so a merging strategy is required to get the final prediction score. To this end, we adopt two popular strategies: averaging and choosing-max. The former one calculates the average score over all the crops' scores, while the latter one constructs the final score by choosing the maximal element over all the inputs

Table 2 The top-1 and top-5 accuracy rates of eight learned CNN models: 10-crop prediction

Model	Top-1(ave)	Top-1(max)	Top-5(ave)	Top-5(max)
AlexNet	59.09 %	58.97 %	81.67 %	81.27 %
AlexNet-Cross	59.85 %	59.62 %	82.18 %	81.69 %
NIN	61.35 %	61.21 %	83.08 %	82.61 %
NIN-Cross	62.44 %	62.41 %	84.01 %	83.62 %
VGG-11	67.02 %	66.76 %	87.56 %	87.05 %
VGG-11-Cross	67.29 %	67.04 %	87.80 %	87.27 %
GoogLeNet	70.35 %	70.39 %	89.84 %	89.66 %
GoogLeNet-Cross	71.58 %	71.59 %	90.76 %	90.64 %

Table 3 The top-1 and top-5 accuracy rates of eight learned CNN models: 18-crop prediction

Model	Top-1(ave)	Top-1(max)	Top-5(ave)	Top-5(max)
AlexNet	58.98 %	58.97 %	81.65 %	81.18 %
AlexNet-Cross	59.76 %	59.73 %	82.17 %	82.61 %
NIN	61.29 %	61.29 %	83.15 %	82.63 %
NIN-Cross	62.52 %	62.44 %	84.01 %	83.61 %
VGG-11	66.98 %	66.80 %	87.58 %	87.02 %
VGG-11-Cross	67.27 %	67.06 %	87.81 %	87.22 %
GoogLeNet	70.33 %	70.39 %	89.82 %	89.61 %
GoogLeNet-Cross	71.64 %	71.60 %	90.76 %	90.65 %

at each dimension. Let s_1, s_2, \dots, s_N denote the prediction scores from N different crops, the final score calculated using the averaging strategy is

$$s_{ave} = \frac{1}{N} \sum_{i=1}^N s_i, \quad (3)$$

and the j -th element of choosing-max prediction score s_{max} is

$$s_{max,j} = \max\{s_{i,j}\}, i = 1, 2, \dots, N, j = 1, 2, \dots, 1000. \quad (4)$$

Table 1 lists the top-1 and top-5 accuracy rates of eight learned CNN models using 1-crop prediction. For AlexNet, NIN and GoogLeNet, it can be seen from Table 1 that the cross-level strategy results in a rise of about 1 % in terms of both top-1 and top-5 accuracy rates. In particular, the performance improvement of GoogLeNet is the most significant. From our perspective, this is mainly because the proportion of levels which are influenced by the cross-level strategy in GoogLeNet is the highest among these three networks. The improvement of VGG-11 is relatively small (about 0.25 %) since the proportion of the influenced level is small, but the progress is still considerable in image classification task. We will show later that the number of weights in VGG-11-Cross just increases slightly with respect to its original network VGG-11.

Tables 2 and 3 list the classification performance of eight CNN models using 10-crop and 18-crop prediction, respectively. The results with averaging and choosing-max strategies are both provided. Comparing with the results in Table 1, we can see that all the measured accuracy rates clearly increase after applying multi-crop prediction approach. The difference between the performances of 10-crop and 18-crop prediction is generally small for all the eight models. Moreover, the averaging strategy usually outperforms the choosing-max strategy in terms of both top-1 and top-5 accuracy rates. Most importantly, for all the four architectures, the improvement on all the measured rates after applying the proposed

Table 4 The size of physical memory taken by the above eight CNN models

Network	without cross-level	with cross-level	ratio
AlexNet	243.9MB	320.6MB	1.314
NIN	30.4MB	40.4MB	1.329
VGG-11	531.5MB	539.0MB	1.014
GoogLeNet	53.5MB	64.5MB	1.206

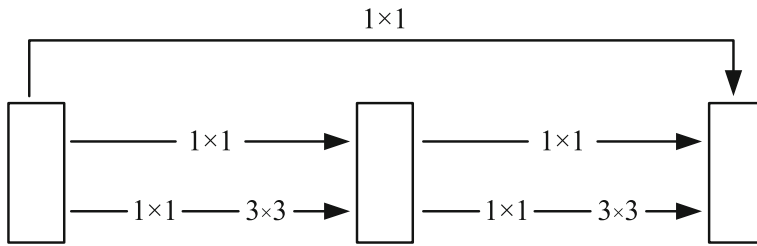


Fig. 6 The core structure of the new designed network

cross-level strategy is still very obvious, which is generally similar to the situation shown in Table 1.

Table 4 lists the size of physical memory taken by the above eight CNN models. For AlexNet, NIN and GoogLeNet, the size of a CNN model increases about 20 %–30 % after applying the cross-level strategy, which is generally acceptable in practice. For VGG-11, the increasing percentage is less than 2 %, but the classification accuracy of the VGG-11 network also obtains a growth of more than 0.2 % on each measure after applying the cross-level strategy, as listed in the previous three tables. This further confirms the effectiveness of our cross-level strategy for the heavy models. As mentioned before, the number of feature maps generated by a cross layer is normally set as half number of the original maps in that level in our method. When the output of a cross layer is connected to a fully-connected layer (e.g., the AlexNet), the increased 50 % number of feature maps will have a considerable effect on the final model size, which is the main factor that increases the model size. In this work, due to the reason that training a CNN model is very time-consuming, the increased percentage of feature maps and the position that the a cross layer added are not fully studied. This issue will be further studied in the future.

4.2 Application to new network design

In addition to the existing networks, the cross-level strategy can be also used for the design of new networks. To verify this point, as well as to further demonstrate the effectiveness of the cross-level strategy from another point of view, we design a new CNN architecture by referring to the GoogLeNet [33]. Specifically, we just remove two branches in the Inception of GoogLeNet, while all the other structures remain the same, mainly including the depth of network and the number of feature maps each branch generates. We apply the cross-level strategy to this new network just as the way to the GoogLeNet. The core structure of the designed network is shown in Fig. 6, in which only the 1×1 and 3×3 branches are

Table 5 The classification performance of the new designed network

Measure	GoogLeNet	New network
1-view: top-1	68.93 %	68.74 %
1-view: top-5	88.90 %	88.78 %
10-view: top-1(ave)	70.35 %	70.26 %
10-view: top-5(ave)	89.84 %	89.81 %
Model size	53.5MB	48.2MB

preserved. The training conditions and testing approaches adopted to this new network are the same as those to the GoogLeNet.

Table 5 lists the classification performance of the new network as well as the GoogLeNet for comparison. We can see from Table 5 that the performance of this new network is very close to that of GoogLeNet, but the model size decreases by about 10 % from 53.5MB to 48.2MB.

5 Conclusion

In this paper, we present a novel strategy called cross-level for CNN-based image classification. The basic idea is to establish a convolutional layer between two nonadjacent levels in the network, which aims to learn more sufficient feature representations for a better classification performance. Experimental results on four popular convolutional neural networks demonstrate the effectiveness of the proposed cross-level strategy. We also exhibit the potential of the cross-level strategy used for the design of new networks. In the future, we will conduct more experiments to further study the impact of the increased percentage of feature maps and the position that the a cross layer added. Furthermore, we will also verify the effectiveness of cross-level strategy used for other CNN-based vision applications, such as object detection and face recognition.

Acknowledgments The authors would like to thank the editors and anonymous reviewers for their constructive comments and valuable suggestions. This work was supported by the National Natural Science Foundation of China (No. 61472393 and No. 61303150), the National Science and Technology Major Project of the Ministry of Science and Technology of China (No. 2012GB102007), and the Anhui Province Initiative Funds on Intelligent Speech Technology and Industrialization (No. 13Z02008). The authors greatly acknowledge the support of IFLYTEK CO.,LTD.

Compliance with Ethical Standards

Conflict of interests The authors declare that they have no conflict of interest.

References

1. Bengio Y, Courville A, Vincent P (2013) Representation learning: a review and new perspectives. *IEEE Trans Pattern Anal Mach Intell* 35:1798–1828
2. Caffe website: <http://caffe.berkeleyvision.org>
3. Caffe model zoo: http://caffe.berkeleyvision.org/model_zoo.html
4. Caffe model zoo wiki page: <https://github.com/BVLC/caffe/wiki/Model-Zoo>
5. Cortes C, Vapnik V (1995) Support-vector networks. *Mach Learn* 20(3):273–297
6. Dalal N, Triggs B (2005) Histograms of oriented gradients for human detection. In: *IEEE Conference on computer vision and pattern recognition (CVPR)*, vol 1, pp 886–893
7. Desimone R, Duncan J (1995) Neural mechanisms of selective visual attention. *Ann Rev Neurosci* 18:193–222
8. Fan J, Xu W, Wu Y, Gong Y (2010) Human tracking using convolutional neural networks. *IEEE Trans Neural Netw* 21:1610–1623
9. Freund Y, Schapire R (1995) A decision-theoretic generalization of on-line learning and an application to boosting. In: *Computational learning theory*, pp 23–37
10. He K, Zhang X, Ren S, Sun J (2014) Spatial pyramid pooling in deep convolutional networks for visual recognition. In: *European conference on computer vision (ECCV)*, pp 346–361
11. He K, Zhang X, Ren S, Sun J (2015) Delving deep into rectifiers: surpassing human-level performance on imageNet classification. arXiv:1502.01852

12. Hinton GE, Salakhutdinov RR (2006) Reducing the dimensionality of data with neural networks. *Science* 313:504–507
13. Hinton GE, Srivastava N, Krizhevsky A, Sutskever I, Salakhutdinov RR (2012) Improving neural networks by preventing co-adaptation of feature detectors. arXiv:1207.0580
14. ImageNet Website: <http://www.image-net.org/>
15. Jia Y, Shelhamer E, Donahue J, Karayev S, Long J, Girshick R, Guadarrama S, Darrell T (2014) Caffe: convolutional architecture for fast feature embedding. In: ACM International conference on multimedia, pp 675–678
16. Krizhevsky A, Sutskever I, Hinton GE (2012) ImageNet classification with deep convolutional neural networks. In: Advances in neural information processing systems (NIPS), vol 25, pp 1106–1114
17. Lazebnik S, Schmid C, Ponce J (2006) Beyond bags of features: spatial pyramid matching for recognizing natural scene categories. In: IEEE Conference on computer vision and pattern recognition (CVPR), vol 2, pp 2169–2178
18. LeCun Y, Boser B, Denker JS, Henderson D, Howard RE, Hubbard W, Jackel LD (1989) Backpropagation applied to handwritten zip code recognition. *Neural Comput* 1:541–551
19. LeCun Y, Bottou L, Bengio Y, Haffner P (1998) Gradient-based learning applied to document recognition. *Proc IEEE* 86:2278–2324
20. LeCun Y, Kavukcuoglu K, Farabet C (2010) Convolutional networks and applications in vision. In: IEEE International symposium on circuits and systems, pp 254–256
21. Lee C, Xie S, Gallagher P, Zhang Z, Tu Z (2014) Deeply-supervised networks. arXiv:1409.5185
22. Li FF, Fergus R, Perona P (2007) Learning generative visual models from few training examples: an incremental bayesian approach tested on 101 object categories. *Comput Vis Image Understand* 106:59–70
23. Lin M, Chen Q, Yan S (2013) Network in network. arXiv:1312.4400
24. Liu Y, Yin B, Yu J, Wang Z (2015) Cross-level: a practical strategy for convolutional neural networks based image classification. In: CCF Chinese conference on computer vision, pp 398–406
25. Long X, Lu H, Li W (2014) Image classification based on nearest neighbor basis vectors. *Multimed Tools Appl* 71:1559–1576
26. Lowe D (2004) Distinctive image features from scale-invariant keypoints. *Int J Comput Vis* 60:91–110
27. Qu Y, Wu S, Liu H, Xie Y, Wang H (2014) Evaluation of local features and classifiers in BOW model for image classification. *Multimed Tools Appl* 70:605–624
28. Sermanet P, LeCun Y (2011) Traffic sign recognition with multi-scale convolutional networks. In: International joint conference on neural networks, pp 2809–2813
29. Simonyan K, Zisserman A (2014) Very deep convolutional networks for large-scale image recognition. arXiv:1409-1556
30. Sivic J, Zisserman A (2003) Video google: a text retrieval approach to object matching in videos. In: International conference on computer vision (ICCV), pp 1470–1477
31. Spirkovska L, Reid MB (1992) Robust position, scale, and rotation invariant object recognition using higher-order neural networks. *Pattern Recog* 25:975–985
32. Sun Y, Wang X, Tang X (2014) Deep learning face representation from predicting 10,000 classes. In: IEEE International conference on computer vision and pattern recognition (CVPR), pp 1891–1898
33. Szegedy C, Liu W, Jia Y, Sermanet P, Reed S, Anguelov D, Erhan D, Vanhoucke V, Rabinovich A (2014) Going deeper with convolutions. arXiv:1409-4842
34. Wang JJ, Yang JC, Yu K, Lv FJ, Huang T, Gong YH (2010) Locality-constrained linear coding for image classification. In: IEEE Conference on computer vision and pattern recognition (CVPR), pp 3360–3367
35. Yang JC, Yu K, Gong YH, Huang T (2009) Linear spatial pyramid matching using sparse coding for image classification. In: IEEE Conference on computer vision and pattern recognition (CVPR), pp 1794–1801
36. Zeiler MD, Fergus R (2014) Visualizing and understanding convolutional networks. In: European conference on computer vision (ECCV), Part I, pp 818–833



Yu Liu received his B.S. degree in automation from University of Science and Technology of China, in 2011. He is currently a Ph.D. candidate in the Department of Automation at University of Science and technology of China. His research interests include image processing, computer vision and information fusion. He is a student member of the IEEE Signal Processing Society.



Baocai Yin received his B.S. and M.S. degrees in department of automation, University of Science and Technology of China, in 2012 and 2015. He is currently a research assistant at University of Science and Technology of China. His research interests include computer vision and machine learning.



Jun Yu received his Ph.D. degree from the University of Science and Technology of China, Hefei, China, in 2010. He is a research associate of the Department of Automation, University of Science and Technology of China. His research interests include human computer interaction and intelligent robot. He has published more than 50 journal and conference papers. He is a member of the IEEE Signal Processing Society. He has undertaken 8 programs from Natural Science Foundation of China, China Postdoctoral Science Foundation, etc.



Zengfu Wang was born in 1960. He received his B.S. degree in electronic engineering from University of Science and Technology of China, in 1982 and his Ph.D. degree in control engineering from Osaka University, Japan, in 1992. He is currently a professor of both Institute of Intelligent Machines, Chinese Academy of Sciences and University of Science and Technology of China. His research interests include computer vision, human computer interaction and intelligent robots. He has published more than 180 journal articles and conference papers.