On Generalization of Deep Neural Networks for Visual Recognition Tasks

Takayuki Okatani

okatani@vision.is.tohoku.ac.jp
Graduate School of Information Sciences, Tohoku University / RIKEN Center for AIP, Aramaki-Aza-Aoba, Sendai, Japan

Keywords: Deep learning, Computer vision, Image restoration, Out-of-distribution detection

ABSTRACT

This article discusses generalization ability of deep neural networks (DNNs) for visual recognition. It is known that DNNs easily fail for images to which noises are added, when they have not learned the noisy images. We discuss how to cope with such limitation of DNNs.

1 INTRODUCTION

The emergence of convolutional neural networks has reshaped research in the field of computer vision in the past years. Their employment has brought about solutions to unsolved problems or contributed to (sometimes significant) improvements in performance (e.g., inference accuracy, computational speed etc.). It was claimed in the past years that CNNs can even surpass human vision in several visual recognition tasks, in particular, the task of object category classification.

However, we should be precise about the meaning (or underlying condition) of such claims. Each of them is made based on experiments that compare CNNs and humans on a recognition task using a *particular* dataset. The experimental results merely indicate that CNNs are better in terms of recognition accuracy than humans on a *closed* set of test inputs. In other words, CNNs may correctly classify inputs that are sampled from the same distribution as the training data they have learned but will wrongly classify inputs sampled from a different distribution. The two distributions usually need to be very close; their difference is called *domain shift,* which is known as one of major causes that impede applications of CNNs to real-world problems.

This article considers how to cope with this issue with a particular focus on the case where the domain shift occurs due to image distortion. We encounter various types of distorted images such as noisy images in the real world. CNNs trained only using clean images for visual recognition tasks will not work properly for such distorted images due to the aforementioned domain shift. We will discuss how we can make CNNs recognize distorted images properly in what follows.

2 METHODS

2.1 Approaches based on Image restoration

There are three approaches to visual recognition from distorted images. The first approach, which is conceptually the simplest, is to train the CNN using not only clean images but noisy images. Then, the CNN will accurately recognize noisy inputs. However, this approach is often impossible to employ, since it requires to have training data of distorted images (i.e., noisy images in the aforementioned case), which need to be given labels (i.e., material or object categories), as well as to perform training on a larger dataset.

The second approach is to make the CNN more robust to image distortion, so that it can correctly recognize distorted images even though it is trained only on clean images. This may be the most difficult one of the three approaches. In fact, there is only a few studies pursuing this approach. For example, our study [1] shows that a type of activation functions mitigates decrease in recognition accuracy due to distortion of input images. However, the improvements are limited for real-world applications.

The third approach is to use another CNN model to restore quality of input images with distortions. We insert this CNN for image restoration before the CNN for classification (or other purposes); the first CNN estimate clean version of the input distorted image, which is fed to the second CNN for classification. We can use the CNN trained only on clean images for the second CNN. Instead, it is necessary to train the second CNN, which requires pairs of a distorted image and its clean version. On the other hand, it is not necessary to give the input distorted images labels for classification, which is advantageous. Moreover, the cost for creating the training data for image restoration (i.e., the second CNN) tends to be lower than the classification task. We have shown through several studies that this approach works well, where the primary concern is how to restore the original image with better quality.

2.2 Better CNN architecture for image restoration

In [2], we pursue better architectural design of networks, particularly the design that can be shared across different distortion types. We pay attention to the effectiveness of paired operations on various image processing tasks. In [4], it is shown that a CNN iteratively performing a pair of up-sampling and down-sampling contributes to performance improvement for image super-resolution. In [5], Suganuma et al. employ evolutionary computation to search for a better design of convolutional autoencoders for several tasks of image

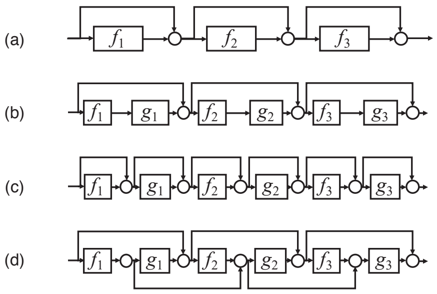

Figure 1. Different designs of residual networks with basic modules.

restoration, showing that network structures repeatedly performing a pair of convolutions with a large- and small-size kernels (e.g., a sequence of conv. layers with kernel size 3, 1, 3, 1, 5, 3, and 1) perform well for image denoising.

Assuming the effectiveness of such repetitive paired operations, we consider implement them in deep networks to exploit their potential; we are specifically interested in how to integrate them with the structure of residual networks. The basic structure of residual networks is shown in Fig.1(a), which have become an indispensable component for the design of modern deep neural networks.

There have been several explanations for the effectiveness of the residual networks. A widely accepted one is the “unraveled” view proposed by Veit et al. [7]: a sequential connection of n residual blocks is regarded as an ensemble of many sub-networks corresponding to its implicit 2^n paths. A network of three residual blocks with modules f_1, f_2, and f_3, shown in Fig.1(a), has $2^3 = 8$ implicit paths from the input to output, i.e., $f_1 \rightarrow f_2 \rightarrow f_3$ $f_1 \rightarrow f_2$, $f_1 \rightarrow f_3$, $f_2 \rightarrow f_3$, f_1, f_2, f_3 and 1. Veit et al. also showed that each block works as a computational unit that can be attached/detached to/from the main network with minimum performance loss.

Considering such a property of residual networks, how should we use residual connections for paired operations? Denoting the paired operations by f and g the most basic construction will be to treat (f_i, g_i) as a unit module, as shown in Fig.1(b). In this connection style, f_i and g_i are always paired for any i in the possible paths. Then, we consider another connection style shown in Fig.1 (d), dubbed “dual residual connection.” This style enables to pair f_i and g_j for any i and j such that $i \leq j$. In the example of Fig.1(d), all the combinations of the two operations emerge in the possible paths. We conjecture that this increased number of potential interactions between $\{f_i\}$ and $\{g_j\}$ will contribute to improve performance for image restoration tasks. Note that it is guaranteed that the components f’s and g’s are always paired in the possible paths. This is not the case with other connection styles such as the one depicted in Fig.1(c).

We call the building block for implementing this dual residual connections *Dual Residual Block* (DuRB); see Fig.2. DuRB is a generic structure that has two containers for the paired operations, and the users choose two operations for them. For each task, we specify the paired operations of DuRBs as well as the entire network. Our experimental results support the effectiveness of our approach.

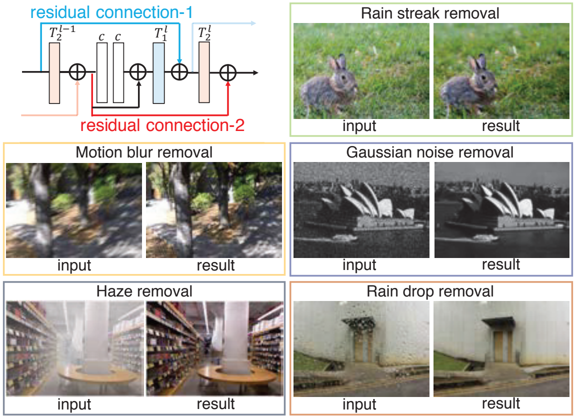

Figure 2. Upper left: structure of the dual residual block. Other panels: results of five image restoration tasks.

2.3 Handling images with unknown distortion types

There are many types of image distortion, such as Gaussian/salt-and-pepper/shot noises, defocus/motion blur, compression artifacts, haze, raindrops, etc. Then, there are two application scenarios for image restoration methods. One is the scenario where the user knows what image distortion that he/she wants to remove; an example is a deblurring filter tool implemented in a photo editing software. The other is the scenario where the user does *not* know what distortion(s) the image undergoes but wants to improve its quality, e.g., applications to vision for autonomous cars and surveillance cameras.

In [3], we consider the latter application scenario. Most of the existing studies are targeted at the former scenario, and they cannot be directly applied to the latter. Considering that real-world images often suffer from a combination of different types of distortion, we need image restoration methods that can deal with combined distortions with unknown mixture ratios and strengths.

There are few works dealing with this problem. A notable exception is the work of Yu et al. [6], which proposes a framework in which multiple lightweight CNNs are trained for different image distortions and are adaptively applied to input images by a mechanism learned by deep reinforcement learning. Although their method is shown to be effective, we think there is room for improvements. One is its limited accuracy; the accuracy improvement gained by their method is not so large, as compared with application of existing methods for a single type of distortion to images with combined distortions. Another is its inefficiency; it uses multiple distortion specific CNNs in parallel, each of which also

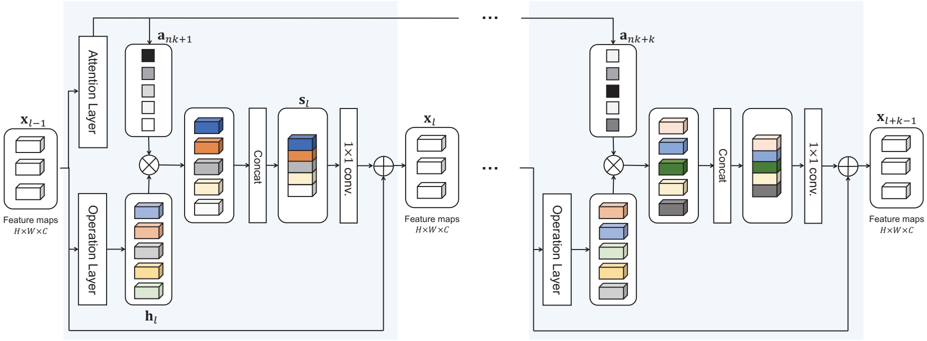

Figure 3. Our operation-wise attention layer designed for restoration of images with unknown combined distortion factors.

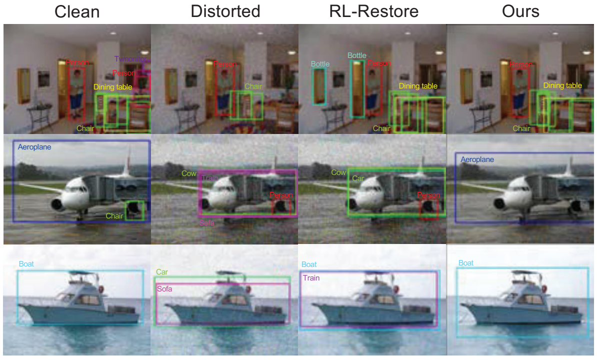

Figure 4. Examples of image restoration and subsequent object detection. Our method can restore high-quality images from distorted ones with unknown distortion types, resulting in better object detection.

needs pretraining.

In [3], we show that a simple attention mechanism can better handle aforementioned combined image distortions. We design a layer that performs many operations in parallel, such as convolution and pooling with different parameters; see Fig.3. We equip the layer with an attention mechanism that produces weights on these operations, intending to make the attention mechanism to work as a switcher of these operations in the layer. Given an input feature map, the proposed layer first generates attention weights on the multiple operations. The outputs of the operations are multiplied with the attention weights and then concatenated, forming the output of this layer to be transferred to the next layer. We call the layer *operation-wise attention* layer. This layer can be stacked to form a deep structure, which can be trained in an end-to-end manner by gradient descent; hence, any special technique is not necessary for training. We evaluate the effectiveness of our approach through several experiments. Examples of image restoration by this method followed by object detection are shown in Fig.4.

2.4 Universal network for unknown distortion typess

As explained above, it is desirable to enable to deal with input images with unknown degradation factor(s). It should be noted that a few studies including ours explained above tackled this problem. However, their performances are no so high; they are significantly lower than those of dedicated networks used in the ideal case when they are applied to images having the assumed degradation factor.

How can we restore images with unknown degradation factors to a higher level? We can think of two approaches. One is to build a universal network that can deal with any degradation factors; it is a network equipped with a single input receiving a degraded image and a single output yielding a restored image. However, such a design has difficulty. A very successful approach to image restoration so far is to have a network predict only the difference from the input image to the desired high-quality image. The residual images tend to have considerably different statistical properties for different factors, making it difficult to have a single output deal with them.

The other approach is to cascade multiple networks, each of which is a dedicated network for a single factor. It should perform well if each component network works well on the target factor and also does no harm on the other factors as well as restored clean images. This approach, however, will suffer from a high memory consumption; the total number of parameters in the cascade increases proportionally with the number of factors we consider, leading to excessive memory use in the case of many factors.

In [7], we consider yet another (the third) approach, which may be considered as an intermediate solution between the above two. We consider a single network with multiple input and output branches, as shown in Fig.5. Each output branch forms a pair with one of the input branches, which is used for removing a single degradation factor. Then, we recurrently use this network, that is, we feed the output from an output branch back to one of the input branches and repeat this procedure for the number of factors with a different input-output pair at each time. (This may be viewed as a cascading repetition of the same network with each cascading point connected with a different input-output pair.) To train the network, we consider a two-step method; in the first step,

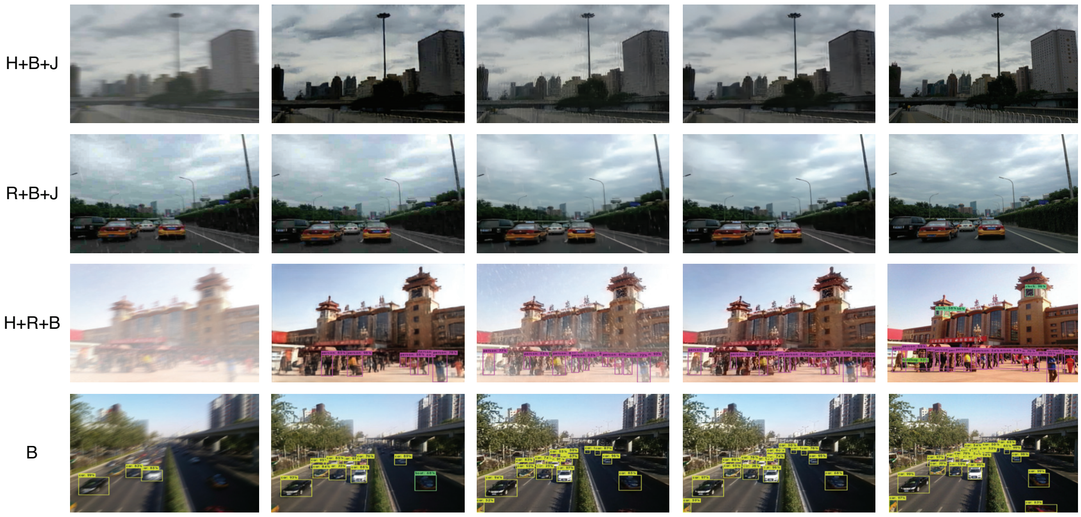

Figure 6. Qualitative evaluation of our method (R-MBN and R-MBN*) on images with mixed distortion factors. From left to right, an input image, Suganuma et al. [3], R-MBN, R-MBN*, and the ground truth. R, B, J and H means rain-streak, motion blur, JPEG artifacts, and haze. Object detection is performed by YOLO-v3.

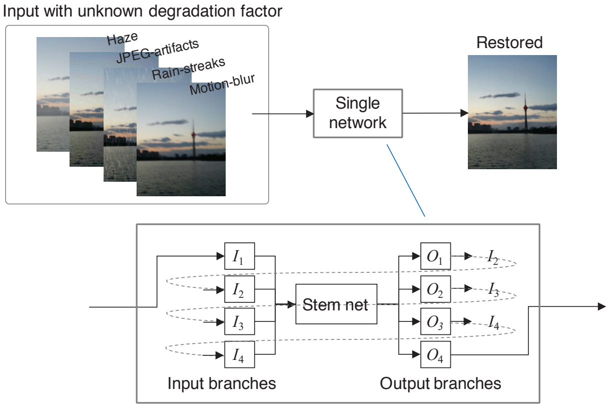

Figure 5. Illustration of our universal network. To restore a clean image from an input with unknown distortion factor, we use a network with multiple input and output branches. Receiving an input image from I_1, it yields an improved image in terms of a single degradation factor from O_1 It is then fed to I_2 and improved in terms of another factor, outputting the improved image from O_2 This is repeated until O_4, yielding the final output. Each box inside the bottom large box represents a sub-network.

we train it in a multi-task learning (MTL) framework, and in the second step, we fine-tune it so that it can be used in a recurrent manner.

This approach resolves the difficulties with the above two approaches. As the network has multiple input and output branches, we can employ the standard formulation of predicting a residual image for each factor. Our training scheme enables us to equip the network with the property that it does not affect the factors other than the target one including clean images. Figure 6 shows examples of image restoration in the presence of unknown mixed distortion factors and subsequent object detection from the restored images.

REFERENCES

[1] X. Liu, M. Suganuma, Z. Sun, and T. Okatani, “Feature quantization for defending against distortion of images,” Proc. Computer Vision and Pattern Recognition (2018).

[2] X. Liu, M. Suganuma, Z. Sun, and T. Okatani, “Dual Residual Networks Leveraging the Potential of Paired Operations for Image Restoration,” Proc. Computer Vision and Pattern Recognition (2019).

[3] M. Suganuma, X. Liu, and T. Okatani, “Attention-Based Adaptive Selection of Operations for Image Restoration in the Presence of Unknown Combined Distortions,” Proc. Computer Vision and Pattern Recognition (2019).

[4] M. Haris, G. Shakhnarovich, and N. Ukita, “Deep Back-Projection Networks for Super-Resolution,” Proc. Computer Vision and Pattern Recognition (2018).

[5] M. Suganuma, Masanori, M. Ozay, and T. Okatani, “Exploiting the Potential of Standard Convolutional Autoencoders for Image Restoration by Evolutionary Search,” Proc. International Conference on Machine Learning (2018).

[6] K. Yu and C. Dong and L. Lin and L. C., Change, “Crafting a Toolchain for Image Restoration by Deep Reinforcement Learning,” Proc. Computer Vision and Pattern Recognition (2018).

[7] A. Veit, M. Wilber, J. Michael J. and S. Belongie, Residual Networks Behave Like Ensembles of Relatively Shallow Networks, Proc. International Conference on Neural Information Processing Systems (2016).