Analyzing the Dependency of ConvNets on Spatial Information

Yue Fan(✉), Yongqin Xian(✉), Max Maria Losch(✉), and Bernt Schiele(✉)

Max Planck Institute for Informatics, Saarbrücken, Germany
{yfan,yxian,mlosch,schiele}@mpi-inf.mpg.de

Abstract. Intuitively, image classification should profit from using spatial information. Recent work, however, suggests that this might be overrated in standard CNNs. In this paper, we are pushing the envelope and aim to investigate the reliance on spatial information further. We propose to discard spatial information via shuffling locations or average pooling during both training and testing phases to investigate the impact on individual layers. Interestingly, we observe that spatial information can be deleted from later layers with small accuracy drops, which indicates spatial information at later layers is not necessary for good test accuracy. For example, the test accuracy of VGG-16 only drops by 0.03% and 2.66% with spatial information completely removed from the last 30% and 53% layers on CIFAR-100, respectively. Evaluation on several object recognition datasets with a wide range of CNN architectures shows an overall consistent pattern.

1 Introduction

Despite the impressive performances of convolutional neural networks (CNNs) on computer vision tasks [9,10,16,18,25], their inner workings remain mostly obfuscated to us, especially how the information is encoded throughout layers. Generally, the majority of modern CNNs for image classification utilize a collection of filters with local receptive fields to capture hierarchical patterns across all the convolutional layers [10,16,25]. Such design choices are based on the assumption that spatial information remains important at every convolutional layer, and better representations can be attained by gradually enlarging the receptive field to incorporate more contexts. This further leads to lots of approaches that help capture spatial correlations between features in order to improve model performance [1,13,26]. For example, a popular class of those methods is the visual attention mechanism [15,19] which enables more powerful representations by enhancing the most salient region of the image.

However, recent works on restricting the receptive field of CNN architectures for scrambled inputs [2] or using wavelet feature networks of shallow depth [20],

Electronic supplementary material The online version of this chapter (https://doi.org/10.1007/978-3-030-71278-5_8) contains supplementary material, which is available to authorized users.

Z. Akata et al. (Eds.): DAGM GCPR 2020, LNCS 12544, pp. 101–115, 2021.
https://doi.org/10.1007/978-3-030-71278-5_8

have all found it to be possible to acquire competitive performances on the respective tasks. This raises doubts on the necessity of spatial information for classification and whether the network can still maintain the performance when the spatial information is completely removed from the training process.

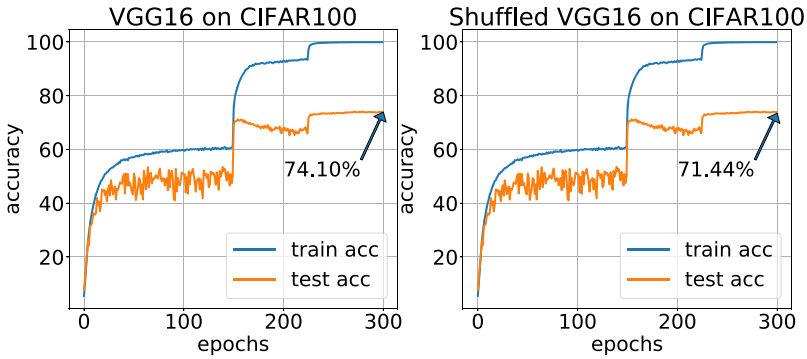

Fig. 1. Shuffling the feature maps from the last 54% layers in VGG-16 randomly and spatially only reduces the final test accuracy by 2.66% (from 74.10% to 71.44%) on CIFAR-100, and the training processes look surprisingly similar, which implies that spatial information may not be necessary for good classification accuracy.

In this work, we re-design the structure of the network to separate the spatial information and channel-wise information independently, with the goal of analyzing the dependency of the network on them. Spatial information refers to the spatial ordering on the feature map. To this end, we propose *channel-wise shuffle* to eliminate channel information, and *spatial shuffle*, *patch-wise spatial shuffle* and *GAP+FC* to eliminate spatial information. Surprisingly, we find that the spatial information is not necessary at later layers, and the modified CNNs, i.e. without accessing any spatial information at later layers, can still achieve competitive results on several object recognition datasets. As an example, Fig. 1 shows the training processes of a standard VGG-16 and a modified VGG-16 with spatial shuffle on CIFAR-100. In the shuffled VGG-16, feature maps must first go through a random spatial shuffle operation before convolved with the filters from the last 54% layers. Interestingly, the test accuracy only drops 2.66%, and the training process is nearly identical to the standard VGG-16. This observation generalizes to various CNN architectures: removing spatial information from the last 30% layers gives a surprisingly little test accuracy decrease within 1% across architectures and datasets, and the accuracy decrease is still within 7% even if the last 50% layers are manipulated. This indicates that spatial information is overrated for standard CNNs and not necessary to reach competitive performances. Finally, our investigation on the detection task shows that although the unavailability of spatial information at later layers does hinder the CNN to localize objects, the impact is not as fatal as expected; at the same time, the classification ability of the model is not affected.

The main contributions of our work are as follows: we find that spatial information at later layers is not really necessary for good classification test accuracy and that even though the depth of the network plays an important role, later layers do not require spatial integration. As a side effect, GAP+FC leads to a smaller model with fewer parameters with small test accuracy drops.

2 Related Work

Intuitively, object recognition benefits from gradually enlarged receptive field and spatial integration. For that reason extensive efforts have been made to enhance the aggregation of spatial information in the decision-making progress of CNNs. [5,32] have made attempts to generalize the strict spatial sampling of convolutional kernels to allow for globally spread out sampling, and [31] have spurred a range of follow-up work on embedding global context layers with the help of spatial down-sampling. Another emerging interest of augmenting CNNs with self-attention has also made progress in several vision tasks. [27] presents a non-local operation that computes the response at a position as a weighted sum of the features at all positions to capture long-range dependencies and shows that self-attention is an instantiation of their non-local operations. [3] show improvements on image classification and achieve state-of-the-art results on video action recognition tasks with a variant of non-local operations. Even a fully attentional model is verified to be effective for various visual tasks [21].

While all of these works have improved on a related classification metric in some way, it is not entirely evident whether the architectural changes alone can be credited, as there is an increasing number of work on questioning the importance of the extent of spatial information for common CNNs. One of the most recent observations by [2] indicates that the VGG-16 architecture trained on ImageNet is invariant to scrambled images to a large extent. Furthermore, they construct a modified ResNet architecture with a limited receptive field as small as 33×33, similar to the style of the traditional Bag-of-Visual-Words and reach competitive results on ImageNet. In contrast to their work, we make a clear distinction between first and last layers, and we show empirically spatial information at last layers are not necessary for good test accuracy.

[23] assumes that current CNNs do not respect the spatial information due to the pooling operation; CNNs look for features in the image without paying attention to their pose during prediction. This limitation motivates the work of [23] where they make use of dynamic routing among capsules to encode the spatial information. Moreover, the widely used global average pooling in most recently proposed architectures [10,17] implies that collapsing spatial information at the very end does not affect the test accuracy. On a related note, [8] indicates that models trained solely on ImageNet do not learn shape sensitive representations with constructing object-texture mismatched images, which would be expected to require global spatial information. Instead, the models are mostly sensitive to local texture features.

Our work aims to push the envelope further to investigate the necessity of spatial information in the processing pipeline of CNNs. While related work has

put the attention mainly on altering the input and does not differentiate between last and first layers, we are interested in taking measures that remove the spatial information at different intermediate layers to shed light on how CNNs process spatial information, evaluating its importance and providing insights for architectural design choices.

3 Methods and Experimental Setup

In this section, we design methods to systematically study the phenomenon found in Fig. 1 that spatial information appears to be neglectable to some extent. We test how information is represented throughout the network's layers by discarding spatial or channel information in different ways in intermediate layers and applying them to well-established architectures. Experiments are conducted on object recognition and detection tasks. Section 3.1 elaborates details on our approaches, and the experimental setup is discussed in Sect. 3.2.

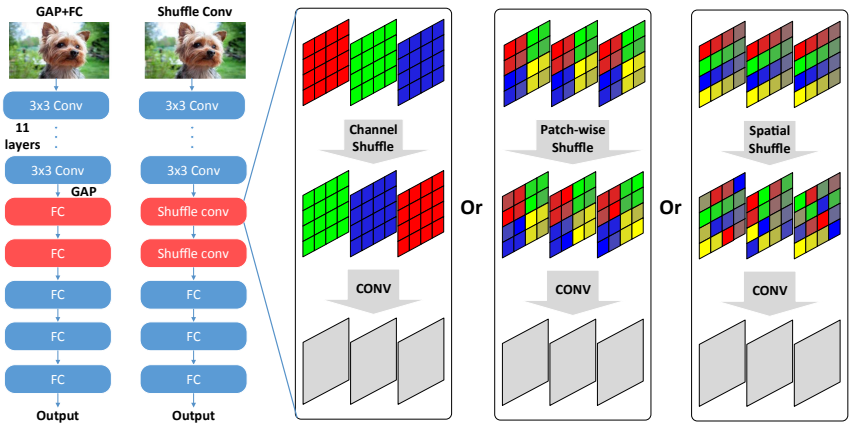

Fig. 2. An example of VGG-16 modified by our methods. The leftmost architecture shows the modification (in red) from *GAP+FC*, where the last two convolutional layers are replaced by fully-connected layers after a GAP layer. The middle architecture shows the modification (in red) from shuffle conv, where the last two convolutional layers are replaced by one of the shuffling methods and an ordinary convolution. *Spatial shuffle* randomly and independently permutes pixels on each feature map at a global scale in the sense that a pixel can end up anywhere on the feature map. *Patch-wise shuffle* first divides the feature map into grids; then it randomly permutes the pixel locations within each grid independently. *Channel shuffle* randomly permutes the order of feature maps, leaving the spatial ordering unchanged. (Color figure online)

3.1 Approaches to Constrain Information

We propose four different methods, namely *channel-wise shuffle*, *spatial shuffle*, *patch-wise spatial shuffle*, and *GAP+FC*, to remove either spatial or channel information from the training. Spatial information here refers to the awareness of the relative spatial position between activations on the same feature map, and channel information stands for the dependency across feature maps. The left part of Fig. 2 illustrates an example of VGG-16 with its last two layers modified by GAP+FC or any of the three shuffle methods.

Spatial Shuffle extends the ordinary convolution operation by prepending a random spatial shuffle operation to permute the input to the convolution. As illustrated in Fig. 2: Given an input tensor of size $c \times h \times w$ with c being the number of feature maps for a convolutional layer, we first take one feature map from the input tensor and flatten it into a 1-d vector with $h \times w$ elements, whose ordering is then permuted randomly. The resulting vector is finally reshaped back into $h \times w$ and substitute the original feature map. This procedure is independently repeated c times for each feature map so that activations from the same location in the previous layer are misaligned, thereby preventing the information from being encoded by the spatial arrangement of the activations. The shuffled output becomes the input of an ordinary convolutional layer in the end. Even though shuffling itself is not differentiable, gradients can still be propagated through in the same way as pooling operations. Therefore it can be embedded into the model directly for end-to-end training. As the indices are recomputed within each forward pass, the shuffled output is also independent across training and testing steps.

Images in the same batch are shuffled in the same way for the sake of simplicity since we find empirically that it does not make a difference whether the images in the same batch are shuffled in different ways.

Patch-Wise Spatial Shuffle is a variant of *spatial shuffle*. In contrast, patch-wise spatial shuffle does not perform on a global scale but a local scale by dividing the feature map into grids. Each patch in the grid is subsequently shuffled independently. Afterwards, an ordinary convolution is performed as usual. Note that the two operations are equivalent when the patch size is the same as the feature map size. Figure 2 demonstrates an example of patch-wise spatial shuffle with a 2×2 patch size, where the random permutation of pixel locations is restricted within each patch.

Channel-Wise Shuffle is used to investigate the importance of channel information which is normally deemed as essential [28–30]. It keeps the spatial ordering of activations and randomly permutes the ordering of feature maps to prevent the model from utilizing channel information. An illustration can be seen in Fig. 2, channel-wise shuffle is also performed independently across training and testing steps.

GAP+FC denotes Global Average Pooling and Fully Connected Layers. *Spatial Shuffle* is an intuitive way of destroying spatial information. However, shuffling

introduces undesirable randomness into the model; non-deterministic feature maps from an image lead to fluctuations in the model prediction, so an evaluation needs multiple forward passes to acquire an estimate of the mean of the output. A simple deterministic alternative achieving a similar goal is to deploy Global Average Pooling (GAP) after an intermediate layer, and all the subsequent ones are substituted by fully connected layers. Compared to *Spatial Shuffle* that introduces an extra computational burden at each forward pass, it is a much more efficient way to avoid learning spatial information at intermediate layers because it shrinks the spatial size of all subsequent feature maps to one; therefore, the number of FLOPs and parameters are also reduced.

3.2 Experimental Setup

This section details the experimental setup for the classification and object detection tasks. We test different architectures on three datasets: CIFAR-100, Small-ImageNet-32x32 [4], and Pascal VOC 2007 + 2012. Small-ImageNet-32x32 is a down-sampled version of the original ImageNet (from 256×256 to 32×32). We report top-1 accuracy and mAP [6,7] in classification and detection experiments respectively. We will take an existing architecture and apply the modification to different layers. The rest of the setup and hyper-parameters for modified architectures remain the same as the original architectures.

Classification: For the VGG architecture, the modification is only performed on the convolutional layers, as illustrated in Fig. 2. For the ResNet architecture, one bottleneck sub-module is considered as a single piece, and the modification is applied onto the 3×3 convolutions within the sub-module since they are the only operations with spatial extent. Features that go through the skip connection branch are also shuffled in the shuffle experiments to prevent the model from learning to ignore the information from the residual branch. The rest of the configuration remains the same (see supplemental material for an example of modified ResNet-50 architecture).

For CIFAR-100 and Small-ImageNet-32x32 experiments, the original ResNet architecture down-samples the input image by a factor of 32 and gives 1×1 feature maps at last layers, therefore shuffling is noneffective. To make shuffling non-trivial, we set the first convolution in ResNet to 3×3 with stride 1 and the first max-pooling layer is removed so that the final feature map size is 4×4.

To alleviate the effect of mismatched training details, we first reproduce the reported results for all experiments and then train our modified architectures under the same training setting. All models in the same set of experiments (e.g. VGG-16 on CIFAR-100) use the same set of hyper-parameters, and they share the same initialization from the same random seed. During testing, we make sure to use a different random seed than during training.

Detection: We use the training set and validation set of VOC 2012+2007 as the training data and report mAP on VOC 2007 test set. We shuffle the last layer in the backbone model to test the robustness of localization against the absence of spatial information.

4 Results

We first compare the test accuracy of VGG-16 on CIFAR-100 with spatial or channel information missing from a different number of last layers in Sect. 4.1. An in-depth study of our main observations on CIFAR-100 and Small-ImageNet-32x32 for VGG-16 and ResNet-50 is conducted in Sect. 4.2. In Sect. 4.3, we investigate the model robustness against the loss of spatial information in various degree by controlling the amount of spatial information that passes through the network. Finally, we present the detection results on VOC datasets in Sect. 4.4.

4.1 Spatial and Channel-Wise Shuffle on VGG-16

In this section, we first investigate the invariance of pre-trained models to the absence of the spatial or channel information at test time, then we impose this invariance at training time with methods in Sect. 3.1.

Shuffle the Last 30% Layers Channel-Wise: Our baseline is a VGG-16 trained on CIFAR-100 that achieves 74.10% test accuracy. We first test its robustness against the absence of the channel information at test time by substituting the last 30% convolutional layers with the channel-wise shuffle convolution. As is expected, the test accuracy drops to 1.04% (Table 1), which is the same as the random guessing on CIFAR-100. Following the same training scheme of the baseline, we then train another VGG-16 with channel-wise shuffle added to its last 30% convolutional layers. This model can reach around 67% test accuracy no matter whether channel-wise shuffle is applied at test time. However, it still performs significantly worse than the baseline, which indicates that the expressiveness of the model is much limited without utilizing the ordering of feature maps even though the spatial information is preserved.

Table 1. Top-1 accuracy of VGG-16 on CIFAR-100 with spatial/channel-wise shuffle enabled at either training or test time for the last 30% layers. A model from standard training does not possess robustness against spatial shuffle (23.49%) and channel-wise shuffle (1.04%). However, when imposed in training, the model achieves 74.07% test accuracy for spatial shuffle and 67.56% for channel-wise shuffle, showing impressive robustness to the loss of spatial information.

Train scheme	No shuffle	Channel shuffle	Channel shuffle	No shuffle	Spatial shuffle	Spatial shuffle	No shuffle
Test scheme	No shuffle	Channel shuffle	No shuffle	Channel shuffle	Spatial shuffle	No shuffle	Spatial shuffle
Top-1(%)	74.10	67.56	67.80	1.04	74.07	73.74	23.49

Shuffle the Last 30% Layers Spatially: As a comparison to channel shuffle, we repeat the same experiment on spatial shuffle, and the result is presented in the second half of Table 1. No shuffle → spatial shuffle of the pre-trained VGG-16 gives 23.49% test accuracy, which is similar to the test accuracy of a one-hidden-layer perceptron (with 512 hidden units and ReLU activation) on CIFAR-100 (25.61%) when evaluated with the random spatial shuffle. However, if the spatial shuffle is infused into the model at training time, then the baseline test accuracy can be retained no matter whether random spatial shuffle appears at test time (74.07% for spatial shuffle → spatial shuffle and 73.74% for spatial shuffle → no shuffle).

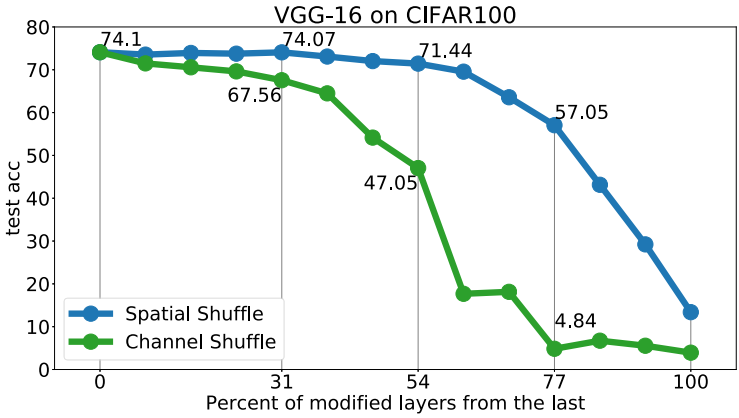

Fig. 3. Classification accuracy of VGG-16 on CIFAR-100 with different shuffle schemes. The very slow decrease of the test accuracy of spatial shuffle implies a far less important role of spatial information for classification. The test accuracy is not much affected, given that the spatial shuffle modifies 31% of its layers. Even with 54% later layers shuffled spatially, the test accuracy only decreases by 2.66%, and the same number of the test accuracy decrease in channel-wise shuffle happens when the last layer is modified.

Shuffle Other Layers: To systematically study the impact of spatial and channel information, we gradually increase the number of modified layers from the last in VGG-16 and report the corresponding test accuracy in Fig. 3. All models are trained with the same setup, and shuffling is performed both at training and test time; the x-axis is the percentage of modified layers counting from the last layer on with 0 referring the baseline.

Besides an overall decreasing trend for both shuffling with the increase of the percent of modified layers, the test accuracy of spatial shuffle drops unexpectedly slowly, e.g. merely 2.66% test accuracy drop when up to 54% of layers from the last are shuffled spatially. Likewise, when spatial information is removed from the last 77% layers, it still has a reasonable test accuracy (57.05%), whereas the test accuracy of channel-wise shuffle is only 4.84%.

Discussion: This indicates that although a standard model makes use of both spatial dimension and channel dimension to encode information, the spatial information plays a surprisingly less pivotal role than the channel information. The model is even able to adapt to the complete absence of spatial information at later layers if spatial information is removed explicitly at training time, which strengthens the claims from [2,23] that CNNs intrinsically possess invariance to the spatial relationship among features to some extent. Moreover, the unsuccessful adaptation to channel-wise shuffle implies that the large model capacity may mainly come from the channel order and shuffling the channel order causes unrecoverable damage to the model.

4.2 Spatial Information at Later Layers is Not Necessary

In this section, we design more experiments to study the reliance of different layers on spatial information: we modify the last convolutional or bottleneck layers of VGG-16 or ResNet-50 by *Spatial Shuffle* (both at training and test time) and *GAP+FC* such that the spatial information is removed in different ways. Our modification on the baseline model always starts from the last layer and is consecutively extended to the first layer. The modified networks are then trained on the training set with the same setup and evaluated on the hold-out validation set.

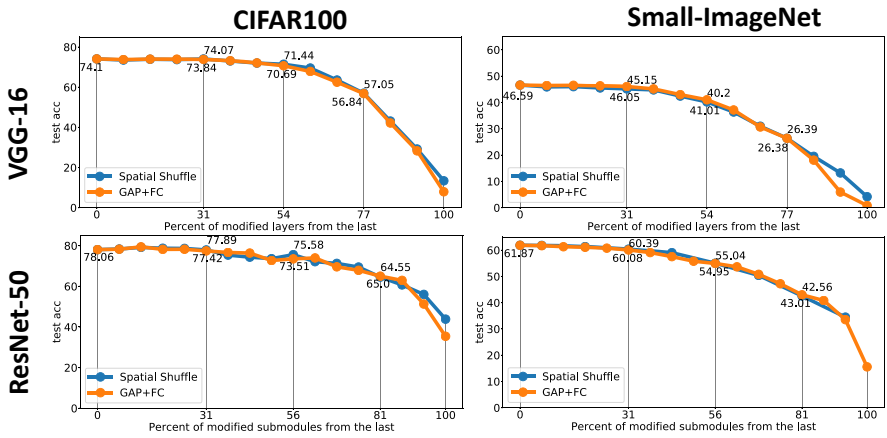

Fig. 4. Classification results of GAP+FC and spatial shuffle for VGG-16 and ResNet-50 on CIFAR-100 and Small-ImageNet-32x32. The x-axis is the percent of modified layers/sub-modules counting from the last one. Models on the same dataset are trained with the same setup. It can be observed consistently across experiments that the baseline test accuracy is preserved for a long time even though spatial information is eliminated from the last several layers by spatial shuffle or GAP+FC, suggesting that spatial information at later layers is not necessary for good test accuracy. The difference between the baseline models and the models whose latter half of the layers are modified by GAP+FC or spatial shuffle is, however, still in a reasonable range between 2.48% (ResNet-50 with spatial shuffle on CIFAR-100) to 6.92% (ResNet-50 with GAP+FC on Small-ImageNet-32x32).

Results on CIFAR-100 and Small-ImageNet-32x32: Results of VGG-16 and ResNet-50 on CIFAR-100 and Small-ImageNet-32x32 are shown in Fig. 4. The x-axis is the percent of modified later layers, and 0 is the baseline model test accuracy without modifying any layer.

As we can see, *Spatial Shuffle* and *GAP+FC* have a similar overall behaviour consistently across architectures and datasets: the baseline test accuracy is retained for a long time before it starts to decrease with the increase of the percent of modified layers. When the last 30% layers are modified by GAP+FC or spatial shuffle, there is no or little test accuracy decrease across experiments (0.17% for ResNet-50 on CIFAR-100 and 1.44% for VGG-16 on Small-ImageNet with spatial shuffle). And the test accuracy decrease is still in a reasonable range (2.48% with spatial shuffle on CIFAR-100 and 6.92% for GAP+FC on Small-ImageNet-32x32 for ResNet-50), even with around half of the last layers modified. At 77% to 81% of the modified later layers, the test accuracy just starts to show a significant difference to the baseline in the range of 8.58% (ResNet-50 with spatial shuffle on CIFAR-100) to 20.21% (VGG-16 with GAP+FC on Smalll-ImageNet-32x32).

Our experiments here clearly show that spatial information can be neglected from a significant number of later layers with no or small test accuracy drop if the invariance is imposed at training, which suggests that *spatial information at last layers is not necessary for good test accuracy*. We should, however, notice that it does not indicate that models whose prediction is based on spatial information can not generalize well. Besides, unlike the common design manner that layers at different depth inside the network are normally treated equally, e.g. the same module is always used throughout the architecture [12,14,24], our observation implies it is beneficial to have different designs for different layers since there is no necessity to encode spatial information in the later layers. As a side effect, GAP+FC can reduce the number of model parameters with little test accuracy drop. For example, GAP+FC achieves nearly identical results (46.05%) to the VGG-16 baseline (46.59%), while reducing the number of parameters from 37.70M to 29.31M on Small-ImageNet-32x32.

4.3 Patch-Wise Spatial Shuffle

In this section, we study the relation between the model test accuracy and the amount of spatial information that propagates throughout a network. The latter is controlled by patch-wise spatial shuffle with different patch sizes. The larger the patch size is, the less the preserved spatial information. Patch-wise spatial shuffle reduces to spatial shuffle when the patch size is the same as the feature map size, in which case no spatial information remains. Our experiments are conducted on CIFAR-100 for VGG-16 and ResNet-50, and we only shuffle a single layer at a time since the model is not able to recover the "damage" caused by shuffling an early layer (see more in the supplemental material).

The result of patch-wise spatial shuffling of different patch sizes is shown in Fig. 5. We can see that the patch size does not make much difference in terms of the test accuracy at later layers, e.g. results of patch size 2, 4 and 8 for ResNet-50

at 8–14 layers are similar. However, the test accuracy has a rapid decrease with the increase of the patch size at first layers, indicating a relatively important role of spatial information at first layers. Nevertheless, this role might not be as much important as what is commonly believed, as the ResNet-50 still has 40.76% test accuracy when the input image is completely shuffled.

4.4 Detection Results on VOC Datasets

Object detection should intuitively suffer more from spatial shuffling than classification since the spatial information should help to localize objects. In this section, we show some initial results on Pascal VOC [6,7].

We design an analogue to YOLO [22] as our detection model. The architecture consists of a backbone and a detection head; the backbone is a ResNet-50 without the classifier, and the detection head has three bottlenecks and a 3×3 convolutional layer whose outputs is in the same format as [22]. Different to [22], we deploy a 3×3 convolution instead of a fully connected layer in the end to output the final detection results. The latter gives the model potential access to the object feature, which may be exploited by the model to predict its location. In order to prevent the undesirable shortcut, we use a 3×3 convolution so that the prediction of a bounding box at a certain location does not depend on all activation on the feature map.

By using a pre-trained ResNet-50 on ImageNet, we can reach 66% mAP on VOC2007 test set after fine-tuning, which is the same as the number in [22]. To avoid pretraining a spatially shuffled model on ImageNet, we compare a spatially shuffled model and a non spatially shuffled model, both trained from

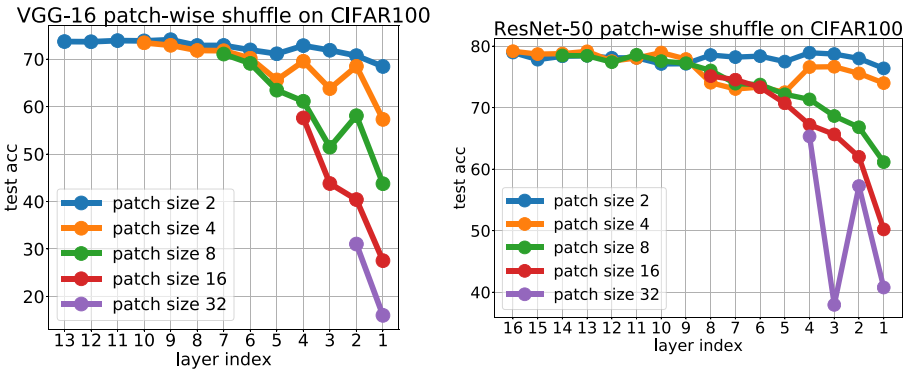

Fig. 5. The result of patch-wise spatial shuffling of VGG-16 and ResNet-50 on CIFAR-100. Only a single layer is shuffled at a time. Layer index 13 and 16 stand for the last layer of VGG-16 and ResNet-50, respectively. With the increase of the patch size, the test accuracy decreases faster at first layers than that at last layers. It is interesting to see that both models' test accuracy do not fall into the random guess (16.02% for VGG-16 and 40.76% for ResNet-50) at layer index one and patch size 32, where the input image is completely shuffled.

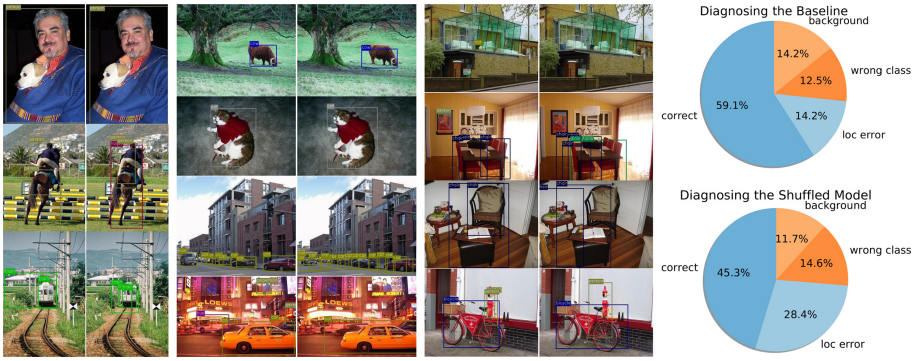

Fig. 6. Left: Qualitative detection results on the VOC 2007 test set. Examples are the first 11 images in the test set. The left result is from the baseline, and the right result is from the shuffled model. Right: Detection error analysis of our baseline and the shuffled model shows a doubled localization error in the shuffled model and the rest types of error are in the same level as the baseline.

scratch on VOC. Our models are trained for 500 epochs with exponentially decaying learning rate starting from 0.001. Our baseline model achieves 50% mAP on VOC2007 test set without using an ImageNet pre-trained backbone. The result of the shuffled model, where we apply random shuffle to the last layer of the backbone, is 34%. While this sounds like a large drop, it turns out that the classification performance is essentially preserved and only the localization performance is suffering. To analyze this effect in detail, we use the method and tools proposed in [11]. The diagnosis tool classifies each prediction from the model as either correct prediction or a type of error based on its class label and IoU with the ground truth. More details can be found in [11].

The results in Fig. 6 right show that the misclassification to the wrong class and background are of similar percents for both models, and the localization error doubles for the shuffled model (an increase from 14.2% to 28.4%). Though random shuffling indeed affects the model's localization ability, it is unexpected that the effect is not fatal. Because random shuffling switches features, it is highly likely the model trained with spatial shuffle has to predict the correct bounding box for one object based on some other features. We should also notice that a prediction is counted as a localization error if it has the correct class label and the IoU to the ground truth is less than 0.5. Therefore, classification-wise speaking, the shuffled model got 73.7% (45.3% + 28.4%) of its predictions correct, which is at the same level as the baseline (73.3% = 59.1% + 14.2%).

Qualitative Results: Figure 6 left shows some qualitative results from both models. Those examples are the first 11 images in the VOC2007 test set. We can see that the localization error actually mainly comes from small objects for which the shuffled model tends to predict several bounding boxes on one object, and the bounding box of the relatively big object is not really off, e.g. the shuffled model managed to localize the dining table in the middle right image and the horse in the middle left image while the baseline can not.

5 Conclusion

To conclude, we empirically show that a significant number of later layers of CNNs are robust to the absence of spatial information, which is commonly assumed to be important for object recognition tasks. Modern CNNs can tolerate the loss of spatial information from the last 30% of layers at around 1% accuracy drop; and the test accuracy only decreases by less than 7%, when spatial information is removed from the last half of layers on CIFAR-100 and Small-ImageNet-32x32. Though the depth of the network is essential for good test accuracy, later layers do not require spatial integration.

References

1. Bell, S., Lawrence Zitnick, C., Bala, K., Girshick, R.: Inside-outside net: detecting objects in context with skip pooling and recurrent neural networks. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 2874–2883 (2016)
2. Brendel, W., Bethge, M.: Approximating CNNs with bag-of-local-features models works surprisingly well on imageNet. In: International Conference on Learning Representations (2019). https://openreview.net/forum?id=SkfMWhAqYQ
3. Chen, Y., Kalantidis, Y., Li, J., Yan, S., Feng, J.: Aˆ 2-nets: double attention networks. In: Advances in Neural Information Processing Systems, pp. 352–361 (2018)
4. Chrabaszcz, P., Loshchilov, I., Hutter, F.: A downsampled variant of imagenet as an alternative to the cifar datasets. arXiv preprint arXiv:1707.08819 (2017)
5. Dai, J., Qi, H., Xiong, Y., Li, Y., Zhang, G., Hu, H., Wei, Y.: Deformable convolutional networks. In: Proceedings of the IEEE International Conference on Computer Vision, pp. 764–773 (2017)
6. Everingham, M., Van Gool, L., Williams, C.K.I., Winn, J., Zisserman, A.: The PASCAL Visual Object Classes Challenge 2007 (VOC 2007) Results. http://www.pascal-network.org/challenges/VOC/voc2007/workshop/index.html
7. Everingham, M., Van Gool, L., Williams, C.K.I., Winn, J., Zisserman, A.: The PASCAL Visual Object Classes Challenge 2012 (VOC 2012) Results. http://www.pascal-network.org/challenges/VOC/voc2012/workshop/index.html
8. Geirhos, R., Rubisch, P., Michaelis, C., Bethge, M., Wichmann, F.A., Brendel, W.: Imagenet-trained CNNs are biased towards texture; increasing shape bias improves accuracy and robustness. In: International Conference on Learning Representations (2019). https://openreview.net/forum?id=Bygh9j09KX
9. Girshick, R., Donahue, J., Darrell, T., Malik, J.: Rich feature hierarchies for accurate object detection and semantic segmentation. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 580–587 (2014)
10. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: CVPR (2016)
11. Hoiem, D., Chodpathumwan, Y., Dai, Q.: Diagnosing error in object detectors. In: Fitzgibbon, A., Lazebnik, S., Perona, P., Sato, Y., Schmid, C. (eds.) ECCV 2012. LNCS, vol. 7574, pp. 340–353. Springer, Heidelberg (2012). https://doi.org/10.1007/978-3-642-33712-3_25
12. Howard, A.G., et al.: MobileNets: efficient convolutional neural networks for mobile vision applications. arXiv e-prints arXiv:1704.04861 (Apr 2017)

13. Hu, J., Shen, L., Sun, G.: Squeeze-and-excitation networks. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 7132–7141 (2018)
14. Iandola, F.N., Moskewicz, M.W., Ashraf, K., Han, S., Dally, W.J., Keutzer, K.: SqueezeNet: AlexNet-level accuracy with 50x fewer parameters and <1mb model size. ArXiv abs/1602.07360 (2017)
15. Jaderberg, M., Simonyan, K., Zisserman, A., et al.: Spatial transformer networks. In: Advances in Neural Information Processing Systems, pp. 2017–2025 (2015)
16. Krizhevsky, A., Sutskever, I., Hinton, G.E.: Imagenet classification with deep convolutional neural networks. In: Proceedings of the 25th International Conference on Neural Information Processing Systems - Volume 1, pp. 1097–1105. NIPS 2012, Curran Associates Inc., USA (2012). http://dl.acm.org/citation.cfm?id=2999134.2999257
17. Lin, M., Chen, Q., Yan, S.: Network in network. CoRR abs/1312.4400 (2013)
18. Long, J., Shelhamer, E., Darrell, T.: Fully convolutional networks for semantic segmentation. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 3431–3440 (2015)
19. Mnih, V., Heess, N., Graves, A., et al.: Recurrent models of visual attention. In: Advances in Neural Information Processing Systems, pp. 2204–2212 (2014)
20. Oyallon, E., Belilovsky, E., Zagoruyko, S.: Scaling the scattering transform: deep hybrid networks. In: Proceedings of the IEEE International Conference on Computer Vision, pp. 5618–5627 (2017)
21. Ramachandran, P., Parmar, N., Vaswani, A., Bello, I., Levskaya, A., Shlens, J.: Stand-alone self-attention in vision models. arXiv preprint arXiv:1906.05909 (2019)
22. Redmon, J., Divvala, S., Girshick, R., Farhadi, A.: You only look once: unified, real-time object detection. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 779–788 (2016)
23. Sabour, S., Frosst, N., Hinton, G.E.: Dynamic routing between capsules. In: Guyon, I., Luxburg, U.V., Bengio, S., Wallach, H., Fergus, R., Vishwanathan, S., Garnett, R. (eds.) Advances in Neural Information Processing Systems, vol. 30, pp. 3856–3866. Curran Associates, Inc. (2017)
24. Sandler, M., Howard, A., Zhu, M., Zhmoginov, A., Chen, L.C.: Mobilenetv2: inverted residuals and linear bottlenecks. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 4510–4520 (2018)
25. Simonyan, K., Zisserman, A.: Very deep convolutional networks for large-scale image recognition. arXiv preprint arXiv:1409.1556 (2014)
26. Szegedy, C., et al.: Going deeper with convolutions. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 1–9 (2015)
27. Wang, X., Girshick, R., Gupta, A., He, K.: Non-local neural networks. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 7794–7803 (2018)
28. Xie, S., Girshick, R., Dollár, P., Tu, Z., He, K.: Aggregated residual transformations for deep neural networks. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 1492–1500 (2017)
29. Zhang, T., Qi, G.J., Xiao, B., Wang, J.: Interleaved group convolutions. In: Proceedings of the IEEE International Conference on Computer Vision, pp. 4373–4382 (2017)
30. Zhang, X., Zhou, X., Lin, M., Sun, J.: ShuffleNet: an extremely efficient convolutional neural network for mobile devices. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 6848–6856 (2018)

31. Zhao, H., Shi, J., Qi, X., Wang, X., Jia, J.: Pyramid scene parsing network. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 2881–2890 (2017)
32. Zhu, X., Hu, H., Lin, S., Dai, J.: Deformable convnets v2: more deformable, better results. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 9308–9316 (2019)