SPATIAL INFORMATION IS OVERRATED FOR IMAGE CLASSIFICATION

Anonymous authors
Paper under double-blind review

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 further investigate the reliance on and necessity of spatial information. We propose and analyze three methods, namely Shuffle Conv, GAP+FC and 1x1 Conv, that destroy spatial information during both training and testing phases. We extensively evaluate these methods on several object recognition datasets (CIFAR100, Small-ImageNet, ImageNet) with a wide range of CNN architectures (VGG16, ResNet50, ResNet152, MobileNet, SqueezeNet). Interestingly, we consistently observe that spatial information can be completely deleted from a significant number of layers with no or only small performance drops.

1 INTRODUCTION

Despite the fantastic performances of convolutional neural networks (CNNs) on computer vision tasks, their inner workings remain mostly obfuscated to us and analyzing them results often in surprising results.

Generally, the majority of modern CNNs for image classification learn spatial information across all the convolutional layers: every layer in AlexNet, VGG, Inception, and ResNet applies 3×3 or larger filters. Such design choice is based on the assumption that spatial information remains important at every convolutional layer to consecutively increase the access to a larger spatial context. This is based on the observations that single local features can be ambiguous and should be related to other features in the same scene to make accurate predictions Torralba et al. (2003); Hoiem et al. (2008).

Recent work on restricting the receptive field of CNN architectures, scrambling the inputs (Brendel & Bethge, 2019) or using wavelet feature networks resulting in networks with shallow depth (Oyallon et al., 2017) have all found it to be possible to acquire competitive performances on the respective classification tasks. This raises doubts on whether common CNNs learn representations of global context as small local features appear to be sufficient for classification.

We add to the list of surprising findings surrounding the inner workings of CNNs and present a rigorous investigation on the necessity of spatial information in standard CNNs by avoiding learning spatial information at multiple layers. To this end, we propose three methods i.e., *shuffle conv*, *GAP+FC* and *1x1Conv*, to eliminate the spatial information. Surprisingly, we find that the modified CNNs i.e., without the ability to access any spatial information at last layers, can still achieve competitive results on several object recognition datasets. This indicates that the spatial information is overrated for standard CNNs and not necessary to reach competitive performances.

In our experiments, the last layers of standard CNNs can be simplified by substituting them with our proposed GAP+FC or 1x1Conv layers which ignore spatial information, leading to a smaller model with less parameters. Moreover, our novel simplifications can be adapted to a wide range of CNN architectures and maintain state-of-the-art performance on various image classification datasets.

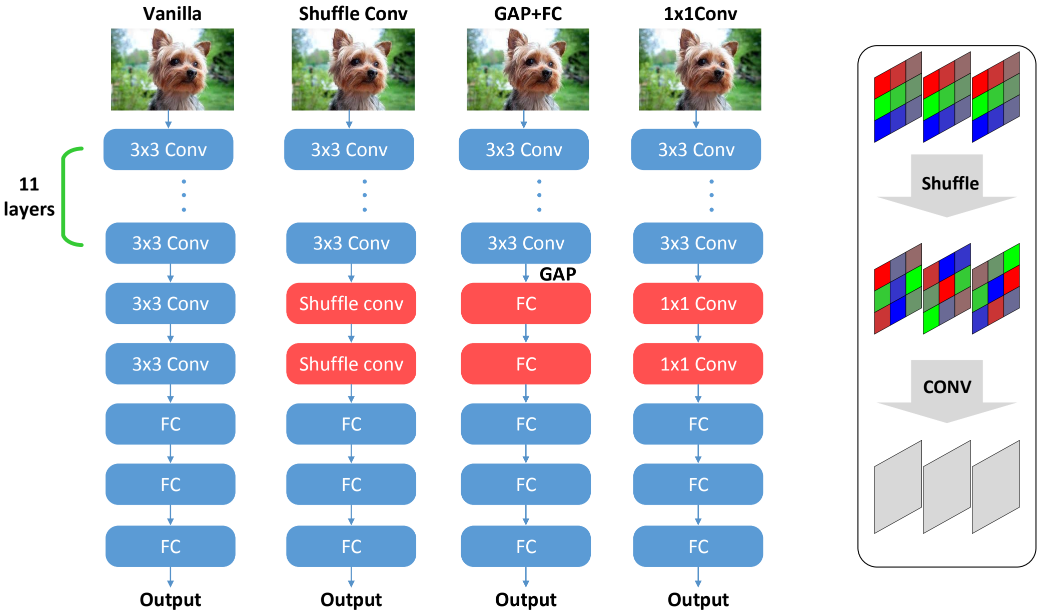

Figure 1: Left: A demonstration of *Shuffle Conv*, *GAP+FC* and *1x1Conv* on a VGG-16 architecture, where last 2 conv layers are modified accordingly. Right: The detail of the *shuffle conv*. Each feature map from the input tensor will be randomly and independently shuffled before being fed into an ordinary convolution.

2 Related Work

Training models for the task of object recognition, our intuitive understanding would be that global image context is beneficial for making accurate predictions. For that reason extensive efforts have been made to enhance the aggregation of spatial information in the decision-making progress of CNNs. Dai et al. (2017); Zhu et al. (2019) have made attempts to generalize the strict spatial sampling of convolutional kernels to allow for globally spread out sampling and Zhao et al. (2017) have spurred a range of follow-up work on embedding global context layers with the help of spatial down-sampling.

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 Brendel & Bethge (2019) for example indicate that the VGG-16 architecture trained on ImageNet is invariant to scrambled images to a large extent, e.g. they reported only a drop of slightly over 10% points top-5 accuracy for a pre-trained VGG-16. Furthermore, they were also able to construct a modified ResNet architecture with a limited receptive field as small as 33×33 and were able to reach competitive results on ImageNet, similar to the style of the traditional Bag-of-Visual-Words. The latter was also explicitly incorporated into the training of CNNs in the works by Mohedano et al. (2016); Feng et al. (2017); Cao et al. (2017), the effect of neglecting global spatial information by design had surprisingly little effect on performance values.

On a related note, Geirhos et al. (2019) has indicated with constructing object-texture mismatched images that models trained solely on ImageNet do not learn shape sensitive representations, which would be expected to require global spatial information, but instead are mostly sensitive to local texture features.

Our work is motivated to push the envelope further in order to investigate the necessity of spatial information in the process pipeline of CNNs. While the related work has put the attention mainly on altering the input, we are interested in taking measures that remove the spatial information in intermediate layers to shed light on how CNNs process spatial information, thus evaluating its importance and make suggestions for architectural design choices.

3 METHODS

In order to test how spatial information is processed in the CNN processing pipeline, we propose three approaches: *shuffle convolution*, *GAP+FC* and *1x1Conv* that neglect spatial information in different ways in intermediate layers and apply these to well established architectures. The evaluation is primarily done with comparing the classification accuracy for models that have been increasingly constrained with respect to how much spatial information can be propagated throughout the network. Section 3.1 elaborates details on our approaches and the experimental setup is discussed in section 3.2.

3.1 APPROACHES TO NEGLECT SPATIAL INFORMATION

Shuffle Convolution extends the ordinary convolution operation by prepending a random spatial shuffle operation, so that the input to the convolution is permuted. As illustrated in Fig. 1 right: Assume 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 Max Pooling. 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 within the same batch are shuffled in the same way for the sake of simplicity since we find empirically that it doesn't make a difference whether the images inside the same batch are shuffled in different ways. Instead of shuffling a single layer, we shuffle all layers from the last to the specific depth (last 2 convolutional layers are shuffled in Fig. 1) in order to prevent the model from remembering encountered permutations. Memorization of random patterns is something that deep networks have been shown to be powerful at Zhang et al. (2017).

Global Average Pooling and Fully Connected Layers: *Shuffle convolution* is an intuitive way of destroying spatial information but it also makes it impossible to learn correlations across channels for a particular spatial location. Furthermore, shuffling introduces undesirable randomness into the model so that during evaluation multiple forward passes are needed to acquire an estimate of the mean of the output. A simple deterministic alternative achieving a similar goal is what we call *GAP+FC*. The deployment of Global Average Pooling (GAP) after an intermediate layer, and substitute all the subsequent ones by fully connected layers. Compared to *shuffle conv*, it is a much more efficient way to avoid learning spatial information at intermediate layers because it shrinks the spatial size of feature maps to one. Fig. 1 demonstrates a toy example of a CNN with the last two convolutional layers modified by *GAP+FC*.

1x1 Convolution: *GAP+FC* collapses the spatial information to a size of 1. However, reducing the spatial size potentially influences the expressive ability of the model. For example, the point-wise difference of two consecutive 7×7 feature maps lies in the 49 dimension space while the difference of two 1×1 feature maps is just a single value, so if the information would be conveyed by the order of the feature maps, larger feature map size tends to be more expressive. In order to retain the information kept in the spatial dimensions but restrict the model to be invariant to the relationships between spatial locations, we propose as an alternative the use of 1x1 convolutions, which replaces the 3x3 convolutions at last layers in a network. It differs from shuffle conv in that the activation at the same spatial location is aligned. Fig. 1 gives a small demonstration where the last 2 layers from a toy CNN are modified. It is worth noting that ResNets use stride-two convolution to downsample the feature maps at the end of bottleneck. Such down-sampling strategy is not ideal for 1x1 convolution because it ignores more than $3/4$ of the input. So we use max or average pooling with $2x2$ windows as our down-sampling method instead.

3.2 Experimental Setup

We test different architectures with shuffle conv, GAP+FC and 1x1Conv on 3 datasets: CIFAR100, Small-ImageNet-32x32 Chrabaszcz et al. (2017) and ImageNet. We measure in each experiment the top-1 accuracy and the number of model parameters. We will take an existing model and apply the modification to layers from the last layer on. The rest of the setup and hyper-parameters remain the same as the baseline model. By shuffle conv or GAP+FC or 1x1Conv, our modification on the baseline model always starts from the last layer and is consecutively extended to the first layer. We denote as $\boldsymbol{K}$ the number of modified convolutional layers or sub-modules counting from the last layer on. The rest of the operations, like skip connections, and modules remain the same. 2×2 average pooling with stride 2 is used for down-sampling in all experiments due to the ablation of down-sampling methods in section 4.4.

For the VGG-16 architecture, the modification is only performed on the convolutional layers as illustrated in Fig. 1. $\boldsymbol{K}$ varies from 0 (representing the baseline) to 13 since 13 out of the 16 layers are convolutional. For the ResNet-50 architecture with 16 bottleneck sub-modules, one bottleneck is considered as one layer and the modification is only applied onto the 3×3 convolutions inside since they are the only operation with spatial extent, the rest of the configuration remains the same as in the baseline model (see Appendix foran example of modified ResNet-50 architecture).

For CIFAR100 and Small-ImageNet-32x32 experiments, the first convolution in ResNet is set to 3×3 with stride 1 and the first max pooling layer is removed so that the final feature map size is 4×4. For each architecture, we first reproduce the original result on the benchmark as our baseline, and then the same training scheme is directly used to train our models. All models in the same set of experiments are trained with the same setup from scratch and they are initialized by the same random seed. During testing, we make sure to use a different random seed than during training.

4 Results

We first present an in-depth study of our main observations on CIFAR100 for VGG-16 and ResNet-50 in section 4.1 and then verify them on other datasets and architectures in section 4.3. Finally, the influence of the depth and receptive field size is discussed in section 4.4.

4.1 Spatial information at last layers is not necessary thus models can be simplified

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

Contradicting to the common sense, recent works suggest a less important role of spatial information in image classification task. Here we take a further step to study the robustness of the model against the absence of the spatial information at test time by applying *Shuffle Conv*. More specifically, we substitute the last 3 convolutional layers (see Appendix A.4 for more results on other layers) of a pre-trained VGG-16 with shuffle conv at test time on CIFAR100 such that the spatial information is neglected in those layers. Because random shuffle is independent at each forward pass, the final test accuracy will be the average of 200 evaluations and the standard deviation is also present. The left table in 2 clean → shuffle shows the model from the clean training scheme gives around 1% test accuracy, which is the same as random guess on CIFAR100, when evaluated with random shuffle. However, if the shuffle conv is infused into the model at training time, then the baseline performance can be achieved no matter whether random shuffle appears at test time as shown in the left table of 2 (73.67% for shuffle → shuffle and 73.57% for shuffle → clean).

We thus design the following experiment: we modify the last $\boldsymbol{K}$ convolutional or bottleneck layers of VGG-16 or ResNet-50 on CIFAR100 by *Shuffle Conv* (both at training and test time), *GAP+FC*, and *1x1Conv* 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 with different $\boldsymbol{K}$ are then trained on the training set with the same setup and evaluated on the hold-out validation set of CIFAR100.

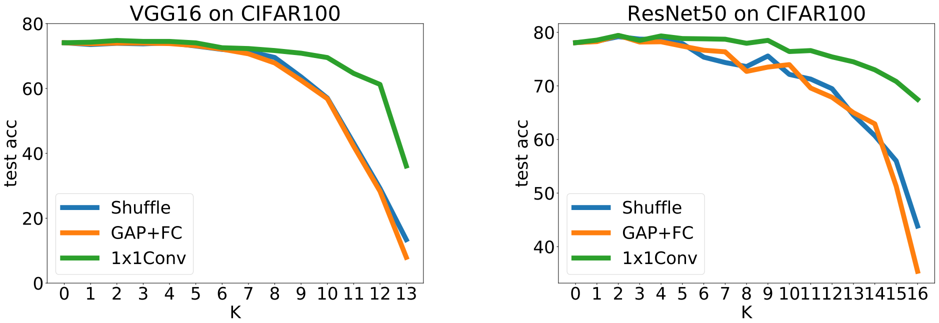

Figure 2: Classification results for VGG-16 and ResNet-50 on CIFAR100. ***K*** is the number of modified last layers, which refer to convolutional layers for VGG-16 and bottlenecks for ResNet-50. VGG-16 has 13 convolutional layers and ResNet-50 has 16 bottleneck sub-modules. All models are trained with the same setup. Curves from GAP+FC look similar to shuffle conv. Test accuracy can be preserved even the last several layers are modified by shuffle conv or GAP+FC or 1x1Conv, suggesting that spatial information at last layers is not necessary for a good test accuracy.

Model	VGG16					ResNet50				
Method	baseline	GAP+FC		1x1Conv		baseline	GAP+FC		1x1Conv	
K	-	5	2	5	2	-	5	2	9	2
Top-1(%)	74.12	73.21	74.00	74.06	74.80	78.06	77.42	79.42	78.49	79.42
#Params(M)	34.02	23.53	29.82	23.53	29.82	23.71	16.37	19.51	14.26	19.51

Table 1: Table summarizes the top-1 accuracy and the number of parameters of different ***K*** on CIFAR100 for VGG-16 and ResNet-50 with GAP+FC and 1x1Conv. ***K*** is defined as the number of modified layers counting from the last layer. The first column for each modification method shows the most compressed model within 1% accuracy difference to the corresponding baseline model and the second column presents the best performed model for each modification method. We can see that 1x1Conv gives even a slightly higher test accuracy while having fewer parameters.

The results on CIFAR100 for VGG-16 and ResNet-50 are shown in Fig. 2. The x-axis is the number of modified layers ***K***, ranging from 0 to the total number of convolutional or bottleneck layers. $\boldsymbol{K} = 0$ is the baseline model performance without modifying any layer. As we can see in the right of Fig. 2, with the increasing number of modified layers, the performance of ResNet-50 drops surprisingly slowly for our three methods consistently. For example, Shuffle conv can modify up to the last 5 layers of ResNet-50 while maintaining similar baseline performance i.e., Shuffle conv(***K***=5) achieves 77.89% accuracy v.s. 78.06% accuracy of the baseline (***K***=0). 1x1Conv and GAP+FC can preserve the baseline performance until $K = 5$ and $K = 9$, where the feature map size is 8 and 16, respectively. For VGG-16, as shown in the left of Fig. 2, a similar trend can be observed. Shuffle conv, GAP+FC, and 1x1Conv are able to tolerate modification of the last 5 layers without losing any accuracy. This is in strong contrast to the common belief that the spatial information is essential for object recognition tasks.

One obvious advantage of our methods is that 1x1Conv and GAP+FC can reduce the number of model parameters without affecting the performance. Table 1 summarizes how many parameters our GAP+FC and 1x1Conv can reduce for VGG16 and ResNet50. We observe that our 1x1Conv (K=5) achieves nearly identical results (74.06%) to the VGG-16 baseline (74.12%), while reducing the number of parameters from 34.02M to 23.53M. For ResNet50, our 1x1Conv (K=2), with only 19.51M parameters, even outperforms the ResNet50 baseline with 23.71M parameters by 1.36%. Similar results can be observed with our GAP+FC. This implies that CNNs may be easily simplified by substituting last layers with 1x1Conv or GAP+FC with no performance drop.

As a side effect, we find that GAP+FC and 1x1Conv have a regularization effect that can lead to improved generalization performance when data augmentation is not applied. Fig. 2 shows the test accuracy of modified ResNet-50 via GAP+FC and 1x1Conv trained with and without data augmentation. While the models trained with data augmentation show similar test accuracy, we observe a significant performance improvement over the baseline on ResNet-50 trained without data augmentation, e.g 1x1Conv outperforms the baseline by 8.01% on CIFAR100 when several last layers are modified. Unfortunately, this effect doesn't hold across other architectures and datasets.

Schemes		Top-1(%)
Train	Test	
shuffle	shuffle	73.67±1.03
shuffle	clean	73.57±0.97
clean	shuffle	1.06±1.15
clean	clean	74.10

Config. for ResNet-50		Top-1(%)
w/ DataAug	baseline	78.06
	GAP+FC	79.42
	1x1Conv	79.42
w/o DataAug	baseline	65.64
	GAP+FC	68.40
	1x1Conv	**73.65**

Table 2: Left: Top-1 accuracy of VGG-16 with random shuffle enabled at either training and test time for the last 3 layers. Shuffled model is robust to the standard test scheme while the test accuracy of a standard VGG-16 drops to the random guess level if evaluated with shuffling. Right: Effect of data augmentation on classification results for ResNet-50 on CIFAR100. The data augmentation here is the random flipping and the random cropping. We present here the best performed model for each method. We can see that modified models reach higher test accuracy when data augmentation is not applied. ResNet-50 with 1x1Conv trained without data augmentation shows a significant performance improvement over the baseline from 65.64% to 73.65% on CIFAR100.

Model	VGG16					ResNet50				
Method	baseline	GAP+FC		1x1Conv		baseline	GAP+FC		1x1Conv	
K	-	4	2	7	4	-	3	1	6	1
Top-1(%)	46.59	46.05	46.50	45.44	46.64	61.87	61.11	61.72	61.00	61.95
#Params(M)	37.70	29.31	33.50	25.64	29.31	25.55	19.26	23.45	17.68	23.45

Table 3: Image classification results on Small-ImageNet for VGG16 and ResNet50 with GAP+FC and 1x1Conv. ***K*** is defined as the number of modified layers counting from the last layer.

4.2 DISCUSSION

Our experiments in Table 2 left clearly show that ordinary models by default don't possess the invariance to the absence of the spatial information. In contrast to the common wisdom, we find that spatial information can be neglected from a significant number of last layers without any performance drop if the invariance is imposed at training, which suggests that *spatial information at last layers is not necessary for a good performance*. We should however notice that it doesn't indicate that models whose prediction is based on the spatial information can't 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, our observation implies it is beneficial to have different designs for different layers since there is no necessity to encode spatial information in the last layers (see Appendix A.3 for discussion on first layers), therefore reducing the model complexity.

Comparing our three methods, we observe that 1x1Conv is more robust to the absence of the spatial information while Shuffle Conv and GAP+FC perform similarly for both VGG-16 and ResNet-50. This implies that CNNs can still benefit from the larger size of activation maps even though its spatial information is not presented.

4.3 GENERALIZATION TO OTHER DATASETS AND ARCHITECTURES

Since CIFAR100 is a relatively easy dataset with centered objects belonging to only 100 classes, we conduct in the following experiments on more complex inputs: small-ImageNet and ImageNet, whereas small-ImageNet is a down-sampled version of the latter (from 256×256 to 32×32). The results on Small-ImageNet are summarized in the Table 3 (see more details in the Appendix). GAP+FC and 1x1Conv present a similar behavior as on CIFAR100 dataset. And the gap between the performance of GAP+FC and 1x1Conv increases, the maximal number of layers that can be modified on ResNet50 for GAP+FC and 1x1Conv are 3 and 6. This implies that spatial information at last layers of CNNs are not necessary for good performance on the datasets with enough complexity.

Furthermore, we conduct experiments for different architectures on full ImageNet with an input image size of 224×224. We first reproduce baselines as in the original papers and then apply the same training scheme directly to train our models. Here we only evaluate 1x1Conv due to its superiority over GAP+FC and due to its excessive computational overhead training on the full ImageNet dataset. In Table 4, we observe that spatial information can be ignored at last layers

Model	ResNet152		ResNet50		VGG16		MobileNetV2		SqueezeNet	
Method	base.	1x1Conv	base.	1x1Conv	base.	1x1Conv	base.	1x1Conv	base.	1x1Conv
K	-	19	-	5	-	1	-	2	-	2
Top-1(%)	77.66	77.82	75.22	75.53	72.96	72.51	72.29	72.06	60.07	60.64
#Params(M)	60.19	45.51	25.55	18.22	37.70	35.60	3.50	3.48	1.25	1.23

Table 4: ImageNet classification results for ResNet-152, ResNet-50, VGG-16 MobilenetV2 and SqueezeNet with 1x1Conv. The best performed models are selected for 1x1Conv. We observe that our modification reduces the number of parameter without loss of the test accuracy.

without losing any test accuracy on the ImageNet. For example, the last 19 layers of ResNet152 can be modified into 1x1 convolution (the feature map size is 14×14) while attaining the same performance. Moreover, we find that the number of spatial invariant layers of ResNet50 becomes smaller compared to ResNet152 i.e., $k = 5$ v.s. $k = 19$. Recall that last 6 layers on ResNet-50 can be modified by the 1x1Conv on Small-ImageNet resulting in a 8×8 final ferule map size, it is surprising that this number becomes 5 on ImageNet where the final feature map is 14, considering the large difference in the ability of expressiveness.

So far, we have evaluated our methods with large models that have been shown to have incredible capacity to learn even from random labels Zhang et al. (2017). A hypothesis could be that the models we test are very complex to begin with such that it is of no surprise that they learn the relevant representations in earlier layers and can encode the information necessary to classify in very few dimensions. To approach this question, we deploy our experiments on architectures that have been specifically designed to be of minimal complexity in order to save memory and reduce the number of floating point operations. Hence, we evaluate MobileNetV2 Sandler et al. (2018) with $3.5M$ parameters and SqueezeNet Iandola et al. (2017) with $1.25M$ parameters, both of which are able to reach competitive performance on ImageNet. MobileNetV2 uses the inverted residual bottleneck as their building block where the input tensor is first expanded along the channel dimension and then a 3×3 depth-wise convolution is performed before the number of channels is reduced to the output dimension by the final 1×1 convolution. In our modification we simply remove the 3×3 depth-wise convolution together with its ReLU and batch normalization. SqueezeNet is composed of fire modules, which leverage the strategies from Iandola et al. (2017) to reduce the model parameters. It first squeezes the number of channels by a 1×1 convolution and then expands by a mixture of 1×1 convolutions and 3×3 convolutions. In our modification, we replace all 3×3 convolutions in the expand phase by 1×1 convolutions. The results in Table 4 show that the last two conv layers of both MobileNetV2 and SqueezeNet are also spatial invariant i.e., neglecting the spatial information at those 2 last layers does not affect the performance at all, despite the minimal model complexity.

The experiments on Small-ImageNet and ImageNet confirm again the claim in section 4.2 that the spatial information at last layers is not necessary for a good performance and its generalizability across architectures can lead to a further reduction of the number of model parameters even on models that are already very efficient, e.g. MobileNetV2 and SqueezeNet.

4.4 Effect of depth and receptive field size

In the previous section, we observed that 1x1Conv gives the best performance in the sense that spatial information of more layers can be neglected without affect the test accuracy. Here we investigate whether these modified layers are of importance at all or whether they can be stripped of the architecture entirely. The relationship between the receptive field size of a layer and whether it can be modified without performance impact is evaluated subsequently.

Importance of the Depth. We saw previously that 1x1Conv gives the best performance in the sense that spatial information at more layers can be neglected without affect the overall test accuracy. Here we ask whether those modified layers can be neglected altogether, effectively reducing the depth. We first pick the most compressed ResNet-50 with the same test accuracy as the baseline on Small-ImageNet, last 6 sub-modules are modified by 1x1Conv. We then strip off one modified layer at a time from the last layer on, resulting in 6 different models which are trained with the same configuration. The result is shown in Fig. 3 left. With the increase of the number of 1x1 convolutional layers, the test accuracy also increases. So even though the spatial information at last layers is not necessary, those last layers are still essential for good performance. It appears

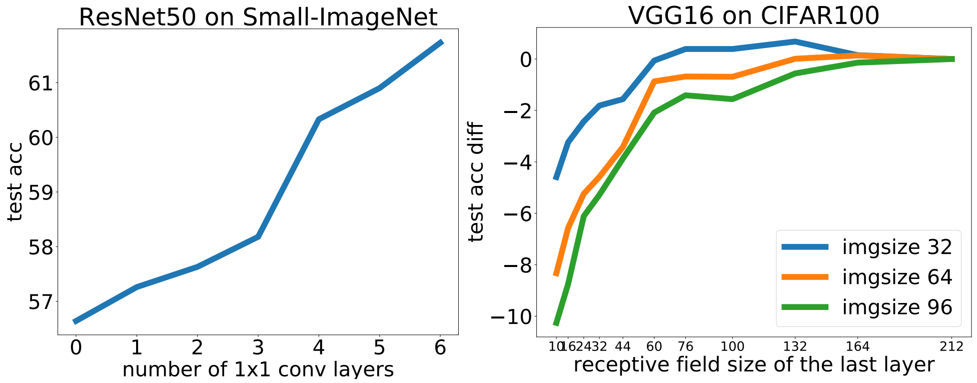

Figure 3: Left: The test accuracy of ResNet-50 on Small-ImageNet increases monotonically with the increase of the number of 1x1 convolutional layers. Right: The relation between the receptive field size and the test accuracy difference to the baseline for different image size on VGG-16 over CIFAR100 shows that the test accuracy saturates with the increase of the receptive field size for a given image size. The minimal required receptive field tends to be larger for larger image size and this minimum is normally larger than the actual image size. The exact relation is however unclear.

that the spatial information is marginalized out at some particular depth and the resulting non-linear transformations are solely used to disentangle the depth wise information.

Relationship to the Receptive Field. A reason for that marginalization of spatial information could be hypothesized to be related to the receptive field size of a particular layer. If the receptive field size of a layer is greater or equal to the size of the image, does that tell us whether all following layers can be manipulated? We choose to ablate VGG-16 because the receptive field for a multi-branch network is not properly defined as it can only state a theoretical upper bound and do so on CIFAR100 as each object normally occupies the entire image. We replace the 3×3 convolutional layers in VGG-16 by 1×1 convolutional layers from the last layer on and until the first layer, thereby varying the receptive field size of the last convolutional layer in our model. Results are shown in Fig. 3 right. Y-axis is the test accuracy difference between the modified model and the baseline model.

We can see that the test accuracy saturates with the increase of the receptive field size for a given image size. In order to reach the saturation, it seems that the minimal required receptive field size has to exceeds the actual image size by a relatively large margin and this margin increases for larger image size. For example, the model reaches approximately the same test accuracy as a vanilla VGG-16 with receptive field being 50 for 32×32 input image, and the same number becomes around 120 for 64×64 input image. This is maybe because the effective receptive field is normally smaller than the theoretical receptive field Luo et al. (2016). However, it is still not really possible to tell a quantitative relation between the required receptive field size and the image size since there are too few data points and it is hard to confirm if an architecture with a specific final receptive field size is sufficient to obtain the baseline performance.

5 Conclusion and future work

To conclude, we empirically show that last layers of CNNs are robust to the absence of the spatial information, which is commonly assumed to be important for object recognition tasks. Our proposed methods, without accessing any spatial information at last layers of modern CNNs, are able to achieve competitive results on several object recognition datasets incuding CIFAR100, Small-ImageNet and ImageNet. We suggest a good rule of thumb for CNN architectures: using 1x1 convolution or fully connected layers at last layers reduces the number of parameters without affecting the performance. An interesting future direction is to study whether our methods can generalize to other computer vision tasks, e.g., object detection and pose estimation where the spatial relationships are vital for localizing objects.

REFERENCES

Wieland Brendel and Matthias Bethge. Approximating CNNs with bag-of-local-features models works surprisingly well on imagenet. In *International Conference on Learning Representations*, 2019. URL `https://openreview.net/forum?id=SkfMWhAqYQ`.

Jiewei Cao, Zi Huang, and Heng Tao Shen. Local deep descriptors in bag-of-words for image retrieval. In *Proceedings of the on Thematic Workshops of ACM Multimedia 2017*, pp. 52–58. ACM, 2017.

Patryk Chrabaszcz, Ilya Loshchilov, and Frank Hutter. A downsampled variant of imagenet as an alternative to the cifar datasets. *arXiv preprint arXiv:1707.08819*, 2017.

Jifeng Dai, Haozhi Qi, Yuwen Xiong, Yi Li, Guodong Zhang, Han Hu, and Yichen Wei. Deformable convolutional networks. In *Proceedings of the IEEE international conference on computer vision*, pp. 764–773, 2017.

Jiangfan Feng, Yuanyuan Liu, and Lin Wu. Bag of visual words model with deep spatial features for geographical scene classification. *Computational intelligence and neuroscience*, 2017, 2017.

Robert Geirhos, Patricia Rubisch, Claudio Michaelis, Matthias Bethge, Felix A. Wichmann, and Wieland Brendel. Imagenet-trained CNNs are biased towards texture; increasing shape bias improves accuracy and robustness. In *International Conference on Learning Representations*, 2019. URL `https://openreview.net/forum?id=Bygh9j09KX`.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, 2016.

Derek Hoiem, Alexei A Efros, and Martial Hebert. Putting objects in perspective. *International Journal of Computer Vision*, 80(1):3–15, 2008.

Forrest N. Iandola, Matthew W. Moskewicz, Khalid Ashraf, Song Han, William J. Dally, and Kurt Keutzer. Squeezenet: Alexnet-level accuracy with 50x fewer parameters and ¡1mb model size. *ArXiv*, abs/1602.07360, 2017.

Wenjie Luo, Yujia Li, Raquel Urtasun, and Richard Zemel. Understanding the effective receptive field in deep convolutional neural networks. In *Advances in neural information processing systems*, pp. 4898–4906, 2016.

Eva Mohedano, Kevin McGuinness, Noel E O'Connor, Amaia Salvador, Ferran Marques, and Xavier Giro-i Nieto. Bags of local convolutional features for scalable instance search. In *Proceedings of the 2016 ACM on International Conference on Multimedia Retrieval*, pp. 327–331. ACM, 2016.

Edouard Oyallon, Eugene Belilovsky, and Sergey Zagoruyko. Scaling the scattering transform: Deep hybrid networks. In *Proceedings of the IEEE international conference on computer vision*, pp. 5618–5627, 2017.

Mark Sandler, Andrew Howard, Menglong Zhu, Andrey Zhmoginov, and Liang-Chieh Chen. Mobilenetv2: Inverted residuals and linear bottlenecks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 4510–4520, 2018.

J Springenberg, Alexey Dosovitskiy, Thomas Brox, and M Riedmiller. Striving for simplicity: The all convolutional net. In *ICLR (workshop track)*, 2015.

Antonio Torralba, Kevin P Murphy, William T Freeman, and Mark A Rubin. Context-based vision system for place and object recognition. 2003.

Chiyuan Zhang, Samy Bengio, Moritz Hardt, Benjamin Recht, and Oriol Vinyals. Understanding deep learning requires rethinking generalization. In *International Conference on Learning Representations*, 2017. URL `https://openreview.net/forum?id=Sy8gdB9xx`.

Hengshuang Zhao, Jianping Shi, Xiaojuan Qi, Xiaogang Wang, and Jiaya Jia. Pyramid scene parsing network. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 2881–2890, 2017.

Xizhou Zhu, Han Hu, Stephen Lin, and Jifeng Dai. Deformable convnets v2: More deformable, better results. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 9308–9316, 2019.

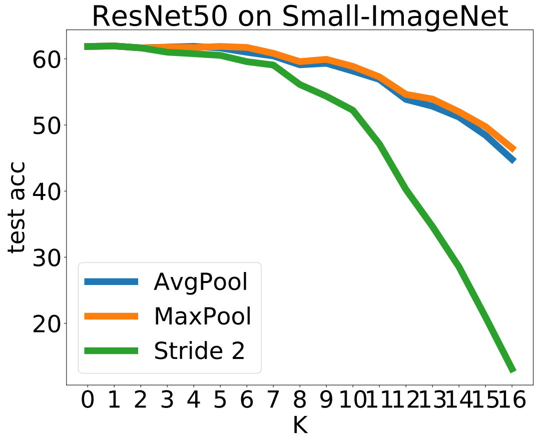

Figure 4: Pooling methods give better test accuracy than convolution with stride 2 as down-sampling method for ResNet-50 on Small-ImageNet.

A Appendix

A.1 Effect of Down-Sampling Method

Convolution with stride 2 was suggested by Springenberg et al. (2015) for 3×3 filters as a replacement for pooling layers as the down-sampling method. For example, ResNets use 1×1 convolution with stride 2 to reduce the feature map size. However, a direct adaptation leads to a failure for our 1x1Conv. In figure 4, we observe a more rapid decrease of the test accuracy for stride 2 down-sampling than average pooling and max pooling on ResNet50 over Small-ImageNet. With the same test accuracy as the baseline, the number of modifiable layers is 3 for convolution with stride 2 and 6 for average pooling. The reason for the failure of the stride 2 case may lie in the fact that 1x1 convolution doesn't have the spatial extent, so a down-sampling will ignore 75% of the activations even they may convey the majority of the information. In an ordinary bottleneck that performs down-sampling, the lost information in the main branch can be replenished from the skip connection where 3×3 convolution is deployed to ensure the information at each location is processed. In our modification, however, the skip connection branch will suffer from the loss of the information as well due to 1x1 convolution.

Average pooling or max pooling on the other hand doesn't have this problem and their performance according to the plot doesn't have significant difference to each other.

A.2 Results on Small-ImageNet

We test the necessity of spatial information by GAP+FC and 1x1Conv for VGG-16 and ResNet-50 on Small-ImageNet. Experimental setup is the same as the CIFAR100 experiment. Results are shown in Fig. 5. Within 1% test accuracy difference, GAP+FC manages to replace the last 4 layers in VGG-16 and 1x1Conv can replace the last 7 layers (46.05% and 45.44% compared to the baseline performance 46.59%, respectively). Similarly, the test accuracy can be preserved until $\boldsymbol{K} = 3$ and $\boldsymbol{K} = 6$ for GAP+FC and 1x1Conv, which confirms the better performance of 1x1Conv over GAP+FC. This indicates spatial information at last layers is not necessary for a good performance.

A.3 Single Layer Shuffle

Previous experiments always apply shuffle conv from one specific layer until the last layer in a network. We test here the impact of random shuffle at different depth by applying shuffle conv at one single layer at a time. The result of VGG-16 on CIFAR100 is summarized in Fig. 6 where the x-axis is the layer index (VGG-16 has 13 convolutional layers). We plot the baseline performance with an horizontal line alongside the modified models in order to show a clearer comparison. We can see an overall similar trend as multiple layer shuffle in Fig. 2, the test accuracy drops slowly

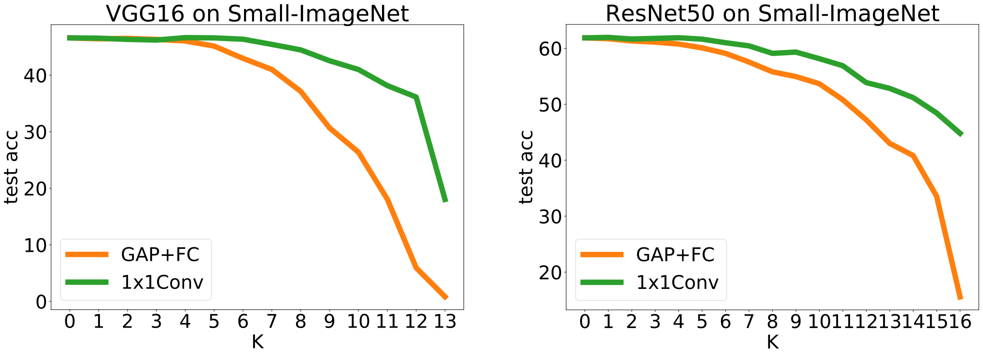

Figure 5: Classification results for VGG-16 and ResNet-50 on Small-ImageNet. $\boldsymbol{K}$ is the number of modified last layers and 0 indicates the baseline performance. We observe that test accuracy can be preserved even the last several layers are modified by GAP+FC or 1x1Conv, suggesting that spatial information at last layers is not necessary for a good test accuracy. All models are trained with the same setup.

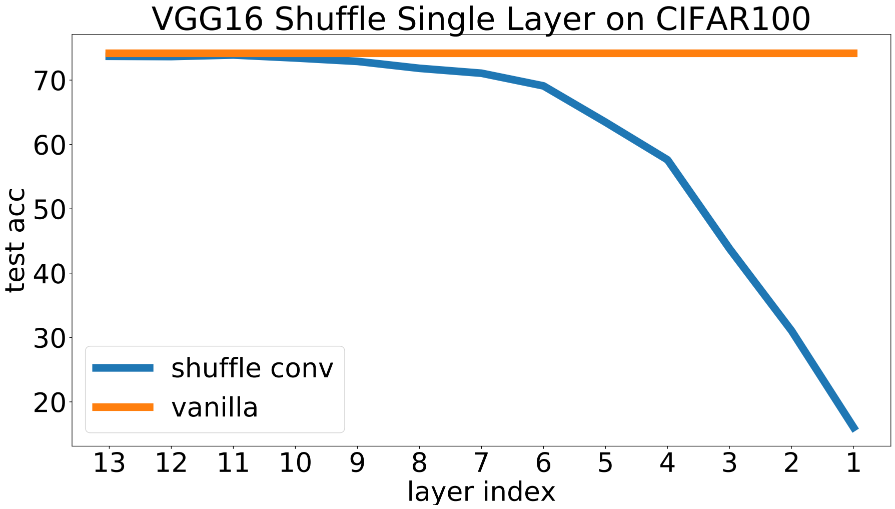

Figure 6: The orange curve is the test accuracy of the vanilla VGG-16 i.e. the baseline. The blue curve shows the test accuracy of the VGG-16 with a single convolutional layer modified by shuffle conv at different depth. The x-axis is the layer index with 13 being the last convolutional layer in VGG-16. Random shuffle is applied both at training and test time. The result implies random shuffle has a larger impact at first layers than the last layers.

with the decrease of the layer index. The baseline performance is maintained for the last 4 layers, which implies random shuffle has a larger impact at first layers than the last layers.

A.4 MISMATCHED TRAINING AND TEST SCHEMES

In Table. 2 left, we presented the test accuracy of a specific model whose last 3 layers are replaced by shuffle conv under mismatched training and test schemes. We show here the complete results of models with different $\boldsymbol{K}$ in Fig. 7. The green curve which is obtained by evaluating the baseline with different $\boldsymbol{K}$ at test time falls to random guess on CIFAR100, compared to the red curve which represents the baseline with clean training and test schemes. And the shuffled models which maintain the baseline accuracy have very similar behavior (the overlapped part of orange curve and blue curve) no matter whether random shuffle appears during evaluation. However, there is a gradually increasing gap between these 2 curve when the shuffled model can't preserve the baseline performance, that

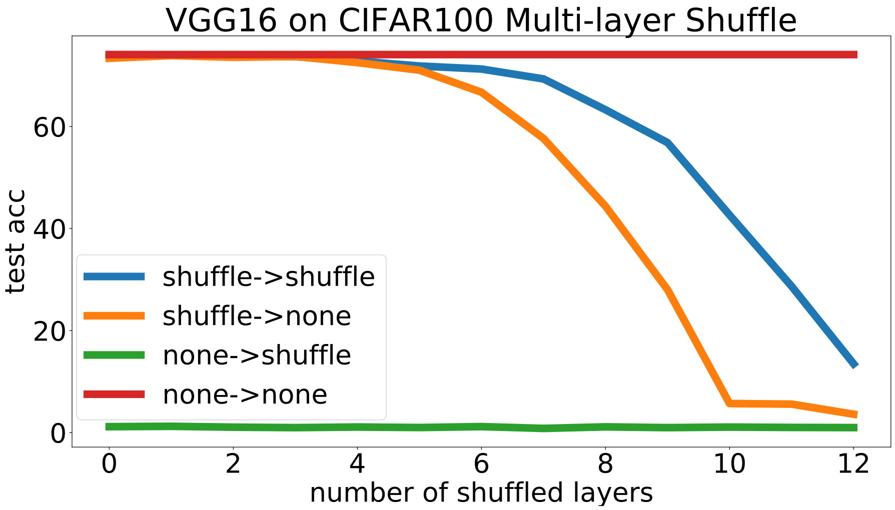

Figure 7: VGG-16 test accuracy with mismatched training and test schemes. The performance of a standard VGG-16 drops to the random guess level if evaluated with shuffling while shuffled models at last layer are invariant to this.

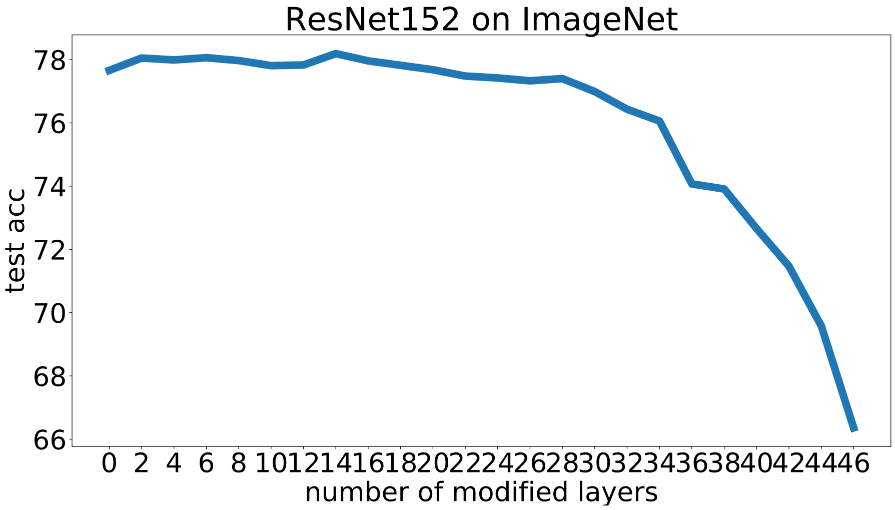

Figure 8: Classification results for ResNet-152 modified by 1x1Conv on ImageNet. The most compressed model without affecting the test performance has the last 19 layers being modified by 1x1Conv.

consistent schemes gives significant higher accuracy than the inconsistent one. Unfortunately, the reason is not fully understood.

A.5 Results on ImageNet

Fig. 8 and 9 show the complete results of test accuracy of ResNet-152 and ResNet-50 being modified by 1x1Conv on ImageNet. All models are trained with the same scheme as in He et al. (2016) from scratch. The claim that spatial information is not necessary at last layers generalizes well on ImageNet.

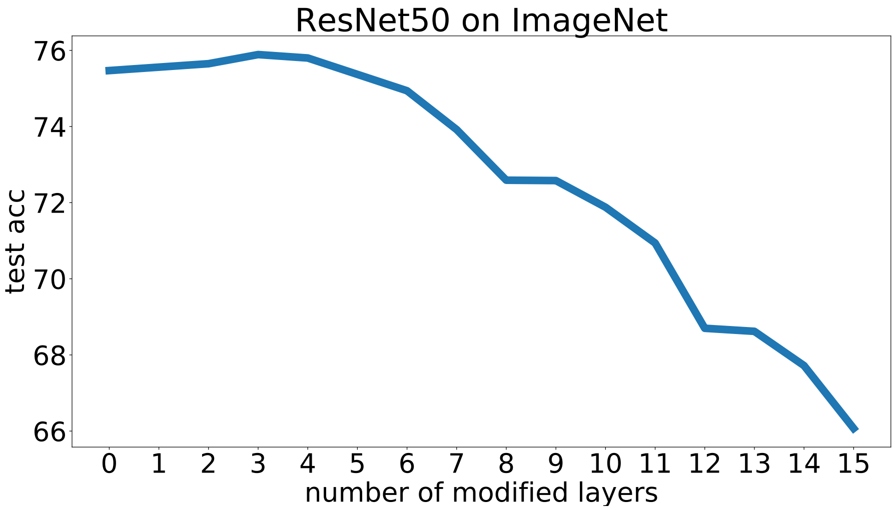

Figure 9: Classification results for ResNet-50 modified by 1x1Conv on ImageNet. The most compressed model without affecting the test performance has the last 5 layers being modified by 1x1Conv.

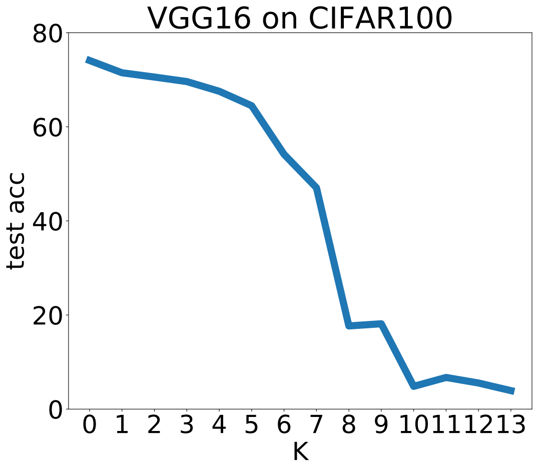

Figure 10: In contrast to random spatial shuffle, VGG-16 doesn't seem to be robust to the channel shuffle. The test accuracy drops from 74.10% to 71.49% with only the last layer being shuffled.

A.6 Channel Shuffle

We test here another type of random shuffle along the depth of feature maps. It randomly swaps the order of the feature maps along the channel dimension in each forward pass and is denoted as *channel shuffle*. The experiments are run for VGG-16 on CIFAR100. Fig. 10 shows the change of the test accuracy with the number of layers K that is modified by channel shuffle increasing. Besides an overall decreasing trend, the test accuracy drops much faster than that from random spatial shuffle (74.10% to 71.49% with only the last layer being shuffled), which implies a much weaker robustness of the model against channel shuffle. We therefore assume a more important role of the order of the feature maps in encoding the information at last layers.

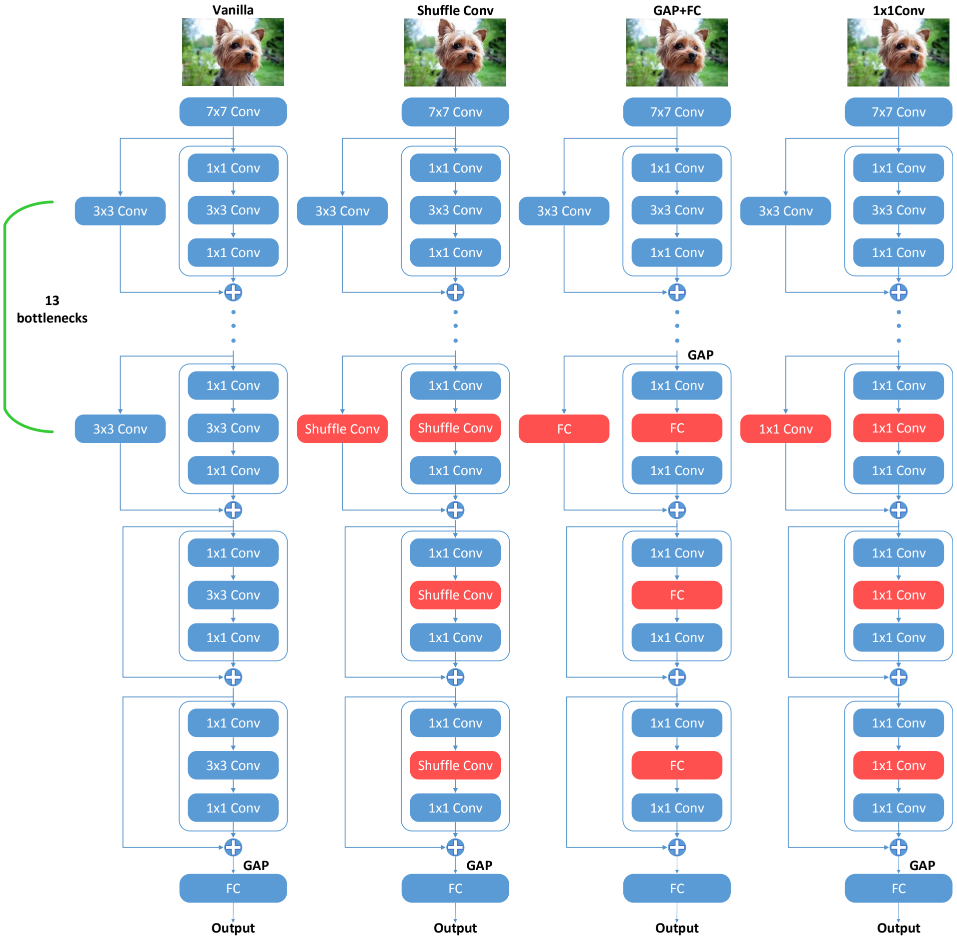

Figure 11: An example of ResNet-50 with the last 3 bottlenecks being modified by shuffle conv, GAP+FC and 1x1Conv.

A.7 RESNET50 ARCHITECTURE

Fig. 11 shows an example of ResNet-50 with the last 3 bottlenecks being modified by shuffle conv, GAP+FC and 1x1Conv. Our modification is applied only the 3 × 3 convolution in side each bottleneck since it is the only operation that has the spatial extent.