

Peer Review

Review of: "Convolutional Neural Networks Do Work with Pre-Defined Filters"

Evgenii Arsentev¹

1. Independent researcher

1. What the work claims

The authors build a Pre-defined Filter Module (PFM; Section 3.1, Figure 1) shaped like a depthwise convolution. Each input channel is convolved with exactly one fixed 1x3x3 kernel from a pool of 16 edge kernels (8 "uneven" and 8 "even"; Section 3.4, Figure 2). Only the following 1x1 convolution is trained, so the network learns combinations of fixed filter outputs. The kernels are normalized to zero sum and unit absolute sum, and only 8 of the 16 are linearly independent (Section 3.4).

PFNet18 replaces every convolution with a kernel size above 1 in ResNet18 by PFMs (Section 3.3, Tables 1 and 2). It has 1.46 million parameters, 13% of the 11.23 million of ResNet18 (Section 3.3). Both are trained from scratch, five seeds each, on six datasets (Section 4, Table 4). PFNet18 is reported better on Caltech101, FGVC-Aircraft, and Flowers102, close on Stanford Cars, and worse on CIFAR10 and CUB-200-2011 (Section 5.1, Table 4). Edge filters beat random fixed filters (Section 5.5, Table 7), PFNet18 is sensitive to aliasing in the strided skip connections, fixed by a 3x3 Gaussian blur (Section 5.4), and its feature visualizations look more shape-like than those of ResNet18 (Section 5.2, Table 5). They conclude that training spatial kernels is unnecessary for reasonable accuracy, and that the power of deep networks lies mainly in learning how to combine filters (Section 7).

2. Whether the conclusion holds on the data

The paper has a clear, simple, and testable idea, and I appreciate that the whole network uses fixed kernels, not only the first layer (Section 1, Section 3.3). Six datasets with five seeds and mean plus spread (Table 4), an ablation that separates the choice of filters from the restriction itself (Table 7), and public

code (Abstract) make the work easy to check. The aliasing analysis (Section 5.4) is an honest negative finding, and I value it.

On the main comparison, I would correct a few numbers in the text. In Table 4, Caltech101 gives PFNet18 65.60 ± 0.66 vs. ResNet18 57.19 ± 0.78 , a difference of 8.41 points. Section 5.1 says "almost 10%" and Section 7 says "an absolute increase of 10%". FGVC-Aircraft is 75.49 ± 0.29 vs. 73.32 ± 1.06 , only 2.17 points, so "10%" in Section 7 does not describe all three datasets. Flowers102 is 80.66 ± 0.35 vs. 73.40 ± 0.34 , a difference of 7.26, which matches Section 5.1. Stanford Cars (77.06 ± 0.42 vs. 77.90 ± 0.37), CIFAR10 (92.15 ± 0.18 vs. 94.33 ± 0.11), and CUB-200-2011 (53.00 ± 0.55 vs. 58.51 ± 0.53) go to ResNet18, so PFNet18 wins on three of six datasets.

On the Flowers102 comparison, the PFNet18 value 80.66 ± 0.35 is the "No aliasing, default" row of Table 7 (blur in the three strided skip connections, Section 5.4), while ResNet18 73.40 ± 0.34 is the "Default" row. Table 7 also has "ResNet18, No aliasing" at 74.36 ± 0.57 , so a like-for-like gap is $80.66 - 74.36 = 6.30$ points, not 7.26. Without blur, PFNet18 reaches 72.40 ± 0.59 , below the default ResNet18. It would help to state whether the Table 4 rows for the other five datasets use the blurred PFNet18 against a non-blurred ResNet18.

On statistics, there are five seeds per cell (Section 4.2), but the $\pm$ values are not labelled as standard deviation or standard error, and no significance test is reported, although Section 6 calls the gains "significant". For Caltech101, the split is 20 training and 10 test images per class, random (Section 4.1), so about 1010 test images. I would add whether the split changes with the seed. No separate validation set is described for any dataset, and for Flowers102, the official validation set is merged into training (Section 4.1). It would help to say how the hyperparameters (Section 4.2) were chosen.

The ablation is done only on Flowers102 (Section 5.5), one of the two datasets with the largest reported gains. There, random frozen filters give 73.98 ± 1.24 , trainable random 74.54 ± 0.88 , trainable edge-initialized 80.11 ± 0.62 , and first ReLU removed 75.21 ± 0.23 (Table 7). I would repeat at least part of this on a second dataset. The feature visualizations were "picked by hand" (Section 5.2), so the shape-versus-texture statement (Section 5.2, Section 6) rests on examples without a quantitative measure. As baselines, only ResNet18 is used. I would add a parameter-matched network (a ResNet of about 1.46 million parameters) and a MobileNet or EfficientNet, since Section 7 names mobile devices as a use case. The trainable-depthwise variant exists only on Flowers102 (Table 7).

The evidence supports that fixed edge kernels throughout a ResNet18-shaped network can match or beat a from-scratch ResNet18 on some small fine-grained datasets. The wider statement in Section 7 about

the power of deep networks needs more than this.

3. Reproducibility

The code is on GitHub (Abstract), and the datasets are public; the official split is used for CUB-200-2011 and Flowers102, and the provided training and test sets for FGVC-Aircraft and Stanford Cars (Section 4.1). The training setup in Section 4.2 is well described: Kaiming normal initialization, Lamb optimizer, cross-entropy, batch size 64, 300 epochs, initial learning rate 0.003 reduced step-wise to $lr_{init} \times 10^{-2}$, five seeds, random crop and horizontal flip, PyTorch, one NVidia GTX 1080 Ti. The kernel normalization rule (Section 3.4) and the CIFAR10 stem changes for both models (Section 4.1) are also clear.

A few details are missing or ambiguous, and I would add them. Section 4.2 states "The weight decay is 1." Is this the intended value, or a typo? Please confirm or correct it. It would help to give the epochs at which the learning rate steps, the Lamb betas and epsilon, the crop size and resize for each dataset (Table 6 uses 224x224 input), the input normalization, the PyTorch and CUDA versions, and the seed values. For Caltech101, please say which images form the random split and whether it is fixed across seeds. Section 5.5 does not give the range of the uniform distribution for random kernels, and Section 5.4 names a 3x3 Gaussian filter without the exact blur kernel parameters. The 16 kernel values are shown only as an image (Figure 2); a numeric table would remove doubt. The Table 6 timings depend on the implementation and the GPU, and the software versions behind 37 ms vs 27 ms forward pass are not given.

The Acknowledgments say the Version of Record was presented at IJCNN 2023 (doi 10.1109/IJCNN54540.2023.10191449). I review the Qeios version, so it would help to state whether anything differs.

With these details added, an independent rerun would be straightforward.

4. What specifically to fix

1. Section 5.1 says "almost 10%" for Caltech101, and Section 7 says "an absolute increase of 10%" for Caltech101, FGVC-Aircraft, and Flowers102. Table 4 gives $65.60 - 57.19 = 8.41$, $75.49 - 73.32 = 2.17$, and $80.66 - 73.40 = 7.26$ points. I would put these actual differences in the text.
2. In Table 4, the PFNet18 value of 80.66 on Flowers102 comes from the blurred "No aliasing, default" model, but ResNet18 73.40 is the "Default" one. Table 7 lists ResNet18 "No aliasing" at 74.36 ± 0.57 . It would help to report both models with the same skip-connection blur on all six datasets in Table 4, or to say clearly which variant each row uses.

3. Section 4.2 states "The weight decay is 1". Could you confirm or correct this value? The learning-rate step epochs and the Lamb settings are missing too, and I would add them.
4. The $\pm$ values in Table 4 and Table 7 need a label: standard deviation or standard error over five seeds. In Section 6, "significant" could be replaced by a test or by neutral wording. A paired comparison across the five seeds would be enough.
5. For Caltech101 (Section 4.1), you take 10 test images per class from a random split. I would fix this split and publish it, and say whether it changes with the seed. More test images per class would also help, since each test image moves accuracy by about 0.1 point (1 of about 1010).
6. The ablation (Section 5.5, Table 7) runs only on Flowers102. It would be good to repeat at least the "16 frozen filters, random init." and "16 trainable filters, edge init." rows on one dataset where PFNet18 loses, for example, CUB-200-2011 (53.00 vs 58.51, Table 4) or CIFAR10.
7. For the baselines, I would add a parameter-matched ResNet (about 1.46 million parameters) and one light network such as MobileNet or EfficientNet, trained with the same recipe. Section 7 names mobile devices as a use case, and Table 6 shows a PFNet18 forward pass of 37 ms vs. 27 ms for ResNet18 and GPU memory of 3.9 vs. 3.0 GB. So the efficiency case needs this context.
8. The visualizations in Section 5.2 and Table 5 were "picked by hand". I would either soften the shape-versus-texture claim in Section 6 or support it with a quantitative test, for example, accuracy on shape-preserving versus texture-preserving image variants.
9. A few small items. Table 1 calls the blocks "Double HF Residual Module," while the text speaks of Basic Blocks with PFMs (Section 3.3), so one name would help the reader. Please give the 16 kernel values as numbers next to Figure 2. I would also mention the IJCNN 2023 Version of Record (Acknowledgments) near the start, with a note on any differences.

5. Recommendation

Minor revisions. The idea is sound and simply tested, and the core finding, that fixed edge kernels in every layer still give competitive accuracy, is worth publishing. PFNet18 uses 13% of ResNet18 parameters (Section 3.3) and wins on three of six datasets (Table 4). The fixes are mostly text, labeling, and a fairer aliasing comparison. I would check the "10%" statements against Table 4 (Sections 5.1, 7), state the weight decay value (Section 4.2), label the spread, and reword "significant" (Section 6), compare blurred and default fairly on Flowers102 (Table 4 vs. Table 7, 74.36 for ResNet18 without aliasing), and extend the ablation beyond Flowers102 (Section 5.5). This is a promising paper, and I look forward to the revised version.

Declarations

Potential competing interests: No potential competing interests to declare.