Supplementary Material for
Spatio-temporal Filter Analysis Improves 3D-CNN For Action Classification

Takumi Kobayashi, Jiaxing Ye
National Institute of Advanced Industrial Science and Technology
1-1-1 Umezono, Tsukuba, Japan
{takumi.kobayashi,jiaxing.you}@aist.go.jp

A. Analyzing temporal filters

In Sec. 2.1, we show the distribution of temporal filters which are L_2-normalized 3-d vectors thus being distributed on a sphere. For ease of analyzing the distribution, we construct Cartesian coordinates composed of physically interpretable axes as shown in Figure A; the average vector $\propto [1, 1, 1]^\top$, the 1st differential vector $\propto [-1, 0, +1]^\top$ and the 2nd differential vector $\propto [-1, +2, -1]^\top$. In the main manuscript, Figure 1 shows the distribution by projecting the temporal-filter samples from a sphere into a plane depicted by gray color in Figure A. In order to visual further details of the temporal filter distributions, Figure B shows the distributions on two types of planes which are perpendicular to each other; one is spanned by the average and 1st-differential vectors, and the other is by the 1st- and 2nd-differential vectors. Note that as the signs of temporal filters can be arbitrarily given in SVD (1), we plot both the temporal filter $\boldsymbol{u}_i$ and its opposite one $-\boldsymbol{u}_i$ for describing the distribution in Figure 1&B. The visualization in Figure B further supports our findings discussed in Sec. 2.1.

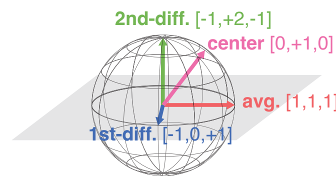


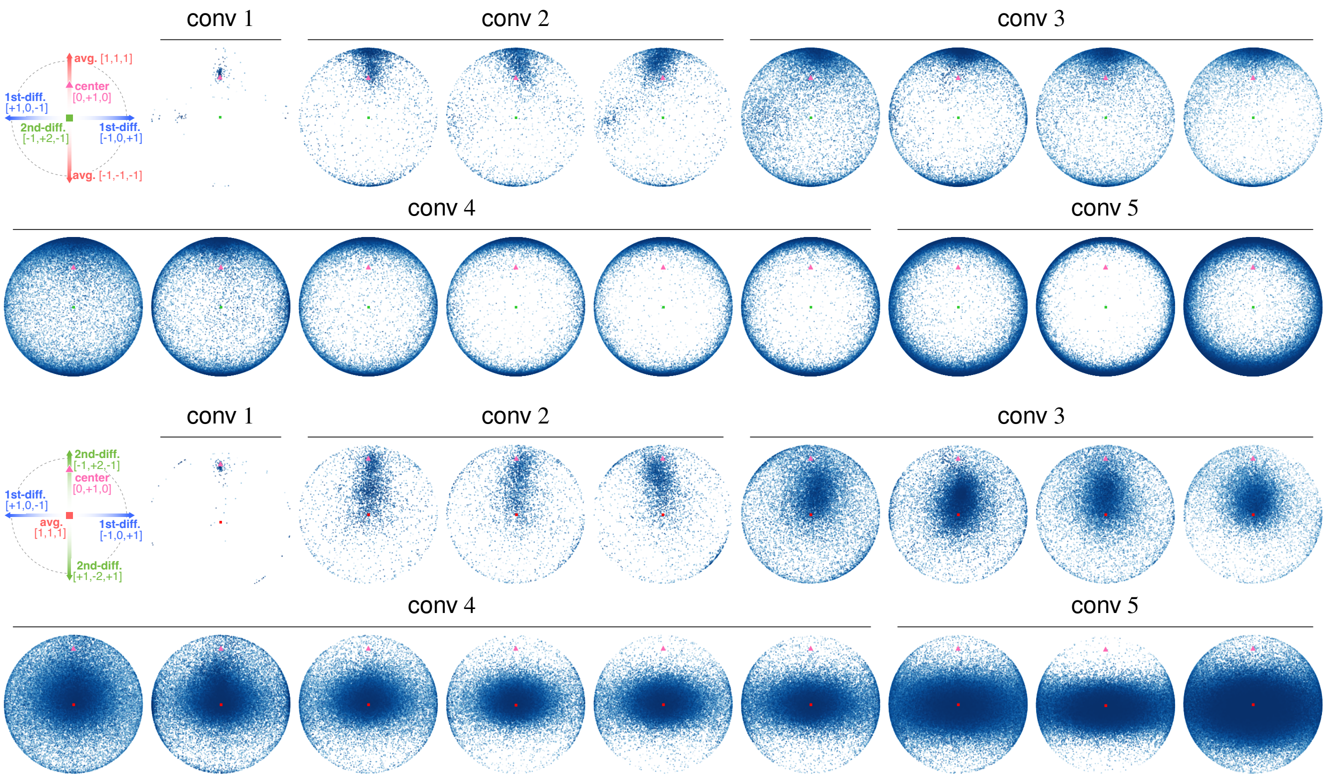
Figure A. Cartesian coordinates of physically interpretable axes.

B. Detailed procedure to train models

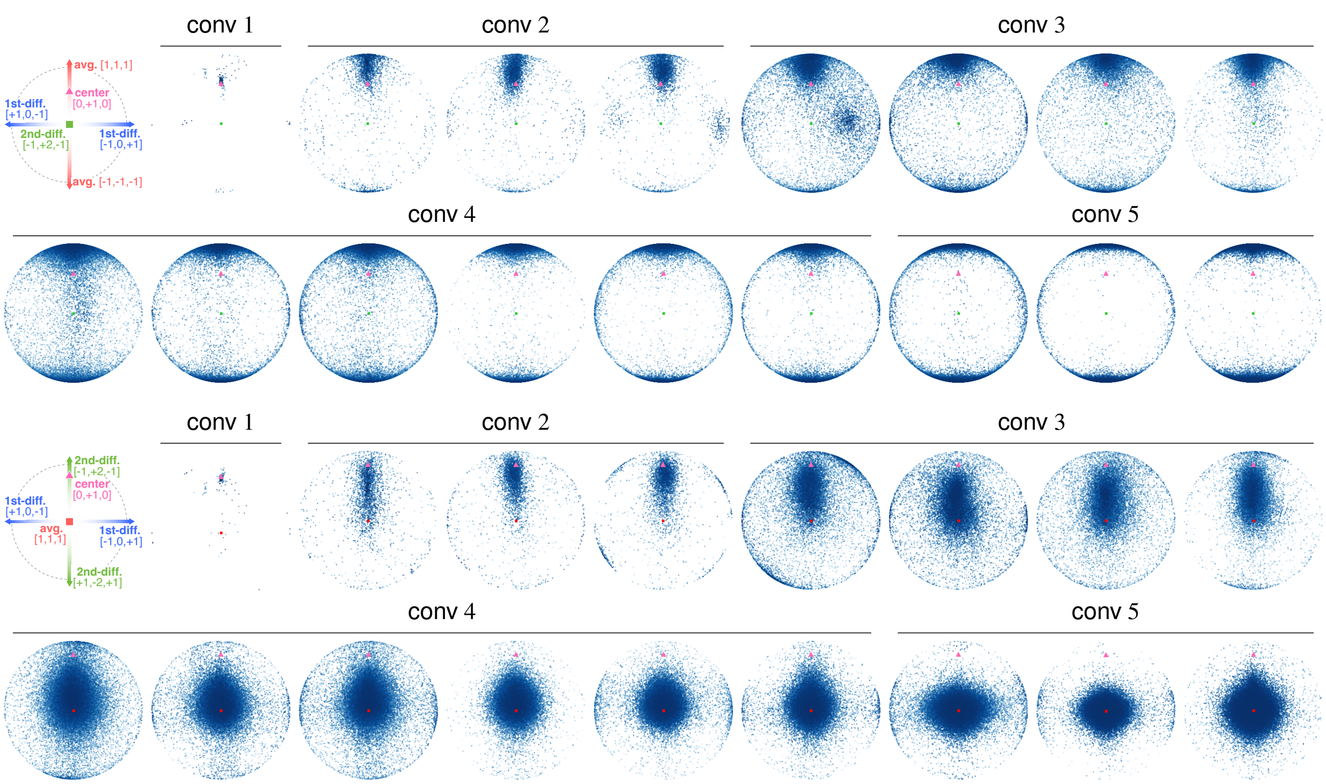
We detail the training and evaluation procedure on respective datasets in Table A. To train 3D-CNNs, we apply SGD optimizer with momentum of 0.9, weight decay of 0.0005 and the other parameters shown in Table A to video sub-clips of $32 \times 224 \times 224$ sampled by random cropping in a spatio-temporal domain. For evaluation, we extract several clips from an input video sequence at fixed positions. Video sub-clips of $32 \times 256 \times 256$ are uniformly sampled in the spatio-temporal domain with the numbers of clips shown in Table A to cover whole a video volume. The classification scores are summed up across those clips to produce the final classification.

Table A. Details of training procedures on respective datasets.

Dataset	SSv2 [1]	Mini-SSv2	Kinetics-400 (K400) [2]	Mini-K400
Training samples	168,913	81,663	241,181	121,802
Test samples	24,777	11,799	19,877	9,934
Classes	174	87	400	200
batch size	32	24	32	32
initial learning rate	0.01	0.01	0.01	0.01
learning rate schedule	cosine decay	$\times 0.1$ at $15, 30$-epochs	cosine decay	$\times 0.1$ at $15, 30$-epochs
training epochs	100	35	100	45
Evaluation clips	$3_{(spatial)} \times 3_{(temporal)} = 9$ clips		$3_{(spatial)} \times 10_{(temporal)} = 30$ clips	



(a) Pretrained on SSv2 dataset: on a plane of average and 1st differential (top), and of 1st and 2nd differential filters (bottom).



(b) Pretrained on K-400 dataset: on a plane of average and 1st differential (top), and on 1st and 2nd differential filters (bottom)

Figure B. Distributions of the primary temporal filters embedded in I3D-ResNet-50 which is pretrained on (a) SSv2 [1] and (b) K-400 [2] datasets. The temporal filters are normalized in unit L_2 norm to distribute on a *sphere* (Figure A).

C. Effective receptive field

Following [3], we measure the effective receptive field of a 3D-CNN as follows.

1. Randomly draw an input volume by $\boldsymbol{I} = \{I_{cthw} \sim \mathcal{N}(0,1)\}_{c,t,h,w}^{3,32,224,224} \in \mathbb{R}^{3\times 32\times 224\times 224}$.

2. Inject gradients to the center neuron on the last feature map. Let $\boldsymbol{X} \in \mathbb{R}^{2048\times 32\times 7\times 7}$ be the last feature map produced by I3D-ResNet-50 and $\boldsymbol{W} \in \mathbb{R}^{2048\times 32\times 7\times 7}$ be a binary map which activates at the center neuron; $W_{cthw} = 1$ for $(t,h,w) = (17,4,4)$, $\forall c$ and $W_{cthw} = 0$ for the others. Thereby, we can design a loss of $\ell = \langle \boldsymbol{W}, \boldsymbol{X} \rangle$, the element-wise multiplication and summation, i.e., inner-product between tensors.

3. The gradients $\boldsymbol{G_I}$ on an input $\boldsymbol{I}$ are computed through back-propagation based on the loss ℓ.

4. Repeat the above three steps 10 times and average the gradient values to measure the effective receptive field $\bar{\boldsymbol{G}} = [\mathrm{E}_{\boldsymbol{I}}(\boldsymbol{G}_{\boldsymbol{I}}^2)]^{\frac{1}{2}}$ where square and square-root operates on tensors in an element-wise manner.

References

[1] Raghav Goyal, Samira Ebrahimi Kahou, Vincent Michalski, Joanna Materzyńska, Susanne Westphal, Heuna Kim, Valentin Haenel, Ingo Fruend, Peter Yianilos, Moritz Mueller-Freitag, Florian Hoppe, Christian Thurau, Ingo Bax, and Roland Memisevic. The 'something something' video database for learning and evaluating visual common sense. In *ICCV*, pages 5843–5851, 2017. 1, 2

[2] Will Kay, Joao Carreira, Karen Simonyan, Brian Zhang, Chloe Hillier, Sudheendra Vijayanarasimhan, Fabio Viola, Tim Green, Trevor Back, Paul Natsev, Mustafa Suleyman, and Andrew Zisserman. The kinetics human action video dataset. *arXiv,* 1705.06950, 2017. 1, 2

[3] Wenjie Luo, Yujia Li, Raquel Urtasun, and Richard Zemel. Understanding the effective receptive field in deep convolutional neural networks. In *NeurIPS*, pages 9446–9454, 2016. 3