Supplementary Materials

The supplementary material provides intuitive explanations of our approach (Section A), network dissection results to understand the change in feature redundancy/expressiveness (Section B), deep metric learning performance to understand the generalizability (Section C), proof of Lemma 1 (Section D) and visualizations of filter similarities (Section E).

A. Intuitive Explanations of our Approach

We analyze a convolution layer which transforms input X to output Y with a learnable kernel K: $Y = \text{Conv}(K, X)$ in CNNs. Writing in linear matrix-vector multiplication form $Y = \mathcal{K}X$ (Fig.2(b) of the paper), we simplify the analysis from the perspective of linear systems. We do not use *im2col* form $Y = K\widetilde{X}$ (Fig.2(a) of the paper) as there is an additional structured linear transform from X to $\tilde{X}$ and this additional linear transform makes the analysis indirect. As we mentioned earlier, the kernel orthogonality does not lead to a uniform spectrum.

The spectrum of $\mathcal{K}$ reflects the scaling property of the corresponding convolutional layer: different input X (such as cat, dog, and house images) would scale up by $\eta = \frac{\|Y\|}{\|X\|}$. The scaling factor η also reflects the gradient scaling. Typical CNNs have highly non-uniform convolution spectrum (Fig.1(b) of the paper): for some inputs, it scales up to 2; for others, it scales by 0.1. For a deep network, these irregular spectrums add up and can potentially lead to gradient exploding and vanishing issues.

Features learned by CNNs are also more redundant due to the non-uniform spectrum issues (Fig.1(a) of the paper). This comes from the diverse learning ability to different images and leads to feature redundancy. A uniform spectrum distribution could alleviate the problem.

To alleviate the problem, we propose to make convolution orthogonal by enforcing $\mathcal{K}$ orthogonal. Orthogonal convolution regularizer in CNNs (OCNNs) leads to uniform $\mathcal{K}$ spectrum as expected. It further reduces the feature redundancy and improves the performance (Fig.1(b)(c)(d) of the paper).

Besides classification performance improvements, we also observe improved visual features, both in high-level (image retrieval) and low-level (image inpainting) tasks. Our OCNNs also generates realistic images (Section 4.6) and is more robust to adversarial attacks (Section 4.7).

B. Network Dissection

We demonstrate in Section 4 that our orthogonal convolutions reduce the feature redundancy by decorrelating different feature channels and enhancing the feature expressiveness with improved performance in image retrieval, inpainting and generation. Network dissection [6] is utilized to further evaluate the feature expressiveness across different channels.

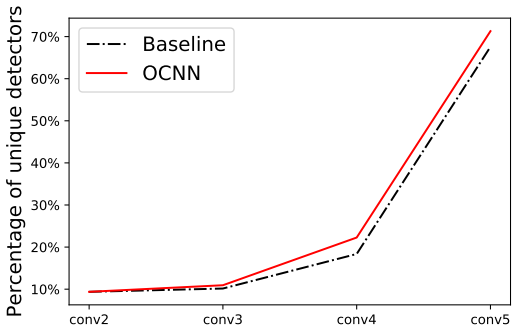

Figure 10. Percentage of unique detectors (mIoU $\geq$ 0.04) over different layers. Our OCNN has more unique detectors compared to baseline ResNet34 [24] at each layer.

Network dissection [6] is a framework that quantifies the interpretability of latent representations of CNNs by evaluating the alignment between individual hidden units and a set of semantic concepts. Specifically, we evaluate the baseline and our OCNN with backbone architecture ResNet34 [24] trained on ImageNet. The models are evaluated on Broden [6] dataset, where each image was annotated with spatial regions of different concepts, including cat, dog, house, etc. The concepts are further grouped into 6 categories: scene, object, part, material, texture and color. Network dissection framework compares the mean intersection over union (mIoU) between network channel-wise activation of each layer and ground-truth annotations. The units/feature channels are considered as "effective" when mIoU$\geq$ 0.04. We denote them as "unique detectors".

Our OCNN (Table 10 and Fig.10) have more unique detectors over different layers of the network. Additionally, the distribution of 6 concept categories is more uniform for our OCNN (Fig.11). The results imply that orthogonal convolutions reduce feature redundancy and en chance the feature expressiveness.

Table 10. Number of units/feature channels with mIoU $\geq$ 0.04 comparisons on ImageNet ILSVRC [14].

	conv2	conv3	conv4	conv5
ResNet34 [24]	6	13	47	346
OCNN (ours)	6	14	57	365

C. Deep Metric Learning

We evaluate the generalizability and performance of our our orthogonal regularizer in deep metric learning tasks. Specifically, following the training/evaluation settings in [41], we perform retrieval and clustering on Cars196 dataset [34] and summarize the results in Table 11). We observe performance gain when orthogonal regularizer is added.

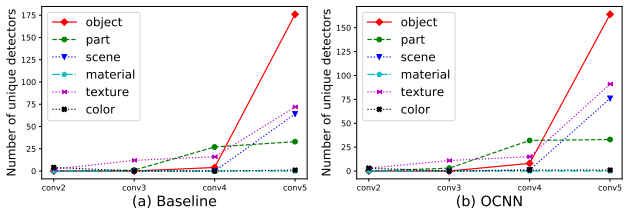

Figure 11. Distribution of concepts of unique detectors of different layers. Our OCNN has more uniform concept distribution compared to baseline ResNet34 [24].

Table 11. Retrieval/clustering performance on Cars196 (%).

	NMI	F1	Recall@1	@2	@4	@8
Triplet loss [28]	61.9	27.1	61.4	73.5	83.1	89.9
ProxyNCA [41]	62.4	29.2	67.9	78.2	85.6	90.6
[41]+Kernel orth	63.1	29.6	67.6	78.4	86.2	91.2
[41]+OCNN	**63.6**	**30.2**	**68.8**	**79.0**	**87.4**	**92.0**

D. Proof of the Orthogonality Equivalence

Here we provide a proof for the lemma 1: The row orthogonality and column orthogonality are equivalent in the MSE sense, i.e. $\|\mathcal{K}\mathcal{K}^T - I\|_F^2 = \|\mathcal{K}^T\mathcal{K} - I'\|_F^2 + U$, where U is a constant. A simple motivation for this proof is that when $\mathcal{K}$ is a square matrix, then $\mathcal{K}\mathcal{K}^T = I \iff \mathcal{K}^T\mathcal{K} = I'$. So we can hope to generalize this result and provide a more convenient algorithm. The following short proof is provided in the supplementary material of [37]. We would like to present it here for the reader's convenience.

Proof. It's sufficient to prove the general result, where we choose $\mathcal{K} \in \mathbf{R}^{M\times N}$ to be an arbitrary matrix[2]. We denote $\|\mathcal{K}\mathcal{K}^T - I_M\|^2$ as L_r and $\|\mathcal{K}^T\mathcal{K} - I_N\|^2$ as L_c.

$$
\begin{aligned}
L_r &= \|\mathcal{K}\mathcal{K}^T - I_M\|_F^2 \\
&= tr\left[(\mathcal{K}\mathcal{K}^T - I_M)^T(\mathcal{K}\mathcal{K}^T - I_M)\right] \\
&= tr(\mathcal{K}\mathcal{K}^T\mathcal{K}\mathcal{K}^T) - 2tr(\mathcal{K}\mathcal{K}^T) + tr(I_M) \\
&= tr(\mathcal{K}\mathcal{K}\mathcal{K}^T\mathcal{K}) - 2tr(\mathcal{K}^T\mathcal{K}) + tr(I_N) + M - N \\
&= tr\left[\mathcal{K}^T\mathcal{K}\mathcal{K}^T\mathcal{K} - 2\mathcal{K}^T\mathcal{K} + I_N\right] + M - N \\
&= tr\left[(\mathcal{K}^T\mathcal{K} - I_N)(\mathcal{K}^T\mathcal{K} - I_N)\right] + M - N \\
&= \|\mathcal{K}^T\mathcal{K} - I_N\|_F^2 + M - N \\
&= L_c + U
\end{aligned}
$$

where $U = M - N$.

□

E. Filter Similarity visualizations

As shown in Fig.1, filter similarity increases with depth of the network. We visualize the guided back-propagation patterns to understand this phenomenon.

[2]Here M and N are just some constant, different from the the ones used in the main text.

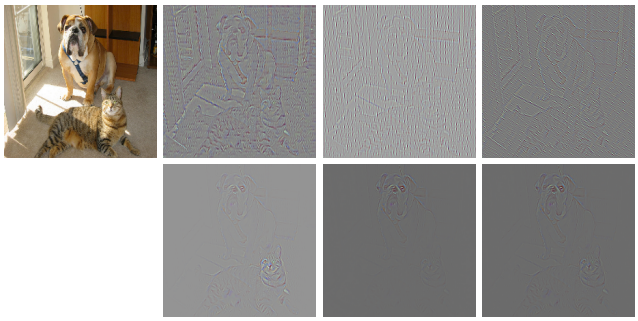

Figure 12. Guided back-propagation patterns of the input image (first column) with a ResNet34 model. The first row depicts patterns of the first 3 channels from layer 7, while the second row depicts patterns of the first 3 channels from layer 33. The filter similarity increases with depth.

For the ResNet34 trained on ImageNet, we plot guided back-propagation patterns of an image in Fig.12. The first row depicts patterns of the first 3 channels from layer 7, while the second row depicts patterns of the first 3 channels from layer 33. Patterns of different channels from earlier layers are more diverse, while patterns of different channels from later layers usually focus on certain regions. The filter similarity increases with depth.