# WHY CONVOLUTIONAL NETWORKS LEARN ORIENTED BANDPASS FILTERS: THEORY AND EMPIRICAL SUPPORT

**Anonymous authors**
Paper under double-blind review

## ABSTRACT

It has been repeatedly observed that convolutional architectures when applied to image understanding tasks learn oriented bandpass filters. A standard explanation of this result is that these filters reflect the structure of the images that they have been exposed to during training: Natural images typically are locally composed of oriented contours at various scales and oriented bandpass filters are matched to such structure. We offer an alternative explanation based not on the structure of images, but rather on the structure of convolutional architectures. In particular, complex exponentials are the eigenfunctions of convolution. These eigenfunctions are defined globally; however, convolutional architectures operate locally. To enforce locality, one can apply a windowing function to the eigenfunctions, which leads to oriented bandpass filters as the natural operators to be learned with convolutional architectures. From a representational point of view, these filters allow for a local systematic way to characterize and operate on an image or other signal. We offer empirical support for the hypothesis that convolutional networks learn such filters at all of their convolutional layers. While previous research has shown evidence of filters having oriented bandpass characteristics at early layers, ours appears to be the first study to document the predominance of such filter characteristics at all layers. Previous studies have missed this observation because they have concentrated on the cumulative compositional effects of filtering across layers, while we examine the filter characteristics that are present at each layer.

## 1 INTRODUCTION

### 1.1 MOTIVATION

Convolutional networks (ConvNets) in conjunction with deep learning have shown state-of-the-art performance in application to computer vision, ranging across both classification (e.g., Krizhevsky et al. (2012); Tran et al. (2015); Ge et al. (2019)) and regression (e.g., Szegedy et al. (2013); Eigen & Fergus (2015); Zhou et al. (2017)) tasks. However, understanding of how these systems achieve their remarkable results lags behind their performance. This state of affairs is unsatisfying not only from a scientific point of view, but also from an applications point of view. As these systems move beyond the lab into real-world applications better theoretical understanding can help establish performance bounds and increase confidence in deployment.

Visualization studies of filters that have been learned during training have been one of the key tools marshalled to lend insight into the internal representations maintained by ConvNets in application to computer vision, e.g., Zeiler & Fergus (2014); Yosinski et al. (2015); Mahendran & Vedaldi (2015); Shang et al. (2016); Feichtenhofer et al. (2018). Here, an interesting repeated observation is that early layers in the studied networks tend to learn oriented bandpass filters, both in two image spatial dimenstions, $(x, y)^\top$, in application to single image analysis as well as in three spatiotemporal dimensions, $(x, y, t)^\top$, in application to video. An example is shown in Fig. 1. Emergence of such filters seems reasonable, because local orientation captures the first-order correlation structure of the data, which provides a reasonable building block for inferring more complex structure (e.g., local measurements of oriented structure can be assembled into intersections to capture corner structure, etc.). Notably, however, more rigorous analyses of exactly why oriented bandpass filters might be

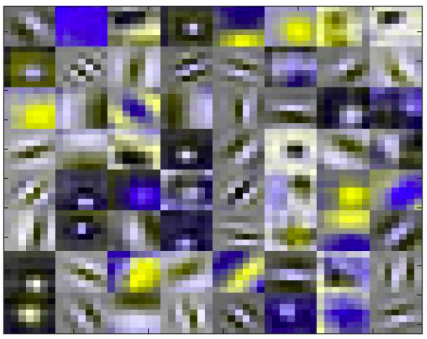

Figure 1: Visualization of pointspread functions (convolutional kernels) previously observed to be learned in the early layers of ConvNets. Brightness corresponds to pointwise function values. The majority of the plots show characteristics of oriented bandpass filters in two spatial dimensions, i.e., oscillating values along one direction, while remaining relatively constant in the orthogonal direction, even as there is an overall amplitude fall-off with distance from the center. The specific examples derive from the early layers of a ResNet-50 architecture He et al. (2016) trained on ImageNet Russakovsky et al. (2015).

learned has been limited. This state of affairs motivates the current paper in its argument that the analytic structure of ConvNets constrains them to learn oriented bandpass filters.

### 1.2 Related research

Visualization of receptive field profiles (i.e., pointspread functions Lim (1990)) of the convolutional filters learned by contemporary ConvNets is a popular tool for providing insight into the image properties that are being represented by a network. A notable trend across these studies is that early layers appear to learn oriented bandpass filters in both two spatial dimensions, e.g., Zeiler & Fergus (2014); Springenberg et al. (2015); Yosinski et al. (2015); Shang et al. (2016), as well as three spatiotemporal dimensions, e.g., Feichtenhofer et al. (2018). Indeed, earlier studies with architectures that also constrained their filters to be convolutional in nature, albeit using a Hebbian learning strategy MacKay (2003) rather than the currently dominant back-propagation approach Rumelhart et al. (1986), also yielded filters that visualized as having oriented bandpass filter characteristics Linsker (1986). Interestingly, biological vision systems also are known to show the presence of oriented bandpass filters at their earlier layers of processing in visual cortex; see Hubel & Wiesel (1962) for pioneering work along these lines and for more general review DeValois & DeValois (1988).

The presence of oriented bandpass filters in biological systems often has been attributed to their being well matched to the statistics of natural images Field (1987); Olshausen & Field (1996); Karklin & Lewicki (2009); Simoncelli & Olshausen (2001), e.g., the dominance of oriented contours at multiple scales. Similar arguments have been made regarding why such filters are learned by ConvNets. Significantly, however, studies have shown that even when trained with images comprised of random noise patterns, convolutional architectures still learn oriented bandpass filters Linsker (1986). These latter results suggest that the emergence of such filter tunings cannot be solely attributed to systems being driven to learn filters that were matched to their training data. Similarly, recent work showed that randomly initialized networks serve well in image restoration problems Ulyanov et al. (2018).

Some recent multilayer convolutional architectures have specified their earliest layers to have oriented bandpass characteristics, e.g., Bruna & Mallat (2013); Jacobsen et al. (2016); Hadji & Wildes (2017); indeed, some have specified such filters across all layers Bruna & Mallat (2013); Hadji & Wildes (2017). These design decisions have been variously motivated in terms of being well matched to primitive image structure Hadji & Wildes (2017) or providing useful building blocks for learning higher-order structures Jacobsen et al. (2016) and capturing invariances Bruna & Mallat (2013). Other work has noted that purely mathematical considerations show that ConvNets are well suited to realizing filter designs for capturing multiscale, windowed spectra Bruna et al. (2016); however, it did not explicitly established the relationship to eigenfunctions of convolution nor offer an explanation for why deep-learning yields oriented bandpass filters when applied to ConvNets. It also did not provide empirical investigation of exactly what filter characteristics are learned at each convolutional layer of ConvNets.

### 1.3 CONTRIBUTIONS

In the light of previous research, the present work appears to be the first to offer an explanation of why ConvNets learn oriented bandpass filters, independently of the input, by appeal to the inherent properties of their architectures in terms of their eigenfunctions. By definition, the convolutional layers of a ConvNet are governed by the properties of convolution. For present purposes, a key property is that the eigenfunctions of convolution are complex exponentials. Imposing locality on the eigenfunctions leads to oriented bandpass filters, which therefore are the appropriate filters to be learned by a ConvNet. Indeed, these theoretical considerations suggest that oriented bandpass filters should be learned at all layers of a ConvNet, not just at early layers. We provide empirical support for this observation by examining filters across all convolutional layers of three standard ConvNets (AlexNet Krizhevsky et al. (2012), ResNet He et al. (2016) and VGG16 Simonyan & Zisserman (2014)) and show that both numerically and visually they are well characterized as having learned oriented bandpass filters at all their convolutional layers. Our empirical study is distinct from earlier visualization efforts, which concentrate on the cumulative compositional results of filtering across layers that typically show emergence of complicated structures in the layerwise feature maps, while we focus on the complementary question of what primitive filter characteristics have been learned at each individual layer and offer both numerical as well as visualization analyses.

## 2 THEORY

This section details a novel explanation for why ConvNets learn oriented bandpass filters. The first two subsections largely review standard material regarding linear systems theory Oppenheim et al. (1983) and related topics Kaiser (2011); Kusse & Westwig (2006), but are necessary to motivate properly our explanation. The final subsection places the material in the context of ConvNets.

### 2.1 EIGENFUNCTIONS OF CONVOLUTION

Let $\mathcal{L}$ be a linear operator on a function space. The set of eigenfunctions $\phi_n$ associated with this operator satisfy the condition Kusse & Westwig (2006)

$$\mathcal{L}\phi_n = \lambda_n \phi_n. \tag{1}$$

That is, the operator acts on the eigenfunctions simply via multiplication with a constant, $\lambda_n$, referred to as the eigenvalue. It sometimes also is useful to introduce a (positive definite) weighting function, $w$, which leads to the corresponding constraint

$$\mathcal{L}\phi_n = \lambda_n w \phi_n. \tag{2}$$

For cases where any function in the space can be expanded as a linear sum of the eigenfunctions, it is said that the collection of eigenfunctions form a complete set. Such a set provides a convenient and canonical spanning representation.

Let $\mathbf{x} = (x_1, x_2, \ldots, x_n)^\top$, $\mathbf{a} = (a_1, a_2, \ldots, a_n)^\top$ and $\mathbf{u} = (u_1, u_2, \ldots, u_n)^\top$. For the space of convolutions, with the convolution of two functions, $f(\mathbf{x})$ and $h(\mathbf{x})$ defined as

$$f(\mathbf{x}) * h(\mathbf{x}) = \int_{-\infty}^{\infty} f(\mathbf{x} - \mathbf{a}) h(\mathbf{a})\, d\mathbf{a} \tag{3}$$

it is well known that functions of the form $f(\mathbf{x}) = e^{i\mathbf{u}^\top \mathbf{x}}$ are eigenfunctions of convolution Oppenheim et al. (1983), i.e.,

$$\int_{-\infty}^{\infty} e^{i\mathbf{u}^\top(\mathbf{x}-\mathbf{a})} h(\mathbf{a})\, d\mathbf{a} = e^{i\mathbf{u}^\top \mathbf{x}} \int_{-\infty}^{\infty} e^{-i\mathbf{u}^\top \mathbf{a}} h(\mathbf{a})\, d\mathbf{a} \tag{4}$$

with the equality achieved via appealing to $e^{i\mathbf{u}^\top(\mathbf{x}-\mathbf{a})} = e^{i\mathbf{u}^\top \mathbf{x}} e^{-i\mathbf{u}^\top \mathbf{a}}$ and subsequently factoring $e^{i\mathbf{u}^\top \mathbf{x}}$ outside the integral as it is independent of $\mathbf{a}$. The integral on the right hand side of (4),

$$\int_{-\infty}^{\infty} e^{-i\mathbf{u}^\top \mathbf{a}} h(\mathbf{a})\, d\mathbf{a}, \tag{5}$$

is the eigenvalue, referred to as the modulation transfer function (MTF) in signal processing Oppenheim et al. (1983). Noting that $e^{i\mathbf{u}^\top \mathbf{x}} = \cos(\mathbf{u}^\top \mathbf{x}) + i \sin(\mathbf{u}^\top \mathbf{x})$ leads to the standard interpretation of $\mathbf{u}$ in terms of frequency of the function (e.g., input signal).

Given the eigenfunctions of convolution are parameterized in terms of their frequencies, it is useful to appeal to the Fourier transform of function $f(\mathbf{x})$, where we use the form Horn (1986)

$$\mathcal{F}(\mathbf{u}) = \int_{-\infty}^{\infty} f(\mathbf{x}) e^{-i\mathbf{u}^\top \mathbf{x}} \, d\mathbf{x}, \tag{6}$$

because any convolution can be represented in terms of how it operates via simple multiplication of the eigenvalues, (5), with the eigenfunctions, $e^{i\mathbf{u}^\top \mathbf{x}}$, with $\mathbf{u}$ given by (6). Thus, this decomposition provides a canonical way to decompose $f(\mathbf{x})$ and explicate how a convolution operates on it.

### 2.2 Imposing locality

Understanding convolution purely in terms of its eigenfunctions and eigenvalues provides only a global representation of operations, as notions of signal locality, $\mathbf{x}$, are lost in the global transformation to the frequency domain, $\mathbf{u}$. This state of affairs often is unsatisfactory from a representational point of view because one wants to understand the structure of the signal (e.g., an image) on a more local basis (e.g., one wants to detect objects as well as their image coordinates). This limitation can be ameliorated by defining a windowed Fourier transform Kaiser (2011), as follows Jahne & Hausbecker (2000).

Let $w(\mathbf{x})$ be a windowing function that is positive valued, symmetric and monotonically decreasing from its center so as to provide greatest emphasis at its center. A Windowed Fourier Transform (WFT) of $f(\mathbf{x})$ can then be defined as

$$\mathcal{F}(\mathbf{u}_c, \mathbf{x}; w) = \int_{-\infty}^{\infty} f(\mathbf{a}) w(\mathbf{a} - \mathbf{x}) e^{-i\mathbf{u}_c^\top \mathbf{a}} \, d\mathbf{a}. \tag{7}$$

Making use of the symmetry constraint that we have enforced on the windowing function allows for $w(\mathbf{x}) = w(-\mathbf{x})$ so that the WFT, (7), can be rewritten as

$$\mathcal{F}(\mathbf{u}_c, \mathbf{x}; w) = \int_{-\infty}^{\infty} f(\mathbf{a}) w(\mathbf{x} - \mathbf{a}) e^{i\mathbf{u}_c^\top (\mathbf{x} - \mathbf{a})} e^{-i\mathbf{u}_c^\top \mathbf{x}} \, d\mathbf{a}, \tag{8}$$

which has the form of a convolution

$$f(\mathbf{x}) * \left( w(\mathbf{x}) e^{i\mathbf{u}_c^\top \mathbf{x}} \right) \tag{9}$$

with the inclusion of an additional phase component, $e^{-i\mathbf{u}_c^\top \mathbf{x}}$.

To provide additional insight into the impact the WFT convolution, (9), has on the function, $f(\mathbf{x})$, it is useful to examine the pointspread function, $w(\mathbf{x}) e^{i\mathbf{u}_c^\top \mathbf{x}}$, in the frequency domain by taking its Fourier transform (6), i.e., calculate its MTF. We have

$$\int_{-\infty}^{\infty} w(\mathbf{x}) e^{i\mathbf{u}_c^\top \mathbf{x}} e^{-i\mathbf{u}^\top \mathbf{x}} \, d\mathbf{x}, \tag{10}$$

which via grouping by coefficients of $\mathbf{x}$ becomes

$$\int_{-\infty}^{\infty} w(\mathbf{x}) e^{-i(\mathbf{u} - \mathbf{u}_c)^\top \mathbf{x}} \, d\mathbf{x}. \tag{11}$$

Examination of (11) reveals that it is exactly the Fourier transform of the window function, cf. (6), as shifted to the center frequencies, $\mathbf{u}_c$. Thus, operation of the WFT convolution, (9), on a function, $f(\mathbf{x})$, passes central frequency, $\mathbf{u}_c$, relatively unattenuated, while it suppresses those that are further away from the central frequency according to the shape of the window function, $w(\mathbf{x})$, i.e., it operates as a bandpass filter. Thus, convolution with a bank of such filters with varying central frequencies, $\mathbf{u}_c$, has exactly the desired result of providing localized measures of the frequency content of the function $f(\mathbf{x})$.

Returning to the pointspread function itself, $w(\mathbf{x}) e^{i\mathbf{u}_c^\top \mathbf{x}}$, and recalling that $e^{i\mathbf{u}^\top \mathbf{x}} = \cos(\mathbf{u}^\top \mathbf{x}) + i \sin(\mathbf{u}_c^\top \mathbf{x})$, it is seen that in the signal domain, $\mathbf{x}$, the filter will oscillate along the direction of $\mathbf{u}_c$ while remaining relatively constant in the orthogonal direction, even as there is an overall amplitude fall-off with distance from the center according to the shape of $w(\mathbf{x})$, i.e., we have an oriented bandpass filter.

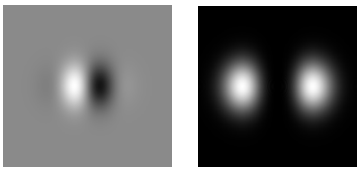

Figure 2: Visualization of an analytically defined oriented bandpass filter (12). The left panel shows the pointspread function corresponding to the odd symmetry ($\sin$) component, while the right panel shows its power in the frequency domain. Brightness corresponds to pointwise function values.

As a specific example Jahne & Hausbecker (2000), taking $w(\mathbf{x})$ to be an n-dimensional Gaussian-like function, $g(\mathbf{x};\sigma) = \kappa\, e^{-\|\mathbf{x}\|^2/\sigma^2}$, with $\sigma$ the standard deviation and $\kappa$ a scaling factor, yields an n-dimensional Gabor-like filter,

$$g(\mathbf{x};\sigma)e^{i\mathbf{u}_c^\top \mathbf{x}} = g(\mathbf{x};\sigma)\left(\cos(\mathbf{u}_c^\top \mathbf{x}) + i\sin(\mathbf{u}_c^\top \mathbf{x})\right), \tag{12}$$

which provides good joint localization of signal content in the signal and frequency domains Gabor (1946). Indeed, visualization of these filters in two spatial dimensions (Figure 2) provides strikingly similar appearance to those presented in Figure 1, if in an idealized form. In particular, their pointspread functions oscillate according to a frequency $\|\mathbf{u}_c\|$ along the direction, $\frac{\mathbf{u}_c}{\|\mathbf{u}_c\|}$, while remaining relatively constant in the orthogonal direction, even as there is an overall amplitude fall-off with distance from the center. In the frequency domain, they have peak power at $\mathbf{u}_c$ with a fall-off following a Gaussian-like shape with standard deviation, $1/\sigma$, that is the inverse of that used in specifying the window, $w(\mathbf{x})$. These observations hold because we already have seen, (11), that the frequency domain representation of such a function is the Fourier transform of the window function, $w(\mathbf{x})$, shifted to the center frequencies, $\mathbf{u}_c$; furthermore, the Fourier transform of a function of the form $g(\mathbf{x};\sigma)$ has a similar form, albeit with an inverse standard deviation Bracewell (1986).

### 2.3 Implications for ConvNets

Convolutions in ConvNets serve to filter the input signal to highlight its features according to the learned pointspread functions (convolutional kernels). Thus, convolution with the oriented filters shown in Figure 1 will serve to highlight aspects of an image that are correspondingly oriented and at corresponding scales. The question at hand is, "Why did the ConvNet learn such filters?" The previous parts of this section have reviewed the fact that complex exponentials of the form $e^{i\mathbf{u}^\top \mathbf{x}} = \cos(\mathbf{u}^\top \mathbf{x}) + i\sin(\mathbf{u}^\top \mathbf{x})$ are the eigenfunctions of convolution. Thus, such frequency dependent functions similarly serve as the eigenfunctions of the convolutional operations in ConvNets. In particular, this result is a basic property of the convolutional nature of the architecture, independent of the input to the system. Thus, for any convolution in a ConvNet the frequency dependent eigenfunctions, $e^{i\mathbf{u}^\top \mathbf{x}}$, provide a systematic way to represent their input.

As with the general discussion of locality presented in Subsection 2.2, for the specifics of ConvNets it also is of interest to be able to characterize and operate locally on a signal. At the level of convolution, such processing is realized via pointspread functions that operate as bandpass filters, (9). Like any practical system, ConvNets will not capture a continuous range of bandpass characteristics, as given by $\mathbf{u}_c$ and the sampling will be limited by the number of filters the designer allows at each layer, i.e., as a metaparameter of the system. Nevertheless, making use of these filters provides a systematic approach to representing the input signal.

Overall, the very convolutional nature of ConvNets inherently constrains and even defines the filters that they learn, independent of their input or training. In particular, learning bandpass filters provides a canonical way to represent and operate on their input, as these serve as the localized eigenfunctions of convolution. As a ConvNet is exposed to more and more training data, its representation is optimized by spanning as much of the data as it can. Within the realm of convolution, in which ConvNet conv layers are defined, oriented bandpass filters provide the solution. They arise as the locality constrained eigenfunctions of convolution and thereby have potential to provide a span of any input signal in a localized manner. Thus, ConvNets are optimized by learning exactly such filters. Notably, since this explanation for why ConvNets learn oriented bandpass filters is independent of training data, it can explain why such filters emerge even when the training data lacks such pattern

structure, including training on random input signals, e.g., Linsker (1986). Moreover, the explanation is independent of the learning algorithm as any algorithm driving its learned representation to span the space of input signals achieves its goal in the eigenfunctions of the convolutional architecture, i.e., oriented bandpass filters. Sec. 1.2 reviewed work showing that both back propagation and Hebbian learning yield oriented bandpass filters as their learned convolutional representations.

Our analysis of oriented bandpass filters as the localized eigenfunctions of convolution is not specific to early ConvNet layers, but rather applies to any convolutional layer. The result thereby makes a theory-based prediction that oriented bandpass filters should be learned at all conv layers in a ConvNet. As reviewed in Sec. 1.2, previous studies have demonstrated that filters learned at early ConvNet layers visualize as oriented bandpass filters; however, it appears that little attention in previous studies has examined the pointspread functions of learned convolutional filters deeper in a network. Instead, studies of learned filters deep in a ConvNet have focused on the cumulative effect of filtering across all layers upto and including a particular layer under consideration. In the following, we present a complementary study that empirically examines filters that have been learned at a given layer without reference to previous layers to see whether they appear as oriented bandpass filters.

## 3 EMPIRICAL SUPPORT

In this section, we present empirical support for the theory-based prediction that oriented bandpass filters should be learned at all layers in a ConvNet. We examine three standard ConvNets (AlexNet Krizhevsky et al. (2012), ResNet50 He et al. (2016) and VGG16 Simonyan & Zisserman (2014)) from both numerical and visualization perspectives. In all cases, we make use of publicly available implementations of the architectures AlexNet; ResNet; VGG16.

### 3.1 NUMERICAL STUDIES

To study numerically whether ConvNets learn oriented bandpass filters, we perform a least-squares fit of the derived oriented bandpass filter, (12), to all learned convolutional filters at all layers of each model. While other oriented bandpass filter models can be used here, the model considered, (12), is a natural choice in the present context, as it results from our theoretical analysis in Sec. 2. In particular, we fit the free parameters of a 2D instantiation of the model, (i.e., the center frequencies, $\mathbf{u}_c$, and the standard deviation, $\sigma$), to the learned pointspread values of every convolutional filter at every layer of AlexNet Krizhevsky et al. (2012), ResNet50 He et al. (2016) and VGG16 Simonyan & Zisserman (2014). Finally, we take the root-mean-square (RMS) error residual between each individual fit of the model and the corresponding learned pointspread function as indicative of how well the learned filter is captured by an oriented bandpass filter, with smaller error indicative of a better fit. Results are plotted in Fig. 3.

For all three architectures the histograms of residuals collapsed across layers shows that in all cases the fitting errors mostly lie below $0.1$, with the bulk of errors residing below $0.01$ and lower. These small residuals indicate generally good fits of the learned filters to the oriented bandpass model.

A more detailed look can be had by considering box plots of errors by layer. Here, the results for AlexNet, for example, show that the median fitting error is under $0.04$ at layer one and subsequently decreased to under $0.02$ for all subsequent layers as well as for the aggregated fit across all layers. Moreover, $95\%$ of the data lies under $0.15$ at layer 1 and under $0.04$ at all other layers as well as the aggregate across layers. To place these numbers in perspective, we perturbed the otherwise analytically defined pointspread function, (12), with various amounts of noise and then compared to the same function without corruption to see how much corruption would yield various RMS errors. Results are summarized in Fig. 4. For example, we find that random noise within only $\approx 6\%$ of the distribution of the uncorrupted filter values yields $0.04$ residual, which demonstrates that the discrepancy from the purely analytic form is very small, indicating that the observed fits are quite good. Still, as indicated by the raw histograms, Fig. 3, some outliers are observed, where the errors approach $0.1$ and beyond. Visual inspection of these cases indicates that they arise when the learning process apparently has failed, as the learned filter has an essentially constant valued pointspread function (i.e., is flat) or else has no discernible structure.

Results for ResNet50 and VGG16 show similar patterns to those of AlexNet, with the main difference being that the distributions are shifted to larger values at their first layers; however, they subsequently conform to values similar to those of AlexNet thereafter. Also, it is seen that the layer

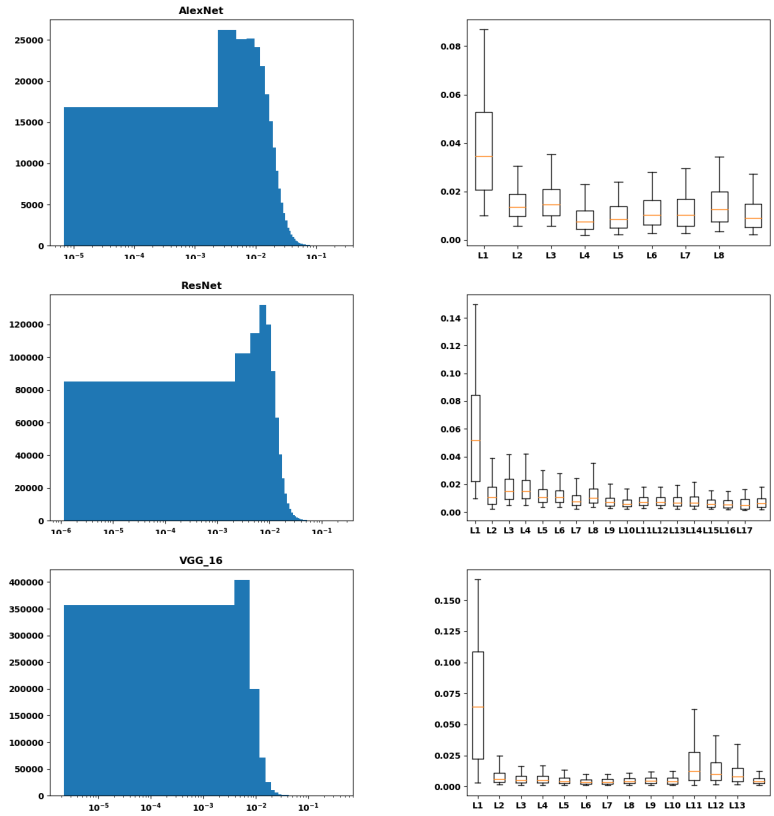


Figure 3: RMS errors from fitting an oriented bandpass model to convolutional filters learned by AlexNet Krizhevsky et al. (2012) (top row), ResNet50 He et al. (2016) (middle row) and VGG16 Simonyan & Zisserman (2014) (bottom row). Left column shows histograms of residuals collapsed across all layers for each architecture: Residual values and count along the abscissa and ordinate, resp. Right column shows box plots Tukey (1977) of residuals by layer along abscissa, with the last plot for each architecture showing results across all layers: The box in each column encompasses the observed interquartile range, the horizontal line in each box indicates the median value, the whiskers above and below the box extend to the $95^{th}$ and $5^{th}$ percentiles, resp.

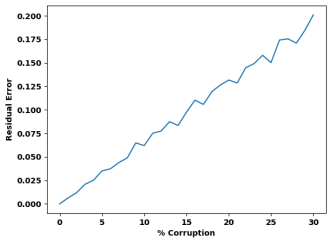


Figure 4: RMS error versus amount of corruption added to an analytically defined oriented bandpass filter to yield that error compared to the uncorrupted filter.

one distribution for VGG16 is shifted upward relative to ResNet50. Perhaps this pattern arises because the layer one filter sizes are largest for AlexNet and smallest for VGG16, with ResNet50 lying in between, while at layer two and beyond VGG16 and ResNet50 have the same size filters and all three architectures have the same size filters at layer four and beyond. Interestingly, for all architectures it is seen that there is a marked decrease in the residual values between layers one and two. This result parallels observations made in earlier visualization studies that learned filters typically begin to show more strongly oriented structure after the first layer, e.g., Linsker (1986); Springenberg et al. (2015). Overall, these numerical experiments indicate that all three of the considered architectures learn oriented bandpass filters at all layers.

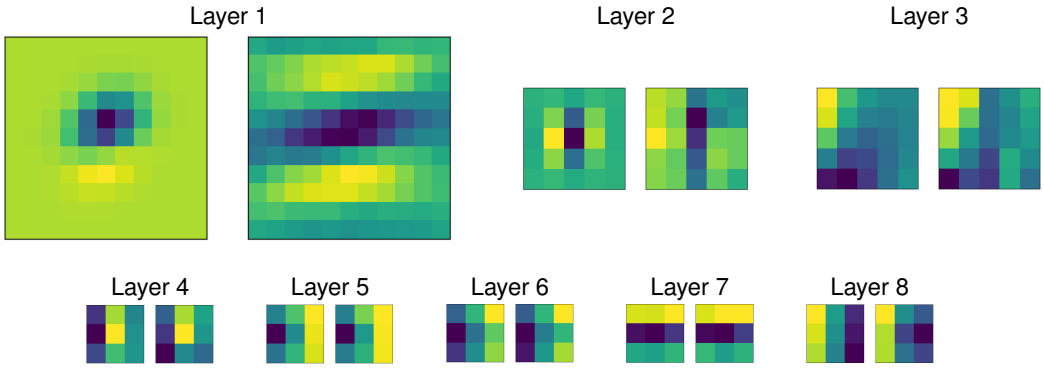


Figure 5: Visualization of learned pointspread functions for AlexNet Krizhevsky et al. (2012). Shown are visualizations of representative functions that have median residual values at each layer. For each layer, the plot on the left shows the learned function and the plot on the right shows the correspondingly fit function. Brightness corresponds to pointwise function values.

### 3.2 Visualization studies

For visualization studies, we plot the pointspread functions for a selection of the learned filters at each layer and display them as images. In the interest of space, here we focus on AlexNet; although, the visualization results for the other architectures similarly are supportive of oriented bandpass filters, as would be expected from the numerical results of Sec. 3.1. In particular, Fig. 5 shows plots of learned pointspread functions from all layers of AlexNet. The shown pointspread functions are representative of the median residual value for each layer, and they are paired with a visualization of the corresponding fit to the oriented bandpass model, (12). Inspection of these plots show that in all cases oriented structure is visible in the learned pointspread functions: In particular, proceeding from layers 1 to 8 orientations are manifest approximately along slight diagonal upper right to lower left, vertical, diagonal upper right to lower left, vertical, vertical, diagonal upper left to lower right, horizontal and vertical, resp. Moreover, the learned and fit pointspread functions are qualitatively very similar. Moreover, the visualizations suggest improved fits between the learned and fit models as layer increases, similar to what is seen in the numerical results of Sec. 3.1. Overall, these visualization results corroborate the numerical results indicating that oriented bandpass filters are indeed learned at all layers of the considered ConvNets.

## 4 Summary

Previous studies have demonstrated that learned filters at the early layers of convolutional networks visualize as oriented bandpass filters. This phenomenon typically is explained via appeal to natural image statistics, i.e., natural images are dominated by oriented contours manifest across a variety of scales and oriented bandpass filters are well matched to such structure. We have offered an alternative explanation in terms of the structure of convolutional networks themselves. Given that their convolutional layers necessarily operate within the space of convolutions, learning oriented bandpass filters provides the system with the potential to span possible input, even while preserving a notion of locality in the signal domain. Notably, our work is applicable to not just early ConvNet layers, but to all conv layers in such networks. We have provided empirical support for this claim, showing that oriented bandpass filters are indeed learned at all layers of three standard ConvNets. These results not only provide new insights into the operations and representations learned by ConvNets, but also suggest interesting future research. In particular, our work motivates investigation of novel architectures that explicitly constrain their convolutional filters to be oriented bandpass in a learning-based framework. In such a framework, it would not be necessary for the training process to learn the numerical values for each and every individual filter value (i.e., each filter tap), but rather would merely need to learn a much smaller number of parameters, e.g., the values of the center frequency, $\mathbf{u}_c$, and the standard deviation, $\sigma$, associated with the Gabor-like filter derived above, (12), or some other suitably parameterized filter. Such a constrained learning approach would require a much less intensive training procedure (e.g. involving far less data) compared to learning values for all individual filter taps, owing to the drastically reduced number of parameters that need to be estimated, even while being able to tune to the specifics of the task that is being optimized.

## REFERENCES

AlexNet. https://github.com/BVLC/caffe/tree/master/models/bvlc_alexnet.

R. Bracewell. *The Fourier Transform and Its Applications*. McGraw-Hill, NY, NY, 1986.

J. Bruna and S. Mallat. Invariant scattering convolution networks. *IEEE Trans. Pattern Anal. Mach. Intell.*, 35:1872–1886, 2013.

J. Bruna, S. Chintala, Y. LeCun, S. Piantino, A. Szlam, and M. Tygert. A mathematical motivation for complex-valued convolutional networks. *Neural Computation*, 28:815–825, 2016.

R. DeValois and K. DeValois. *Spatial Vision*. Oxford University Press, NY, NY, 1988.

D. Eigen and R. Fergus. Predicting depth, surface normals and sematic labels with a common muli-scale convolutional architecture. In *ICCV*, 2015.

C. Feichtenhofer, A. Pinz, R. Wildes, and A. Zisserman. What have we learned from deep representations for action recognition? In *CVPR*, 2018.

D. Field. Relations between the statistics of natural images and the response properties of cortical cells. *Journal of the Optical Society of America A*, 4:2379, 1987.

D. Gabor. Theory of communication. *Journal of the Institute of Electrical Engineers*, 93:429–457, 1946.

W. Ge, X. Lin, and Y. Yu. Weakly supervised complementary parts models for fine-grained image classification from the bottom up. In *CVPR*, 2019.

I. Hadji and R. Wildes. A spatiotemporal oriented energy network for dynamic texture recognition. In *ICCV*, 2017.

K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *CVPR*, 2016.

B. Horn. *Robot Vision*. MIT Press, Cambridge, MA, 1986.

D. H. Hubel and T. N. Wiesel. Receptive fields, binocular interaction and functional architecture in the cat's visual cortex. *The Journal of Physiology*, 160:106–154, 1962.

J H. Jacobsen, J V. Gemert, Z. Lou, and A W.M. Smeulders. Structured receptive fields in cnns. In *CVPR*, 2016.

B. Jahne and H. Hausbecker. *Computer Vision and Applications*. Academic Press, San Diego, CA, 2000.

G. Kaiser. *A Friendly Guide to Wavelets*. Modern Birkhauser Classics, Switzerland, 2011.

Y. Karklin and M. Lewicki. Emergence of complex cell properties by learning to generalize in natural scenes. *Nature*, 457:83–86, 2009.

A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. In *NIPS*, 2012.

B. Kusse and W. Westwig. *Mathematical Physics*. Wiley-VCH, Weinheim, FRG, 2006.

J. Lim. *Two-Dimensional Signal and Image Processing*. Prentice Hall, Upper Saddle River, NJ, 1990.

R. Linsker. From basic network principles to neural architecture. *Proc. National Academy of Sciences USA*, 83:7508–7512, 8390–8394, 8779,8783, 1986.

D. MacKay. *Information Theory, Inference, and Learning Algorithms*. Cambridge University Press, Cambridge, UK, 2003.

A. Mahendran and A. Vedaldi. Understanding deep image representations by inverting them. In *CVPR*, 2015.

B. Olshausen and D. Field. Emergence of simple-cell field properties by learning a sparse code for natural images. *Nature*, 381:607–609, 1996.

A. Oppenheim, A. Willsky, and I. Young. *Signals and Systems*. Prentice Hall, Upper Saddle River, NJ, 1983.

ResNet. `https://github.com/fchollet/deep-learning-models/releases/tag/v0.1`.

D. Rumelhart, G. Hinton, and R. Williams. Learning representations by back-propagating errors. *Nature*, 323:533–536, 1986.

O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A. Berg, and L. Fei-Fei. Imagenet large scale visual recognition challenge. *IJCV*, 2015.

W. Shang, K. Sohn, and H. Lee D. A. Enlitic. Understanding and improving convolutional neural networks via concatenated rectified linear units. In *ICML*, 2016.

E. Simoncelli and B. Olshausen. Natural image statistics and neural representation. *Annual Review of Neuroscience*, 24:1193–1216, 2001.

K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. In *ICLR*, 2014.

J. Springenberg, A. Dosovitskiy, T. Brox, and M. Riedmiller. Striving for simplicity: The all convolutional net. In *ICLR Workshop*, 2015.

C. Szegedy, A. Toshev, and D. Erhan. Deep neural networks for object detection. In *NIPS*, 2013.

D. Tran, L. Bourdev, R. Fergus, L. Torresani, and M. Paluri. Learning spatiotemporal features with 3d convolutional networks. In *ICCV*, 2015.

J. Tukey. *Exploratory Data Analysis*. Addison-Wesley, 1977.

D. Ulyanov, A. Vedaldi, and V. Lempitsky. Deep image prior. In *CVPR*, 2018.

VGG16. `https://github.com/fchollet/deep-learning-models/releases/tag/v0.1`.

J. Yosinski, J. Clune, A. Nguyen, T. Fuchs, and H. Lipson. Understanding neural networks through deep visualization. In *ICML workshops*, 2015.

M. Zeiler and R. Fergus. Visualizing and understanding convolutional networks. In *ECCV*, 2014.

T. Zhou, M. Brown, N. Snavely, and D. Lowe. Unsupervised learning of depth and ego-motion from video. In *CVPR*, 2017.