Prune and distill: similar reformatting of image information along rat visual cortex and deep neural networks

Prune and distill: similar reformatting of image information along rat visual cortex and deep neural networks

Paolo Muratore[1], **Sina Tafazoli**[1,2], **Eugenio Piasini**[1], **Alessandro Laio**[1,3], **Davide Zoccolan**[§,1]

[1]International School for Advanced Studies (SISSA), Trieste, Italy
[2]Princeton Neuroscience Institute, Princeton University, Princeton, NJ, United States of America
[3]Abdus Salam International Centre for Theoretical Physics (ICTP), Trieste, Italy

[§]To whom correspondence should be addressed: zoccolan@sissa.it

Abstract

Visual object recognition has been extensively studied in both neuroscience and computer vision. Recently, the most popular class of artificial systems for this task, deep convolutional neural networks (CNNs), has been shown to provide excellent models for its functional analogue in the brain, the ventral stream in visual cortex. This has prompted questions on what, if any, are the common principles underlying the reformatting of visual information as it flows through a CNN or the ventral stream. Here we consider some prominent statistical patterns that are known to exist in the internal representations of either CNNs or the visual cortex and look for them in the other system. We show that intrinsic dimensionality (ID) of object representations along the rat homologue of the ventral stream presents two distinct expansion-contraction phases, as previously shown for CNNs. Conversely, in CNNs, we show that training results in both distillation and active pruning (mirroring the increase in ID) of low- to middle-level image information in single units, as representations gain the ability to support invariant discrimination, in agreement with previous observations in rat visual cortex. Taken together, our findings suggest that CNNs and visual cortex share a similarly tight relationship between dimensionality expansion/reduction of object representations and reformatting of image information.

1 Introduction

Deep Convolutional Neural Networks (CNNs) currently stand as our best class of models of visual processing in the brain [1, 2, 3], showing success in: (1) predicting the tuning of individual neurons [4] and bold responses [5] at various stages of the ventral stream; (2) accounting for the ability of ventral stream neurons to encode a variety of object properties [6]; and (3) controlling their activity via synthetic stimuli inferred through model inversion [7, 8]. This suggests that the objective-optimization framework of deep learning offers a parsimonious explanation of the inner workings of complex, hierarchical brain circuits [9], although the latter are likely shaped by very different learning processes (e.g., unsupervised adaptation to the spatiotemporal statistics of the visual input [10, 11]. Despite this success, key differences between biological and artificial hierarchical networks exist (e.g., in sensitivity to noise or adversarial examples [12, 13]), possibly highlighting core dissimilarities in how information is processed in the two systems.

In this study, we investigated whether a similar reformatting of image information takes place along rat visual cortex and a representative CNN (VGG-16). We started from the observation that the

36th Conference on Neural Information Processing Systems (NeurIPS 2022).

3UXQH DQG GLVWLOO VLPLODU UHIRUPDWWLQJ RI LPDJH LQIRUPDWLRQ DORQ

2ULJLQDO
3UXQH DQG GLVWLOO VLPLODU UHIRUPDWWLQJ RI LPDJH LQIRUPDWLRQ DOR
QHWZRUNV 0XUDWRUH 3 7DID]ROL 6 3LDVLQL (/DLR $ =RFFRODQ '
SUHVHQWDWR DO FRQYHJQR WK &RQIHUHQFH RQ $GYDQFHV LQ 1HXUDO ,QI
WHQXWRVL D 1HZ 2UOHDQV /RXLVLDQD QHO 1RYHPEHU 'HFHPEHU

$YDLODELOLW\
7KLV YHUVLRQ LV DYDLODEOH DW VLQFH 7 =

3XEOLVKHU

3XEOLVKHG
'2,

7HUPV RI XVH

7HVWR GHILQLWR GDOOüDWHQHR UHODWLYR DOOH FODXVROH GL FRQFHVVLR

3XEOLVKHU FRS\ULJKW

$H(X) = -\sum_{x \in \mathcal{X}} p_X(x) \log p_X(x)$. The final estimate of the information conveyed by the units of a given layer about the feature metric was computed as $U^\ell(X|Y) = \mathbb{E}_i \left[I_i^\ell \left(X_i^\ell; Y_i^\ell \right) / H_i^\ell \left(X_i^\ell \right) \right]$, where $\mathbb{E}_i$ is the expected value over all units i of layer ℓ. Importantly, although such unit-averaging was performed on a sub-population of $\mathcal{O}\left(10^2\right)$ units, the variability of U^ℓ across independent experiment realizations (different units and stimuli) was very low, as shown by the error bars reported in Figures 3 and 5. The limited sampling bias for the mutual information was corrected with the the Panzeri-Treves method [22, 23]. Finally, we stress here how U^ℓ, being a single-unit information estimate, is not bound by the data processing inequality and can in general express non-monotonic behaviours as a function of layer depth ℓ.

2.3 Definition of the metrics to quantify visual features

In our analysis, each image patch $\mathsf{Image}_{\mathrm{RF}}$ falling within the RF of a unit was quantified by an array of four different visual properties of increasing complexity: 1) luminosity; 2) contrast; 3) orientation of the dominant edge (if any); and 4) orientations of the two dominant edges (if any), which is a proxy for the orientation and width of the dominant corner. Therefore $\mathsf{feat} \in \{\mathsf{luminosity}, \mathsf{contrast}, \mathsf{orientation}, \mathsf{corner}\}$.

Luminosity can be easily defined as the average pixel-intensity in the image path: $\mathsf{luminosity} = \mathsf{mean}\left(\mathsf{Image}_{\mathrm{RF}}\right)$. Contrast quantifies the amount of luminosity variation in the patch and was computed as: $\mathsf{contrast} = \mathsf{mean}\left(\mathsf{Sobel} * \mathsf{Image}_{\mathrm{RF}}\right)$, where Sobel denotes the Sobel kernel and $*$ is the convolution operator (the Sobel transform is a standard approach to compute image gradients [24]).

The dominant orientation of in an image patch is less straightforward to quantify, because of the large variation in RF size across the layers of the network and the complexity of the natural scenes in ImageNet. At very low resolution, such as for individual units in early layers in VGG-16 (which have 3×3 RF size), no meaningful orientation can be computed. For units in late layers, which process the entire scene, multiple prominent orientations might coexist or not exist at all. More generally, image patches span a spectrum of scene orientation strength, ranging from those containing one or more sharp edges to those featuring none. To deal with such variability, we developed a two-stage, compute-and-filter approach. The orientation estimation routine is based on Fourier Analysis and defines the dominant orientation of the patch $\theta^\star$ as the angle of highest power of its Fourier spectrum (see Algorithm 1 in the Supplementary Material for a detailed pseudo-code of the pipeline). In addition, the function provides an orientation strength index $\xi \in [0, 1]$, which peaks for images containing at least one very sharp edge. Before measuring orientation information, we ranked the pool of sampled units in each layer by computing, for each unit, the average of the orientation strength index ξ across the full set of 1500 input images. Out of the initial population of 250 units we only retained the 200 units with the largest average index. In addition, for each selected unit, we only considered the 500 images with the largest index ξ.

The corner feature was quantified as the pair of orientations of the two most prominent edges in the image patch. Specifically, the corner estimation routine applies Fourier analysis and peak-finding algorithms to identify the two dominant orientations $\theta_1^\star$ and $\theta_2^\star$ in a patch, along with a corner strength index $\zeta \in [0, 1]$ [17, 16], which is large when at least two orientations with similar power are detected in the Fourier spectrum, while it becomes negligible both for no-peak and single-peaked angular spectra (see Algorithm 2 in the Supplementary Materials for a pseudo-code description of the complete pipeline). We used this index following the same rationale as for the ξ index of orientation strength, this time ranking units and input images based on their corner strength ζ and retaining a population of 200 units, each tested with a sample of 400 images.

3 Results

3.1 Intrinsic dimension of object representations along the rat ventral stream

We applied the nonlinear ID estimator Two-NN [20] to compute the intrinsic dimension of object representations in four visual cortical areas (V1, LM, LI and LL) of the rat ventral stream, as a function of the number of units included in the population vector space (Figure 2A, solid lines), up to the maximal number of units available in each area (circles). In addition, we extracted the asymptotic values of the ID (stars in Figure 2B) via power-law fits (dashed lines) to control for

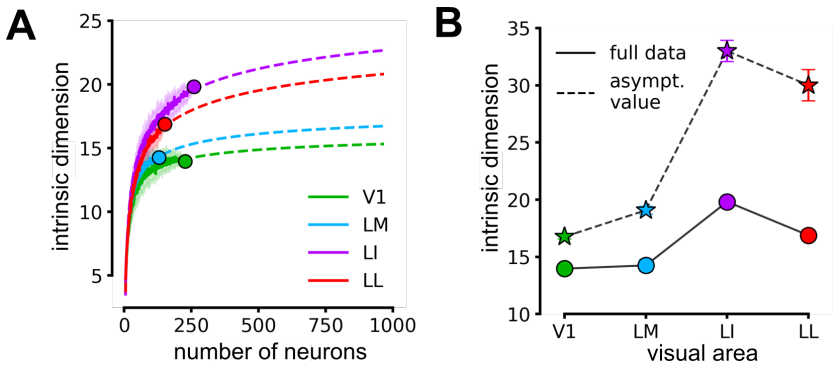

Figure 2: **Intrinsic Dimension of neural representations** (**A**) Estimation of the ID as a function of the number of neurons considered in the four visual areas (solid lines). Shadings correspond to the SD of multiple estimates with randomly sampled neuronal sub-populations, while circles mark the estimates obtained with the full populations in each area. Dashed lines are power-law fits to the data. (**B**) The ID estimates obtained for the full populations (circles) and for the asymptotic values of the fits (stars) are plotted as a function of the rank of the areas along the rat ventral stream. Error bars are the standards deviation of the values returned by the fit.

finite-size effects. At any population size considered, the ranking of the visual areas in terms of the estimated ID was remarkably stable, with V1 featuring the lowest ID, LI the highest, and LM and LL reaching intermediate values. More importantly, plotting the ID in each area as a function of its rank along the cortical processing hierarchy (Figure 2B) revealed a characteristic "hunchback" profile, with an initial expansion (from V1 to LI), followed by a contraction (from LI to LL). This trend is consistent with the one observed in deep networks (see Figure 1A) by [14], who conjectured that the initial ID expansion was due to the pruning of low-level image information (e.g., luminosity and contrast). Our result strongly supports this intuition, since the alternation of the expansion-contraction phases is now observed along an object processing pathway where such pruning has been shown to take place (see Figure 1B) [15].

3.2 Encoding of low- to middle-level visual features in single units of VGG-16

We now turn to the other question addressed in our study, namely investigating if the information about low-level image features is actively discarded in artificial networks in a manner that resembles the one observed in rats. Having defined a set of metrics to quantify image features of increasing complexity (see Section 2.3), we measured how much information about these features was encoded by the activation of individual units across the layers of VGG-16 (see Section 2.2 for details).

We found that information about luminosity was a monotonic decreasing function of the layer's depth, with training producing a very large luminosity information loss in the very first layer (compare blue and green curves in Figure 3A). Intuitively, this can be explained by the fact that learning spatially structured convolutional kernels will tend to produce both positive and negative weights with balanced, near-zero average, which are poorly sensitive to the mean luminosity within a unit's RF. By contrast, randomly assigned weights will often have the same sign, at least for the small kernels of the early layers, yielding activations that are proportional to the luminous energy falling within a unit's RF. This intuition was confirmed by comparing the distributions of the average weights for the 3×3 kernels of the first layer in the trained and untrained network (bar plot in the inset). The gradual monotonic decrease that was nevertheless observed in the untrained network is explained by the fact that randomly assigned weights, in case of increasingly larger kernels, will progressively tend to the zero-average condition (inset, red line).

If training produces spatially structured kernels, units in early layers should not only lose sensitivity to luminosity, but also become sensitive to image contrast. Our mutual information analysis confirmed this intuition, showing that the units of the initial layers encoded a larger amount of contrast information in the trained network, as compared to the untrained one (Figure 3B, blue vs. green curve). In addition, as a result of training, contrast information grew steadily in the early convolutional layers, reaching a peak in the third one, but then decayed sharply in the following layers, eventually

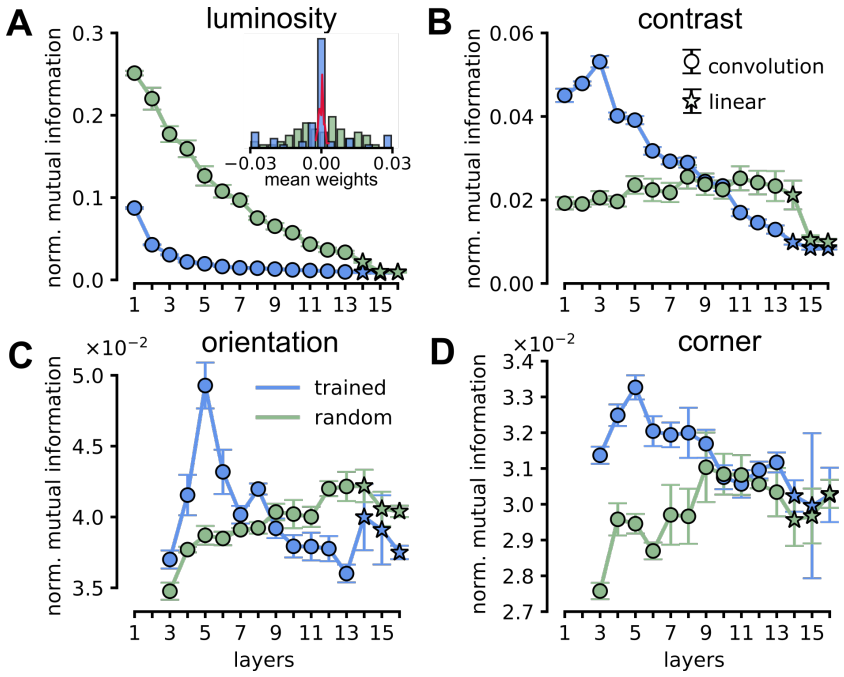

Figure 3: **Distillation and pruning of image information in VGG-16** (**A**) Mean normalized information conveyed by VGG-16 units about image luminosity for a trained (blue) and random (green) network. Error bars are standard deviations over five realizations of the experiment (independent sampling of units, images and random weights). Circles represent convolutional layers, while stars indicate fully connected layers. Inset: distribution of the average weights of the units in the first layer of the trained and random network (blue and green bars) and in the forth convolutional layer of the latter. (**B**-**D**) Same as in **A**, but for the information conveyed by VGG-16 units about image contrast, orientation and corners (i.e., joint orientation of two prominent edges).

attaining values that were lower than those of the untrained network. This suggests that learning representations that are useful to process and classify natural images requires to first distill contrast information in the units of early layers, followed by actively discarding such information in later processing stages. The pruning of contrast and luminosity information matches the results in rat visual cortex [15] (see Figure 1B). We note that the analysis of rat data did not reveal the initial rise of contrast information found in VGG-16, but this is unsurprising, given that the rat dataset did not contain recordings from the processing stages that precede V1, i.e., retina and thalamus, where center-surround contrast detectors first emerge in the visual system and that would correspond to VGG-16 very first layers.

We next considered visual features of increasing complexity, measuring the amount of information encoded by VGG-16 units about the dominant orientations of the image patches falling within their RFs. This analysis was applied only to units for which enough input images existed that contained, in the patch falling within the units' RFs, a sufficiently prominent oriented edge (see Section 2.3). Moreover, we excluded from the analysis the first two layers of the network, because their units have RFs that are too small for the orientation estimate to be meaningful. When visualized as a function of layer depth, orientation information in the trained network followed a hunchback profile (Figure 3C, blue curve), raising sharply and reaching a peak in the fifth convolutional layer, i.e., at a later stage than contrast information (see Figure 3B), consistently with the hierarchically higher nature of the orientation feature. Following the peak, orientation information dropped sharply in the deeper layers. As for luminosity and contrast, this trend was the result of training, as it was not observed in the randomly initialized network (green curve). And again, as for contrast, orientation information, once distilled in individual units of early layers, was then actively discarded in the following processing stages, becoming lower than for the untrained network - a finding consistent with the loss of orientation tuning found along the ventral stream [16] (see Figure 1D, solid lines). Just like for contrast, no initial rise of orientation tuning was observed along the rat ventral stream,

because no data were available from subcortical areas where orientation tuning is known to be much less prominent [25].

Finally, considering features of even greater complexity, we measured the information conveyed by VGG-16 units about the joint orientation of two dominant edges, i.e., the corner information (again, this analysis was applied only to cases where the image patch falling within a unit RF contained a sufficiently prominent corner; see Section 2.3). As for orientation, also corner information varied across the layers of the trained network according to a hunchback profile (Figure 3D, blue curve), again peaking in the fifth convolutional layer, and again being discarded in deeper processing stages, but more gradually than orientation information, reaching a sort of plateau in middle layers. Once more, this trend was not observed in the untrained network (green curve) and was instead consistent with the increase of neurons tuned for pairs of orientations found along the ventral stream [16] (see Figure 1D, dashed lines).

Importantly, all these feature information trends were largely preserved when assessed on the activations following the ReLU non-linearities (see Appendix C), and when tested on other networks of the VGG family (see Appendix D)

3.3 Effective pruning of low-level information requires training

One of the most intriguing findings of our experiments is that training is necessary not only to build sensitivity for low- and middle-level visual features, but also plays the complementary role of pruning this information, once it has been distilled in individual units of early layers. To better understand the extent to which information pruning is actively enforced by training, we considered a hybrid VGG-16 network constructed as follows: layers $\ell \leq \ell^\star$ shared the same weights as the fully-trained (on ImageNet) VGG-16, while weights in layers $\ell > \ell^\star$ were left randomly initialized. By letting $\ell^\star$ vary, one could visualize the effect of random transformations after a given checkpoint ($\ell^\star$) and ask whether the observed decay of feature information (Figure 3) is a direct consequence of training (active information pruning) or is merely the result of architectural constraints. We found that training played an active role in pruning luminosity information (Figure 4A), with the information profile of the fully-trained network (blue curve) being consistently lower with respect to the profiles obtained for hybrid networks with intermediate $\ell^\star$ checkpoints (green curves). The effect of training was even more striking for contrast information (Figure 4B), which, in the hybrid networks, displayed a growing trend through the random convolutional layers, before finally dropping in the second fully-connected layer. In the case of orientation (Figure 4C), results were only partially consistent with those of luminosity and contrast. Training the network only up to layer 5 (i.e., up to the peak of orientation information), still yielded a large drop of information from layer 6 onward when these layers are left randomly initialized. However, orientation information did not reach the same low values attained by the fully-trained network in the last layers: information here remained substantially higher. This indicates that a reduction of orientation information after the peak in layer 5 is achieved

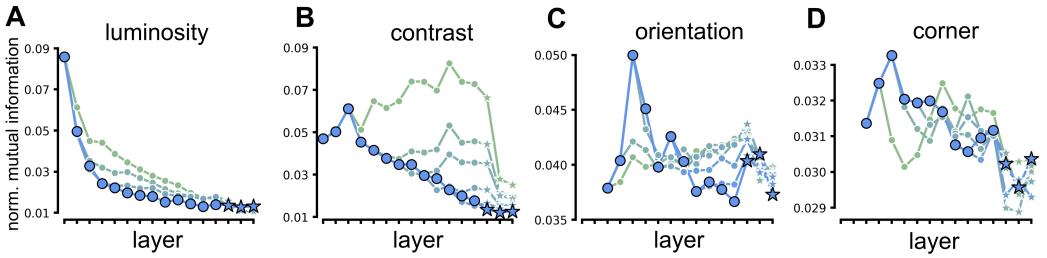

Figure 4: **Training results in active pruning of low-level image information** (**A**) Mean normalized information conveyed by VGG-16 units about image luminosity for a trained network (thick blue line; same curve as in Figure 3A) and for three additional hybrid network configurations (green lines) that have been trained only until layer $\ell^\star$ (with weights in the following layers having been left randomly initialized). The gradients of color (from green to blue) correspond to progressively larger $\ell^\star$ values, i.e., $\ell^\star \in \{1, 2, 3\}$. (**B**) Same as in **A**, but for the information conveyed by VGG-16 units about image contrast. Here the thick blue line is the same curve as in Figure 3B and $\ell^\star \in \{3, 5, 7, 9, 11\}$. (**C**-**D**) Same as in **A**, but for the information conveyed by VGG-16 about image orientation and corner.

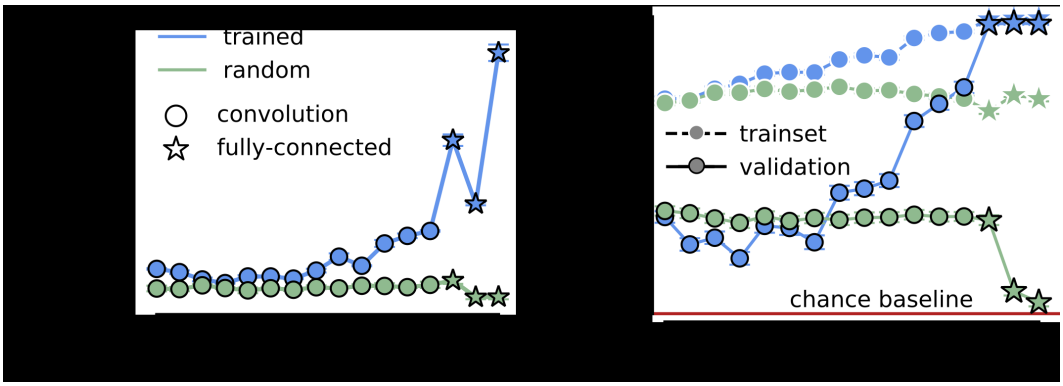

Figure 5: **Evolution of category information across VGG-16 layers** (**A**) Mean normalized information conveyed by VGG-16 units about image category for a trained (blue line) and a random (green line) VGG-16 network. As in Figure 3, error bars are standard deviations over five realizations of the experiment. (**B**) Training and validation performance (dashed and solid lines respectively) of linear SVM classifiers that were trained to predict the labels of images belonging to 10 selected Imagenet categories (250 and 50 images per category were used, respectively, for training and validation), based on the activations of a pool of 250 VGG-16 units sampled from each layer. Data points are averages (± SD) over 200 sub-populations of 100 units that were randomly sampled from each pool of 250 units.

even without training, likely because of architectural constraints (e.g., increase in receptive fields). However, further pruning of orientation information in the last layers of the network still requires training, consistently with the results found for luminosity and contrast information. A qualitatively similar behavior, albeit noisier, was found for corner information (Figure 4D).

3.4 Information on object identity emerges in late layers at both the single unit and population level

The VGG-16 network used in our experiments was pretrained to achieve high classification performance on Imagenet. Thus, all the information about the low- to middle-level visual features explored in our analyses must have been harvested (and then pruned) by the network in the attempt to maximize the separability of image categories in its output layer. Having reported how information about several such features peaked in early convolutional layers, we next asked how information about image category evolved across the network. Intuition suggests that it should peak in the very last layer, where readout takes place. It is however unclear how such information varies along the network depth, especially the one encoded by individual units. In the rat ventral stream, information about object category encoded by single neurons has been found to rise from low to high visual areas, with a matching increase in the ability of neuronal populations to support invariant recognition [15] (see Figure 1C). In [18], using a neighbourhood regularity metric, it was shown how representation-support for image category emerged sharply in late layers of various CNN architectures.

Here, we measured the category information encoded by VGG-16 units by using the label of the 1500 test images as the feature variable X_i^ℓ in Eq. (1). We found that this metric remained low and stable for about half the depth of the trained network, increasing smoothly in the last convolutional layers and then abruptly in the fully connected ones (Figure 5A, blue curve), while no trend was observed for the random network (green curve). This result resonates with that in [18], again indicating a late and sharp rise in image category information.

Next, we investigated how easily accessible was such category information encoded by single units. To this aim, we trained linear SVMs to predict the image labels based on the activity of a pool of 250 units in a layer (Supplementary Material, section B). We found a growth of decoding accuracy (Figure 5B, blue curve) that tracked the increase of category information observed, at the single unit level, in the second half of the convolutional layers (compare to Figure 5A). This suggests that, similarly to what observed along the rat ventral stream, the concentration of category information in single units plays a role in supporting the linear readout of category label at the population level.

Checklist

1. For all authors...
   (a) Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope? [Yes]
   (b) Did you describe the limitations of your work? [Yes]
   (c) Did you discuss any potential negative societal impacts of your work? [N/A]
   (d) Have you read the ethics review guidelines and ensured that your paper conforms to them? [Yes]
2. If you are including theoretical results...
   (a) Did you state the full set of assumptions of all theoretical results? [N/A]
   (b) Did you include complete proofs of all theoretical results? [N/A]
3. If you ran experiments...
   (a) Did you include the code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL)? [Yes]
   (b) Did you specify all the training details (e.g., data splits, hyperparameters, how they were chosen)? [Yes]
   (c) Did you report error bars (e.g., with respect to the random seed after running experiments multiple times)? [Yes]
   (d) Did you include the total amount of compute and the type of resources used (e.g., type of GPUs, internal cluster, or cloud provider)? [No]
4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets...
   (a) If your work uses existing assets, did you cite the creators? [Yes]
   (b) Did you mention the license of the assets? [N/A]
   (c) Did you include any new assets either in the supplemental material or as a URL? [Yes]
   (d) Did you discuss whether and how consent was obtained from people whose data you're using/curating? [N/A]
   (e) Did you discuss whether the data you are using/curating contains personally identifiable information or offensive content? [N/A]
5. If you used crowdsourcing or conducted research with human subjects...
   (a) Did you include the full text of instructions given to participants and screenshots, if applicable? [N/A]
   (b) Did you describe any potential participant risks, with links to Institutional Review Board (IRB) approvals, if applicable? [N/A]
   (c) Did you include the estimated hourly wage paid to participants and the total amount spent on participant compensation? [N/A]

Appendix

A Algorithms for orientation and contrast

We report in this Appendix the detailed pseudo-codes for the computation of the orientation and the contrast features.

The algorithm for the computation of the orientation features (see Algorithm 1) is based on a Fourier analysis of the unit-specific RF-sub patch of the image (denoted $\mathsf{Image}_{\mathrm{RF}}$ in the main text). We first extract the power spectrum by taking the norm of the complex-value 2D- Fourier Transform of the image. We then perform a log-polar transform of the image to make explicit the angular dependence of the power spectrum. By summing over the angular dimension, we obtain the total amount of power present in a given angular direction θ. We define the image orientation to be the angle of maximal power $\theta^\star$. We furthermore produce a quality metrics $\xi \in [0, 1]$ which is the Michelson contrast of the angular power spectrum. Such metric takes high values for strongly peaked spectra (i.e. there exist an angular direction which carries the most amount of power present in the image) and can later be used to score the images and implement an high-pass filter.

Algorithm 1 Estimating orientation for given $\mathsf{Image}_{\mathrm{RF}}$

procedure ORIENTATION($\mathsf{Image}_{\mathrm{RF}}$)
Input: Image tensor of shape $[C, W, H]$
$P_{xy} \leftarrow \| \ \mathsf{FFT2D}\left(\mathsf{Image}_{\mathrm{RF}}\right) \|$ ▷ *Compute power spectrum via real-FFT of* $\mathsf{Image}_{\mathrm{RF}}$
$P_{r\theta} \leftarrow \mathsf{to\_logpolar}(P_{xy})$ ▷ *Convert the power spectrum to Log-Polar coordinates*
$P_\theta \leftarrow \sum_r \left(P_{r\theta}\right)$ ▷ *Sum along the radius dimension*
$\theta^\star \leftarrow \mathrm{argmax}_\theta \left(P_\theta\right)$
$\xi \leftarrow \dfrac{\mathrm{max}P_\theta - \mathrm{min}P_\theta}{\mathrm{max}P_\theta + \mathrm{min}P_\theta}$ ▷ *Compute a quality index for orientation*
return $\theta^\star, \xi$
end procedure

The algorithm for the contrast feature (see Algorithm 2) follows a similar rationale as the one for the orientation. It is again based on a Fourier analysis of the unit-specific RF-sub patch of the image, with the major difference being the need to extract the two most powerful (in terms of the Fourier power spectrum) two orientations. Our analysis relies on the Python scipy.signal implementation of the find_peaks algorithm, which identifies the peaks in a 1D signal, in our case the angular power spectra. To compute a quality metric for the corner feature, we simultaneously measure also the values of the deepest pits of the signal. The final image score $\zeta \in [0, 1]$ is a bimodal selectivity index and takes high values for multi-peaked signals, while being small for no- or singled-peaked signals.

B SVM decoding of object identity from VGG-16 units

Following results on single-unit information about object identity (see Section 3.4 of main text), we investigated how a population-based linear decoder could harvest such information for the object classification task. We used the Python sklearn implementation of a linear SVM as our decoder. The stimulus set was composed of images taken from the ILSVRC2012 ImageNet dataset. Among the vast pool of images categorized into 1000 different classes, we selected 10 random classes and used this subset of ImageNet as out dataset. We then built a training set (sampling from the ImageNet training set) which consisted in a total of 2500 images (250 images per class), while we used all the available 50 images per class of the ImageNet validation set (for a total of 500 images) as our validation dataset. We then recorded the activations of a random sub-population of 250 units from each layer of

Algorithm 2 Estimating corner for given $\mathsf{Image}_{\mathrm{RF}}$

procedure CORNER($\mathsf{Image}_{\mathrm{RF}}$)
Input: Image tensor of shape $[C, W, H]$
$P_{xy} \leftarrow \| \mathsf{FFT2D}\left(\mathsf{Image}_{\mathrm{RF}}\right) \|$ ▷ *Compute power spectrum via real-FFT of* $\mathsf{Image}_{\mathrm{RF}}$
$P_{r\theta} \leftarrow \mathsf{to\_logpolar}(P_{xy})$ ▷ *Convert the power spectrum to Log-Polar coordinates*
$P_{\theta} \leftarrow \sum_{r} \left(P_{r\theta}\right)$ ▷ *Sum along the radius dimension*
$\boldsymbol{\theta}^{\star}, \boldsymbol{\mu}^{\star} \leftarrow \mathsf{find\_peaks}\left(P_{\theta}\right)$ ▷ *Get position* $\boldsymbol{\theta}^{\star}$ *and values* $\boldsymbol{\mu}^{\star}$ *of* P_{θ} *peaks*
$\_\,, \boldsymbol{\nu}^{\star} \leftarrow \mathsf{find\_peaks}\left(-P_{\theta}\right)$
▷ *Get position and value of highest peak*
$i,\ \mu_1 \leftarrow \mathrm{argmax}_{(1)}\left(\boldsymbol{\mu}^{\star}\right), \mathrm{max}_{(1)}\left(\boldsymbol{\mu}^{\star}\right)$
▷ *Get position and value of second-to-highest peak*
$j,\ \mu_2 \leftarrow \mathrm{argmax}_{(2)}\left(\boldsymbol{\mu}^{\star}\right), \mathrm{max}_{(2)}\left(\boldsymbol{\mu}^{\star}\right)$
$\theta_1^{\star}, \theta_2^{\star} \leftarrow \boldsymbol{\theta}^{\star}[i], \boldsymbol{\theta}^{\star}[j]$ ▷ *Get corresponding angle of first and second peak*
▷ *Get values of first two deepest pits*
$\nu_1, \nu_2 \leftarrow \mathrm{min}_{(1)}\left(-\boldsymbol{\nu}^{\star}\right), \mathrm{min}_{(2)}\left(-\boldsymbol{\nu}^{\star}\right)$
$\zeta \leftarrow \dfrac{\mu_2 - \nu_2}{\mu_1 - \nu_1}$ ▷ *Compute a quality index for corner*
return $\theta_1^{\star}, \theta_2^{\star}, \zeta$
end procedure

the (Pytorch implementation of) VGG-16 neural network. We considered both a fully-trained (on ImageNet) VGG-16 network and a randomly-initialized one as control.

We sampled a random sub-population of 100 units among the 250 available in each layer and then fitted a linear-SVM model to predict the classification label based on the activations of the whole sub-population. We then repeated the experiment 200 time with independent samples of the sub-population. The final estimate for the population-based decoding was then measured as the classification accuracy (both on the training and validation set) averaged over the 200 realizations of the experiment.

C Probing Information after the non-linear ReLU activations

In a given layer of a neural network, one can consider unit activations before the non-linear activation gate (ReLU in VGGs), or after the gate. This choice is somewhat arbitrary, because we are interested in a layerwise comparison and both choices allow measuring information and comparing it between layers in a consistent way. Intuitively, they correspond (respectively) to the information received by a neuron from the previous layer or transmitted to the next. In the main text we measured the linear activations of the layer unit (pre-activations), because we speculated that this could be advantageous as ReLU gates maps half of possible values to zero, making it harder to spot interesting patterns in information by decreasing the range over which this can vary between layers. We check here that the

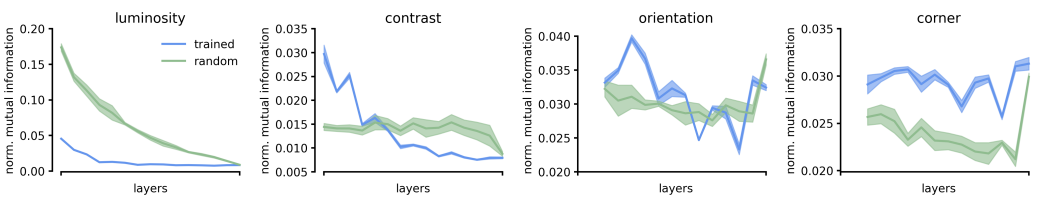

Figure C: **Single unit Mutual Information after ReLU activation** Mean normalized information conveyed by VGG-16 units when probed after the layer activation (ReLUs). Visual features and color conventions are the same as in Figure 3 of main text. Shaded area are standard deviations over five realizations of the experiment (independent sampling of units, images and random weights).

choice between the two alternatives does not qualitatively affect the results. Indeed, the trends for the single unit mutual information when probed after the non-linearity are qualitatively similar to those presented in the main text (compare Figure C with Figure 3 of main text).

D Mutual Information trends in other VGG networks

We report the measured single-units mutual information trends for the same visual features (luminosity, contrast, orientation and corner) in two different networks of the VGG family: VGG-11 and VGG-19. The observed profiles are very similar to the ones presented for VGG-16, exhibiting the complementary pruning and distilling phenomena described in the main text.

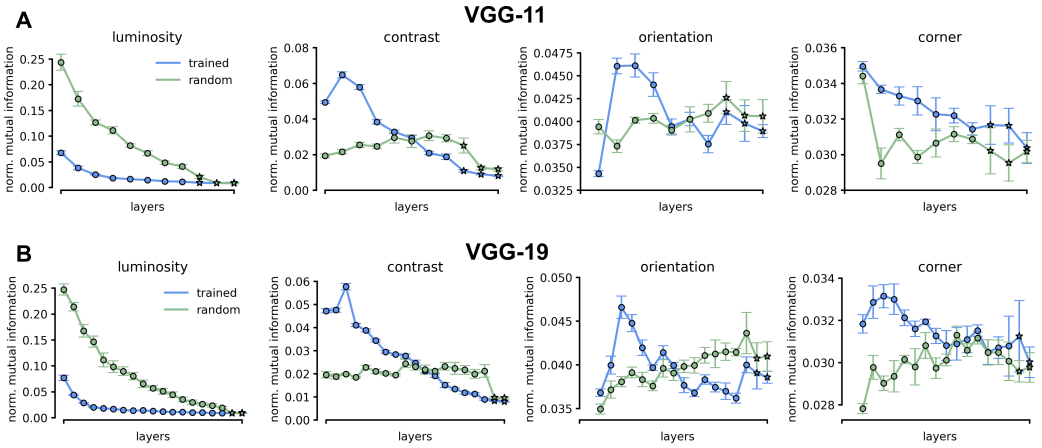

Figure D: **Mutual Information in VGG-11 and VGG-19** (**A**) Mean normalized information conveyed by VGG-11 units about image luminosity, contrast, orientation and corner. Color and marker conventions are the same of Figure 3 of main text. Error bars are standard deviations over five realizations of the experiment. (**B**) Same as in (**A**) but for units in a VGG-19 network.