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 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 Image$_{\mathrm{RF}}$

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

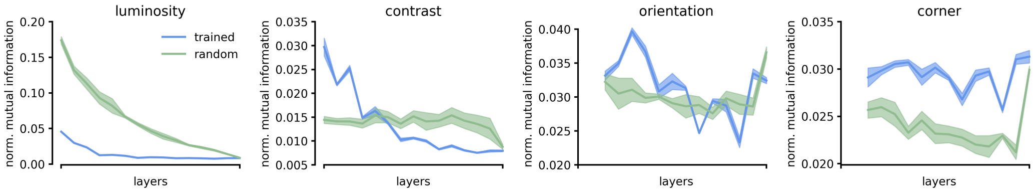

Figure C: **Single unit Mutual Information after ReLU activation** Mean normalized information conveyed by VGG-16 units when probed after the layer activation (ReLUs). Visual features and color conventions are the same as in Figure 3 of main text. Shaded area are standard deviations over five realizations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

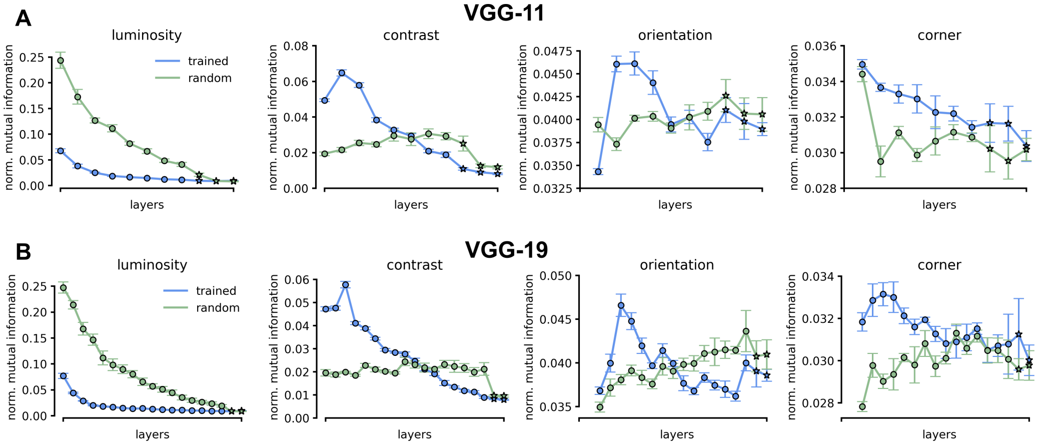

Figure D: **Mutual Information in VGG-11 and VGG-19** (**A**) Mean normalized information conveyed by VGG-11 units about image luminosity, contrast, orientation and corner. Color and marker conventions are the same of Figure 3 of main text. Error bars are standard deviations over five realizations of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