A feedforward mechanism for human-like contour integration

Fenil R. Doshi, Talia Konkle & George A. Alvarez
Department of Psychology, Harvard University, Cambridge, Massachusetts
Kempner Institute for the Study of Natural and Artificial Intelligence,
Harvard University, Cambridge, Massachusetts

Deep neural network models provide a powerful experimental platform for exploring core mechanisms underlying human visual perception, such as perceptual grouping and contour integration — the process of linking local edge elements to arrive at a unified perceptual representation of a complete contour. Here, we demonstrate that feedforward, nonlinear convolutional neural networks (CNNs) can emulate this aspect of human vision without relying on mechanisms proposed in prior work, such as lateral connections, recurrence, or top-down feedback. We identify two key inductive biases that give rise to human-like contour integration in purely feedforward CNNs: a gradual progression of receptive field sizes with increasing layer depth, and a bias towards relatively straight (gradually curved) contours. While lateral connections, recurrence, and feedback are ubiquitous and important visual processing mechanisms, these results provide a computational existence proof that a feedforward hierarchy is sufficient to implement gestalt "good continuation" mechanisms that detect extended contours in a manner that is consistent with human perception.

INTRODUCTION

As we shift our gaze from location to location, our perceptual system effortlessly transforms millions of disconnected points of light into coherent representations. These representations support a myriad of downstream tasks, from object recognition to social perception, to visually guided reaching and navigation within dynamic environments. To enable this rich behavioral repertoire, the visual system capitalizes on spatial and temporal regularities to parse and organize the visual field into meaningful holistic units – contours, surfaces, shapes, agents – with predictable behaviors and interaction affordances (Biederman & Ju, 1988; Biederman, 1987; Marr & Nishihara, 1978; Nakayama & Shimojo, 1992; Nakayama et al., 1995; Von der Heydt, 2015).

The process of forming such holistic visual representations was most famously studied by the Gestalt psychologists (Wertheimer, 1912; Köhler, 1967; Koffka, 1922) who emphasized the importance of configural relationships between parts, whereby the perceptual system groups together elements that are likely from the same physical object in the environment to represent meaningful, structured “wholes.” Gestalt psychologists documented several laws that dictate which elements will be grouped together, such as proximity, similarity, connectedness, continuity, parallelism, closure, and symmetry (Wertheimer , 1923; Wertheimer, 1938; Vezzani et al., 2012; Wagemans et al., 2012; Ellis, 2013). While these grouping phenomena were initially examined through the lens of subjective perceptual experience, efforts have been made to

objectively measure and model them in well-controlled experiments with stimuli that offer parametric control (Wertheimer , 1923; Kubovy, 1994; Hochberg & Hardy, 1960; Quinlan & Winton, 1998; Rock & Brosgole, 1964). However, progress integrating these ideas into a broader theory of vision has been hindered by the lack of image-computable models of high-level visual processing that can manifest these phenomena.

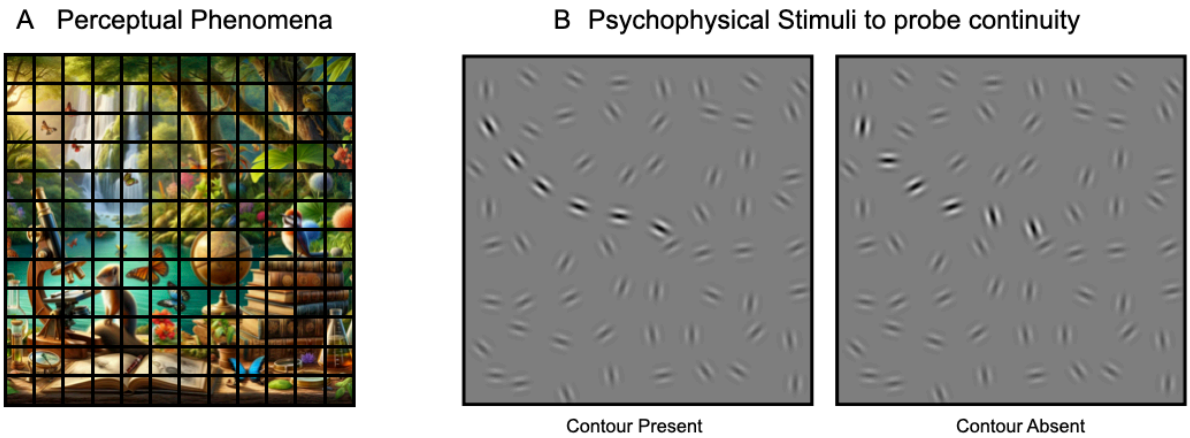

Fig. 1. Visual Grouping and Contour Integration in Human Perception. (A) *Demonstration of the phenomena where observers can group boundaries from the same object despite occlusions.* ***(B)*** *Psychophysical stimuli to examine the 'continuity' principle underlying contour integration (Field et al., 1993). The left panel displays a contour-present stimulus containing a subset of gabor patches (i.e. contour elements) systematically aligned to suggest an extended, coherent contour amidst randomly oriented background patches. The right panel displays a contour-absent stimulus with identical gabor patches, except the contour patches are also randomly oriented, eliminating the perception of the continuous contour. Contour patches are highlighted for illustrative clarity.*

One hundred years after Gestalt Psychology started, deep neural networks (DNNs) have emerged as powerful image computable models that can perform high-level visual tasks, and thus have the potential to fill this void. DNN models are promising not only because they are highly performant on challenging visual tasks (Krizhevsky et al., 2012; Russakovsky et al., 2015), but also because they predict human perceptual judgments (Kubilius et al., 2016; Ritter et al., 2017; Feinman & Lake., 2018; Peterson et al., 2018, Battleday et al., 2020) and activity in visual cortex (Khaligh-Razavi & Kriegeskorte, 2014; Yamins & DiCarlo, 2016 ; Güçlü & Van Gerven, 2015; Schrimpf et al., 2018; Rajalingham et al., 2018; Konkle & Alvarez., 2022) better than any previous hand-designed models. Despite these strengths, however, these models appear to lack human-like structure in their internal representations: DNNs favor local features like color and texture over global shape-related features (Alvarez & Konkle, 2022; Baker et al., 2018b; Geirhos et al., 2018; Malhotra et al., 2020), lack sensitivity to configural relationships between object parts (Malhotra et al., 2023; Puebla & Bowers., 2022; Baker et al., 2023), and show limited or mixed success in capturing perceptual grouping phenomena (Jacob et al., 2021; Bowers et al., 2022; Biscione & Bowers, 2023). While these shortcomings have been well-documented, it remains unclear the extent to which the lack of structured representations within

these models reflects aspects of the visual diet, model architecture, task objective, and/or regularization pressures that shape representations (Hermann & Lampinen., 2020; Bowers et al., 2022; Richards et al., 2019).

To gain some traction on these questions, we focus our investigation on a specific perceptual grouping operation known as contour integration (Elder & Goldberg., 2002; Geisler et al., 2001; Field et al., 1993). Contour integration is the process of stitching together local edge representations (Hubel & Wiesel, 1968; Field, 1987, 1993), which may appear fragmented in the retinal image due to factors like clutter, occlusion, and contrast differences between the figure and ground (Elder et al., 2003; see **Fig. 1A**). Contour integration relies particularly on the *continuity principle* (i.e., gestalt "good continuation") and makes an interesting case study because it is amenable to pychophysical experiments (Field et al., 1993), and has been proposed to rely on lateral mechanisms (Field et al., 1993; Kapadia et al., 1995) as well as feedback and recurrence (Hansen & Neumann, 2009; Li, 1998; Roelfsema, 2006; Piëch et al., 2013; Linsley et al., 2018; Veerabadran & de Sa., 2019; Linsley et al., 2020; Khan et al., 2020; Boutin et al., 2021). Moreover, prior work has assumed that these mechanisms are likely required because the combinatorial explosion of possible contour configurations prevent a feed-forward solution (Field et al., 1993) — an assumption we can now test with modern DNN models.

Here we examine whether purely feed-forward deep convolutional neural networks can implement contour integration, and identify the key inductive biases necessary to achieve human-like contour detection abilities. We demonstrate that bottom-up feedforward neural networks without lateral connections, recurrence, or top-down feedback mechanisms can in fact support contour detection through the integration of locally aligned elements. Second, we establish that this capacity is supported by the architectural inductive bias of non-linear processing over progressively increasing receptive field sizes. Third, we find that our model quantitively predicts human sensitivity to different global contour curvatures (Field et al., 1993) best when tuned to integrate local elements with small orientation differences — a bias towards straight edges or gradual curves. Collectively, these findings provide compelling computational evidence that purely feedforward architectures can support contour integration in a manner that is consistent with human psychophysical data, but that additional inductive biases are needed for models to show a fully emergent alignment with human perception. More broadly, the finding that a hierarchy of progressively increasing receptive fields enables the extraction of large scale structure (e.g., an extended contour) from relatively fine-grained properties of the input (e.g., alignment of relatively high frequency elements) suggests that purely feed-forward processing can play a more significant role in higher-order grouping operations than prior work has assumed. We propose an integrated view in which lateral and recurrent processes refine the inputs to the feed-forward hierarchy, enabling the feed-forward mechanisms to perform the crucial integration operations over distilled inputs.

RESULTS

Feedforward deep convolutional neural networks (DCNNs) can support fine-grained detection of extended contours

Here we assess how well a classic, feed-forward, deep convolutional neural network architecture – the Alexnet – can support contour detection via contour integration. While these models can

successfully categorize objects among a thousand categories, they are infamously "texture-biased" (Geirhos et al, 2018), and can even recognize objects successfully even when images are globally scrambled (Brendel & Bethge, 2019). Additionally, these models lack mechanisms such as within-layer association (self-attention) or lateral recurrence operations, which explicitly enable associative processing of image patches within a layer. Thus, we hypothesized that feed-forward convolutional models, due to these architectural limitations, would perform poorly at the task of detecting an extended contour of local gabor elements hidden amidst a grid of randomly-oriented gabor elements.

To test this idea, we first developed a controlled set of contour stimuli for a contour detection task, drawing directly from prior human psychophysics research (Field et al. 1993). Each display consists of a jittered grid of 256 gabor elements (arranged on a 16 x 16 grid in images of size 512 x 512 pixels). Half the displays contain an embedded contour, where the orientations of 12 local gabor elements are aligned to form an extended path (see **Fig. 1B**). We refer to these elements (i.e. the contour elements) as having aligned orientations, because they form paths with varying degrees of global curvature. The remaining background gabor elements in the display are all randomly oriented. These 'contour-present' displays contain a hidden contour. For each contour-present display we also generate matched 'contour-absent' displays which contain identical background gabor elements as their corresponding contour-present images, while the contour elements in these displays are misaligned with random orientations. This careful stimulus construction prevents detecting contours via other cues of proximity or density, requiring integration across the aligned co-circular elements to distinguish contour-present from contour-absent displays. **See Methods and Supplementary Fig. 1** for more details on stimuli generation.

To examine the capacity of the Alexnet model's ability to detect these contours, we compared three critical model variants. (1) An untrained model with randomly initialized weights, to test whether the model architecture alone, without training, enables contour integration. (2) A model trained on ImageNet 1000-way image classification, to examine whether the features learned over natural images to support object recognition show improved contour integration. (3) A model that is first trained on ImageNet to do 1000-way classification, and then subsequently fine-tuned end-to-end to support contour integration. This final variant explores the maximum capacity that could be supported by the architecture by directly fine-tuning on the contour detection task. And, for all three models, we probed each layer, to examine whether contour information is more accessible at earlier or later stages of the hierarchy.

To quantify the contour detection capacity of any given model layer, we trained a separate linear classifier over the output activations of each layer to distinguish between the contour-present and contour-absent displays. Stimuli included displays with contour curvatures spanning a broad range of curvature (from gradual to irregular). Classification accuracy was assessed on a held-out set of test images, as our measure of the contour-detection capacity at each layer (**see Methods** for details on contour-detection linear probing).

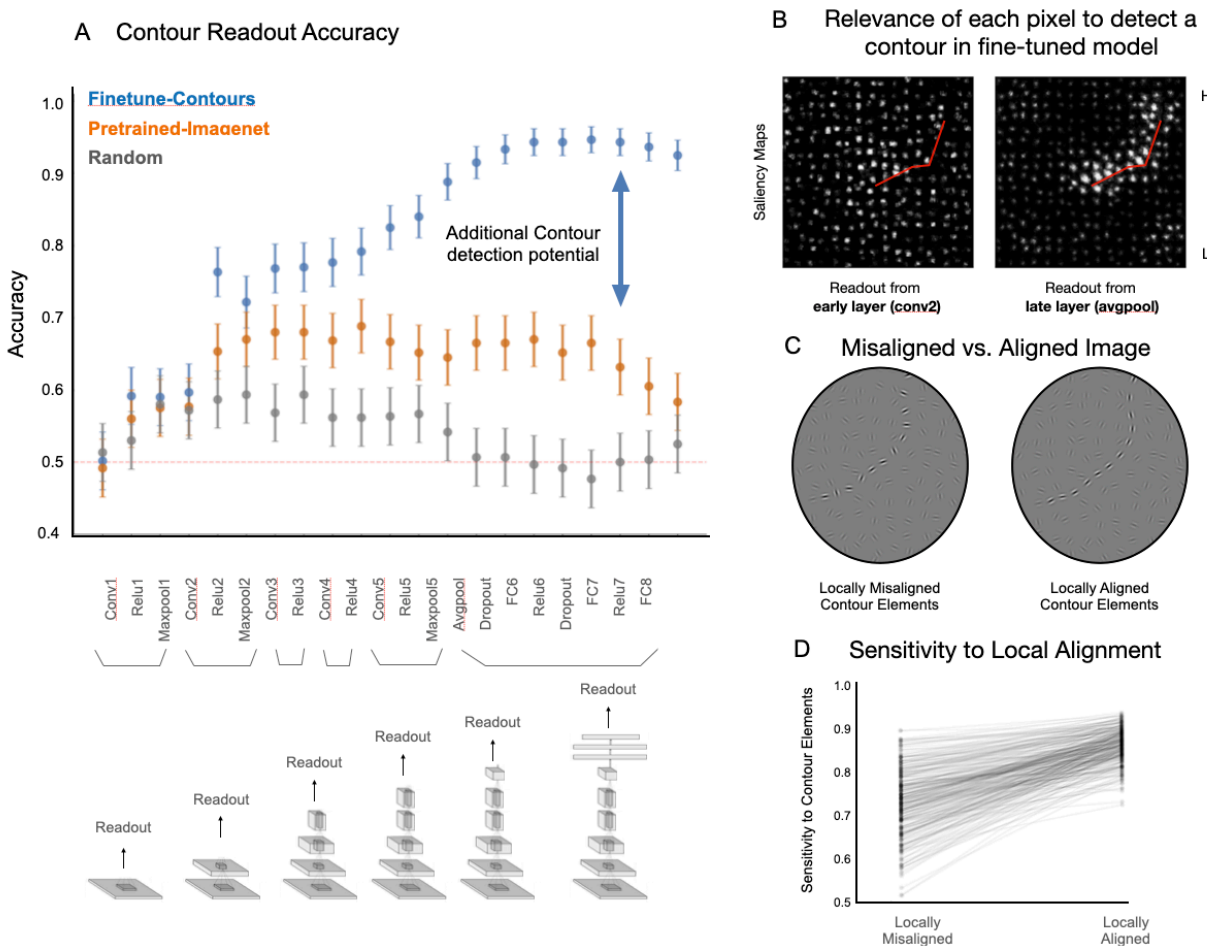

Fig. 2. Contour Integration Capacity in Feedforward CNNs. (A) *Accuracy on contour detection across different readout layers – gray indicating randomly initialized models, orange for models pretrained on Imagenet for object recognition, and blue for finetuned models on the held-out test set. Error bars denote 95% confidence intervals for readout accuracy.* ***(B)*** *Saliency maps from two fine-tuned models, highlighting pixel relevance for detecting the contour within an example image; location of the contour is highlighted in red for illustrative purposes.* ***(C)*** *Example image pair with misaligned and aligned contour elements.* ***(D)*** *Plot showing the fine-tuned model's sensitivity to elements making up the contour (for aligned display), or elements at the same locations in misaligned display. Each pair is connected via a gray line. Overall the plot depicts the sensitivity to local alignment of contour elements.*

The results of all three model variations are shown in **Figure 2A**. First, the untrained model shows poor contour detection accuracy, which never exceeds 60% for any layer of the model (gray dots). This result indicates that the convolutional inductive bias alone provides minimal support for contour integration. The model trained on ImageNet classification (**Figure 2A**, orange dots) did show higher contour detection accuracy than untrained models, particularly within deeper layers of the model. This result indicates that the features learned over natural images to support object recognition do in fact improve the contour integration capacity of the

Alexnet model, conferring better contour detection at relatively later stages of the hierarchy. However, the contour detection accuracy in the Imagenet trained model peaked at a modest 68%.

Surprisingly, the model that was subsequently fine-tuned on the contour detection task led to a substantially greater capacity for contour integration (**Figure 2A**, blue dots). This was true particularly within the deeper layers, where accuracy is consistently above 90% with a peak of 94.3%. This result reveals that models with purely feed-forward convolutional processing are in fact architecturally-capable of accurately detecting these extended contours.

While our initial plan was to begin adding in other mechanisms that enable convolutional neural networks to do contour integration, we instead explored how it was possible that these feed-forward networks could succeed at the task at all. Note that with the success of these fine-tuned models, we are not proposing a learning story where the visual system learns contour integration through training on the task explicitly (or anything of that sort). Instead, our aim next is to clarify how this purely feed-forward convolutional architecture is succeeding at this task. This is an important research direction for two reasons. First, current theoretical accounts of contour integration generally assume that local associative mechanisms —such as lateral connections or top-down feedback from subsequent processing stages—are *necessary*. Therefore, understanding how such a feed-forward model achieves this competence without such mechanisms will reveal new insights for the task of contour detection. Second, it's possible the model's mechanism is also at play in the human visual system and understanding how the model is achieving this could therefore provide new insights into this phenomenon in human vision.

Fine-tuned networks detect contours via integration of locally aligned elements

How is it that our models trained over natural images, and then fine-tuned on contour detection, can successfully perform the contour detection task by their later processing stages? One potential concern about these results is that these models have found some short-cut or confound present in the stimulus displays, which allows them to perform well on the contour task without actually detecting the contour per se. Do these fine-tuned models (**Figure 2B**, blue dots) actually integrate information across the entire contour? We first address this potential issue, by leveraging tools from the field of feature-visualization, to quantify the extent to which the model isolates the contour elements from the background elements. Specifically, we used a feature visualization technique called Guided Backpropagation (Springenberg et al., 2014) to highlight the parts of the image that significantly contribute to the contour present/absent decision (**see Methods** for details on computing saliency maps). This method produces a "saliency map", where the relevance of each pixel to the final "contour-present" decision is highlighted.

In **Fig. 2B**, we show the saliency maps over an example image where the location of the contour is highlighted with a red line for visualization purposes. We focus on read-out from two fine-tuned models, where the linear-classifier is attached to the "conv2" layer (early stage) and the "avg pool" layer (late-stage). The "avg pool" layer (the final stage of the convolutional backbone, before the fully-connected read-out layers), which is relatively late stage of the Alexnet model as compared to the conv2 layer, shows higher contour detection accuracy (91.5% - avgpool layer; 59.6% - conv2 layer). It is clear that the avgpool model is leveraging more

information from the contour elements along the path as compared the background while successfully detecting the contour. In contrast, a linear read-out head attached to an earlier stage (conv2), shows little-to-no concentrated salience along the embedded contour. (See **Methods** and **Supplementary Fig. 2** for quantitative results for all the layers). These results indicate that deeper layers in the feedforward architecture, but not shallow layers, have features that do in fact leverage the contour elements along the path to perform the contour detection task.

We next explore an even more nuanced potential concern. While the saliency maps depict sensitivity to contour elements, is successful performance actually arising from the *alignment* of these elements, and not just their locations, since the placement of contour elements is non-random? (**see Methods; Stimulus Generation**). To investigate this question, we compared the sensitivity scores for contour elements across image pairs (contour-present and contour-absent images) that differed only in whether the contour elements were aligned (forming a contour) or misaligned (no contour), maintaining the same locations for all elements as shown in **Fig. 2C**. The model (fine-tuned end-to-end from the avgpool layer) showed higher sensitivity to contour elements when they were aligned, as depicted in **Fig. 2D**. This result confirms that the model relies on fine-grained collinearity between contour elements to isolate the contour from the background. This pattern was also observed in models fine-tuned from other layers of Alexnet, with a greater emphasis on local alignment evident in models fine-tuned from deeper layers (refer to **Supplementary Fig. 2**).

Taken together, these results confirm the result that purely feedforward models have significant contour-detection capabilities, and can effectively support contour integration by stitching together elements based on local alignment cues.

Progressive receptive field growth with layer depth underlies the model's ability to support contour integration

Another potentially deflationary account of this result is that success is trivial: any deep neural network can support contour integration because deep neural networks are universal function approximators. However, this property of universality only holds in the limit and does not necessarily apply to all networks (especially smaller ones). Consequently, it is possible for models to be architecturally incapable of successfully performing the contour detection task. We next aimed to design models exactly with this property in mind, in order to identify what aspect of the standard DCNN architecture was enabling successful contour detection.

Specifically, we hypothesized that the progressively increasing receptive field sizes of the Alexnet model architecture were critical for its capacity to detect contours. This hypothesis derived from the result that the read out from later stages, not the earlier stages, showed the strongest task performance. To examine this possibility, we designed a new model architecture, dubbed "PinholeNets, which are architecturally similar, but with a much more restricted field of view in their convolutional layers. This "cousin" model class allows us to directly investigate the impact of receptive field size progression on a model's ability to integrate extended contours, while holding as many other architectural aspects as we could constant.

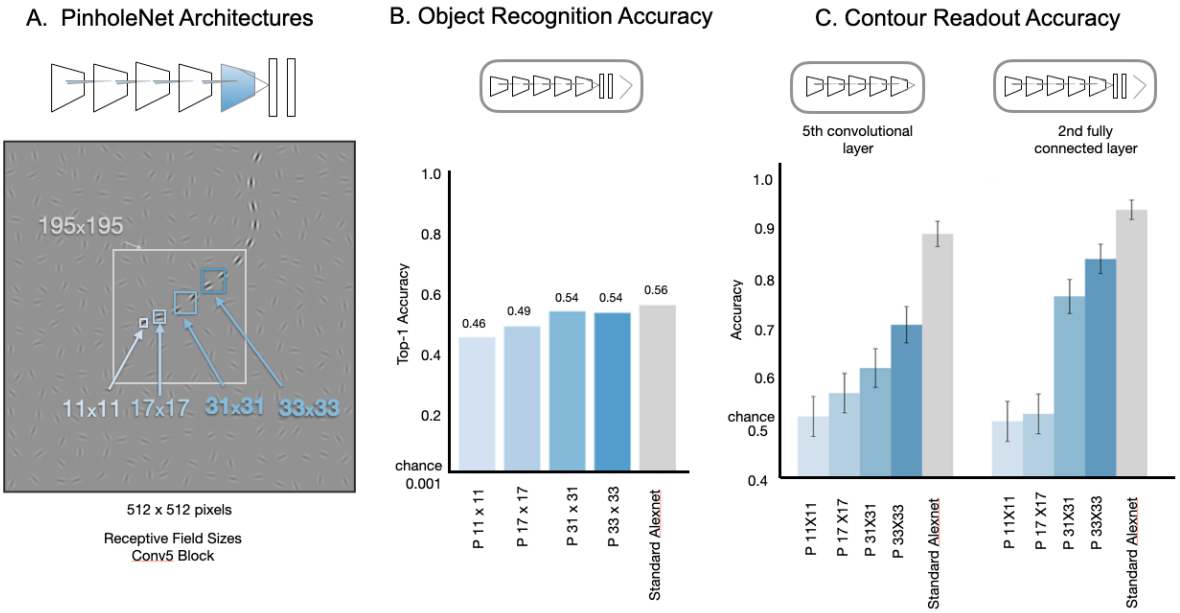

Fig. 3. Impact of Receptive Field Size and Progression on Contour Integration in Feedforward Models. (A) *At the top, a schematic representation of PinholeNet architectures, highlighting the restricted field of view within the convolutional layers. The bottom images visualizes the receptive field size (blue squares) for units in the fifth convolutional block of the different PinholeNet variants. For comparison, the grey square shows the size for a unit in the fifth convolutional layer of a standard Alexnet model.* ***(B)*** *Top-1 object recognition accuracy on the ImageNet validation set for PinholeNet models with varying receptive field sizes (blue bars), compared to the standard Alexnet model (gray bar), which serves as a performance benchmark.* ***(C)*** *Contour detection accuracy for readout from the 5th convolutional layer (left) and 2nd fully-connected layer (right) in PinholeNets (blue bars) and the standard Alexnet model (grey bar) on the held-out test set. The error bars denote the 95% confidence intervals for readout accuracy.*

We built four variants of PinholeNet models, here referred to as P11, P17, P31, and P33, named based on the receptive field size of units in the last (i.e. fifth) convolutional block. Hence the P11 model corresponds to an architecture whose unit in the fifth convolutional block would have a receptive field size of only 11 x 11 pixels. By comparison, the original Alexnet model has receptive field sizes of 195 x195px in conv5. **Fig. 3A** shows the receptive field of these units over the stimuli containing an extended contour. **See Methods** and **Supplementary Fig.3A** for a detailed description of the PinholeNet architecture. It is critical to note that the number of kernels and hierarchical (non-linear) stages across all models is kept constant, and the only varying factor is the receptive field size of the units within the network.

Before testing whether these PinholeNets fail at contour integration, for comparative purposes it is important to first establish that these models are capable of learning useful representations at all, e.g. for classifying objects. Indeed, prior work has demonstrated that models with a restricted field of view can still be trained successfully on object recognition (Brendel & Bethge, 2019), revealing that very local cues in the ImageNet dataset are sufficient to support object

classification (Baker et al., 2018b; Geirhos et al., 2018). We trained the PinholeNets on ImageNet for object recognition by pooling the final convolutional output across space, and then attaching the standard Alexnet classifier head (three fully connected layers). The terminal fully connected layer provided the 1000-way probability vector for ImageNet object categories.

We found that all PinholeNets attained a level of accuracy that is on par with the standard Alexnet model, with **Fig. 3B** showing their top-1 performance on the held-out ImageNet validation set (P11 – 45.63%; P17 – 49.34%; P31 – 54.45%; P33 – 53.94%). PinholeNets are illustrated in varying shades of blue (lighter shades correspond to models with smaller receptive fields), while the standard Alexnet model is represented in grey. Even though the standard Alexnet has the highest accuracy, it is just 10% more than the worst performing Pinholenet (in a task where chance accuracy is 1/1000, or 0.1%). Thus, despite their limited field of view in the convolutional backbone, PinholeNets are capable of classifying objects at similar levels as their Alexnet cousin, and learn features that are useful for object recognition.

We next evaluated their capacity for contour integration. To do so, we initialize with weights that support object recognition and then fine-tune the models in an end-to-end fashion following the same procedures as on the Alexnet model. To simplify the comparison between models, we focus on the results for readout from the 5th convolutional layer, which is the final layer with restricted field of view. In this layer (**Fig. 3C-left)**, performance on the contour detection task increases as the model's receptive field size increases, with all PinholeNets falling well short of the standard Alexnet model (converging at P11 – 52.33%; P17 – 57%; P31 – 61.83%; P33 – 70.66%; standard Alexnet 88.83% on the held-out test set). Thus, these results show that there exist a class of feedforward models which are architecturally limited to detect extended contours due to smaller receptive fields, while still able to succeed at object recognition.

We next probe these PinholeNets from the 2nd fully connected layer—this is an interesting test because at this stage, the units in the layer all have full-field receptive fields, even for PinholeNets (**Fig. 3C-right)**. Interestingly, there still remains a significant gap between the performance of PinholeNets and standard Alexnet. This gap is especially pronounced when the readout is attached to the smallest PinholeNets backbone architectures (accuracy converging at P11 – 51.3%; P17 – 52.8% on the held-out test set). Thus, these results also show that it is not sufficient to have full-field receptive fields in order to be able to perform the contour detection task well. Instead, this result indicates that progressively increasing receptive field sizes enroute to those fully-connected layers is important.

Finally, we address the concern that maybe Alexnet's superior contour detection performance is just a byproduct of its slightly superior object recognition accuracy. To do so, we trained a standard Alexnet for half the normal number of epochs (50), at which point the model achieves a top-1 object recognition accuracy of 50.23%, aligning closely with the accuracy exhibited by the completely trained PinholeNets (between 45.63% and 53.94%). We then proceeded to finetune this half-trained standard AlexNet model on the contour detection task. We found that, even with a reduced accuracy in ImageNet object recognition, this model still exhibited superior performance in contour detection (89.33%; indicated by the black dotted line in **Supplementary Fig. 3C**) compared to the PinholeNets (between 52.33% and 70.66%).

Taken together, these results provide clear evidence that having progressively increasing receptive field sizes through a feed-forward processing hierarchy is a sufficient architectural inductive bias that enables the ability for successful contour detection.

Human sensitivity to global path curvature

The previous results established the contour-integration capabilities of (certain) feedforward deep neural networks models, but it is unclear whether any of these models show human-like perceptual signatures of contour integration. To address this question, we examined whether the fine-tuned Alexnet models show a drop in contour detection accuracy as global curvature increases, which is a signature of human contour-integration. To do so, we had both humans and models perform a contour detection task for contours of various curvatures, replicating the human psychophysics paradigm introduced by Field et al., 1993. Then, we compared contour detection performance on a trial-by-trial basis across humans and models. To do so, we also developed a principled method for comparing noisy human responses and noiseless model responses based on signal detection theory (Green & Swets, 1966).

For this behavioral experiment, we generated a set of 2000 images (1000 contour present/absent controlled pairs), and varied the contour curvature across displays. The degree of curvature of the path in the contour-present image is described by the angle between successive contour elements along the path ($\pm \beta$). This curvature level was fixed for a path in an image (15°, 30°, 45°, 60°, or 75°), with smaller β values resulting in displays with straighter contours, and larger β values leading to curvier and more irregular contours (see **Fig. 4A**). For the contour-absent image, contour elements were positioned at exactly the same locations but were randomly oriented. The background elements were identical for the contour-present and contour-absent images in a given present/absent pair (see **Methods section** for additional details on stimulus generation). The dataset was randomly split into five sets of 200, with 40 pairs for each beta condition in each set, and each participant was shown trials from one of these five sets selected at random.

Participants performed a 2-interval forced-choice (2-IFC) contour detection task, where they were presented with two displays (one with the contour-present image and another with the corresponding contour-absent image) and had to identify which one contained an extended contour. A typical trial is depicted in **Fig. 4B**. Over the course of the experiment, participants encountered 200 trials with contours at all levels of curvature (with trials counterbalanced across ten conditions: five distinct β values, presented in either the first or second interval of the trial). Different trial-types were uniformly distributed over the full experiment. See the **Methods section for** behavioral experimental design. To ensure the reliability of our results, we excluded participants whose performance was beyond 2.5 standard deviations of the group mean accuracy or whose individual performance trends across all β conditions did not correlate well with the group trend (see exclusion criteria in **Methods** and **Supplementary Fig. 4A**). Our final analysis was based on a refined pool of 78 participants who met these exclusion criteria (**Supplementary Fig. 4C**).

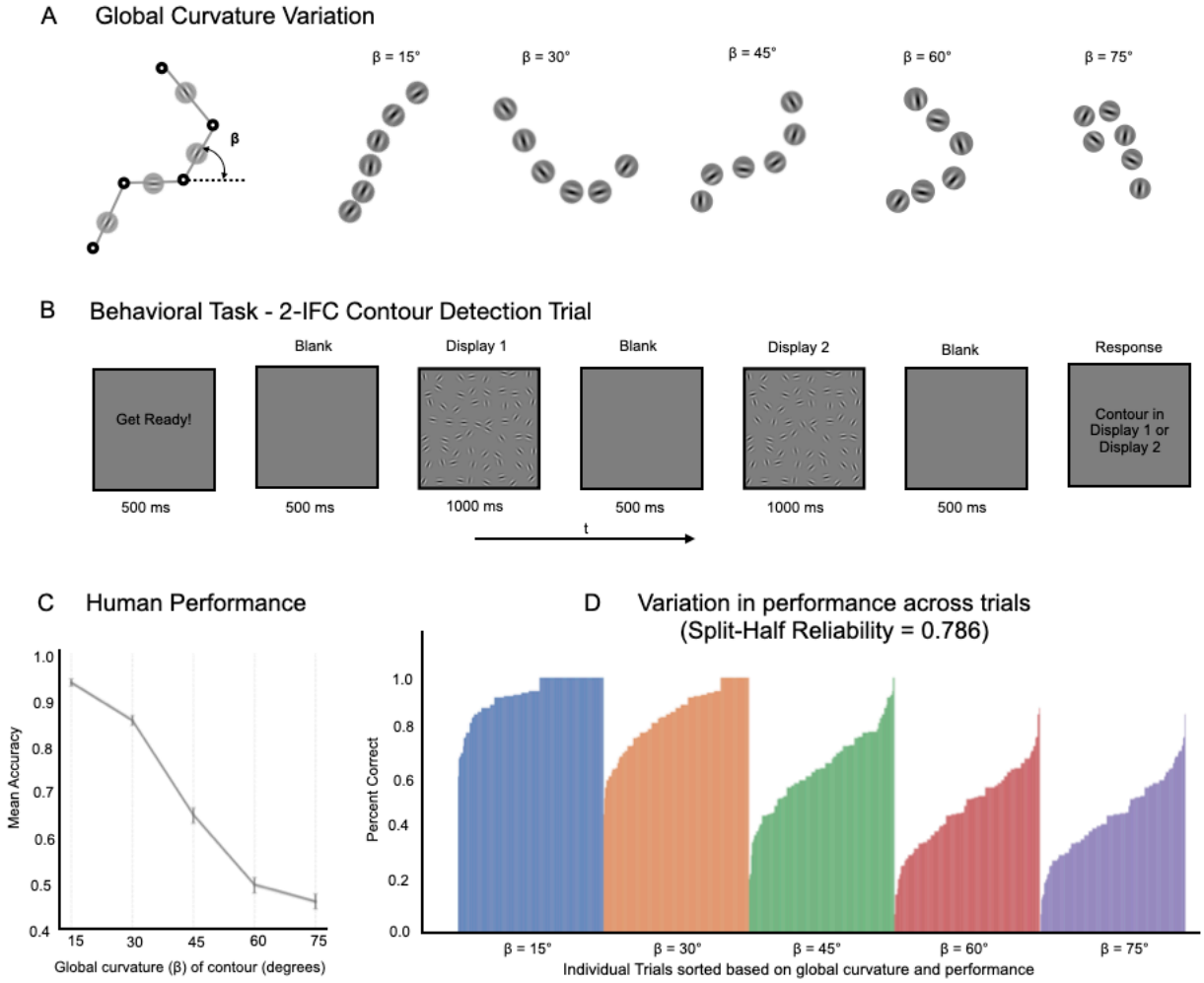

Fig. 4. Human Sensitivity to Global curvature. (A) *Variation in global curvature (β) across a range of contour stimuli used in the study, with β values set at 15°, 30°, 45°, 60°, and 75°, demonstrating straighter to more curved contours.* ***(B)*** *Sequence of a 2-IFC contour detection trial where participants identify the display containing the contour.* ***(C)*** *Mean accuracy of participants for contour detection across varying β conditions, with error bars representing 95% confidence intervals bootstrapped across participants.* ***(D)*** *Bar graph showing the variability in human performance across individual trials within each β condition.*

We plot the mean accuracy across the five curvature levels (β conditions) for these participants in **Fig. 4C**. The error bars denote the 95% CI of the mean accuracy for each beta condition bootstrapped across the 78 participants. The average results across curvature conditions shows that participants were able to accurately detect relative straight contours, and that the task becomes increasingly difficult as the beta value increases (**Fig. 4C**).

However, there is additional richness in these data revealed by plotting the performance for individual trials (measured as percent correct across participants) as shown in **Fig. 4D**. Here, we find substantial variation across the trials within displays of the same curvature level (see **Supplementary Fig. 5 for examples**). And, these differences are highly consistent across participants, with the split half reliability across participants at r = 0.65 and the Spearman-Brown adjusted estimate of reliability at 0.786, 95% CI [0.777 to 0.794]. Critically, the reliability of these trends enables us to compare the strength of contour representations at the level of individual trials (i.e. each image pair).

Estimating decision strength in deterministic models

One challenge in comparing human and model decisions on a trial-by-trial basis is that these models, like most deep neural network models, are deterministic: they will give the same response to an image every time it is presented, and thus their decisions can only yield a binary outcome (correct or incorrect), which can mask the fact that some contours are easier to detect than others.

To address this challenge, we use a signal-detection framework to compare model and human decisions. The idea here is to estimate, separately for both models and humans, what the contour signal strength is for each individual trial. For models, the estimate of the contour signal strength was derived by comparing the activation on the "contour present" node for a contour-present image and its matched contour-absent image from the stimulus dataset used in the behavioral experiment. The stronger the "contour present" response is for a contour-present image relative to contour-absent image, the stronger the "contour signal" for that specific image-pair. This is estimated in equation 1 (see signal detection analysis in **Methods and Supplementary Fig. 6**) where *response* is measured in the contour-present node for each image pair.

$$\frac{response_{contour-present\ image} - response_{contour-absent\ image}}{\sqrt{2}} \qquad (1)$$

For humans, percent correct on a 2AFC task can be considered a measure of signal strength (Green & Swets, 1966). Thus, of principal interest is whether the strength of contour signals in the model predict human percent correct for the same image pair.

Fine-tuning to detect low curvatures gives strongest alignment with human behavioral data

Do models fine-tuned to perform contour detection have the same curvature perceptual signatures as humans? Recall the human behavioral data indicated that detecting contours with high global curvature is challenging. We first considered a model that was fine-tuned to perform contour detection over all curvatures spanning a broad range (global curvature parameter β varying broadly as shown in **Fig. 5A**, with the classifier attached to the avgpool layer). Importantly, by doing so, we are testing the capacity of the model to perform contour detection across a broad range of contour curvatures, but this does not guarantee that it will be equally performant across the full range — it's possible that this model will also show the same characteristic dropoff in detection accuracy with curvature that people do. However, at a trial-by-

trial level, this model shows only a weak correlation with human percent correct across the 1000 trials/image-pairs (**Figure 5B** red scatter plot; Pearson's $r = 0.1907$, 95% CI [0.13,0.35], $p < 0.001$). On average, this model fails to capture the characteristic dropoff in contour detection performance as curvature increases (**Figure 5C,** line plot, red vs. dark gray lines). Instead, it was capable of detecting curvatures better than humans, particularly for the more extreme curvatures. This correspondence failure also occurs in models that were fine-tuned from other layers of Alexnet (**Supplementary Fig. 7**).

What would it take for the model to show a more human-like pattern of contour detection? The human behavioral data indicate that detecting contours with high global curvature is challenging, and is best for straight lines, with a gradual fall off with curvature. Thus, we hypothesized that if we limit the model's fine-tuning training to only straight line contours, we might naturally recapitulate this human-like characteristic fall off, and potentially account for the trial-by-trial variation as well. However, it is also possible that the model will over fit to the training curvatures and show overly sharp tuning to that specific training curvature, and fail to capture the gradual tuning/fall off with curvature seen in people. In the following analyses we explored a range of fine-tuning protocols which control the exposure to different degrees of curvature. It's important to note that we are not directly fit the model to human data by training on the human data in any way. Instead, we are exploring a range of possible hypothesis about what curvatures the model system needs to be fine-tuned to, and seeking the fine-tuning protocol which shows the best emergent human-like pattern of successes and failures.

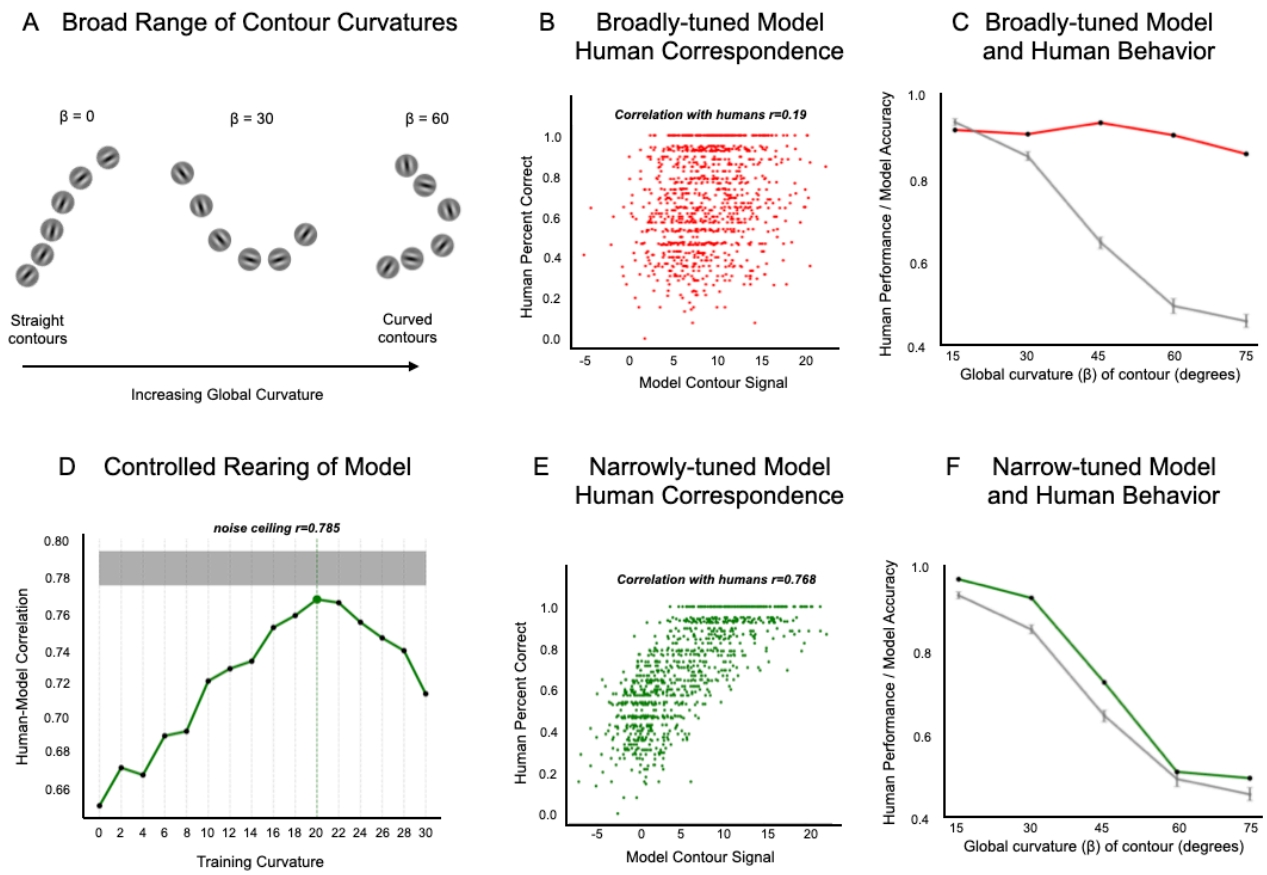

***Fig. 5. Model and Human Behavioral Correspondence for Contour Integration*. (A)** Contour stimuli containing global curvatures (β) spanning a broad range. **(B)** Scatter-plot depicting the correlation between the broadly-tuned model's contour signal strength and human percent correct across trials, showing weak correspondence (Pearson's r = 0.1907). **(C)** Line plot illustrating the broadly-tuned model's performance against human performance for different global curvature levels, highlighting the model's insensitivity to increasing curvature. **(D)** Line plot illustrating the correlation of models, that were trained on curvatures within a specific narrow range (resulting in narrowly-tuned models), with humans, peaking at β = 20° and approaching noise ceiling (r=0.785). **(E)** Scatter-plot depicting the correlation between the narrowly-tuned model's contour signal strength and human percent correct across trials, showing strong correspondence (Pearson's r = 0.768). **(F)** Line plot illustrating the narrowly-tuned model's performance against human performance for different global curvature levels, highlighting the human-like sensitivity to curvature.

Starting with a baseline model trained on Imagenet, we created 16 new model variations that were fine-tuned to detect contours, each at a single curvature (β) level, from 0° (perfectly straight) to 30°, in 2° increments. Note in all these models, the contour classifier was attached to the avgpool layer. **Fig. 5D** plots how strongly each of these fine-tuned models correlated with human trial-by-trial responses. Contrary to our expectations that the model trained on purely

straight edges would best recapitulate the human data, we instead find that models fine-tuned to detect curvatures around β=20° showed the highest agreement (r = 0.77, 95% CI [0.7413, 0.7923], p < 0.001). Indeed, the level of correspondence approached the human data noise ceiling (r = 0.786). **Figure 5E** shows this trial-by-trial scatter plot, with the 20°-fine-tuned model's estimate of contour signal strength on the x-axis, and the human's percent correct for that the trial on the y-axis. **Figure 5F**, shows that, aggregating across trials with the same curvature, the model shows the same characteristic dropoff in performance as in the human data. Finally, we also find that fine-tuning on higher curvatures (45°, 60°, or 75°) results in lower correlations with humans (r = 0.2, r = -0.49, r = -0.63), as detailed in **Supplementary Fig. 8**. These results also generalize to linear-classifier probes trained on other layers (see **Supplementary Fig. 9**), consistently showing peak human correspondence when fine-tuned with contours whose curvature is around 20 degrees (± 4 degrees).

Note that this emergent fit is by no means guaranteed. It quite reasonably could have been the case that fine-tuning a model on only one curvature level would show little generalization to other curvatures, with a steep fall off in accuracy for out-of-distribution curvatures (as is often the case with deep net fine-tuning). For example, the model trained on 20° curvature might show poor contour detection accuracy at 0°, 10°, 15°, and 25° etc. In this case we might have had to explore fine-tuning over ranges of curvatures, perhaps with different frequencies for each level. However, our analyses revealed a relatively simple and strong finding: fine-tuning solely on 20 degrees of contour curvature ends up capturing human-like contour detection variation across all images of varying curvatures, at near noise-ceiling of the highly reliably human data we collected. And, as result, the model also demonstrates the right slope in the average performance across different curvature levels.

Broadly, this emergent match for the 20° fine-tuned model suggests that human contour integration mechanisms might actually be optimized for this relatively narrow band of gradually curving contours, where feedforward CNNs can emulate these signatures of human perception when their contour-integration mechanisms are biased towards encoding relatively gradual curves.

Assessing Emergent Contour Detection Capabilities in Other Network Architectures

In an auxiliary analyses, we also explored how other standard and potentially more powerful vision model architectures would fare on the contour detection task. Specifically we focused ResNet-50 (He at al., 2016) and a Vision Transformer (Dosovitskiy et al., 2021; Steiner et al., 2021; Wightman , 2019), testing whether these pre-trained models (without additional fine-tuning) exhibit better emergent contour detection accuracy than the standard Alexnet model we used. ResNet-50 was selected as the current industry standard deep CNN architecture, with more layers and residual (skip forward) connections, and has superior ImageNet performance than the classic Alexnet architecture. The Vision Transformer was chosen because it has within-layer self-attention mechanisms, learning how to associate information across different patches of the image, which could facilitate lateral information flow and potentially enhance contour detection.

To carry out these analyses, we kept the backbone features fixed, and trained linear probes to test whether contour presence could be read-out from each of the hierarchical layer blocks (relu

blocks of the Resnet50; and the class token representation for vision transformer). To train the linear readouts, we showed each model contours with curvatures spanning a broad range, as we did in our first analysis with the Alexnet architecture, to assess the degree to which detecting an extended contour is possible. For vision transformers, note that we resized our contour displays to fit the vision transformer front end size of 224 x 224, which meant that each input patch had access to an individual gabor element from the display, plus sometimes a partial view of another gabor element depending on the jitter. (**see Methods for additional details**).

Do these models, in their pre-trained state, inherently support the detection of extended contours based on their learned representations? Performance accuracy for contour detection across intermediate layers of these models is shown in **Supplementary Fig. 10**. ResNet-50's contour detection performance is comparable to that of the pre-trained AlexNet model, with a maximum accuracy of approximately 69%, but remains far below the fine-tuned AlexNet model shown in blue. This result suggests that even the modern DCNN pre-trained models lack the strong inductive biases required to integrate local information for detecting extended contours. The Vision Transformer, starting at chance level in the earlier layers, shows a gradual but marginal increase in performance through the later transformer blocks, reaching a maximum accuracy of 58%. This finding is important because, despite containing explicit associative processing between image patches, pre-trained Vision Transformers did not exhibit a significantly enhanced capacity for contour detection. This limited capacity could also be attributed to factors such as insufficient training data or suboptimal hyperparameters, like the optimizer or learning rate scheduler, while training the linear readouts. However, since this experiment did not involve fine-tuning, the same hyperparameters were used for linear readouts across all models, simply testing whether relevant information for contour detection was present in the intermediate layers of each network. Taken together, these findings suggest that, while modern pre-trained architectures such as ResNet-50 and Vision Transformers exhibit some capacity for contour detection, they still lack the necessary inductive biases to show emergent contour integration capability.

DISCUSSION

Deep neural networks provide extensive control over factors crucial to learning visual representations, including the visual diet, neural architecture, learning objectives, and regularization pressures, and are particularly well-suited to making computational plausibility arguments — i.e., understanding what factors are necessary and sufficient for different kinds of representations to be learned. As such, these models provide a powerful experimental platform for exploring theories of visual processing, including investigating the mechanisms underlying perceptual grouping and the formation of high-level holistic representations. In the present study, we focused on the contour integration capabilities of deep convolutional neural networks, and found that these purely feedforward systems can support human-like contour detection, provided they have progressively-increasing receptive fields, and a bias toward relatively smooth/straight contours. Moreover, we found that these models performed contour detection using local alignment of features to detect chains of local elements and integrate them into a higher order structure — a contour. Thus, despite being supported by units with large receptive fields, the integration still operates over fine-grained local details, highlighting that large-receptive fields

need-not encode coarse representations. Taken together, these results suggest that feed-forward processing alone can implement sophisticated grouping operations like contour integration.

It is important to emphasize that these results do not contradict or undermine prior work suggesting that lateral connections and feedback play an important role in contour integration (Field et al., 1993; Keemink & Rossum, 2016, Li, 1998; Piëch et al., 2013; Boutin et al., 2021). Instead, they show that lateral connections and feedback mechanisms are not necessary for contour integration, that purely feedforward processing along a hierarchy with progressively increasing receptive fields is sufficient. As an additional point of clarification, these DCNN models do use backpropagation during training to adjust their weights, which might be considered as a kind of feedback. However, here we note that, once trained, the actual processing of the display images, and flow of information en route to a contour detection decision remains strictly feedforward. This means that, once the network is trained, all computations proceed in one direction, from earlier to later layers, without any feedback occurring between them, and this process is sufficient to support successful contour detection in these models.

At the same time, we find that CNN models pre-trained on ImageNet classification alone was not sufficient; these model did not show an emergent match with human behavior. And, this pattern of data is in broad agreement with prior work (Jacob et al., 2021; Bowers et al., 2022; Biscione & Bowers, 2023), Instead, our analyses revealed that fine-tuning with an inductive bias towards gradual curvature was required to see more human-like contour detection decisions. Thus, these results provide a clear direction for future research — to discover new inductive biases that can be built into DNN models so that human-like contour detection emerges without any fine-tuning to the contour task. For example, perhaps more human-like contour detection will naturally emerge in in models trained on even more general visual objective (e.g., self-supervised learning or predictive coding over more 3-dimensional object views.).Thus, our primary claim here is that it is possible for a purely feed-forward hierarchy to perform contour integration in a human-like fashion, but that the models are lacking some key ingredients to show a fully *emergent behavior* that aligns with humans, without any task-based fine-tuning.

A Synergistic View of Feed-Forward, Lateral, & Recurrent Mechanisms

We find evidence that a purely feedforward system can implement human-like contour detection, implementing contour-integration mechanisms that are sensitive to local-collinearity cues and stitching them together across a hierarchy of progressively growing receptive fields. We believe this is an important discovery because the role of the feedforward hierarchy in contour integration and perceptual grouping has been relatively understudied in prior work, which has assumed that relatively low-level mechanisms perform contour integration by amplifying responses to local elements that contain colinear neighboring elements (see **Fig. 6A** for illustration). For instance, models with lateral/top-down connections and/or recurrent units can capture human perception of contours (Field et al., 1993; Keemink & Rossum, 2016), as well as neurophysiological aspects of contour detection (Li, 1998; Piëch et al., 2013; Boutin et al., 2021), including within natural images (Linsley et al., 2018; Linsley et al., 2020; Veerabadran & de Sa., 2019; Khan et al., 2020). However, one conceptual issue with early-stage contour-integration mechanisms is that they serve primarily to boost responses to features that are part of a contour, but those responses are still localized and ambiguous — e.g., a particular neuron

sensitive to a particular orientation can give a strong response due to a number of factors, from increased contrast of the local features, better alignment to the preferred orientation, and/or the presence of colinear elements outside of the classical feed-forward receptive field. Even taken together as a population, these localized responses at best can provide an implicit representation of the contour.

Our current feedforward model suggests an alternative mechanism, where binding and grouping are interwoven within a multi-level hierarchical framework, driven primarily by feed-forward mechanisms which represent the "integrated whole" at the top of the hierarchy. Each convolutional layer in this framework represents similarities and differences over space, with subsequent layers incorporating aspects of non-transitive grouping (Geisler & Super, 2000). This feedforward computation starts by isolating potential candidates of colinear elements over smaller receptive fields and then by processing successively over larger and larger receptive fields chips away accidental collinear elements till it narrows down on the complete contour (see **Fig. 6B** for illustration). It might be argued that the feedforward model's sensitivity is restricted to contours of particular lengths—a generalization recurrent models handle with ease (Linsley et al., 2020). Yet, this distinction reflects the fundamental mechanisms at play rather than the measure of the model's human-like capabilities. We propose an integrated framework in which the early lateral/recurrent operations serve to amplify local responses to elements that are likely components of extended contours, but that feedforward processing plays a primary role in integrating these local features into an explicit representation of the extended contour. On this account, it is the tuning properties of these high-level integration units (and not the previously proposed early-stage mechanisms) that underlies human contour perception by providing an explicit representation of the integrated "whole" beyond the simultaneous localized representation of its elemental "parts" — i.e., making the whole greater than the sum of its parts (Wertheimer, 1938).

Limitations of the approach

We examined contour integration using a relatively constrained stimulus set, specifically Gabor elements with specific parameters and contours comprised of a fixed number of local elements. This approach enabled us to isolate contour-integration mechanisms and provided a high-powered comparison with human psychophysical data. A promising future direction would be to explore how the integration units respond to systematic variations in other stimulus dimensions, such as frequency, scale, phase and symmetry (Dakin & Hess, 1998; Dakin & Hess, 1999; Hess & Dakin, 1997; Field et al., 2000), but in our view this step would be most fruitful following advances in deep neural networks that lead to emergent alignment with humans, as noted above and discussed in detail below. Our investigation also focused on two-dimensional images, but research by Hess & Field, 1995 suggests that the visual system also engages in contour integration across different depth planes. Exploring these dynamics with deep neural network frameworks, particularly those that are depth-aware (Wang & Neumann, 2018; Oquab et al., 2023), could offer new insights into how the visual system transitions from an image-centric representation to one that is more object-centric.

Finally, in the current study we used a linear classifier probe trained directly on contour detection to assess the contour-integration capacity for different model layers and to enable direct comparisons between model and human performance. An alternative approach would be to assess the tuning properties of individual units within each model layer, and to draw parallels between those tuning properties and neurophysiology or behavioral data. For instance, future research could isolate artificial units sensitive to the alignment of local elements and test whether these units exhibit a preference for gradual curvatures in a way that parallels human contour detection sensitivity. We see this approach as complementary to the work presented here, and provide resources for generating stimuli that can be useful for such *artiphysiology* research.

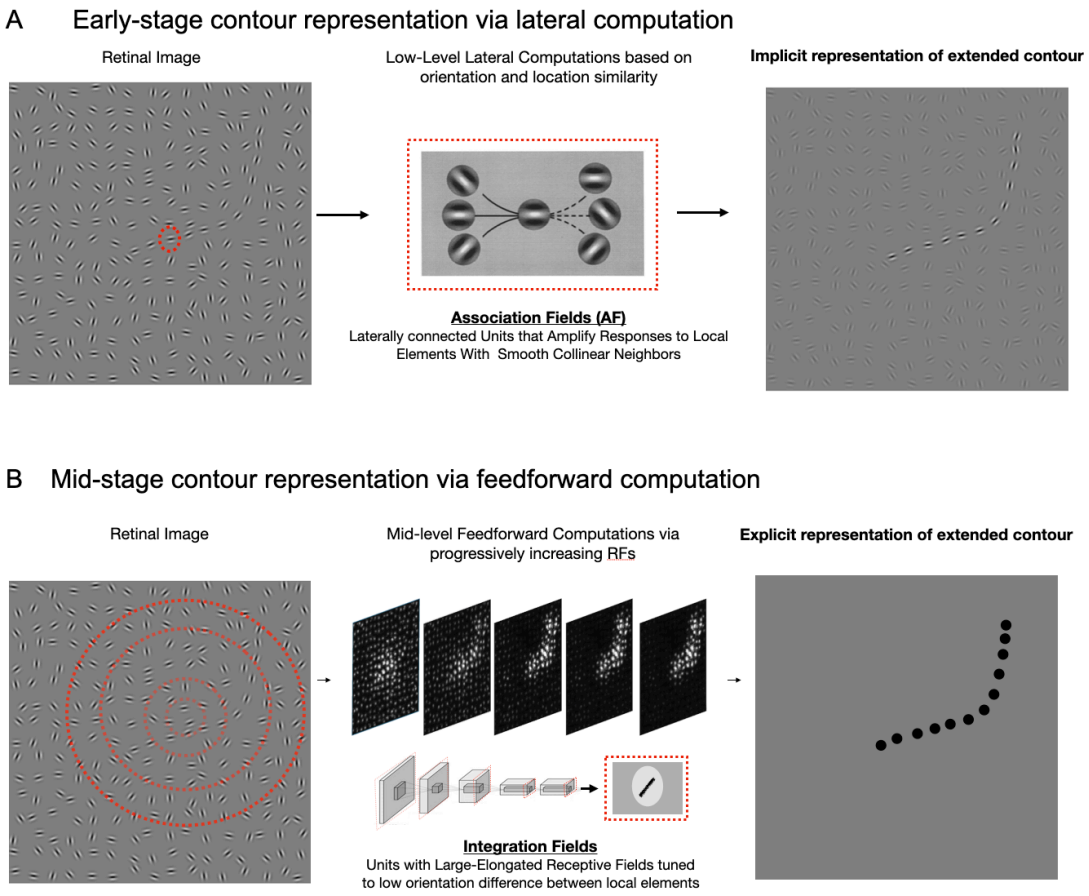

Fig. 6. Computational Mechanisms underlying Human Contour Integration. (A) *Low-level computations amplify responses to local elements that are part of a contour in the retinal image. This is facilitated by lateral connections between units with collinear tunings conceptualized as Association Fields* ***(B****) Mid-level feedforward computations focus on identifying potential candidates for an extended contour in the retinal image. This is facilitated by units with progressively increasing receptive fields (RFs) that are tuned to low orientation differences, allowing for the integration of local features into coherent extended contours.*

Towards Emergent Human-like Behavior

An ambitious goal for vision science is to develop a deep learning framework that, through exposure to a natural visual diet and a biologically inspired architecture and task objective, would inherently exhibit human-like perceptual organization. Such a system would naturally demonstrate human-like perceptual phenomena, including human-like contour integration, when subjected to the same psychophysical tasks as humans, and would not have to be fine-tuned on the probe stimuli. Although training with datasets like Imagenet can enhance these models' contour integration abilities compared to their untrained counterparts, it remains unclear what additional inductive biases are needed for these models to show, without further fine-tuning, the characteristic human degradation with increased contour curvature.

Why are humans more sensitive to relatively straight contours? One possibility is that human perception is optimized for curvatures that are most prevalent in their visual diet, and that relatively straight contours are more prevalent in natural images. Indeed, research indicates a bias in natural image distributions toward relatively straight edges, often with orientation differences peaking near 20 degrees between adjacent elements (Elder & Goldberg, 2002; Geisler et al., 2001) — a value that is intriguingly comparable to the inductive bias that provided the optimal alignment between our fine-tuned models and human contour detection accuracy. On this account, networks that learn over images with this natural image bias may inherently become tuned to relatively straight edges. While ImageNet contains about 1.3 million images, it is unclear whether they are adequately representative of natural image statistics needed to induce this bias. Training networks on more extensive and varied naturalistic datasets (Yalniz et al., 2019) could potentially enable them to develop more nuanced representations of extended contours, more closely approximating human sensitivity to curvature variations. Additionally, using ego-centric datasets like the developmentally realistic SAY-Cam (Sullivan et al., 2021) or EGO4D dataset (Grauman et al., 2022), or high-resolution input images that can seemingly increase shape reliance without data augmentation (Muller et al., 2023) may further enhance the network's ability to learn rich, realistic visual representations.

Nonetheless, we speculate that statistical regularities in the natural visual diet must be combined with strong pressures to form structured representations for models to demonstrate human-like contour integration. Standard computer vision tasks, such as object recognition, or self-supervised objectives, do not appear to require such structured, shape-based representations (Geirhos et al., 2020, 2021), resulting in deep neural network models that prioritize local color and texture information over extended contours and shapes (Baker et al., 2018b; Geirhos et al., 2018, 2021), and generally lack sensitivity to the spatial configurations between object parts (Malhotra et al., 2023; Baker et al., 2023). While some efforts have been made to enhance shape-representation in deep neural network models (Geirhos et al., 2018), improvements in shape-bias have not been accompanied by high-quality, human-like shape-representations (Doshi et al., 2024). We propose that stronger objective pressures to form geometric shape representations (Sablé et al., 2021) are needed for the emergence of human-like contour integration in these models.

In addition to changes in visual diet or task objective, it is also possible that these networks require modified architectures that are better suited towards contour integration. While out-of-

the-box pretrained Vision Transformer did not exhibit enhanced contour integration capacity compared to DCNNs, its self-attention mechanism may still hold promise for future explorations of contour processing, particularly if combined with other mechanisms. One promising direction can be to incorporate ganglion and LGN-like processing in the early layers of the network. These cells are adept at encoding crucial edge and contrast information via their distinct center-surround receptive fields (Schiller & Tehovnik., 2015). Emulating these characteristics can potentially enhance nuanced detection of orientation and alignment changes, essential for detecting extended contours. Consistent with this possibility, previous research has shown that such network modifications lead to better shape processing without necessitating changes in the training visual diet composed of natural image statistics (Malhotra et al., 2020; Muller et al., 2023). Moreover, Lindsey et al. (2019) demonstrated that an architectural bottleneck in early stages, which simulates information compression leaving the eye via the optic nerve, can naturally develop these early characteristic receptive fields. Finally, there are lateral and recurrent connectivity motifs, such as association fields (Field et al., 1993), that could serve to amplify relatively straight contours causing the feed-forward integrative mechanisms themselves to be biased towards straight or gradually curved contours. Such structural changes in neural networks might be required for DNN models to show alignment with human visual processing that emerges without fine-tuning.

Conclusion

While our findings do not in any way discount the important role of early lateral or recurrent mechanisms in forming grouped representations in human vision, they do suggest that a feed-forward visual hierarchy with progressively increasing receptive field sizes is capable of playing a significant role in performing fine-grained contour integration and grouping operations. We propose that there is a synergistic interplay between the early local/recurrent mechanisms focused on in prior work, and the feedforward integrative mechanisms discovered in the current work. Interactions between these mechanisms may result in more sophisticated mechanisms of perceptual grouping than previously explored, or alternatively it maybe be the case that lateral/current operations serve primarily to “distill” the inputs to feed-forward integrative mechanisms. Deep neural network models offer a powerful testbed for instantiating and examining such complex processes, and using these tools to focus on the interplay between levels of processing promises deeper insights into the formation of holistic perceptual representations in both biological and artificial systems.

METHODS

Stimuli Generation

A. Gabor Elements

We used gabor elements as the fundamental local elements in the contour displays. These elements are sinusoidal gratings modulated by a gaussian envelope. The sinusoidal wave component represents the grating pattern, created by the interaction of the spatial wavelength, orientation, and phase offset, while the gaussian envelope, a bell-shaped curve that modulates the visibility of the sinusoidal wave making it fade out gradually from the center to the edges, is parametrized by its standard deviation. The resulting Gabor element combines these two

components, encapsulating spatial frequency and orientation information within a confined spatial region, making them ideal for investigating visual mechanisms due to their compatibility with the receptive fields of neurons in the visual cortex. These elements are characterized by the following equation:

A spatial wavelength (λ) of 8 pixels was chosen. This value dictates the wavelength of the sinusoidal wave – i.e., the number of stripes. A smaller λ value results in more stripes, whereas a larger λ value results in fewer stripes. The sinusoidal wave has a phase offset of -90 degrees. This parameter affects the positional shift between the dark and light stripes in the wave, hence controlling the symmetry of the gabor pattern. The standard deviation, which controls the spread of the gaussian envelope and determines how quickly the sinusoidal wave will fade out in the gabor pattern, was set to 4 pixels. Each gabor element is embedded in an image patch of size 28 x 28 pixels. The orientation (θ) of the Gabor elements is variably controlled and generated algorithmically (see the **Contour Path Algorithm**). The orientation decides the angle at which the stripes in the Gabor pattern are tilted. The orientation is particularly crucial as it determines the angle of the sinusoidal wave within the Gabor patch. The contrast was maintained so as to maximally ensure optimal visibility and effectuality in probing the visual system.

B. Contour Path Algorithm

The contour path generation follows the algorithm initially proposed by Field et al., 1993. The image, sized at 512x512 pixels, is divided into a 16x16 grid, resulting in cells of 32x32 pixels each. Hence the image grid contains 256 total cells, and each cell contains a gabor element. **See Supplementary Fig. 1A-bottom**.

A starting point is randomly determined, situated 64 pixels from the image's center. A vector is projected from the starting point towards the image's center, adjusted by an angle of $\pm \beta$. The subsequent anchor point is located at a distance, D, from the starting point, where D is set to 32 pixels. A Gabor element is designated to be placed between these two points, marking the specific grid as occupied. The orientation of the Gabor element, denoted as α, will be described in the next section. If this grid is already pre-occupied, the vector's length is incremented by ΔD, which is set to 25% of D (8 pixels). Using the previously identified anchor point as a reference, the vector is once again adjusted by $\pm \beta$. An offset of $\Delta\beta$, which is a random integer between -10 and 10 degrees sampled from a uniform distribution, is added to these adjustments. This process is repeated to determine the next anchor points and Gabor element positions. As the display is comprised of 12 elements, this procedure is carried out until the 13th anchor point is identified. The β value is adjusted experimentally: a lower β yields straighter contours, while a higher β produces more curved paths. **Supplementary Figure 1A** illustrates the parameters involved in this path generation process. The figure below it depicts the placement of anchor points and the targeted locations for gabor elements.

C. Contour Stimuli

In our study, we generated a well-controlled pair of images as the contour stimuli. Each pair consists of two images: one with an extended contour ('contour-present' image) and the other without the extended contour ('contour-absent image). The distinction between these images lies in the alignment principles governing the placement of Gabor elements on the contour path. To elucidate, each image embeds two sets of Gabor elements—contour and background elements.

For the ‘contour-present’ image, Gabor elements are positioned based on the locations determined by the path generation algorithm. These Gabor patches, measuring 28x28 pixels, are situated within grids of size 32x32 pixels. We can adjust their orientation (characterized by the alpha parameter) to manipulate the degree of local alignment of these gabor elements with respect to the overall contour path. Conversely, in the ‘contour-absent’ image, gabor patches are placed in identical locations, but their local orientation (alpha parameter) is randomized. This procedure is depicted in **Supplementary Figure 1B (top)**. Following the placement of contour elements, we place background gabor elements (each 28x28 pixels in size) in the grid cells unoccupied by the foreground elements (the path generation algorithm marked the grids containing the foreground elements as being occupied). These patches are approximately centered within their respective grid cells. However, to introduce variability and prevent uniformity, a random jitter within one-third of the gabor image size (i.e. ~ 9 pixels) was added to their positions. Their local orientation (alpha) is also random. However, the position and orientation of each background gabor element remains identical between the ‘contour-present’ and ‘contour-absent’ images, as illustrated in **Supplementary Figure 1B (bottom**). Hence each image is divided into 256 cells, of which 12 of the cells are filled with a gabor contour element and the remaining 244 cells contain the gabor background element. **Supplementary Figure 1C** displays a well-controlled representative pair of a contour-present and a contour-absent image.

D. Contour and Background Masks

Binary masks are made by thresholding the gaussians at the specific location where a gabor element was places. This is done separately for the contour and background elements giving 2 masks. The gaussians vary in a range of 0-255. A threshold of 140 was chosen to create the masks.

<u>Contour-detection Linear Probing</u>

A. Contour Readout

We examined the Alexnet's contour integration capability by training a separate linear classifier on outputs from each of its 21 layers, as depicted in Fig. 2A. Each layer's output was probed to quantify contour-detection capacity. To train the linear classifier, we first flatten the outputs from a layer and pass it as input to the readout classifier. This classifier linearly transforms the input and outputs if a contour is present or not, represented as two values in a one-hot encoding format. The readout classifier has the weight parameter which is of shape [size of output from a layer , 2] and bias terms for each of the two output nodes. For the random and pretrained models (gray and blue points in **Fig. 2A**), we kept the backbone weights frozen—randomized for the former and pretrained (for object recognition on Imagenet) for the latter—during the contour readout training phase. Conversely, finetuned models (blue points in **Fig. 2A**) were trained end-to-end, starting with pretrained weights at initialization, thus providing an assessment of the network's architectural capacity to support contour detection.

B. Training Stimuli

We constructed a comprehensive dataset (for training purposes) comprising pairs of contour images. Each pair was created for 91 distinct beta values (global curvature) ranging from 0 to 90 degrees, generating 2,500 pairs per beta value. With the alpha value (local orientation) fixed at 0 degrees, this dataset encompassed a total of 455,000 images. The parameters for the gabors and

contour path adhered to the specifics mentioned previously. Specific images were chosen to tailor the 'visual diet' for our contour detection models —those designed to process a broad range of curvatures (shown in **Fig. 2 and 3**) were trained on 4,992 images spanning six beta values (0, 15, 30, 45, 60, and 75 degrees) to ensure uniform distribution. Each beta condition was represented by 832 images, yielding 2,496 pairs of contour stimuli (contour-present and contour-absent images). In contrast, the narrowly-focused models (depicted in Fig. 5) were trained on approximately 5,000 images from a single or a limited range of beta values.

C. Test Stimuli

Our held-out test dataset mirrored the diversity of the training set, featuring 50 pairs per beta value, for all the 91 beta values, totaling 9,100 images. Models trained on a broad spectrum of beta values were evaluated against 600 images distributed equally across these values (0, 15, 30, 45, 60, and 75 degrees). Narrowly-focused models were tested on 50 pairs strictly from their training beta values, aligning the testing to the training approach and ensuring the evaluation's fidelity to the model's training experience.

D. Hyperparameters for training contour readouts

We used stochastic gradient descent (SGD) as the optimizer with a learning rate of 0.0001, momentum of 0.9, and no weight decay to train the readout models. The training loss was computed using a cross-entropy loss function, which is appropriate for this binary classification task. The batch size was set to 8 samples in each batch. We used Pytorch's OnecycleLR as the learning rate scheduler which allowed the learning rate to vary over the course of training. The maximum learning rate was set to 0.0001, with 40% of the total training steps dedicated to increasing the learning rate, followed by a gradual decrease. The scheduler and optimization to train readout models was configured for 100 total epochs, with the learning rate adjusted dynamically based on the training progress.

Model sensitivity to alignment of local elements

A. Saliency Maps via Guided Backpropagation

To compute the saliency maps, we pass an image forward through the model and get the output predictions. This forward pass comprises passing the image though the deep net backbone up to a certain layer and then passing the activations from that layer through the attached readout layer. We then compute the gradient, with respect to the output prediction being "contour present", and backpropagate it through the network. A critical manipulation, however, is to modify this backpropagated gradient, at every Relu layer that is involved in the forward pass, by zeroing out all the negative gradients. This ensures that only positive contributions will finally be interpreted. We apply this modification repeatedly at every intermediate relu layer till the gradient is backpropagated at the pixel-level input. At this stage, the gradients can be visualized as saliency maps, providing us with insights about which parts of the image finally influences the model to predict the presence of a contour.

B. Sensitivity to location and alignment of local contour elements

For every image, we use the binary masks and the saliency map to separate the saliency values into two distributions, one group for the contour elements and another group for the background elements. The difference in the mean of these two distributions can be interpreted as the

sensitivity of the model to the contour gabor patches. This value is bounded between -1.0 and 1.0 where a value of -1.0 means completely attending to the background patch relative to the contour and a value of 1.0 would mean completely attending to the contour relative to the background.

While the difference in means provides a straightforward understanding of model sensitivity towards foreground versus background elements, it lacks depth in capturing the nuances of their respective saliency distributions. More specifically, solely relying on mean differences can be misleading, as it doesn't account for the ranking and variance within these distributions. A mean-centric measure could yield similar values for distributions that are inherently different, potentially obscuring significant variations in sensitivity that are crucial for our analysis. Hence, we used another metric, called A' which is derived from the area under the Receiver Operating Characteristic (ROC) curve, providing a scalar value between 0 and 1 that indicates the probability that a randomly selected contour element's saliency value is ranked higher than a randomly selected background saliency value. In other words, a value of 0 suggests that the distribution of contour saliency values is consistently ranked lower than that of the background values, indicating no sensitivity to the contour elements. Conversely, a value of 1 suggests perfect sensitivity, with contour elements consistently ranked higher than background elements. This measure effectively characterizes the separability and overlap between the contour and background elements' saliency distributions, providing a more granular understanding of the model's attention mechanism without making assumptions about the normality of the underlying distributions.

To measure the model's sensitivity to local alignment of contour elements, we measure the contour location sensitivity separately for image pairs which contain aligned or misaligned contour elements but at the exact same location. The difference of each pair lets us asses how much a model is influenced by the alignment of contour elements when detecting an extended contour. We measure the average difference (along with and the 95% CI of the differences to quantify the sensitivity to alignment of the contour elements.

PinholeNet Models

These networks, akin in architecture to Alexnet, deviate primarily by the removal of maxpool operations within their convolutional layers. Notwithstanding this alteration, the count of channels post each convolutional layer (C1 through C5) remains constant across all iterations of PinholeNets, maintaining at 64, 192, 384, 256, and 256 channels respectively. Each convolutional layer engages in a convolutional operation followed by a relu non-linearity. Where these models diverge is in their kernel size and stride during convolutional operations, rendering their deep net units with a restricted field-of-view, culminating in effective receptive field sizes (in the C5 units) of 11, 17, 31, and 33 pixels for the P11, P17, P31, and P33 models respectively (refer to **Fig. 3A** for clarity). These models are structurally similar to Alexnet but differ primarily in the omission of maxpool operations in their convolutional layers. Despite this change, the number of channels after each convolutional block (from C1 to C5) remains consistent across all PinholeNet versions, with 64, 192, 384, 256, and 256 channels in the C1, C2, C3, C4, and C5 blocks respectively. Each block engages in a convolutional operation followed by a relu non-linearity. The distinct characteristic of these models is the variation in kernel size and stride during convolutions, rendering their deep net units with a restricted field-of-view. This results in

the receptive field sizes of 11, 17, 31, and 33 pixels for deep net units in the fifth convolutional block. (see **Supplementary Fig. 3A** for a visual depiction of the architecture).

Behavioral Experiment

A. Experimental Design

Each participant undertook a 2-IFC (Two-Alternative Forced Choice) contour detection task, as illustrated in **Fig. 4B**. Participants were prompted with a 'Get Ready!' indicator, followed by a blank screen that lasted for 500 ms. Sequentially, two displays were presented, each for 500 ms, interspersed by a blank screen with a duration of 1000 ms. Their task was to discern which of the two displays (Display 1 or Display 2) contained an extended contour by registering their response via keypress. Over the course of the experiment, each participant encountered 200 total trials from one of the 5 file splits (i.e., each trial containing 200 paired contour displays). These trials are counterbalanced across the 5 beta conditions and the 2 displays (i.e. equal number of contour-present images are shown in Display 1 and Display 2). The experiment is segmented into five blocks, with each block presenting 40 counterbalanced trials (spanning all beta values and display conditions, hence 4 trials spanning the 10 total conditions). A resting period followed each block, with participants advancing to subsequent blocks by pressing the spacebar when ready. Each block is further subdivided into two mini-blocks. Both mini-blocks contain a counterbalanced set of trials, and they transition without any intervening breaks. The rationale behind this design is to ensure that while conditions are perfectly counterbalanced throughout the experiment and within each block, their presentation within a block isn't skewed. Specifically, without the use of these mini-blocks, there could be unintended randomness where certain conditions predominantly appear at the beginning or at the end of a trial block.

B. Exclusion Criteria

Initially, participants were recursively removed until the remaining participants' overall accuracy (or mean percent correct) for all the images that the participants viewed lay within 2.5 times the standard deviation. This step led to the exclusion of 11 participants. **Supplementary Fig. 4A-top** shows the mean overall accuracy before and after the exclusion. However, simply aligning with overall accuracy might not guarantee that individual participants reflect the general trend observed across all beta conditions. To address this, for each participant, we calculated their mean accuracy across all beta conditions, resulting in a five-dimensional vector (each representing one of the 5 beta conditions). We then assessed the correlation between an individual's vector and the average of the vectors across the remaining participants (excluding the target participant) to measure the degree to which a particular participants' performance trended with the group. Any participant whose correlation values lay outside 2.5 times the standard deviation of these correlation values was recursively excluded, leading to the further exclusion of 19 participants. **Supplementary Fig. 4A-bottom** shows the mean overall accuracy before and after the exclusion This analysis resulted in the exclusion of a total 30 participants from the initial set of 108 participants, giving a set of 78 participants who met both our overall accuracy and beta-condition trend criteria. The participants were recruited using Prolific (www.prolific.com).

Contour Signal Strength in Humans and Models

To quantify the contour signal in humans and models, we employed a signal detection theory (SDT) framework which allowed us to discern the strength and reliability of contour signals across a range of image displays.

A. Human Percent Correct

For human participants, the signal detection framework was operationalized by calculating the percent correct across trials. Each participant was subjected to a 2-interval forced-choice (2-IFC) task, wherein they were presented with pairs of displays (one contour-present and one contour-absent) and asked to identify the display containing the extended contour. Their performance was recorded as a binary correct/incorrect outcome for each trial. The human signal strength, therefore, was represented by the percentage of correct responses, with higher percentages reflecting greater sensitivity to contour presence. These individual performance metrics were aggregated to form a group-level performance profile against which the model's signal strength could be compared.

B. Model Contour Signal

The model's signal strength was determined by assessing the activation levels at the "contour present" node within the model's architecture. We measured the activation responses of the "contour present" node when processing image pairs from the psychophysics dataset. For each pair — one image with a contour (contour-present) and one without (contour-absent) — we calculated the difference in activation levels at the "contour present" node. This node's response provides an index of the model's confidence in the presence of a contour within the image. The signal strength for each image pair was evaluated by plotting these differential activations on a 2D scatter plot, where the x-axis represented activations for contour-present images, and the y-axis for contour-absent images. The critical measure of signal strength is the distance of each point from the identity line (diagonal) on this plot. Points farther from the diagonal represent a greater discrepancy between the model's responses to contour-present versus contour-absent images, thus indicating a stronger contour signal. This distance-from-diagonal metric effectively serves as a signal-to-noise ratio, capturing the model's ability to discriminate between images with and without contours (see **Supplementary Fig. 6**).

Assessing other Network Architectures

A. ***Pre-trained Models***

We used features from two pre-trained architectures: ResNet-50 and a Vision Transformer. The ResNet-50 model was adapted from PyTorch's torchvision library (Paszke et al., 2019) and was pre-trained on ImageNet for object categorization. The Vision Transformer (Dosovitskiy et al., 2021) was adapted from the timm library (Wightman, 2019) and was also pre-trained on ImageNet for object categorization, but with additional augmentation and regularization techniques (Steiner et al., 2021).

B. Model Input and Preprocessing

For the ResNet-50 model, we retained the original size of the contour displays (512 x 512). The adaptive average pooling layer in ResNet-50 allows for flexibility in input resolution, enabling the use of these larger images even though ResNet-50 was originally trained on 224 x 224

images. The images were normalized using the mean ([0.485, 0.456, 0.406]) and standard deviation ([0.229, 0.224, 0.225]) of the ImageNet dataset for each RGB channel.

For the Vision Transformer, we had to resize the images to meet the architectural constraints of the model, which require an input of 224 x 224, divided into 196 patches (14 x 14 grid), with each patch being 16 x 16 pixels. To achieve this, we first resized the 512 x 512 contour images to 224 x 224 pixels using bicubic interpolation, which ensures smooth scaling by considering the nearest 16 pixels during resizing. This method preserves the details of the Gabor elements while maintaining smooth transitions in pixel intensity. The images were normalized using a mean of [0.5, 0.5, 0.5] and a standard deviation of [0.5, 0.5, 0.5] for each channel (RGB). This was done to maintain consistency with the original pre-processing used during the model's pre-training. This resizing process reduced the Gabor elements from 28 x 28 pixels (originally embedded within a 16 x 16 grid in the 512 x 512 pixel image) to approximately 12 x 12 pixels (in the 224 x 224 pixel image broken down into 14 x 14 patches), resulting them to approximately fall within individual patches processed by the Vision Transformer.

C. Training Linear Readout for Contour Detection

For both ResNet-50 and the Vision Transformer, we froze the pre-trained backbone features and trained linear probes to determine whether contour presence could be detected from intermediate layers. For ResNet-50, the output of each ReLU block and the final pooling layer was used to train the linear probes, while for the Vision Transformer, the linear probes were trained using the class token representation (containing 768 features) from each of the 12 transformer blocks.

The hyperparameters and contour "diet" used for training the linear readouts were identical to those used for the AlexNet model layers. The task involved identifying extended contours formed by Gabor elements embedded within a grid of randomly oriented Gabor elements. For each model, accuracy in detecting contour presence was measured by training linear readouts at each layer or block. By keeping the backbone features fixed and only updating the weights in the linear readout layer, we were able to assess the extent to which contour detection was supported by the learned representations in the intermediate layers of these pre-trained models.

Acknowledgments

We would like to thank lab members of the Vision Science Lab for their helpful feedback and support during the writing process. Special thanks to Spandan Madan for insightful comments on the manuscript.

Data and materials availability

All data, training code, analysis code, and figure-plotting code is available on Github: https://github.com/feziodoshi/dnn_contour_integration.

Funding

Kempner Graduate Fellowship to FRD
NSF PAC COMP-COG 1946308 to GAA
NSF CAREER: BCS-1942438 to TK

Supporting Information

Supplementary information file, including Figs 1-10.

REFERENCES

1. Alvarez, G., & Konkle, T. (2022). Isolating Global Form Processing Using Shape Metamers. *Journal of Vision*, *22*(14), 4082-4082.
2. Baker, N., Erlikhman, G., Kellman, P. J., & Lu, H. (2018a). Deep Convolutional Networks do not Perceive Illusory Contours. In *CogSci*.
3. Baker, N., Lu, H., Erlikhman, G., & Kellman, P. J. (2018). Deep convolutional networks do not classify based on global object shape. *PloS computational biology*, *14*(12), e1006613.
4. Baker, N., Garrigan, P., Phillips, A., & Kellman, P. J. (2023). Configural relations in humans and deep convolutional neural networks. Frontiers in Artificial Intelligence, 5, 961595.
5. Battleday, R. M., Peterson, J. C., & Griffiths, T. L. (2020). Capturing human categorization of natural images by combining deep networks and cognitive models. *Nature communications*, *11*(1), 5418.
6. Biederman, I. (1987). Recognition-by-components: a theory of human image understanding. *Psychological review*, *94*(2), 115.
7. Biederman, I., & Ju, G. (1988). Surface versus edge-based determinants of visual recognition. *Cognitive psychology*, *20*(1), 38-64.
8. Biscione, V., & Bowers, J. S. (2023). Mixed evidence for gestalt grouping in deep neural networks. *Computational Brain & Behavior*, *6*(3), 438-456.
9. Bowers, J. S., Malhotra, G., Dujmović, M., Montero, M. L., Tsvetkov, C., Biscione, V., … & Blything, R. (2022). Deep problems with neural network models of human vision. *Behavioral and Brain Sciences*, 1-74.
10. Boutin, V., Franciosini, A., Chavane, F., Ruffier, F., & Perrinet, L. (2021). Sparse deep predictive coding captures contour integration capabilities of the early visual system. PLoS computational biology, 17(1), e1008629.
11. Brendel, W., & Bethge, M. (2019). Approximating cnns with bag-of-local-features models works surprisingly well on imagenet. *arXiv preprint arXiv:1904.00760*.
12. Dakin, S. C., & Hess, R. F. (1998). Spatial-frequency tuning of visual contour integration. *JOSA A*, *15*(6), 1486-1499.
13. Dakin, S. C., Hess, R. F., & Vision, S. (1999). Contour integration and scale combination processes in visual edge detection. *Spatial vision*, *12*(3), 309-328.
14. Doshi, F. R. & Konkle T., Alvarez, G.A. (2024). Quantifying the Quality of Shape and Texture Representations in Deep Neural Network Models. In Vision Science Society, 2024.
15. Dosovitskiy, A. (2020). An image is worth 16x16 words: Transformers for image recognition at scale. arXiv preprint arXiv:2010.11929.
16. Elder, J. H., & Goldberg, R. M. (2002). Ecological statistics of Gestalt laws for the perceptual organization of contours. *Journal of Vision*, *2*(4), 5-5.
17. Elder, J. H., Krupnik, A., & Johnston, L. A. (2003). Contour grouping with prior models. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *25*(6), 661-674.

18. Ellis, W. D. (2013). *A source book of Gestalt psychology*. Routledge.
19. Feinman, R., & Lake, B. M. (2018). Learning inductive biases with simple neural networks. *arXiv preprint arXiv:1802.02745*.
20. Field, D. J. (1987). Relations between the statistics of natural images and the response properties of cortical cells. *Josa a*, *4*(12), 2379-2394.
21. Field, D. J. (1993). Scale-invariance and self-similar 'wavelet'transforms: an analysis of natural scenes and mammalian visual systems. *Wavelets, fractals, and Fourier transforms*, 151-193.
22. Field, D. J., Hayes, A., & Hess, R. F. (1993). Contour integration by the human visual system: evidence for a local "association field". *Vision research*, *33*(2), 173-193.
23. Field, D. J., Hayes, A., & Hess, R. F. (2000). The roles of polarity and symmetry in the perceptual grouping of contour fragments. *Spatial vision*, *13*(1), 51-66
24. Geirhos, R., Narayanappa, K., Mitzkus, B., Bethge, M., Wichmann, F. A., & Brendel, W. (2020). On the surprising similarities between supervised and self-supervised models. arXiv preprint arXiv:2010.08377.
25. Geirhos, R., Narayanappa, K., Mitzkus, B., Thieringer, T., Bethge, M., Wichmann, F. A., & Brendel, W. (2021). Partial success in closing the gap between human and machine vision. *Advances in Neural Information Processing Systems*, *34*, 23885-23899.
26. Geirhos, R., Rubisch, P., Michaelis, C., Bethge, M., Wichmann, F. A., & Brendel, W. (2018). ImageNet-trained CNNs are biased towards texture; increasing shape bias improves accuracy and robustness. *arXiv preprint arXiv:1811.12231*.
27. Geirhos, R., Temme, C. R., Rauber, J., Schütt, H. H., Bethge, M., & Wichmann, F. A. (2018). Generalisation in humans and deep neural networks. *Advances in neural information processing systems*, *31*.
28. Geisler, W. S., & Super, B. J. (2000). Perceptual organization of two-dimensional patterns. *Psychological Review*, *107*(4), 677.
29. Geisler, W. S., Perry, J. S., Super, B. J., & Gallogly, D. P. (2001). Edge co-occurrence in natural images predicts contour grouping performance. *Vision research*, *41*(6), 711-724.
30. Grauman, K., Westbury, A., Byrne, E., Chavis, Z., Furnari, A., Girdhar, R., ... & Malik, J. (2022). Ego4d: Around the world in 3,000 hours of egocentric video. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (pp. 18995-19012).
31. Green, D. M., & Swets, J. A. (1966). Signal detection theory and psychophysics (Vol. 1, pp. 1969-2012). New York: Wiley.
32. Güçlü, U., & van Gerven, M. A. (2015). Deep neural networks reveal a gradient in the complexity of neural representations across the ventral stream. *Journal of Neuroscience*, *35*(27), 10005-10014.
33. Hansen, T., & Neumann, H. (2008). A recurrent model of contour integration in primary visual cortex. Journal of Vision, 8(8), 8-8.
34. He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep residual learning for image recognition. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 770-778).
35. Hermann, K., & Lampinen, A. (2020). What shapes feature representations? Exploring datasets, architectures, and training. *Advances in Neural Information Processing Systems*, *33*, 9995-10006.

36. Hess, R. F., & Dakin, S. C. (1997). Absence of contour linking in peripheral vision. *Nature*, *390*(6660), 602-604.
37. Hess, R. F., & Field, D. J. (1995). Contour integration across depth. Vision Research, 35(12), 1699-1711.
38. Hochberg, J., & Hardy, D. (1960). Brightness and proximity factors in grouping. *Perceptual and Motor Skills*.
39. Hubel, D. H., & Wiesel, T. N. (1968). Receptive fields and functional architecture of monkey striate cortex. *The Journal of physiology*, *195*(1), 215-243.
40. Jacob, G., Pramod, R. T., Katti, H., & Arun, S. P. (2021). Qualitative similarities and differences in visual object representations between brains and deep networks. Nature communications, 12(1), 1872.
41. Kapadia, M. K., Ito, M., Gilbert, C. D., & Westheimer, G. (1995). Improvement in visual sensitivity by changes in local context: parallel studies in human observers and in V1 of alert monkeys. *Neuron*, *15*(4), 843-856.
42. Keemink, S. W., & van Rossum, M. C. (2016). A unified account of tilt illusions, association fields, and contour detection based on elastica. Vision research, 126, 164-173.
43. Khaligh-Razavi, S. M., & Kriegeskorte, N. (2014). Deep supervised, but not unsupervised, models may explain IT cortical representation. *PloS computational biology*, *10*(11), e1003915.
44. Khan, S., Wong, A., & Tripp, B. P. (2020, October). Task-driven learning of contour integration responses in a V1 model. In *NeurIPS 2020 Workshop SVRHM*.
45. Koffka, K. (1922). Perception: An introduction to the Gestalt-theorie. *Psychological bulletin*, *19*(10), 531-585.
46. Köhler, W. (1967). Gestalt psychology. *Psychologische Forschung*, *31*(1), XVIII-XXX.
47. Konkle, T., & Alvarez, G. A. (2022). A self-supervised domain-general learning framework for human ventral stream representation. *Nature communications*, *13*(1), 491.
48. Krizhevsky, A., Sutskever, I., & Hinton, G. E. (2012). Imagenet classification with deep convolutional neural networks. *Advances in neural information processing systems*, *25*.
49. Kubilius, J., Bracci, S., & Op de Beeck, H. P. (2016). Deep neural networks as a computational model for human shape sensitivity. *PloS computational biology*, *12*(4), e1004896.
50. Kubovy, M. (1994). The perceptual organization of dot lattices. *Psychonomic bulletin & review*, *1*(2), 182-190.
51. LeCun, Y., Bottou, L., Bengio, Y., & Haffner, P. (1998). Gradient-based learning applied to document recognition. Proceedings of the IEEE, 86(11), 2278-2324.
52. Li, Z. (1998). A neural model of contour integration in the primary visual cortex. *Neural computation*, *10*(4), 903-940.
53. Lindsey, J., Ocko, S. A., Ganguli, S., & Deny, S. (2019). A unified theory of early visual representations from retina to cortex through anatomically constrained deep CNNs. *arXiv preprint arXiv:1901.00945*.
54. Linsley, D., Kim, J., Ashok, A., & Serre, T. (2020). Recurrent neural circuits for contour detection. *arXiv preprint arXiv:2010.15314*.
55. Linsley, D., Kim, J., Veerabadran, V., Windolf, C., & Serre, T. (2018). Learning long-range spatial dependencies with horizontal gated recurrent units. *Advances in neural information processing systems*, *31*.

56. Malhotra, G., Dujmović, M., Hummel, J., & Bowers, J. S. (2023). Human shape representations are not an emergent property of learning to classify objects. *Journal of Experimental Psychology: General*.
57. Malhotra, G., Evans, B. D., & Bowers, J. S. (2020). Hiding a plane with a pixel: examining shape-bias in CNNs and the benefit of building in biological constraints. *Vision Research*, *174*, 57-68.
58. Marr, D., & Nishihara, H. K. (1978). Representation and recognition of the spatial organization of three-dimensional shapes. *Proceedings of the Royal Society of London. Series B. Biological Sciences*, *200*(1140), 269-294.
59. Muller, N., Groen, I. I., & Scholte, H. S. (2023). Pre-Training on High-Quality Natural Image Data Reduces DCNN Texture Bias.
60. Nakayama, K., & Shimojo, S. (1992). Experiencing and perceiving visual surfaces. *Science*, *257*(5075), 1357-1363.
61. Nakayama, K., He, Z. J., & Shimojo, S. (1995). Visual surface representation: A critical link between lower-level and higher-level vision.
62. Oquab, M., Darcet, T., Moutakanni, T., Vo, H., Szafraniec, M., Khalidov, V., ... & Bojanowski, P. (2023). Dinov2: Learning robust visual features without supervision. arXiv preprint arXiv:2304.07193.
63. Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., ... & Chintala, S. (2019). Pytorch: An imperative style, high-performance deep learning library. Advances in neural information processing systems, 32.
64. Peterson, J. C., Abbott, J. T., & Griffiths, T. L. (2018). Evaluating (and improving) the correspondence between deep neural networks and human representations. *Cognitive science*, *42*(8), 2648-2669.
65. Piëch, V., Li, W., Reeke, G. N., & Gilbert, C. D. (2013). Network model of top-down influences on local gain and contextual interactions in visual cortex. *Proceedings of the National Academy of Sciences*, *110*(43), E4108-E4117.
66. Puebla, G., & Bowers, J. S. (2022). Can deep convolutional neural networks support relational reasoning in the same-different task?. *Journal of Vision*, *22*(10), 11-11.
67. Quinlan, P. T., & Wilton, R. N. (1998). Grouping by proximity or similarity? Competition between the Gestalt principles in vision. *Perception*, *27*(4), 417-430.
68. Rajalingham, R., Issa, E. B., Bashivan, P., Kar, K., Schmidt, K., & DiCarlo, J. J. (2018). Large-scale, high-resolution comparison of the core visual object recognition behavior of humans, monkeys, and state-of-the-art deep artificial neural networks. *Journal of Neuroscience*, *38*(33), 7255-7269.
69. Richards, B. A., Lillicrap, T. P., Beaudoin, P., Bengio, Y., Bogacz, R., Christensen, A., … & Kording, K. P. (2019). A deep learning framework for neuroscience. *Nature neuroscience*, *22*(11), 1761-1770.
70. Ritter, S., Barrett, D. G., Santoro, A., & Botvinick, M. M. (2017, July). Cognitive psychology for deep neural networks: A shape bias case study. In *International conference on machine learning* (pp. 2940-2949). PMLR.
71. Rock, I., & Brosgole, L. (1964). Grouping based on phenomenal proximity. *Journal of Experimental Psychology*, *67*(6), 531.
72. Roelfsema, P. R. (2006). Cortical algorithms for perceptual grouping. Annu. Rev. Neurosci., 29, 203-227.

73. Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., ... & Fei-Fei, L. (2015). Imagenet large scale visual recognition challenge. *International journal of computer vision*, *115*, 211-252.
74. Sablé-Meyer, M., Fagot, J., Caparos, S., van Kerkoerle, T., Amalric, M., & Dehaene, S. (2021). Sensitivity to geometric shape regularity in humans and baboons: A putative signature of human singularity. Proceedings of the National Academy of Sciences, 118(16), e2023123118.
75. Schiller, P. H., & Tehovnik, E. J. (2015). Vision and the visual system. Oxford University Press.
76. Schrimpf, M., Kubilius, J., Hong, H., Majaj, N. J., Rajalingham, R., Issa, E. B., ... & DiCarlo, J. J. (2018). Brain-score: Which artificial neural network for object recognition is most brain-like?. *BioRxiv*, 407007.
77. Springenberg, J. T., Dosovitskiy, A., Brox, T., & Riedmiller, M. (2014). Striving for simplicity: The all convolutional net. arXiv preprint arXiv:1412.6806.
78. Steiger, J. H. (1980). Tests for comparing elements of a correlation matrix. Psychological bulletin, 87(2), 245.
79. Steiner, A., Kolesnikov, A., Zhai, X., Wightman, R., Uszkoreit, J., & Beyer, L. (2021). How to train your vit? data, augmentation, and regularization in vision transformers. arXiv preprint arXiv:2106.10270.
80. Sullivan, J., Mei, M., Perfors, A., Wojcik, E., & Frank, M. C. (2021). SAYCam: A large, longitudinal audiovisual dataset recorded from the infant's perspective. Open mind, 5, 20-29.
81. Veerabadran, V., & de Sa, V. R. (2019). V1Net: A computational model of cortical horizontal connections.
82. Vezzani, S., Marino, B. F., & Giora, E. (2012). An early history of the Gestalt factors of organisation. Perception, 41(2), 148-167.
83. Von der Heydt, R. (2015). Figure–ground organization and the emergence of proto-objects in the visual cortex. *Frontiers in psychology*, *6*, 1695.
84. Wagemans, J., Elder, J. H., Kubovy, M., Palmer, S. E., Peterson, M. A., Singh, M., & Von der Heydt, R. (2012). A century of Gestalt psychology in visual perception: I. Perceptual grouping and figure–ground organization. *Psychological bulletin*, *138*(6), 1172.
85. Wang, W., & Neumann, U. (2018). Depth-aware cnn for rgb-d segmentation. In Proceedings of the European conference on computer vision (ECCV) (pp. 135-150).
86. Wertheimer, M. (1912). Experimentelle studien uber das sehen von bewegung. *Zeitschrift fur psychologie*, *61*, 161-165.
87. Wertheimer, M. (1923). Untersuchungen zur Lehre von der Gestalt, II. [Investigations in Gestalt Theory: II. Laws of organization in perceptual forms]. Psychologische Forschung, 4, 301–350.
88. Wertheimer, M. (1938). Laws of organization in perceptual forms.
89. Wightman, R. (2019). Pytorch image models.
90. Yalniz, I. Z., Jégou, H., Chen, K., Paluri, M., & Mahajan, D. (2019). Billion-scale semi-supervised learning for image classification. arXiv preprint arXiv:1905.00546.
91. Yamins, D. L., & DiCarlo, J. J. (2016). Using goal-driven deep learning models to understand sensory cortex. *Nature neuroscience*, *19*(3), 356-365.