Perceptual misalignment of texture representations in convolutional neural networks

Ludovica de Paolis[1] (ldepaoli@sissa.it), Fabio Anselmi[2] (fabio.anselmi@units.it), Alessio Ansuini[3] (alessio.ansuini@areasciencepark.it), Eugenio Piasini[1] (epiasini@sissa.it)

[1]Neuroscience area, International School for Advanced Studies (SISSA), Trieste, Italy

[2]Department of Mathematics, Informatics and Geosciences, Università degli Studi di Trieste, Trieste, Italy

[3] Department of Data Engineering, Area Science Park, Trieste, Italy

1 Abstract

Mathematical modeling of visual textures traces back to Julesz's intuition that texture perception in humans is based on local correlations between image features (Julesz, 1981). An influential approach for texture analysis and generation generalizes this notion to linear correlations between the nonlinear features computed by convolutional neural networks (CNNs), compiled into Gram matrices (Gatys et al., 2015). Given that CNNs are often used as models for the visual system, it is natural to ask whether such "texture representations" spontaneously align with the textures' perceptual content, and in particular

Correspondence to: Ludovica de Paolis (ldepaoli@sissa.it) and Eugenio Piasini (epiasini@sissa.it)
Code available at: https://github.com/ludovicadepaolis01/perceptual_misalignment

whether those CNNs that are regarded as better models for the visual system also possess more human-like texture representations. Here we quantify the perceptual content captured by feature correlations computed for a diverse pool of CNNs, and we compare it to the models' perceptual alignment with the mammalian visual system as measured by Brain-Score (Schrimpf et al., 2018). Surprisingly, we find that there is no connection between conventional measures of CNN quality as a model of the visual system and its alignment with human texture perception. We conclude that texture perception involves mechanisms that are distinct from those that are commonly modeled using approaches based on CNNs trained on object recognition, possibly depending on the integration of contextual information.

2 Keywords

Textures, Convolutional Neural Networks, Brain-Score, Representational Similarity Analysis

3 Introduction

Visual textures hold a special status in vision neuroscience as stimuli that afford a non-trivial degree of controllability and complexity, unlike natural images or more traditional parametric stimuli (Victor et al., 2017). Because of this, multiple mathematical frameworks for modeling and generating textures have been established, with considerable success. One particularly influential strand of theoretical, computational and empirical work stems from the seminal view of Julesz (Julesz, 1981; Julesz, 1962; Caelli & Julesz, 1978), who proposed that texture perception is controlled by local correlations of visual features within an image. This view was extended by later work that better formalized the notion of textures as statistical objects (Zhu et al., 2000) (but see (L. Liu et al., 2019) for a detailed historical overview). This study demonstrated the effectiveness of using features based on the

neuroscience of vision (Simoncelli, 2000) and developed a connection with natural stimulus statistics through the efficient coding paradigm (Victor et al., 2017; Tkačik et al., 2010; Victor & Conte, 2012; Hermundstad et al., 2014; Tesileanu et al., 2020, 2022; Caramellino et al., 2021; Zanon et al., 2026).

In recent years, following the rise in popularity of deep convolutional neural networks (CNNs) as tools in computer vision, Gatys et al. proposed a machine learning-based operationalization of Julesz's idea (Gatys et al., 2015). The Gatys model effectively defined a texture ensemble based on the Gram matrices formed by spatial correlations between intermediate-layer activations of a CNN trained for object recognition. Texture synthesis is performed by optimizing the image pixels via gradient descent to match the statistical structure of an exemplar texture. This is achieved by minimizing a *Gram loss*, defined as the normalized L2 distance between the Gram matrices of the exemplar image and those of the synthesized image. This approach was rapidly adopted in the budding fields of neural texture synthesis and style transfer (see for instance (Gatys et al., 2016; Li et al., 2017; Ulyanov et al., 2017; Johnson et al., 2016; N. Yu et al., 2019)), and was later improved under multiple aspects: high resolution images processing (Snelgrove, 2017), better methods for capturing long-range correlations within images (G. Liu et al., 2016; Sendik & Cohen-Or, 2017; Berger & Memisevic, 2017), usage of more sophisticated losses capable of capturing stationary properties of the CNN's activations beyond the pairwise correlations in the Gram loss (Heitz et al., 2021; Vacher et al., 2020).

In parallel, and more broadly, CNNs trained for object recognition have become ubiquitous tools in vision neuroscience. Over the years, work from many labs has established these models as functional accounts of the ventral visual stream in monkeys (Yamins et al., 2014; Cadieu et al., 2014; Khaligh-Razavi & Kriegeskorte, 2014; Cadena, Denfield, et al., 2019; Schrimpf et al., 2020; Bashivan et al., 2019) and humans (Storrs et al., 2021; Güçlü & Gerven, 2015), as well as the corresponding visual cortical areas in rodents (Cadena, Sinz, et al., 2019; Nayebi et al., 2023; Muratore et al., 2022, 2025). However, the strong

predictive power of this approach for neural activity is only partly mirrored by the capacity of these artificial neural networks to accurately predict behavioral responses to individual stimuli (Feather et al., 2023; Geirhos et al., 2021; Wichmann & Geirhos, 2023; Rajalingham et al., 2018; Vinken et al., 2020).

Beyond object recognition, the ventral stream is also thought to be important for texture perception. Indeed, the idea that texture perception can be explained by reference to the neural mechanisms in early vision goes at least as far back as Turner, 1986 and Malik and Perona, 1990. More recent results in computational modeling and neurophysiological evidence point generally to the involvement of early-to-intermediate areas in the ventral stream (Simoncelli, 2000; Freeman & Simoncelli, 2011; Freeman et al., 2013; Y. Yu et al., 2015; Okazawa et al., 2017; Ziemba et al., 2016; Henderson et al., 2023; Matteucci et al., 2024; Ziemba et al., 2024). However, overall, our understanding of the neural computations underpinning texture perception is still limited compared to those enabling object recognition. This makes CNN-based methods for texture analysis and synthesis particularly interesting from a neuroscience perspective, despite the recent emergence of approaches based on diffusion models for practical applications, e.g. (Wang et al., 2024). More specifically, it raises the question of whether the existing quality measures devised for CNN-based models of the ventral stream align with the perceptual quality of textures generated based on the internal features of those models. If such alignment exists, it would not only suggest that object recognition and texture perception share a significant amount of functional mechanisms, but also that these mechanisms are among those that are well captured by CNNs trained on object recognition. Conversely, the absence of such a relationship – where increased fidelity to neural or behavioral object recognition data does not lead to improved modeling of human texture perception – would indicate that current CNN architectures or training paradigms lack critical components.

In the present work, we considered a set of popular CNNs and we asked whether the models that are more "brain-like" in object recogntion also tend to represent visual textures

in a way that is closer to human perception. As a metric of model performance on object recognition, we used the standardized benchmarks provided by Brain-Score (Schrimpf et al., 2018). To measure alignment with human texture perception, we evaluated the models on the images of the Describable Texture Dataset (Cimpoi et al., 2014), which are divided in classes according to human annotations. We then assessed whether the "texture representations" provided by the Gram matrices computed from the models' feature maps (as in (Gatys et al., 2015)) are organized in a way that respect the classes subdivision in the dataset.

Our results show that there is no correlation between the goodness of the Gram texture representation – with respect to the clusters formed by human-annotated labels – and Brain-Score. This suggests that the features governing the utility of artificial neural networks as models of visual cortex in an object recognition context are not fully aligned with those that can make them good models for texture processing, therefore highlighting an underexplored path for the improvement of machine learning-based models for texture analysis and synthesis.

4 Methods

4.1 The "Gram representation" for textures

Our study builds on the texture analysis and synthesis method introduced by Gatys et al. (Gatys et al., 2015). This approach models textures using feature representations extracted from a CNN, which are shown to support the generation of perceptually compelling texture samples. In the original implementation, given a natural texture image x, the algorithm starts by feeding x as input to the network and extracting the corresponding activation maps from a predetermined set of layers. For each selected layer l, the feature activations F^l are used to compute a Gram matrix G^l, defined as the inner product between

vector representations of the feature maps:

$$G^l_{ij} = \sum_m F^l_{im} F^l_{jm}, \tag{1}$$

where the index m runs over spatial samples of the feature maps, and i and j label feature channels at layer l. This Gram matrix summarizes the correlations between feature channels, discarding spatial information and yielding a statistical description of textures. These "Gram representations" are thought to capture the elements of the input image that are important for texture perception. In this work, we study the Gram representations generated by a number of different CNNs.

4.2 Texture synthesis

In the present work we provide examples of texture synthesis to help the reader gauge the quality of the images that can be produced based on the representations described in the previous paragraph, despite the fact that our main analyses and conclusions do not depend on the output of any texture synthesis algorithm. For completeness, we report here the details of our synthesis approach.

To generate synthetic textures we followed the method established by Gatys et al., 2015, with minor modifications. Briefly, texture synthesis is performed by iteratively updating the pixels of an initially random image $\hat{x}$, where pixel values are drawn from a Gaussian distribution $\hat{x} \sim \mathcal{N}(0, 1)$. This process leads to match the Gram matrices of $\hat{x}$ to those of the target image x by minimizing the *Gram loss*, defined as a weighted sum of squared L2 distances between corresponding Gram matrices across layers:

$$\mathcal{L}(\hat{x}, x) = \sum_l \frac{1}{4 M_l N_l^2} \|\hat{G}^l - G^l\|_2^2. \tag{2}$$

where N_l is the number of features in layer l and M_l is the number of units within each feature (note that the term $1/4M_lN_l^2$ that we use here to weight the contribution from

each individual layer is not identical to the one in Gatys et al., 2015; see the Appendix for details on our choice of loss weighting for image synthesis). To minimize the Gram loss we used L-BFGS as in Gatys et al., 2015, using the implementation of the optimization algorithm provided by PyTorch (Ansel et al., 2024).

Minimizing this objective in pixel space generates novel images whose texture statistics, as captured by feature correlations in a convolutional neural network, match those of the original image. Importantly, while spatial structure is not preserved, the resulting images retain perceptually salient texture properties, highlighting the role of feature-level statistics in texture representation.

4.3 CNN Layer selection

We implemented the texture representation and the synthesis algorithm described above using PyTorch (Ansel et al., 2024), for a range of CNN architectures trained for object classification with ImageNet-1K (see Table 1). The pre-trained networks were sourced from TorchVision (TorchVision maintainers and contributors, 2016). The convolutional layers of the CNN originally used by Gatys et al., 2015 (VGG-19) have five different spatial scales, corresponding to the five consecutive "convolve-and-pool" blocks of that architecture. Gatys et al., 2015 showed that selecting just one layer for each of these spatial scales was enough to produce perceptually satisfying synthetic textures. Therefore to ensure that the results of our analysis are comparable across different networks, we extracted feature maps from five layers from each architecture, selected in such a way that they are located at comparable relative depth across architectures; see Appendix, Supplementary Tables 3, 4 for details.

4.4 Stimulus dataset

The Describable Textures Dataset (DTD) (Cimpoi et al., 2014) is our input dataset of choice. It consists of 5640 images of naturalistic visual textures collected from the Internet,

Family	Model
	AlexNet (Krizhevsky et al., 1994)
DenseNet (Huang et al., 2017)	DenseNet-121 DenseNet-169 DenseNet-201
InceptionNet (Howard et al., 2017)	Inception-v3 (Szegedy et al., 2016)
ResNet (He et al., 2016)	ResNet18 ResNet34 ResNet50 ResNet101 ResNet152
VGG (Simonyan & Zisserman, 2015)	VGG-16 VGG-19

Table 1: Convolutional Neural Network architectures grouped by family.

spanning a wide range of materials and surface appearances. The dataset is organized into 47 texture classes, each one containing 120 images. The images size ranges between 300x300px and 640x640px. We resized (scaled) every image to 224x224px, in order to match the CNNs' input spatial dimensions and to maintain a stable representation of the textures.

The class labels arise from a human annotation procedure based on the *Texture Lexicon*, a psycholinguistic study introduced in Bhushan et al., 1997 with the goal of constructing a linguistically grounded vocabulary for describing visual textures. The Texture Lexicon identifies a set of texture-related adjectives commonly used by human observers (e.g., *blotchy*, *striped*, *bumpy*, *woven*) and organizes them according to perceptual relevance.

To construct DTD, the authors selected 47 representative texture terms from the Texture Lexicon and used them as labels in a large-scale annotation task involving human participants. Annotators were presented with images and asked to assign one or more texture attributes that best described the visual appearance of the image. Final class labels were determined by aggregating human responses and retaining images for which there was sufficient agreement across annotators. As a result, DTD encodes texture categories that reflect shared perceptual judgments rather than low-level image statistics or semantic

object labels.

This annotation procedure makes DTD particularly suitable for studies of texture perception, as it provides a direct link between image-level texture statistics and perceptually meaningful texture descriptors grounded in human language and cognition. DTD treats textures as entities that can be categorized in a manner analogous to object images, effectively providing human-defined clusters of visual textures with a balanced distribution of images across classes. The richness and diversity of DTD thus constitute a perceptual and linguistic benchmark against which we can evaluate whether texture representations derived from CNNs in combination with the Gatys method are sufficient to preserve class-specific texture distinctions as defined by human perception.

4.5 Clustering

We performed Representation Similarity Analysis (RSA) (Kriegeskorte et al., 2008) on the texture representations extracted with the Gram matrices on our pool of 12 CNNs. RSA was introduced in neuroscience as a method to compare object representations obtained with different means (e.g., functional Magnetic Resonance Imaging vs. Magnetoencephalography), or to compare biological and computational representations (e.g., brain activity vs. CNNs). It relies on Representational Dissimilarity Matrices (RDMs), where each entry is the pairwise dissimilarity between responses to two stimuli. We employed RSA to verify whether the RDMs produced with Gram matrices extracted from DTD show a clustering pattern consistent with the original 47 classes. For each one of the 12 CNNs, we obtained 5 RDMs – one for each of the convolutional layers analyzed – of dimensionality 5640x5640, where each entry is the distance among Gram matrices for one pair of images in DTD.

We tested whether Gram matrix–based texture representations capture the structure of the 47 texture classes in DTD by performing hierarchical clustering with Ward linkage function based on Euclidean distance, as implemented in scikit-learn (Pedregosa et al., 2011).

Hierarchical clustering is an unsupervised technique that groups data points based on their similarity, producing a tree-like representation in which clusters are progressively merged according to their pairwise distances. This hierarchical organization provides insight into the intrinsic structure of the data without relying on class labels.

To evaluate the extent to which the learned texture representations recover the human-defined categories in DTD, we treated the dataset labels as a reference standard. Specifically, we fixed the number of clusters to $k = 47$, matching the number of texture classes in the original dataset, and allowed the clustering algorithm to organize the representations accordingly. This procedure enables a direct comparison between the automatically discovered clusters and the perceptual texture classes defined by human annotations.

4.6 Measuring clustering quality

We quantified texture information carried in the representations within Gram matrices with Mutual Information (MI) (MacKay, 2003). In our analysis MI was computed between the real 47 classes in DTD (*real classes*) and the 47 clusters found by hierarchical clustering (*found clusters*). If X and Y are random variables representing respectively the true label of an image (*real class*) and the label assigned to it by the clustering procedure (*found cluster*), MI is the reduction in uncertainty, quantified as entropy (H), in one of the two variables associated with learning the value of other:

$$\mathrm{MI}(X:Y) = H(X) - H(X|Y) = H(Y) - H(Y|X) = \sum_{x,y} p(x,y)\, \log \frac{p(x,y)}{p(x)\, p(y)}$$

By construction, $\mathrm{MI}(X:Y)$ is upper-bounded by the minimum between $H(X)$ and $H(Y)$. This can be seen by the expression above taking into account that $H(X|Y)$ and $H(Y|X)$ are always nonnegative. Therefore, considering that Y follows an unknown distribution where *found clusters* result from hierarchical clustering, and X is distributed uniformly among the 47 *real classes* of DTD, in our case the MI is upper bounded by the entropy of

X:

$$\mathrm{MI}_{\max} = H(X) = -\sum_{i=1}^{47} p_i \log_2 p_i = \log_2 47 \approx 5.5\text{bits} \tag{3}$$

All empirical estimates for mutual information were corrected for potential undersampling bias using the NSB method (Nemenman et al., 2002), as implemented by the ndd Python package (Marsili, 2021).

We quantified MI between *real classes* and *found clusters* for the *found classes* distribution drawn from each of the 5 layers of the 12 CNNs. Therefore, for each model we obtained 5 MI values.

4.7 Brain-Score correlation

To assess the relationship between the goodness of a CNN as a model for biological object recognition and texture perception, we correlated the best MI value (i.e. the highest out of the 5 MI values computed for each model) of each CNN to Brain-Score. Brain-Score is a composite of multiple behavioral and neural benchmarks that determine how similar artificial neural networks are to the object recognition mechanisms in the human and macaque brain (Schrimpf et al., 2018). In this study, we focus on the following Brain-Score components: *neural vision*, *behavior vision*, and *average vision*, as well as the region-specific neural predictability scores for *V1, V2, V4*, and *IT*. The *neural vision* score summarizes a model's ability to predict neural responses across areas of the primate ventral visual stream, while *behavior vision* quantifies the alignment between model predictions and human behavioral performance on object recognition tasks. The *average vision* score provides an overall summary across neural and behavioral benchmarks. The region-specific scores (*V1, V2, V4, IT*) evaluate how well model representations predict neural activity in individual cortical areas involved in the hierarchical processing of visual information along the ventral stream. For each of the 12 models, we correlated the highest MI with the aforementioned Brain-Score values which we report in Appendix Supplementary Table 5. We run a linear correlation with Pearson's p and set $\alpha = 0.05$.

5 Results

5.1 Representational Similarity Analysis

To understand how texture information is represented across the layers of our CNNs, we started by performing Representational Similarity Analysis. Figure 1 shows Representational Dissimilarity matrices (RDMs) calculated on pairs of Gram matrix representations of DTD images extracted from VGG-19. Performing the same analysis on other networks led to qualitatively similar results (see de Paolis, 2025 for high-resolution RDM plots for all the networks considered in this work). Visual inspection reveals that the structure of the RDMs reflects the existence of at least some of the classes in DTD, and that for this network the clearest structure emerges in the last three layers. To understand these trends more quantitatively, we next moved to performing clustering analysis of the Gram representations.

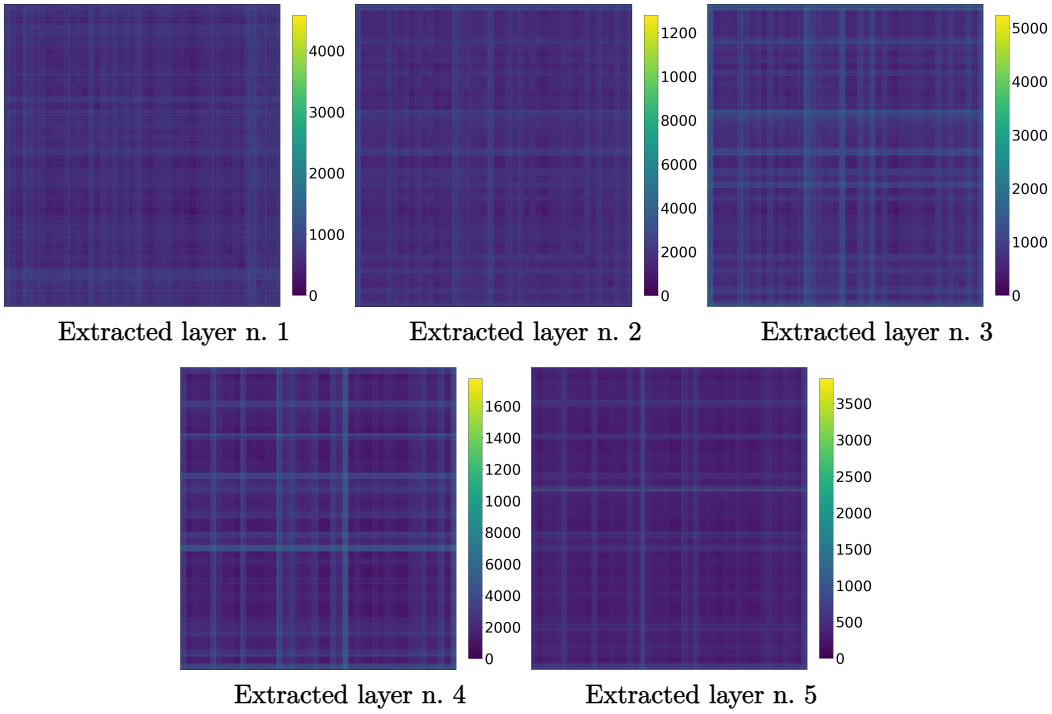


Figure 1: Representational Dissimilarity Matrices obtained from the five layers analyzed for VGG-19. The size of each matrix is 5640×5640, and each entry corresponds to the Euclidean distance – relative to each matrix, as in the colorbars – between the Gram matrix representation of one pair of images in DTD. The entries are sorted such that images belonging to each of the 47 ground-truth categories are placed next to each other.

5.2 Clustering analysis

Figure 2 represents the values of MI by model and by layer found with hierarchical clustering. The plot shows a trend: MI grows along with the depth of the extracted features, consistently with the trend found by visual inspection in the RSA plots. The highest MI is yielded by the last layer for most models, but it does not go beyond ≈ 2.7 bits, which is about half of the maximum achievable MI, according to Equation 3. These numbers show that increasing the depth of the extracted representations leads to an increasing clustering quality.

The results of the correlation among the highest MI value per model and selected Brain-Score benchmarks are reported in Table 2. We did not find any significant correlation between any Brain-Score value and the best MI found in any model, as shown in Figure 3.

Brain-Score metric	**Pearson's r**	**p-value**
average vision	-0.130	0.68
neural vision	-0.165	0.60
behavior vision	-0.029	0.92
V1	-0.229	0.47
V2	-0.262	0.41
V4	-0.063	0.84
IT	-0.104	0.74

Table 2: Correlation between texture representation quality, expressed as MI between the clusters in the Gram representation of textures and the ground-truth labels in DTD, and quality of the networks as models of the ventral visual stream, measured by Brain-Score. For more details on Brain-Score see Section 4.7.

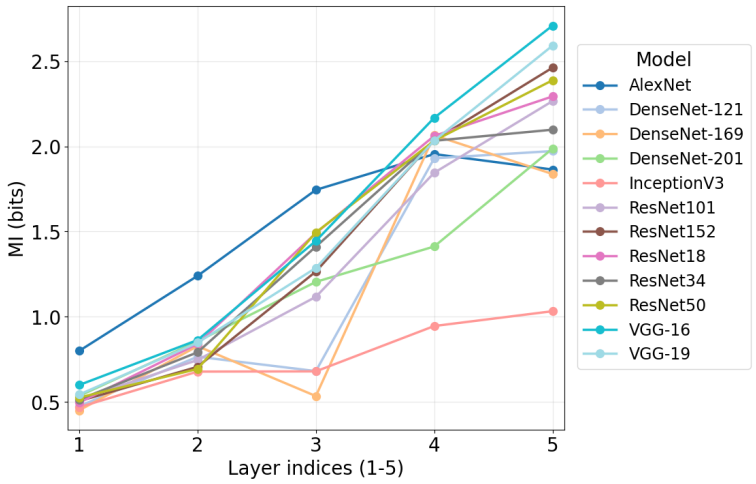


Figure 2: MI values across CNN layers identified by index (1-5). Each line corresponds to one of the 12 models as color-coded in the legend.

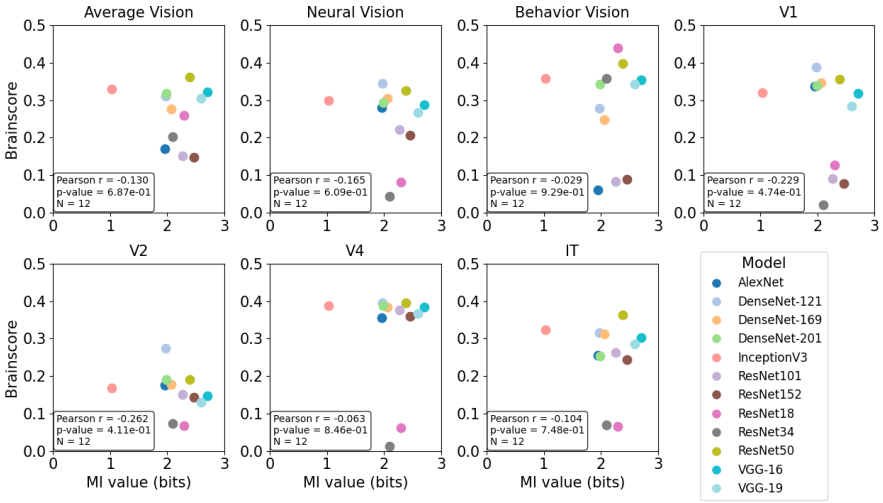


Figure 3: Plots of the correlations with Pearson's p. Each subplot represents the correlation of Brain-Score benchmarks *average vision, neural vision, behavior vision, V1, V2, V4, IT* respectively vs. the layer with the highest MI for each of the 12 CNNs. Each color identifies a model as reported in the legend.

5.3 Texture synthesis

In Figure 4 we report samples of synthesized textures for a subset of selected models. The purpose of this plot is to illustrate the output of the Gatys algorithm, as implemented by us with our pool of CNNs; we underline that texture synthesis is not part of the main analysis pipeline, and in our work is only used to generate example visualizations. The results of the quantitative analyses described in the previous sections do not depend directly from this visual outcome. Each sample belongs to 4 separate DTD classes: *blotchy, striped, matted, scaly.* We observe how some models achieve a better perceptual output in terms of synthesized textures, despite running the same algorithm.

6 Discussion

In this study, we have investigated the quality of the Gram texture representation proposed by Gatys et al., 2015 in terms of its agreement with the categorical grouping in the Describable Texture Dataset (Cimpoi et al., 2014), for a diverse pool of deep convolutional networks. We have compared this measure of perceptual quality with Brain-Score, an independent benchmark for "brain-like" operation of the CNNs in object recognition settings (Schrimpf et al., 2018, 2020).

Our analyses lead to three main conclusions. First, the perceptual quality of the Gram representations tends, for most networks, to grow monotonically with the depth of the layers analyzed (Figure 2). This is generally compatible with recent results suggesting that texture processing is distributed along intermediate areas of the ventral visual stream inclusing V2 and V4 (Okazawa et al., 2017; Ziemba et al., 2024) and possibly IT (Zhivago & Arun, 2014).

Second, the Gram representation is unable to recover the full class structure of the data, as determined by the human-assigned labels. Quantitatively, this is seen by the value of the clustering MI reaching only about half of the theoretical maximum (Figure 2). This

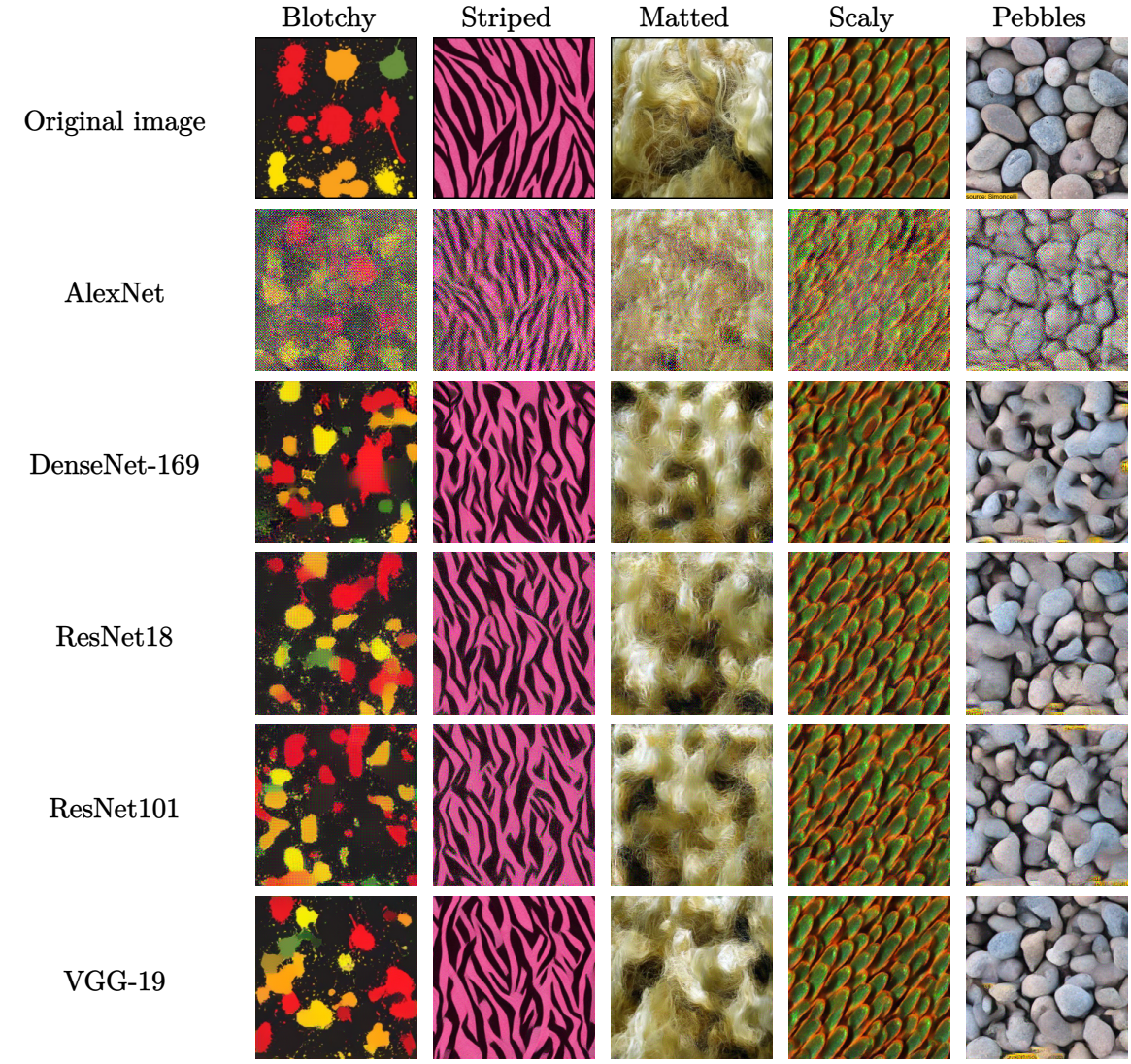


Figure 4: Textures generated with the Gatys algorithm applied to a subsample of CNNs (rows). The 4 original images (top row) belong to 4 different classes in DTD (columns: *blotchy; striped; matted; scaly*). The last column represents one of the images originally used by Gatys et al., which we synthesized as a reference. Textures generated by the whole pool of 12 models can be found in Appendix (Supplementary Figures 1, 2)

highlights that, despite its popularity, the Gram representation is far from perfect. In further work, we aim at exploring whether the perceptual improvements associated with more sophisticated versions of the representation translate to significant improvements in MI on the DTD.

Third, the quality of the CNNs as implicit models for texture perception is not correlated with their quality as models of the ventral stream in an object recognition setting (Figure 3). This is despite the fact that there is substantial variability in both metrics, as is well known for Brain-Score, and as can be also appreciated visually for the textures in Figure 4. It is not simply due for instance to all models having roughly the same texture representation capacity. This result is surprising, because one could have expected that the functional features captured by Brain-Score are general enough to at least partly include those that are necessary to support human-like texture perception. Instead, this appears not to be the case. We conclude that in order to design methods of texture analysis and synthesis that are better aligned with human perception we need to look beyond standard CNNs trained for object recognition, for instance by incorporating unsupervised or self-supervised learning (Matteucci et al., 2024), multimodality (Radford et al., 2021) or different architectures, such as those with attention mechanisms (Vaswani et al., 2017; Guo et al., 2022).

Finally, we note that our conclusions are broadly compatible with another series of works relating human vision and texture encoding in CNNs. In an influential paper, Geirhos et al., 2019 showed that CNNs trained for object classification on ImageNet tend to classify images based on texture information more than shape information, in a way that differs markedly from human behavior. However, subsequent work within that research strand points to the idea that, in fact, shape and texture information do coexist within the representation layers of CNNs, and that this so-called "texture bias" emerges primarily at the level of the decision head, which under standard training conditions learns to prioritize texture information over shape information (Hermann et al., 2020; Islam et

al., 2021). Our notion of "texture information" (the MI between the Gram clusters and the ground-truth DTD labels) increases with layer depth across most models, generalizing the observations in Hermann et al., 2020, which shows (Supplemental Figure C.4) that for AlexNet the texture information increases rapidly in early convolutional layers and then flattens out towards the end of the hierarchy. This is very similar to the trend we report for AlexNet in Figure 2. Another interesting observation in Hermann et al., 2020 is that neurally-motivated architectures like CorNET (Kubilius et al., 2018; designed by the same team as Brain-Score) do not fare better than standard CNNs from the point of view of their texture bias in classification. This echoes our own results showing that architectures that are better models of ventral stream processing in the canonical sense (related to object identity processing, measured by Brain-Score) are not necessarily more human-like in how they process textures.

7 Conclusions

Our work highlights a number of limitations of popular machine learning-based approaches for visual texture analysis and synthesis. At the same time, it provides a simple example of a visual function that is likely performed in the ventral stream but for which the existing quality metrics of neural and behavioral fidelity for models seem to have very little predictive power. In future work, we plan to extend our analyses to different architectures and texture representations, and to develop a more direct and systematic way of testing the perceptual quality of generated textures with large-scale human psychophysics. Going forward, we hope that our approach can provide an additional guide for the development of truly general-purpose models of the mammalian visual system.

8 Acknowledgments

EP was partially supported by by the European Union — NextGenerationEU — PNRRM4C2-I.1.1, in the framework of PRIN Project no. 2022XE8X9E, CUP:G53D23004590001.
AA was supported by the project "Supporto alla diagnosi di malattie rare tramite l'intelligenza artificiale" CUP: F53C22001770002, from the project "Valutazione automatica delle immagini diagnostiche tramite l'intelligenza artificiale", CUP: F53C22001780002, and by the European Union – NextGenerationEU within the project PNRR "PRP@CERIC" IR0000028 - Mission 4 Component 2 Investment 3.1 Action 3.1.1.

References

Julesz, B. (1981). Textons, the elements of texture perception, and their interactions. *Nature*, *290*(5802), 91–97. https://doi.org/10.1038/290091a0

Gatys, L. A., Ecker, A. S., & Bethge, M. (2015). Texture synthesis using convolutional neural networks. *Proceedings of the 28th International Conference on Neural Information Processing Systems*, *1*, 262–270.

Schrimpf, M., Kubilius, J., Hong, H., Majaj, N. J., Rajalingham, R., Issa, E. B., Kar, K., Bashivan, P., Prescott-Roy, J., Geiger, F., Schmidt, K., Yamins, D. L. K., & DiCarlo, J. J. (2018). Brain-score: Which artificial neural network for object recognition is most brain-like? *bioRxiv preprint*. https://www.biorxiv.org/content/10.1101/407007v2

Victor, J. D., Conte, M. M., & Chubb, C. F. (2017). Textures as Probes of Visual Processing. *Annual Review of Vision Science*, *3*(1), 275–296. https://doi.org/10.1146/annurev-vision-102016-061316

Julesz, B. (1962). Visual Pattern Discrimination. *IRE Transactions on Information Theory*, *8*(2), 84–92. https://doi.org/10.1109/TIT.1962.1057698

Caelli, T., & Julesz, B. (1978). On perceptual analyzers underlying visual texture discrimination: Part I. *Biological Cybernetics*, *28*(3), 167–175. https://doi.org/10.1007/BF00337138

Zhu, S., Liu, X., & Wu, Y. (2000). Exploring texture ensembles by efficient Markov chain Monte Carlo-Toward a "trichromacy" theory of texture. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *22*(6), 554–569. https://doi.org/10.1109/34.862195

Liu, L., Chen, J., Fieguth, P., Zhao, G., Chellappa, R., & Pietikäinen, M. (2019). From BoW to CNN: Two Decades of Texture Representation for Texture Classification. *International Journal of Computer Vision*, *127*(1), 74–109. https://doi.org/10.1007/s11263-018-1125-z

Simoncelli, J. P. E. . P. (2000). A parametric texture model based on joint statistics of complex wavelet coefficients. *International Journal of Computer Vision*, *40*, 49–70.

Tkačik, G., Prentice, J. S., Victor, J. D., & Balasubramanian, V. (2010). Local statistics in natural scenes predict the saliency of synthetic textures. *Proceedings of the National Academy of Sciences*, *107*(42), 18149–18154. https://doi.org/10.1073/pnas.0914916107

Victor, J. D., & Conte, M. M. (2012). Local image statistics: Maximum-entropy constructions and perceptual salience. *JOSA A*, *29*(7), 1313–1345. https://doi.org/10.1364/JOSAA.29.001313

Hermundstad, A. M., Briguglio, J. J., Conte, M. M., Victor, J. D., Balasubramanian, V., & Tkačik, G. (2014). Variance predicts salience in central sensory processing. *eLife*, *3*, e03722. https://doi.org/10.7554/eLife.03722

Tesileanu, T., Conte, M. M., Briguglio, J. J., Hermundstad, A. M., Victor, J. D., & Balasubramanian, V. (2020). Efficient coding of natural scene statistics predicts discrimination thresholds for grayscale textures (S. Palmer & T. E. Behrens, Eds.). *eLife*, *9*, e54347. https://doi.org/10.7554/eLife.54347

Tesileanu, T., Piasini, E., & Balasubramanian, V. (2022). Efficient processing of natural scenes in visual cortex. *Frontiers in Cellular Neuroscience*, *16*. Retrieved December 13, 2022, from https://www.frontiersin.org/articles/10.3389/fncel.2022.1006703

Caramellino, R., Piasini, E., Buccellato, A., Carboncino, A., Balasubramanian, V., & Zoccolan, D. (2021). Rat sensitivity to multipoint statistics is predicted by efficient coding of natural scenes. *eLife*, *10*, e72081. https://doi.org/10.7554/eLife.72081

Zanon, M., Lemaire, B. S., Piasini, E., Caramellino, R., Nallet, C., Balasubramanian, V., Gervain, J., Zoccolan, D., & Vallortigara, G. (2026). Predisposed and learned preferences for multipoint visual statistics in visually naive newly hatched chicks. *Proceedings of the Royal Society B: Biological Sciences*, *293*(2066), 20253157. https://doi.org/10.1098/rspb.2025.3157

Gatys, L. A., Ecker, A. S., & Bethge, M. (2016). Image Style Transfer Using Convolutional Neural Networks, 2414–2423. Retrieved September 27, 2022, from https://openaccess.thecvf.com/content_cvpr_2016/html/Gatys_Image_Style_Transfer_CVPR_2016_paper.html

Li, Y., Fang, C., Yang, J., Wang, Z., Lu, X., & Yang, M.-H. (2017). Diversified Texture Synthesis With Feed-Forward Networks, 3920–3928. Retrieved June 10, 2024, from https://openaccess.thecvf.com/content_cvpr_2017/html/Li_Diversified_Texture_Synthesis_CVPR_2017_paper.html

Ulyanov, D., Vedaldi, A., & Lempitsky, V. (2017). Improved texture networks: Maximizing quality and diversity in feed-forward stylization and texture synthesis. *Computer Vision and Pattern Recognition*.

Johnson, J., Alahi, A., & Fei-Fei, L. (2016). Perceptual Losses for Real-Time Style Transfer and Super-Resolution. In B. Leibe, J. Matas, N. Sebe, & M. Welling (Eds.), *Computer Vision – ECCV 2016* (pp. 694–711). Springer International Publishing. https://doi.org/10.1007/978-3-319-46475-6_43

Yu, N., Barnes, C., Shechtman, E., Amirghodsi, S., & Lukac, M. (2019). Texture Mixer: A Network for Controllable Synthesis and Interpolation of Texture, 12164–12173. Retrieved December 10, 2025, from https://openaccess.thecvf.com/content_CVPR_2019/html/Yu_Texture_Mixer_A_Network_for_Controllable_Synthesis_and_Interpolation_of_CVPR_2019_paper.html

Snelgrove, X. (2017). High-resolution multi-scale neural texture synthesis. *SA '17: SIGGRAPH Asia 2017 Technical Briefs*, 1–4.

Liu, G., Gousseau, Y., & Xia, G.-S. (2016). Texture synthesis through convolutional neural networks and spectrum constraints. *2016 23rd International Conference on Pattern Recognition (ICPR)*, 3234–3239. https://doi.org/10.1109/ICPR.2016.7900133

Sendik, O., & Cohen-Or, D. (2017). Deep Correlations for Texture Synthesis. *ACM Trans. Graph.*, *36*(5), 161:1–161:15. https://doi.org/10.1145/3015461

Berger, G., & Memisevic, R. (2017). Incorporating long-range consistency in CNN-based texture generation. *International Conference on Learning Representations*. https://doi.org/10.48550/arXiv.1606.01286

Heitz, E., Vanhoey, K., Chambon, T., & Belcour, L. (2021). A sliced wasserstein loss for neural texture synthesis. *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 9407–9415.

Vacher, J., Davila, A., Kohn, A., & Coen-Cagli, R. (2020). Texture Interpolation for Probing Visual Perception. *Advances in Neural Information Processing Systems*, *33*, 22146–22157. Retrieved June 10, 2024, from https://proceedings.neurips.cc/paper/2020/hash/fba9d88164f3e2d9109ee770223212a0-Abstract.html

Yamins, D. K. L., Hong, H., C. F. Cadieu, C., Solomon, E. A., & DiCarlo, D. S. J. J. (2014). Performance-optimized hierarchical models predict neural responses in higher visual cortex. *Proceedings of the National Academy of Sciences*, *111*(23), 8619–8624.

Cadieu, C. F., Hong, H., Yamins, D. L. K., Pinto, N., Ardila, D., Solomon, E. A., Majaj, N., & DiCarlo, J. J. (2014). Deep neural networks rival the representation of primate it

cortical neurons. *Proceedings of the National Academy of Sciences*, *111*(23), 8519–8524.

Khaligh-Razavi, S.-M., & Kriegeskorte, N. (2014). Deep supervised, but not unsupervised, models may explain it cortical representation. *PNAS*, *111*(23), 8619–8624.

Cadena, S. A., Denfield, G. H., Walker, E. Y., Gatys, L. A., Tolias, A. S., Bethge, M., & Ecker, A. S. (2019). Deep convolutional models improve predictions of macaque v1 responses to natural images. *PLOS Computational Biology*, *15*(4), e1006897. https://doi.org/10.1371/journal.pcbi.1006897

Schrimpf, M., Kubilius, J., Lee, M. J., Murty, N. A. R., Ajemian, R., & DiCarlo, J. J. (2020). Integrative benchmarking to advance neurally mechanistic models of human intelligence. *Neuron*. https://www.cell.com/neuron/fulltext/S0896-6273(20)30605-X

Bashivan, P., Kar, K., & DiCarlo, J. J. (2019). Neural population control via deep image synthesis. *Science*, *364*(6439), eaav9436. https://doi.org/10.1126/science.aav9436

Storrs, K. R., Kietzmann, T. C., Walther, A., Mehrer, J., & Kriegeskorte, N. (2021). Diverse Deep Neural Networks All Predict Human Inferior Temporal Cortex Well, After Training and Fitting. *Journal of Cognitive Neuroscience*, *33*(10), 2044–2064. https://doi.org/10.1162/jocn_a_01755

Güçlü, U., & Gerven, M. A. J. van. (2015). Deep Neural Networks Reveal a Gradient in the Complexity of Neural Representations across the Ventral Stream. *Journal of Neuroscience*, *35*(27), 10005–10014. https://doi.org/10.1523/JNEUROSCI.5023-14.2015

Cadena, S. A., Sinz, F. H., Muhammad, T., Froudarakis, E., Cobos, E., Walker, E. Y., Reimer, J., Bethge, M., Tolias, A. S., & Ecker, A. S. (2019). How well do deep neural networks trained on object recognition characterize the mouse visual system? *Real Neurons & Hidden Units: Future directions at the intersection of neuroscience and AI, NeurIPS 2019 Workshop*.

Nayebi, A., Kong, N. C. L., Zhuang, C., Gardner, J. L., Norcia, A. M., & Yamins, D. L. K. (2023). Mouse visual cortex as a limited resource system that self-learns an ecologically-general representation. *PLOS Computational Biology*, *19*(10), e1011506. https://doi.org/10.1371/journal.pcbi.1011506

Muratore, P., Tafazoli, S., Piasini, E., Laio, A., & Zoccolan, D. (2022). Prune and distill: Similar reformatting of image information along rat visual cortex and deep neural networks. *Advances in Neural Information Processing Systems*, *35*, 30206–30218.

Muratore, P., Alemi, A., & Zoccolan, D. (2025). Unraveling the complexity of rat object vision requires a full convolutional network and beyond [Patterns (N Y)]. *Patterns*, *6*(2), 101149.

Feather, J., Leclerc, G., Mądry, A., & McDermott, J. H. (2023). Model metamers reveal divergent invariances between biological and artificial neural networks. *Nature Neuroscience*, *26*(11), 2017–2034. https://doi.org/10.1038/s41593-023-01442-0

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. Retrieved December 8, 2025, from https://proceedings.neurips.cc/paper/2021/hash/c8877cff22082a16395a57e97232bb6f-Abstract.html

Wichmann, F. A., & Geirhos, R. (2023). Are Deep Neural Networks Adequate Behavioral Models of Human Visual Perception? *Annual Review of Vision Science*, *9*, 501–524. https://doi.org/10.1146/annurev-vision-120522-031739

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. https://doi.org/10.1523/JNEUROSCI.0388-18.2018

Vinken, K., Boix, X., & Kreiman, G. (2020). Incorporating intrinsic suppression in deep neural networks captures dynamics of adaptation in neurophysiology and perception. *Science Advances*, *6*(42), eabd4205. https://doi.org/10.1126/sciadv.abd4205

Turner, M. R. (1986). Texture discrimination by Gabor functions. *Biological Cybernetics*, *55*(2), 71–82. https://doi.org/10.1007/BF00341922

Malik, J., & Perona, P. (1990). Preattentive texture discrimination with early vision mechanisms. *JOSA A*, *7*(5), 923–932. https://doi.org/10.1364/JOSAA.7.000923

Freeman, J., & Simoncelli, E. P. (2011). Metamers of the ventral stream. *Nature Neuroscience*, *14*(9), 1195–1201. https://doi.org/10.1038/nn.2889

Freeman, J., Ziemba, C. M., Heeger, D. J., Simoncelli, E. P., & Movshon, J. A. (2013). A functional and perceptual signature of the second visual area in primates. *Nature Neuroscience*, *16*(7), 974–981. https://doi.org/10.1038/nn.3402

Yu, Y., Schmid, A. M., & Victor, J. D. (2015). Visual processing of informative multipoint correlations arises primarily in v2. *eLife*, *4*, e06604.

Okazawa, G., Tajima, S., & Komatsu, H. (2017). Gradual Development of Visual Texture-Selective Properties Between Macaque Areas V2 and V4. *Cerebral Cortex*, *27*(10), 4867–4880. https://doi.org/10.1093/cercor/bhw282

Ziemba, C. M., Freeman, J., Movshon, J. A., & Simoncelli, E. P. (2016). Selectivity and tolerance for visual texture in macaque v2. *Proceedings of the National Academy of Sciences*, *113*(22).

Henderson, M. M., Tarr, M. J., & Wehbe, L. (2023). A texture statistics encoding model reveals hierarchical feature selectivity across human visual cortex. *The Journal of Neuroscience*, *43*, 4144–4161.

Matteucci, G., Piasini, E., & Zoccolan, D. (2024). Unsupervised learning of mid-level visual representations. *Current Opinion in Neurobiology*, *84*, 102834.

Ziemba, C. M., Goris, R. L. T., Stine, G. M., Perez, R. K., Simoncelli, E. P., & Movshon, J. A. (2024). Neuronal and behavioral responses to naturalistic texture images in macaque monkeys. *Journal of Neuroscience*, *44*(42).

Wang, Y., Holynski, A., Curless, B. L., & Seitz, S. M. (2024). *Infinite Texture: Text-guided High Resolution Diffusion Texture Synthesis* (1). https://doi.org/10.48550/ARXIV.2405.08210

Cimpoi, M., Maji, S., Kokkinos, I., Mohamed, S., & Vedaldi, A. (2014). Describing Textures in the Wild. *2014 IEEE Conference on Computer Vision and Pattern Recognition*, 3606–3613. https://doi.org/10.1109/CVPR.2014.461

Ansel, J., Yang, E., He, H., Gimelshein, N., Jain, A., Voznesensky, M., Bao, B., Bell, P., Berard, D., Burovski, E., Chauhan, G., Chourdia, A., Constable, W., Desmaison, A., DeVito, Z., Ellison, E., Feng, W., Gong, J., Gschwind, M., ... Chintala, S. (2024). PyTorch 2: Faster Machine Learning Through Dynamic Python Bytecode Transformation and Graph Compilation. *Proceedings of the 29th ACM International Conference on Architectural Support for Programming Languages and Operating Systems, Volume 2*, 929–947. https://doi.org/10.1145/3620665.3640366

TorchVision maintainers and contributors. (2016, November). *TorchVision: PyTorch's Computer Vision library*. https://github.com/pytorch/vision

Krizhevsky, A., Nair, V., & Hinton, G. (1994). The cifar-10 dataset. https://www.cs.toronto.edu/~kriz/cifar.html

Huang, G., Liu, Z., Q., L. V. D. M. K., & Weinberger. (2017). Densely connected convolutional networks. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 4700–4708.

Howard, A. G., Zhu, M., Chen, B., Kalenichenko, D., Wang, W., Weyand, T., Andreetto, M., & Adam, H. (2017). Mobilenets: Efficient convolutional neural networks for mobile vision applications. *arXiv preprint arXiv:1704.04861*.

Szegedy, C., Vanhoucke, V., Ioffe, S., Shlens, J., & Wojna, Z. (2016). Rethinking the inception architecture for computer vision. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2818–2826.

He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep residual learning for image recognition. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 770–778.

Simonyan, K., & Zisserman, A. (2015). Very Deep Convolutional Networks for Large-Scale Image Recognition. *International Conference on Learning Representations*. https://doi.org/10.48550/arXiv.1409.1556

Bhushan, N., Rao, A. R., & Lohse, G. L. (1997). The Texture Lexicon: Understanding the Categorization of Visual Texture Terms and Their Relationship to Texture Images. *Cognitive Science*, *21*(2), 219–246. https://doi.org/10.1207/s15516709cog2102_4

Kriegeskorte, N., Mur, M., & Bandettini, P. A. (2008). Representational similarity analysis – connecting the branches of systems neuroscience. *Frontiers in Systems Neuroscience*, *2*, 4. https://doi.org/10.3389/neuro.06.004.2008

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., & Duchesnay, E. (2011). Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, *12*, 2825–2830.

MacKay, D. J. C. (2003). *Information theory, inference, and learning algorithms*. Cambridge University Press.

Nemenman, I., Shafee, F., & Bialek, W. (2002). Entropy and inference, revisited. *Advances in Neural Information Processing Systems 14*, 471–478.

Marsili, S. (2021). Ndd [https://github.com/simomarsili/ndd].

de Paolis, L. (2025, April). Supplementary materials: Representational dissimilarity matrix plots for texture representations in cnns. https://doi.org/10.5281/zenodo.19814745

Zhivago, K. A., & Arun, S. P. (2014). Texture discriminability in monkey inferotemporal cortex predicts human texture perception. *Journal of Neurophysiology*, *112*(11), 2745–2755. https://doi.org/10.1152/jn.00532.2014

Radford, A., Kim, J. W., Hallacy, C., Ramesh, A., Goh, G., Agarwal, S., Sastry, G., Askell, A., Mishkin, P., Clark, J., Krueger, G., & Sutskever, I. (2021). Learning transferable visual models from natural language supervision. *Proceedings of the 38th International Conference on Machine Learning*.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., & Polosukhin, I. (2017). Attention is all you need. *Advances in Neural Information Processing Systems*, *30*.

Guo, S., Deschaintre, V., Noll, D., & Roullier, A. (2022). U-attention to textures: Hierarchical hourglass vision transformer for universal texture synthesis. *Proceedings of the 19th ACM SIGGRAPH European Conference on Visual Media Production*, 1–10.

Geirhos, R., Rubisch, P., Michaelis, C., Bethge, M., Wichmann, F. A., & Brendel, W. (2019). ImageNet-trained CNNs are biased towards texture; increasing shape bias improves accuracy and robustness. https://doi.org/10.48550/ARXIV.1811.12231

Hermann, K., Chen, T., & Kornblith, S. (2020). The origins and prevalence of texture bias in convolutional neural networks. In H. Larochelle, M. Ranzato, R. Hadsell, M. Balcan, & H. Lin (Eds.), *Advances in neural information processing systems* (pp. 19000–19015, Vol. 33). Curran Associates, Inc. https://proceedings.neurips.cc/paper_files/paper/2020/file/db5f9f42a7157abe65bb145000b5871a-Paper.pdf

Islam, M. A., Kowal, M., Esser, P., Jia, S., Ommer, B., Derpanis, K. G., & Bruce, N. (2021). Shape or Texture: Understanding Discriminative Features in CNNs. https://doi.org/10.48550/ARXIV.2101.11604

Kubilius, J., Schrimpf, M., Nayebi, A., Bear, D., Yamins, D. L. K., & DiCarlo, J. J. (2018, September 4). *CORnet: Modeling the Neural Mechanisms of Core Object Recognition*. https://doi.org/10.1101/408385

9 Appendix

9.1 Loss normalization

In this section we derive the weighting rule we used to combine the Gram losses from multiple layers.

Consider the Gram matrix G for a certain layer l. Its (i, j)-th entry is

$$G_{ij} = \sum_{n=1}^{M} F_i^n F_j^n$$

where the index n runs over all elements of the i-th feature map F_i (so for instance in a typical convolutional network M would be the product of the height and width of the feature map). Assume that the individual activations of the network units are roughly normally distributed:

$$F_i^n \sim \mathcal{N}(0, 1)$$

This is a reasonable assumption in presence of batch normalization. Our goal is now to understand how the "Gram loss" for this layer is distributed. The Gram loss is

$$\mathcal{L}_l = \sum_{i=1}^{N} \sum_{j=1}^{N} \left(G_{ij}^{\text{rec}} - G_{ij}^{\text{orig}} \right)^2$$

For a fixed value of i, j and n let's define two new random variables A and B:

$$A = \left(\frac{F_i^n + F_j^n}{\sqrt{2}} \right)^2 \quad , \quad B = \left(\frac{F_i^n - F_j^n}{\sqrt{2}} \right)^2$$

from which we get

$$\frac{A - B}{2} = \frac{1}{2} \left[\frac{F_i^2 + F_j^2 + 2F_iF_j}{2} - \frac{F_i^2 + F_j^2 - 2F_iF_j}{2} \right] = \frac{1}{4} \left[4F_iFj \right] = F_iF_j$$

so we can think of G_{ij} as the sum of M iid terms, each of which is distributed like $(A-B)/2$.

By observing that $(F_i^n + F_j^n)/\sqrt{2} \sim \mathcal{N}(0,1)$, we see that A and B are each simply the square of a standard Normal. Therefore they both follow a Chi-squared distribution with one degree of freedom

$$A \sim \chi_1^2 \quad , \quad B \sim \chi_1^2$$

which means that

$$\mathbb{E}[A] = \mathbb{E}[B] = 1 \quad , \quad \mathrm{Var}[A] = \mathrm{Var}[B] = 2$$

Therefore, $A - B$ is the difference of two independent random variables with mean zero and variance 2. This implies that, for any n,

$$\mathbb{E}[2F_i^n F_j^n] = \mathbb{E}[A - B] = 0$$
$$\mathrm{Var}[2F_i^n F_j^n] = \mathrm{Var}[A] + \mathrm{Var}[B] = 4$$

and therefore

$$\mathbb{E}[F_i^n F_j^n] = 0$$
$$\mathrm{Var}[F_i^n F_j^n] = 2$$

So G_{ij} is the sum of a large number M of iid random variables, each with mean 0 and variance 2. By the central limit theorem, then,

$$G_{ij} \sim \mathcal{N}\left(0, \sqrt{2M}\right)$$

If this is true for both the original and reconstructed Gram, then

$$G_{ij}^{\mathrm{rec}} - G_{ij}^{\mathrm{orig}} \sim \mathcal{N}(0, 2\sqrt{M})$$

and therefore

$$\frac{G_{ij}^{\text{rec}} - G_{ij}^{\text{orig}}}{2\sqrt{M}} \sim \mathcal{N}(0,1)$$

Now, consider the quantity

$$\frac{\mathcal{L}_l}{4M} = \sum_{i=1}^{N}\sum_{j=1}^{N}\left(\frac{G_{ij}^{\text{rec}} - G_{ij}^{\text{orig}}}{2\sqrt{M}}\right)^2$$

It is clear that $\mathcal{L}_l/4M$ is the sum of N^2 squared standard Normal random variables. Therefore,

$$\frac{\mathcal{L}_l}{4M} \sim \chi^2_{N^2}$$

and as a consequence

$$\mathbb{E}\left[\frac{\mathcal{L}_l}{4M}\right] = N^2$$

Finally, this implies that

$$\mathbb{E}\left[\mathcal{L}^l\right] = 4MN^2$$

In other words, if the activations F_i^n are standardized, the expected value of the Gram loss for a layer with N features of size M is $4MN^2$, just as a consequence of the size and shape of the layer. Therefore, to make sure that Gram losses coming from different layers have the same weight in our total loss we should divide each layer-specific Gram loss by $4MN^2$, where M and N are the values specific to that layer.

9.2 Supplementary Tables

Table 3: Extracted layers from each CNN and relative Gram matrix dimensionality. The names refer to the terminology of the models' versions in Torchvision. For models whose layers are not associated with a name, we report the absolute index.

Model	**Extracted layers**	**Gram dimensionality**
AlexNet	`layer idx=[0, 3, 6, 8, 10]`	L0=$64(64+1)/2 = 2080$ L3=$192(192+1)/2 = 18520$ L6=$384(384+1)/2 = 73920$ L8=$256(256+1)/2 = 32896$ L10=$256(256+1)/2 = 32896$

DenseNet-121	`bn_0=features.norm0`
	`bn_1=features.denseblock1.denselayer1.norm2`
	`bn_2=features.denseblock2.denselayer1.norm2`
	`bn_3=features.denseblock3.denselayer1.norm2`
	`bn_4=features.denseblock4.denselayer1.norm2`
		bn0=$64(64+1)/2 = 2080$
		bn1=$128(128+1)/2 = 8256$
		bn2=$128(128+1)/2 = 8256$
		bn3=$128(128+1)/2 = 8256$
		bn4=$128(128+1)/2 = 8256$

DenseNet-169	`bn_0=features.norm0`
	`bn_1=features.denseblock1.denselayer1.norm2`
	`bn_2=features.denseblock2.denselayer1.norm2`
	`bn_3=features.denseblock3.denselayer1.norm2`
	`bn_4=features.denseblock4.denselayer1.norm2`
		bn0=$64(64+1)/2 = 2080$
		bn1=$128(128+1)/2 = 8256$
		bn2=$128(128+1)/2 = 8256$
		bn3=$128(128+1)/2 = 8256$
		bn4=$128(128+1)/2 = 8256$

DenseNet-201	`bn_0=features.norm0`
	`bn_1=features.denseblock1.denselayer1.norm2`
	`bn_2=features.denseblock2.denselayer1.norm2`
	`bn_3=features.denseblock3.denselayer1.norm2`
	`bn_4=features.denseblock4.denselayer1.norm2`
		bn0=$64(64+1)/2 = 2080$
		bn1=$128(128+1)/2 = 8256$
		bn2=$128(128+1)/2 = 8256$
		bn3=$128(128+1)/2 = 8256$
		bn4=$128(128+1)/2 = 8256$

Table 4: (Continued from previous page)

Model	**Extracted layers**	Gram matrix dimensionality

Inception-v3	`bn_0=Conv2d_1a_3x3.bn`
	`bn_1=Conv2d_2a_3x3.bn`
	`bn_2=Conv2d_2b_3x3.bn`
	`bn_3=Conv2d_3b_1x1.bn`
	`bn_4=Conv2d_4a_3x3.bn`
		bn0=$32(32+1)/2 = 561$
		bn1=$32(32+1)/2 = 561$
		bn2=$64(64+1)/2 = 2080$
		bn3=$80(80+1)/2 = 3240$
		bn4=$192(192+1)/2 = 185208$

ResNet18	`bn_0=layer1.0.bn1`
	`bn_1=layer1.0.bn1`
	`bn_2=layer2.0.bn1`
	`bn_3=layer3.0.bn1`
	`bn_4=layer4.0.bn1`
		bn0=$64(64+1)/2 = 2080$
		bn1=$64(64+1)/2 = 2080$
		bn2=$128(128+1)/2 = 8256$
		bn3=$256(256+1)/2 = 32896$
		bn4=$512(215+1)/2 = 131280$

ResNet34	`bn_0=layer1.0.bn1`
	`bn_1=layer1.0.bn1`
	`bn_2=layer2.0.bn1`
	`bn_3=layer3.0.bn1`
	`bn_4=layer4.0.bn1`
		bn0=$64(64+1)/2 = 2080$
		bn1=$64(64+1)/2 = 2080$
		bn2=$128(128+1)/2 = 8256$
		bn3=$256(256+1)/2 = 32896$
		bn4=$512(215+1)/2 = 131280$

ResNet50	`bn_0=layer1.0.bn1`
	`bn_1=layer1.0.bn1`
	`bn_2=layer2.0.bn1`
	`bn_3=layer3.0.bn1`
	`bn_4=layer4.0.bn1`
		bn0=$64(64+1)/2 = 2080$
		bn1=$64(64+1)/2 = 2080$
		bn2=$128(128+1)/2 = 8256$
		bn3=$256(256+1)/2 = 32896$
		bn4=$512(215+1)/2 = 131280$

ResNet101	`bn_0=layer1.0.bn1`
	`bn_1=layer1.0.bn1`
	`bn_2=layer2.0.bn1`
	`bn_3=layer3.0.bn1`
	`bn_4=layer4.0.bn1`
		bn0=$64(64+1)/2 = 2080$
		bn1=$64(64+1)/2 = 2080$
		bn2=$128(128+1)/2 = 8256$
		bn3=$256(256+1)/2 = 32896$
		bn4=$512(215+1)/2 = 131280$

ResNet152	`bn_0=layer1.0.bn1`
	`bn_1=layer1.0.bn1`
	`bn_2=layer2.0.bn1`
	`bn_3=layer3.0.bn1`
	`bn_4=layer4.0.bn1`
		bn0=$64(64+1)/2 = 2080$
		bn1=$64(64+1)/2 = 2080$
		bn2=$128(128+1)/2 = 8256$
		bn3=$256(256+1)/2 = 32896$
		bn4=$512(215+1)/2 = 131280$

VGG-16	`bn idx=[1, 8, 15, 25, 35]`	bn1=$64(64+1)/2 = 2080$ bn8=$128(128+1)/2 = 8256$ bn15=$256(256+1)/2 = 32896$ bn25=$512(215+1)/2 = 131280$ bn35=$512(215+1)/2 = 131280$
VGG-19	`bn idx=[1, 8, 15, 28, 41]`	bn1=$64(64+1)/2 = 2080$ bn8=$128(128+1)/2 = 8256$ bn15=$256(256+1)/2 = 32896$ bn25=$512(215+1)/2 = 131280$ bn35=$512(215+1)/2 = 131280$

Table 5: Summary of the values from Brain-Score benchmarks that we selected for our study (columns), for the 12 CNNs from our pool (rows). In the second column and second row, for models and benchmarks respectively, we report the .csv coordinates of the leaderboard. Our Brain-Score values were exported from the leaderboard website on September 8th 2025.

Model	**Model coordinates**	**average vision**	**neural vision**	**behavior vision**	**V1**	**V2**	**V4**	**IT**
Benchmark coordinates		D2	E2	DB2	F2	AM2	AP2	AV2
AlexNet	B290	0.171	0.281	0.06	0.338	0.176	0.355	0.255
Densenet-121	B128	0.311	**0.344**	0.278	**0.388**	**0.275**	**0.396**	0.315
DenseNet-169	B158	0.276	0.305	0.248	0.346	0.178	0.384	0.312
DenseNet-201	B94	0.318	0.293	0.343	0.34	0.19	0.388	0.253
Inception-v3	B84	0.33	0.3	0.358	0.321	0.169	**0.388**	0.324
Resnet18	B354	0.152	0.221	0.084	0.091	0.151	0.377	0.263
Resnet34	B357	0.147	0.206	0.089	0.077	0.143	0.359	0.244
Resnet50	B60	**0.362**	0.326	**0.397**	0.356	0.191	0.395	**0.363**
Resnet101	B176	0.26	0.081	0.44	0.127	0.068	0.062	0.066
Resnet152	B228	0.202	0.044	0.359	0.021	0.074	0.012	0.069
VGG-16	B104	0.322	0.288	0.354	0.319	0.147	0.385	0.303
VGG-19	B148	0.305	0.267	0.343	0.285	0.13	0.367	0.285

9.3 Supplementary Figures

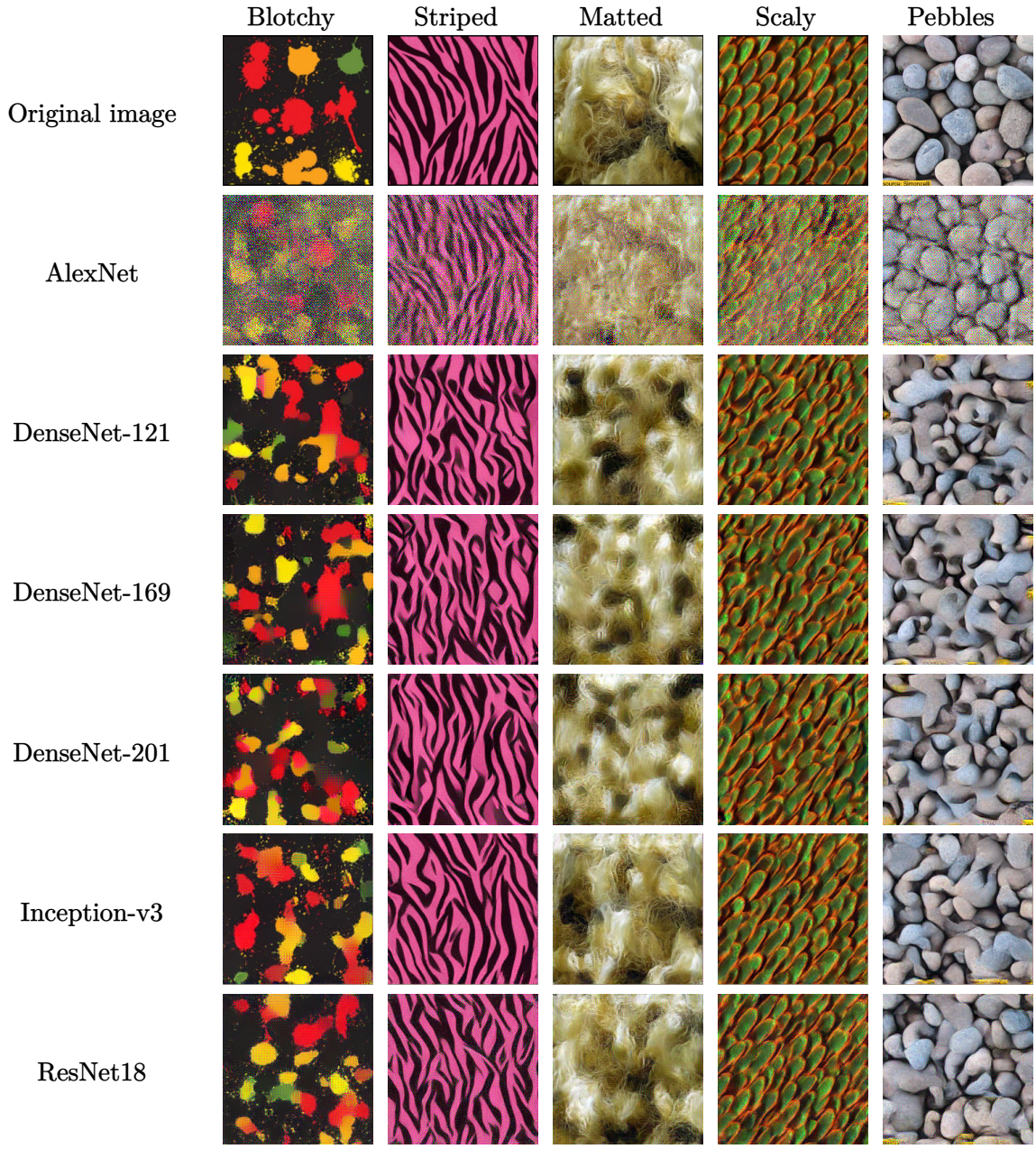


Figure 1: Textures generated with the Gatys algorithm with the 12 CNNs of choice (rows). The 4 original images (top row) belong to 4 different classes in DTD (columns: *blotchy; striped; matted; scaly*). The last column represents one of the images originally used by Gatys et al., which we synthesized as a reference.

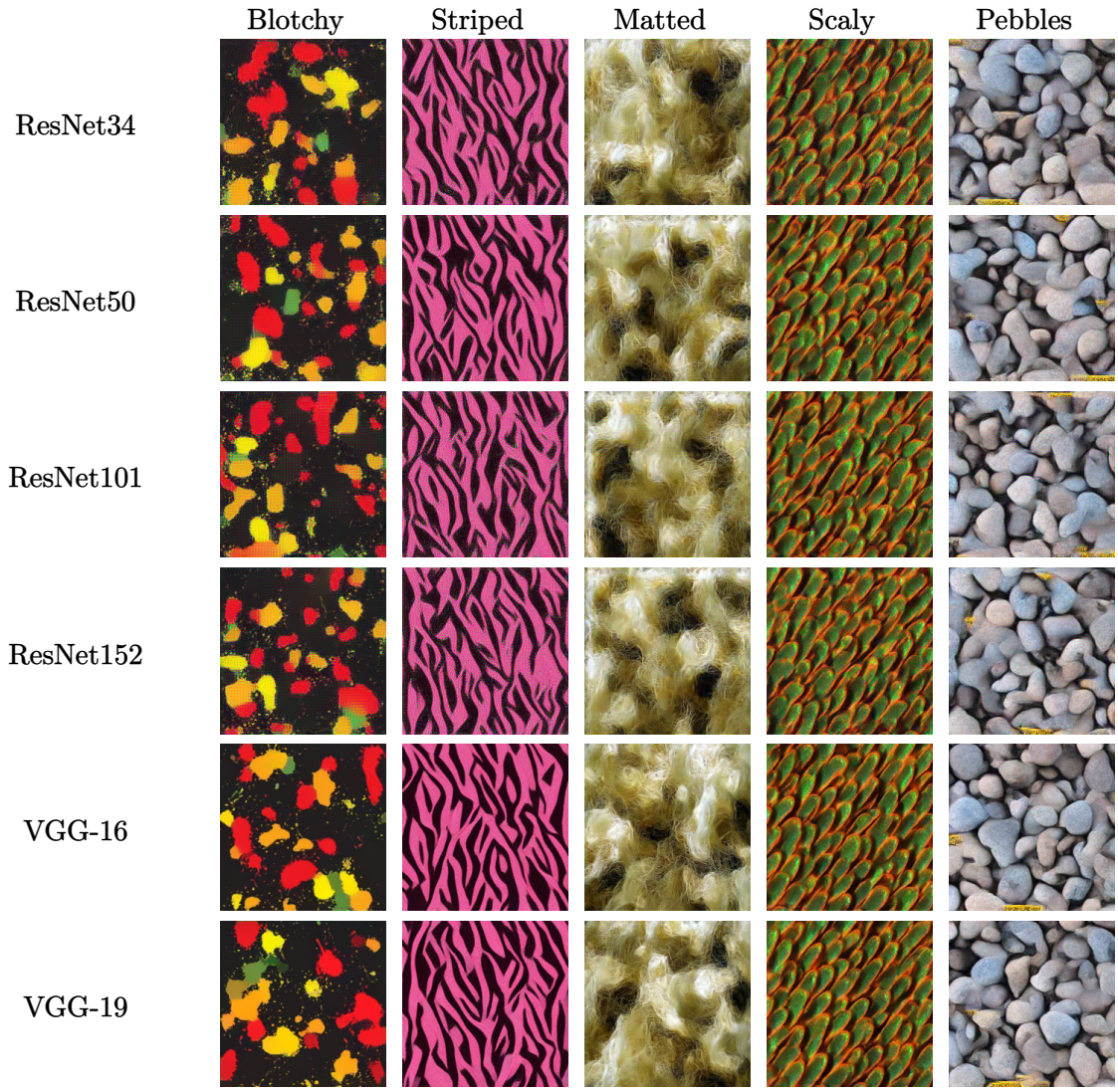


Figure 2: (continued from previous page)