Pattern Recognition 145 (2024) 109889

Contents lists available at ScienceDirect

Pattern Recognition

journal homepage: www.elsevier.com/locate/pr

# Minimum description length clustering to measure meaningful image complexity

Louis Mahon [a,c,*], Thomas Lukasiewicz [b,c]

[a] School of Informatics, University of Edinburgh, UK
[b] Institute of Logic and Computation, Vienna University of Technology, Austria
[c] Department of Computer Science, University of Oxford, UK

ARTICLE INFO

*Keywords:*
Meaningful complexity
Clustering
Image complexity
Minimum description length
Machine learning
Information theory

ABSTRACT

We present a new image complexity metric. Existing complexity metrics cannot distinguish meaningful content from noise, and give a high score to white noise images, which contain no meaningful information. We use the minimum description length principle to determine the number of clusters and designate certain points as outliers and, hence, correctly assign white noise a low score. The presented method is a step towards humans' ability to detect when data contain a meaningful pattern. It also has similarities to theoretical ideas for measuring meaningful complexity. We conduct experiments on seven different sets of images, which show that our method assigns the most accurate scores to all images considered. Additionally, comparing the different levels of the hierarchy of clusters can reveal how complexity manifests at different scales, from local detail to global structure. We then present ablation studies showing the contribution of the components of our method, and that it continues to assign reasonable scores when the inputs are modified in certain ways, including the addition of Gaussian noise and the lowering of the resolution. Code is available at https://github.com/Lou1sM/meaningful_image_complexity.

## 1. Introduction

Pattern recognition and machine learning typically concern the case where we already know that the given data have some pattern, and we want a method that can automatically discover what the pattern is. In this paper, we address the problem of determining whether the data have any meaningful pattern to begin with, or whether it contains no or only very simple systematic structure, a problem that might be called pattern detection. Humans are highly proficient at recognizing patterns such as words in a speech signal or objects in a video, but even in the absence of explicit recognition, we can often detect when there is a pattern there at all. For example, we hear speech in a foreign language that we do not understand, but can still tell that there is some meaningful structure that could be recognized, unlike ambient city noise or white noise on the radio. Similarly, we may see an abstruse technical diagram and realize that there is something meaningfully complex there, even if we do not know what it is, unlike an image of a blank wall. We study this ability to recognize complexity through the development of a complexity metric. Data with a rich pattern should be scored as high complexity, whatever that pattern is, and unstructured or simply structured data should be scored as low complexity.

There is unavoidable subjectivity in measuring complexity quantitatively. This is always the case when defining a new metric. We cannot begin the investigation of a complexity metric by defining what complexity is, that would be to put the cart before the horse. Inevitably, the investigation involves exploring what complexity is, not just how to measure it, that is, the definition of complexity and the specification of a complexity metric are two sides of the same thing, the latter is really an instantiation of the former. For example, if one defines complex images as those in which there is a high variation between all the pixel values, then it is natural to use the entropy of pixel values as a complexity metric; or if one defines complex images as those in which nearby pixel intensities tend to be very different from each other, then another metric is the obvious choice (grey-level co-occurrence matrix; see Section 4). This renders unavailable the standard blueprint for applied machine learning research of showing that a novel method outperforms existing methods on some quantifiable task or benchmark, because the field does not have such a benchmark for measuring image complexity. What we do have is a vague idea of what complexity is, vague but still powerful and important. The task is to translate this vague idea into something computable.

[*] Corresponding author at: School of Informatics, University of Edinburgh, UK.
*E-mail address:* oneillml@tcd.ie (L. Mahon).

https://doi.org/10.1016/j.patcog.2023.109889
Received 27 February 2023; Received in revised form 24 July 2023; Accepted 10 August 2023
Available online 19 August 2023

There are several applications that benefit from being able to measure visual complexity. Remote sensing often gathers large numbers of images, most of which depict nothing interesting, such as empty desert or ocean, but occasionally capture important information, such as the gathering of fauna or sudden change in flora. It is useful to automatically filter out the simple images before manual inspection [1–3]. In the field of psychometrics, there is interest in understanding what humans will find visually interesting or aesthetic, and this often involves a component modelling complexity, e.g., [4,5]. Relatedly, complexity perception influences how humans regard digital interfaces such as graphical websites, and automated complexity measures have been proposed to guide interface design [6]. Being able to distinguish signal from noise is especially relevant to remote sensing, where images often become corrupted by noise due to the sensing equipment or various post-processing steps [7–9]. Much work has been done to reduce noise in remote sensing images [10,11] and to improve the robustness of image processing methods to noise [12,13].

Most existing techniques for quantifying and measuring image complexity (discussed further in Section 2) are based on measuring intricacy, the idea being that the more intricate it is and the more dissimilar its parts, the more complex it is. This is relatively easy to measure, but it is incomplete for two reasons. Firstly, and most importantly, it does not distinguish between meaningful intricacy (signal) and meaningless intricacy (noise). Using intricacy as a measure of complexity means that a white-noise image, where the pixel values are chosen independently at random, is measured as highly, perhaps even maximally, complex, because there is a high degree of difference between neighbouring pixels. Note that this problem even holds for Kolmogorov complexity, where a standard result is that most bitstrings are almost incompressible, and so, with very high probability, a random bitstring will receive near maximum complexity score. There has been theoretical work to divide the Kolmogorov complexity into meaningful information and noise, using, e.g., ‘sophistication’ [14,15] or ‘effective complexity’ [16,17]. Our applied method can be thought of as an instantiation of the high-level idea in these theoretical methods. We return to this comparison in Section 3.3.

A second disadvantage of variation as a complexity measure is that it cannot capture the fact that images can have a different complexity at different scales. A blurry photograph of a complex scene, for example, is locally simple but globally complex, while a finely-detailed but repetitive pattern is the opposite.

Rather than meaning a high degree of variation, we instead conceive of complexity as ‘taking a large number of steps to assemble’. An image can be thought of as being built out of pixels, local groups of pixels are combined to form patches, groups of neighbouring patches are combined to form super-patches etc. Quantifying complexity based on the assembly process is the approach taken in the theory of assembly pathways [18,19], originally for the purpose of quantifying the complexity of molecules to aid in the search for extraterrestrial life [20,21]. The pathway assembly index of an object is the minimum number of combinations needed to produce it from simple parts, where repeated components can be reused without adding to the count. In order to discretize the structure of the image and allow the assembly index to be applied, we employ clustering. For the first level of the hierarchy, we cluster the pixel values and replace them with their cluster index. For higher levels, we cluster the multisets of cluster indices from the level below.

Another advantage of discretizing is that we can then easily compute entropy. Taking the entropy of a continuous image is difficult, we must use some approximation of differential entropy [22,23]. In our case, however, we are dealing with discrete cluster labels, so we need only compute the entropy of a categorical distribution, which is easy. At each scale (i.e., hierarchy level), we compute the entropy of the multisets of cluster indices across the image to quantify complexity. The total complexity score is the sum of this entropy at each scale. We can also examine the entropy for individual scales to get an indication of the local vs. global complexity in the image: low scales (i.e., small patch sizes) measure local complexity, whereas higher scales capture more global structure (as shown in Section 4.4).

At each level of the hierarchy, the cluster indices produced, and hence the complexity score, depend on $K$, the number of clusters in the clustering model. We choose $K$ in a theoretically sound way via the minimum description length (MDL) principle [24]. MDL says that we should choose the model that can completely represent the given data in the fewest number of bits. Clustering can be interpreted as compression, where we encode each point by its cluster index, along with the residual error of how it differs from the centroid of that cluster. Treating each cluster as a probability distribution, and employing the Kraft–McMillan inequality, we see that the residual error for a point $x$ under the cluster probability distribution $p$ can be represented using $-\log p(x)$ bits. Representing the data under the clustering model takes $-\sum_x \log p(x)$ bits, plus the number of bits to represent the cluster indices and the model itself. Increasing $K$ reduces the average residual error, but increases the size of the indices and the model itself. By MDL, we choose $K$ so as to minimize the total size. MDL is a key component in filtering out noise from our complexity measure. In white noise images, where there is no meaningful or consistent pattern between different points, MDL finds only one cluster, because the small reduction in residual error from encoding more is not worth the extra cost, so the image ends up with a very low complexity score. We both prove this mathematically and observe it empirically.

There are two important similarities between the computational method presented here with human visual perception. The first is hierarchical processing. The visual cortex is divided into five areas, V1-V5. Each takes as input the integrated information output from the previous area, and has progressively larger receptive fields [25]. This allows humans to perceive each element of a visual scene as composed of smaller elements, e.g., a photograph is composed of man, road, bicycle; the bicycle is composed of wheels, frame, saddle; the wheels are composed of spokes, tyre, valve etc. Similarly, our method processes progressively larger patches of features and passes the output of each level of the hierarchy to the level above as input. The approach of treating images as hierarchically structured underpins convolutional neural networks, and has also been leveraged for image segmentation [26], face recognition [27], and image inpainting [28].

The second important similarity to human perception is the role of simplicity. Many authors have argued that human perception looks for the simplest interpretation of visual data [29–31]. Similarly, our use of the minimum description length principle allows us to ignore certain parts of the image and group together other parts in a way that produces the most parsimonious representation overall. This is not a feature of CNNs, but there are some existing works that use MDL clustering for other image processing tasks, such as image segmentation [32], shape modelling [33], or key-frame extraction [34]. As well being for different tasks, these methods differ from ours in that they do not exclude certain parts of the image as outliers, and are not hierarchical in the sense of passing the output from lower levels to higher levels as input.

The main contributions of this paper are briefly summarized below.

- We propose a novel theoretically sound measure of image complexity and discuss its relationship to ideas in algorithmic information theory.
- We test our method empirically on seven image datasets, four public and three synthetic datasets that we created. We show that our method performs as desired in distinguishing images from different datasets. In particular, our method is able to correctly assign a low complexity to white noise, in contrast to existing methods, which assign it a high complexity.
- We support these results theoretically by proving that, given normally distributed clusters, MDL will find just a single cluster when the clustering model is fit on white noise, and so our method will assign a low score.

- We conduct a further set of experiments, showing how our method can measure complexity at different scales in the image, how it performs when Gaussian noise is added to the image or the resolution of the image is reduced, and how it responds to an increasing fractal dimension of a fractal image.

The rest of this paper is organized as follows. Section 2 gives an overview of related work. Section 3 describes our method, and Section 4 presents our empirical evaluation. Finally, Section 5 summarizes our findings and suggests directions for future work.

## 2. Related work

### *2.1. Measuring image complexity*

**Fractal dimension** is a property of curves, which in some sense measures their complexity. It can be applied to an image by first binarizing with a threshold, then taking the boundary between white and black pixels as a curve and computing its Minkowski–Bouligand dimension. Lam et al. [35] explore the use of fractal dimension to measure the complexity of satellite images, and Sun et al. [2] consider the application to remote sensing images more generally. Both also contain a detailed account of methods that use fractal dimension for image complexity. Forsythe et al. [36] compare fractal dimension against human judgements of the complexity and beauty of visual art.

**File compression ratio** is the ratio between the size of a compressed file under a chosen compression algorithm, and the size of the uncompressed original. Marin and Leder [37] measure image complexity using the file compression ratio, under two compression algorithms: GIF, which is lossy, and TIFF, which is lossless. The compression ratio was compared to human judgements of complexity, on the International Affective Picture System. It is also used as a complexity measure in Forsythe et al. [36] and by Machado et al. [38]. The former investigate the ability of JPEG-ratio, GIF-ratio, and a novel ‘perimeter detection’ method to predict human judgements of complexity in visual art. The latter explore various combinations of compression algorithms with automated edge detection, and compares the results to human judgements of complexity. The authors find the best results using Sobel and Canny filters, followed by JPEG compression.

Carballal et al. [5] test the accuracy of various supervised machine learning models of complexity by annotating art and non-art images with human judgements of complexity, then regressing these annotations using a machine learning algorithm that includes feature selection. This was repeated a number of times, and the accuracy of a given feature was taken to be the fraction of times it was selected by the feature selection algorithm.

An alternative method is to use the **gradient of pixel intensities** across the image. This is the approach taken by Redies et al. [39]. The gradient is computed separately for each of the RGB channels, and the gradient at a pixel is taken to be the maximum across the three channels. The average gradient across the entire image is then taken as a measure of complexity. This is again applied to quantifying aesthetic judgements of visual art, this time as part of the Birkhoff-like measure [40], which characterizes beauty as the ratio of order and complexity. A final method to consider is **the Fourier transform**, as used by Khan et al. [41]. The idea is that the more high-frequency components present in the power spectrum, the more complex the image. The authors investigate using both the mean and the median of the power spectrum, and find best results for the median. The application in this case is guiding neural architecture search, the claim being that one should first measure the complexity of a given image dataset, and then use the result to inform architecture design.

### *2.2. Relation to other tasks in pattern recognition*

Our method for measuring image complexity begins by assigning a cluster label to each pixel. It can therefore be interpreted as producing a **segmentation** of the image, by defining a segment as a contiguous set of pixels with the same cluster label. There are several common approaches to image segmentation, such as modified graph-cutting algorithms [42] or component trees [26]. Among these, the segmentation provided by our method falls into the category that uses only colour and texture information [43], and also relates closely to those methods that use the minimum description length principle [44]. We do not directly explore the segmentation quality of our method, but Fig. 2 gives a visual indication of the segments produced.

**Clustering** is a fundamental task in pattern recognition and machine learning that learns the structure of data in a fully unsupervised way. Current research topics in clustering include the use of deep neural networks such as CNNs [45,46] or graph neural networks (GNNs) [47], and exploring alternatives to the standard centroid-based clustering, e.g., density-based clustering [48,49]. Our method relates especially to work on reducing the need for hyperparameters such as cluster number [50,51].

**Compression** is of strong theoretical and practical interest to pattern recognition, and has been used specifically to measure data complexity by the works described in Section 2.1. Aside from standard algorithms such as JPEG, common approaches to image compression include deep learning [52,53] and variants of the wavelet transform [54]. By combining clustering with MDL, we treat clustering as a form of compression (see [55] for a discussion of clustering as compression) and thus illustrate the connection between compression and data complexity.

## 3. Method

This section gives an overview of the minimum description length principle as it is used in our method, then describes our method in detail with the aid of a worked example, and compares our approach, on a high level, to existing theoretical work on meaningful data complexity.

### *3.1. Minimum description length patch clustering*

Our measure of complexity uses a form of clustering based on description length (DL), i.e., the number of bits needed to specify the given data. Description length is relative to an encoding scheme, and via the Kraft–MacMillan inequality, this corresponds to a probability distribution. Specifically, the Kraft-MacMillan inequality says that, under the optimal encoding scheme (optimal in the sense of being shortest on average) of a probability distribution $p(\cdot)$, the description length of a point $x$ is $-\log p(x)$. We model the probability distribution with a Gaussian mixture model (GMM), because (a) we seek a distribution-based clustering model, and a GMM is by far the most commonly used distribution-based clustering model, (b) choosing a GMM is equivalent to simply modelling the distribution within each cluster as normal, and this has theoretical justifications in the central limit theorem and maximization of differential entropy [56]. The description length is therefore relative to the means $\mu = (\mu_i)_{1 \le i \le K}$ and the covariances $\Sigma = (\Sigma_i)_{1 \le i \le K}$ of this GMM. The probability of a point $x$ under its assigned component of the mixture model $(\mu, \Sigma)$ is given by

$$p(x, \mu, \Sigma) = \max_{1 \le k \le K} \frac{\exp(-\frac{1}{2}(x - \mu_k)\Sigma_k^{-1}(x - \mu_k))}{\sqrt{(2\pi)^d |\Sigma_k|}} , \qquad (1)$$

where $\mu_k$ and $\sigma_k$ are, respectively, the mean and covariance of the $k$th component, and $d$ is the dimensionality of the data. Specifying $x$ under $p$ requires first indexing the cluster to which $x$ belongs and then encoding $x$ under the probability distribution of that cluster, which we refer to as the residual error. The latter was just shown to take $-\log p(x, \mu, \Sigma)$ bits. Similarly, the length of the former depends on the encoding scheme for, and equivalently the probability distribution over, the indices $1, \dots, K$, which can be taken empirically from the data. Specifically, the length of encoding which cluster $x$ belongs to is $-\log n_k/N$, so the total description length is then

$$\min_{1\leq k\leq K} -\log \tfrac{n_k}{N} + \tfrac{1}{2}(x-\mu_k)\Sigma_k^{-1}(x-\mu_k) + \tfrac{1}{2}\log{(2\pi)^d|\Sigma_k|}\,, \tag{2}$$

where $k$ is the index of the cluster that it belongs to, $n_k$ is the number of points belonging to cluster $k$, and $N$ is the total number of data points. As discussed in Section 3.3, we can conceive of the $-\log n_k/N$ term as the meaningful portion of this description and the remainder as the meaningless portion.

### 3.1.1. Differential description length

Because the multivariate normal distributions composing the GMM are continuous probability density functions (pdf), instead of probability mass functions as in the discrete case, it is possible that $p(x, \mu, \Sigma) > 1$. Note that this is always a possibility for pdfs, e.g., the univariate Normal distribution

$$\mathcal{N}(\mu, \tfrac{1}{5\sqrt{2\pi}})$$

has the value 5 at $x = \mu$. In these cases, the Kraft-MacMillan inequality would seem to suggest that the corresponding encoding scheme can represent $x$ with a strictly negative number of bits, which of course is not possible. The apparent contradiction is resolved by making explicit the precision with which $x$ is to be encoded. Completely specifying any real number is not possible with a finite number of bits, instead one can only specify an extended region $D_x \subset \mathbb{R}^n$, which contains $x$. The number of required bits is then determined by the probability mass inside $D_x$, which is given by

$$p_m(D_x, \mu, \Sigma) = \int_{D_x} p(z, \mu, \Sigma)dz\,. \tag{3}$$

Let $\epsilon$ be the coordinate-wise precision for specifying $x$, i.e., set $D_x$ to be a hypercube of side-length $\epsilon$. The probability mass in $D_x$ is then approximated as $p(x, \mu, \Sigma)\epsilon^d$, giving the description length

$$-d\log\epsilon - \log p(x, \mu, \Sigma) - \log n_k/N\,. \tag{4}$$

The additional term $-d\log\epsilon$ will be higher for smaller $\epsilon$, and will always increase the total description length to be positive even if $-\log(p(x, \mu, \Sigma)) < 0$. That it will be large enough to counterbalance $-\log(p(x, \mu, \Sigma))$ is clear from observing that the probability mass in (3) is never greater than 1. Note that the additional $-d\log\epsilon$ term is independent of the pdf itself. Thus, it can be ignored when using MDL and comparing different pdfs (which correspond to different fit clustering models). That is, when invoking the MDL principle, it is sufficient to look only at the term remaining after the $-d\log\epsilon$ term has been removed:

$$-\log(p(x, \mu, \Sigma)) - \log n_k/N\,. \tag{5}$$

We refer to this remaining quantity as the differential description length. We define the differential description length (DDL) to be the negative logarithm of the probability density. It is the continuous analogue of the description length, just as differential entropy is the continuous analogue of entropy. Similarly to differential entropy, DDL can be negative. This happens precisely when the probability density is greater than 1, as just discussed. DDL is related to the description length as follows: for a point $x$ with DDL $D$, the number of bits required to specify it to a precision $\epsilon$ is $\max(\{0, -D - d\log\epsilon\})$. The max is required to account for the case where the region specified by the precision $\epsilon$ is larger than the interval in which we already know $x$ to lie. For example, if we assume a priori that $x$ is uniformly distributed on $[0, 1]$, in which case all points have DDL 0 under the prior distribution, and then we try to specify to precision 2, we will end up with

$$-D - d\log\epsilon = 0 - \log 2 = -1.$$

Taking the maximum with zero means that, in such cases, we obtain the correct result of 0.

### 3.1.2. Determining outliers

As well as choosing the number of clusters (see Section 3.1.3), we can use the minimum description length (MDL) principle to determine which points are outliers with respect to the given model. An outlier can be defined as one that takes more bits to specify under the model than it does to specify directly, independently of the model. We can always specify (up to finite precision $\epsilon$) any point directly using the same discretizing reasoning as above. First, restrict attention to some bounded region of $\mathbb{R}^n$, which is large enough so that we can assume that it will contain all values the data could have.[1] Once this bounded region is specified, partition it into a set of small regions–hypercubes with side-length $\epsilon$–and then specify a point $x$ by indexing the unique region that contains $x$. The number of possible regions is

$$\left(\frac{a_{max} - a_{min}}{\epsilon}\right)^d,$$

where $d$ is the dimensionality of the data, and $a_{max}$ and $a_{min}$ are the maximum and minimum values, respectively, that appear anywhere in the image. The number of bits to specify a point directly is then

$$\log\left(\frac{a_{max} - a_{min}}{\epsilon}\right)^d = -d\log\epsilon + d\log(a_{max} - a_{min})\,. \tag{6}$$

Again, we can ignore the precision value $\epsilon$, because it will appear equally in both description length under the model and the description length from indexing the hypercube. Instead, we can use the differential description length. The indexing of the $\epsilon$ hypercube in (6) is equivalent, when using the differential description length, to using a uniform prior on $[a_{min}, a_{max}]^d$. Under such a distribution, the DDL of any point is $d\log(a_{max} - a_{min})$. Comparing to the DDL under the model, as in (5), a point is an outlier iff

$$-\log(p(x, \mu, \Sigma)) - \log\frac{n_k}{N} > d\log(a_{max} - a_{min}) \iff \tag{7}$$

$$\iff p(x, \mu, \Sigma)\frac{n_k}{K} < (a_{max} - a_{min})^{-d}\,, \tag{8}$$

where, as above, $n_k$ is the number of points assigned to the same cluster as $x$. We can then define the total DDL of $x$, where $x$ can be specified either directly or using the encoding scheme from the model, as

$$D(x, \mu, \Sigma) = \min\left(d\log(a_{max} - a_{min}), -\log(p(x, \mu, \Sigma)) - \log\frac{n_k}{N}\right)\,. \tag{9}$$

### 3.1.3. Determining the number of clusters

For a given set of independent points, $X = (x_i)_{1\leq i\leq N}$, we have

$$-\log p(X) = -\log\prod_{i=1}^{N} p(x_i) - \sum_{i=1}^{N}\log p(x_i)\,, \tag{10}$$

so the description length of the entire set is the sum of the description lengths of all its points, and the same for the DDL. The description length of $X$ under the GMM depends on the number of clusters in the GMM, and using the MDL principle, we can determine the optimum number of clusters by regarding 'optimum' as meaning 'produces the smallest DDL'.

Let $\mu(X, K), \Sigma(X, K)$ denote the values of $\mu$ and $\Sigma$ with $K$ components, which maximize the probability of $X$:

$$\mu(X, K), \sigma(X, K) = \operatorname*{argmax}_{\mu,\Sigma}\prod_{x\in X} p(x, \mu, \Sigma)\,. \tag{11}$$

Finding these optimal parameters means fitting the GMM to the dataset $X$, and can be performed with the usual expectation–maximization

[1] There are several reasonable choices for such a bounded set: the range of values that can be specified using a standard 32-bit float or the hyperrectangle whose sides are the coordinate-wise ranges across the dataset of patches. We find that the exact choice does not affect results. In our implementation, we choose the hypercube whose sides, in each dimension, run from the minimum to the maximum values across all dimensions in the dataset.

algorithm. Denote by $D(X,K)$ the DDL of $X$ under the optimal encoding corresponding to this fit GMM. Using $D(\cdot)$ from (9), we have

$$D(X,K) = \sum_{x \in P(X)} d(x, \mu(X,K), \sigma(X,K)) . \tag{12}$$

The value of $D(X,K)$ is the description length of the model itself plus the DDL of $X$ under the model. The former, i.e., the description length of a GMM with $K$ parameters, is, for precision $\epsilon$, given by

$$D(K) = Kd \log \left(a_{max} - a_{min}\right) \; + Kd^2 \log \left(a_{max} - a_{min}\right) . \tag{13}$$

Then, the optimal number of clusters $K^*$ is that which minimizes the total description length:

$$K^* = \operatorname*{argmin}_{1 \le K \le |X|} D(X,K) + D(K) . \tag{14}$$

Note that one only needs to consider values of $K$ up to the size of the dataset, as adding more clusters beyond that point can only increase the total description length. In practice, we test only values up to 8, as fitting GMMs with many clusters becomes expensive and, in our experiments, does not change results.

**Theorem 1.** *When clustering white noise in* $[0,1]^m$*, using a GMM with* $k$ *components, the expected DDL of a point is a monotonically increasing function of* $k$.

See the appendix for a proof. This means that, for white noise, we should expect MDL to select the model with just a single cluster, in which case every point will receive the same cluster label and the resulting entropy will be zero.

Determining the outliers and the number of clusters is relevant to measuring complexity, because it will affect the cluster model that is learnt, and so affect the cluster labels that are assigned and, ultimately, our complexity score.

### 3.2. Hierarchical patch entropy

The method described in this section is depicted graphically in Fig. 1. At each level of the hierarchy, we begin with a 3d tensor $X$ of shape $(H,W,C)$ and will cluster the vectors of the last dimension; on the first level, this means clustering 3d vectors specifying the colour intensities for each of the three colour channels at each point. Before clustering, the model computes $K^*$ as in (14), then clusters the last-dimension vectors of $X$ using a mixture model with $K^*$ components. From this clustering, we can form the 2d tensor $A$, of shape $(H,W)$ whose $(i,j)$th entry is the cluster index of the $(i,j)$th pixel in $X$, and $B$, the 3d tensor of shape $(H-m+1, W-m+1, K^*)$ whose $(i,j,k)$th entry is the count of how many times the $k$th cluster appears in the $m \times m$ patch beginning at $(i,j)$ in $A$.

The patch size $m$ is a user-set parameter. We refer to the vector at location $i,j$ in $B$ as the *signature* of the $(i,j)$th patch. Our measure of entropy at this level is the entropy of the categorical distribution of all signatures that appear in $B$.

As an example of how a patch signature in $B$ is formed from the corresponding patch of cluster indices in $A$, consider the top-left coloured patch in $A$, at the bottom-right of Fig. 1. This patch, coloured in dark red, contains three copies of index 2, one copy of index 3, five copies of index 4, and no copies of any other index. Thus, the patch signature is the vector $[0,0,3,1,5,0,0,0,0,0]$. At the bottom-left of Fig. 1 we see this patch signature is then stored at the corresponding location in $B$, also in dark red; note the first channel showing 0, the first element in the patch signature.

To measure complexity at a larger scale, we repeat the above procedure, this time beginning with $B$ instead of $X$. Let subscripts denote the level of the hierarchy, so that $A_i$ and $B_i$ are the tensors formed, as just described, on the $i$th level of the hierarchy. Then, we can say that $B_i$ contains the signatures (i.e., counts vectors) of the patches in $A_i$, and $A_i$ contains the MDL-cluster indices of the last-dimension vectors in $B_{i-1}$. To begin the iteration, $B_0$ is set to $X$, the input image.

The present implementation computes up to $B_4$, and uses larger patch sizes for each level: 4, 8, 16, and 32. Note, however, that this is not the same as simply clustering larger patches of an image. What is clustered at each level is the cluster indices from the level below, so is quite different from the input image. The full method is described in Algorithm 1.

**Algorithm 1** Algorithm for computing the complexity of an image.

```
function MDL_Cluster(D)
    $best\_DL \leftarrow \infty$
    $A \leftarrow$ cluster indices of MDL of $D$, initialized randomly
    for $K \in \{1, \dots, K\_max\}$ do
        fit a GMM with $K$ components to $D$
        $DL \leftarrow$ differential description length of $D$ under this fit GMM, as per (13)
        if DL < best_DL then
            $A \leftarrow$ cluster indices of $D$ under this fit GMM
            $best\_DL \leftarrow DL$
        end if
    end for
    return $A$
end function
function Signatures_Entropy(S)
    $bin\_counts \leftarrow$ hash table whose keys are the unique elements in $S$, and whose values are the number of times that element occurs in $S$
    return $-\sum_{b \in bin\_counts} \frac{bin\_counts[x]}{|S|} \log \frac{bin\_counts[x]}{|S|}$
end function
function Compute_Patch_Signatures(X,m)
    $A \leftarrow$ MDL_Cluster($X$)
    $B \leftarrow$ multisets of cluster indices appearing in all $m \times m$ patches of $A$ (including overlapping)
    return $B$
end function
function Complexity(X,scales)
    $total\_complexity \leftarrow 0$
    for $m \in scales$ do
        $X \leftarrow$ Compute_Patch_Signatures($X, m$)
        $total\_complexity \leftarrow total\_complexity +$ Signatures_Entropy($X$)
    end for
    return $total\_complexity$
end function
```

The method begins with the function MDL_Cluster, which returns the cluster indices of the MDL clustering of each location in the input. The right-hand-side of Fig. 2 shows an example of the output of this function when applied to the image from the left-hand-side of Fig. 2.

### 3.3. Comparison with theoretical measures of meaningful complexity

As mentioned in Section 1, previous works have explored, theoretically, how one might divide the algorithmic information of an object into a meaningful portion and a meaningless portion via sophistication [14,15] and effective complexity [16,17]. The applied method that we present in this paper shares the same high-level approach to these theoretical ideas, namely, to select the description for our data that has shortest overall length, and then, within that shortest description, select the size of the meaningful portion as a measure of the data complexity.

We assume that we have some way of distinguishing meaningful vs. meaningless descriptions. In our case, meaningful descriptions correspond to assignments of cluster labels to different parts of the image, and have length given by the first term in (2); the meaningless descriptions correspond to the residual error in specifying a point

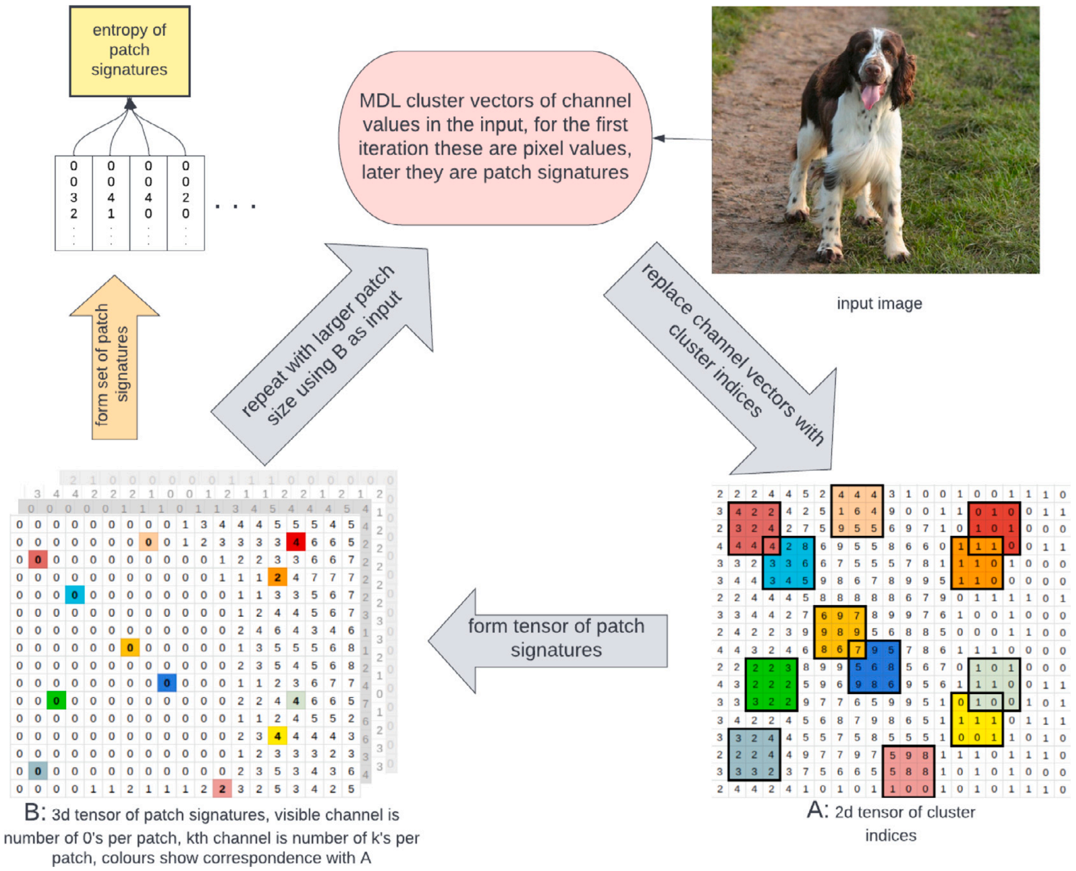
entropy of patch signatures
MDL cluster vectors of channel values in the input, for the first iteration these are pixel values, later they are patch signatures
input image
form set of patch signatures
repeat with larger patch size using B as input
replace channel vectors with cluster indices
form tensor of patch signatures
B: 3d tensor of patch signatures, visible channel is number of 0's per patch, kth channel is number of k's per patch, colours show correspondence with A
A: 2d tensor of cluster indices

**Fig. 1.** Method for computing the entropy of patch signatures as a measure of complexity. Each patch signature is the multiset of MDL cluster indices that appear there.

exactly given its cluster label, as per the second two terms in (2) along with the specification of outliers as per (6). Sophistication and effective complexity, on the other hand, characterize the meaningful portion as a description of a set of which the given data is a typical member, and the meaningless portion corresponds to selecting the given data from within that set. Let $\mathcal{S}$ and $\mathcal{R}$ denote, respectively, the sets of all possible meaningful and meaningless descriptions. Given data $X$, we write

$$D_0, \dots, D_n \vdash X, \text{ where } D_i \in \mathcal{S} \cup \mathcal{R}, \forall 1 \leq i \leq n \tag{15}$$

to mean that descriptions $D_0, \dots, D_n$ together perfectly describe $X$. We might try to characterize the meaningful complexity in $X$ as the length of its shortest meaningful description:

$$\min_{S \in \mathcal{S}} \{l(S) | S \vdash X\}, \tag{16}$$

where $l(\cdot)$ denotes the length of a description. However, this naive approach returns us to the problem of measuring random noise as highly complex, because if we are restricted only to meaningful descriptions, then we would need a very long one to completely describe a piece of noise. Instead, the approach taken both by our work, and by sophistication and effective complexity on the theoretical side, is to make use of the non-meaningful portion, not to count directly towards the complexity score, but in selecting the shortest description. The amount of meaningful complexity in $X$ is measured as

$$l(D^*), \text{ where } (D^*, E^*) = \min_{(D,E) \in \mathcal{S} \times \mathcal{R}} \{l(D) + l(E) | D, E \vdash X\}. \tag{17}$$

This leads to random noise getting a high value of $l(\mathcal{R})$, but a low value of $l(\mathcal{S})$, so even though its overall description length, $l(\mathcal{S}) + l(\mathcal{R})$, might be high, the resulting complexity score is low. In our case in particular, as shown by Theorem 1, the total description length tends to be minimized by having a single cluster, which means that the meaningful description is essentially of zero length and the entirety of the data is specified directly as outliers via (6).

There are important differences between our method and these theoretical works as well: in order to capture local spatial information, we measure the entropy of cluster labels *within patches*, not of individual points; and we repeat our method recursively at different levels, to capture compositionality, as described in Section 1. However, to the problem of correctly measuring the complexity of noise, our method uses, on a high level, the same solution as that explored in the theoretical concepts of sophistication and effective complexity.

### 3.4. Worked example

This section contains a worked example on a randomly chosen image from ImageNet, shown in Fig. 2. The steps of our method are enumerated for each of the four levels of the hierarchy. This shows how the final complexity score is obtained. At each level $i$, the model

1. performs MDL clustering on the set of array elements $B_{i-1}$, and assigns each a cluster label, to form $A_i$ (initially, $B_0$ is an image array of pixels, and then $A_1$ contains a cluster label for each pixel in $B_0$)
2. forms $B_i$ out of patch signatures of multisets of labels in each patch of $A_i$

**Layer 1**: 50246 points to cluster (pixels)

Number of components found by MDL, as per (14): 7

Assign each pixel a label from $0, \dots, 6$, and form patch signatures as multisets of labels inside all $4 \times 4$ patches, which gives 48450 patches, of 1411 different unique values.

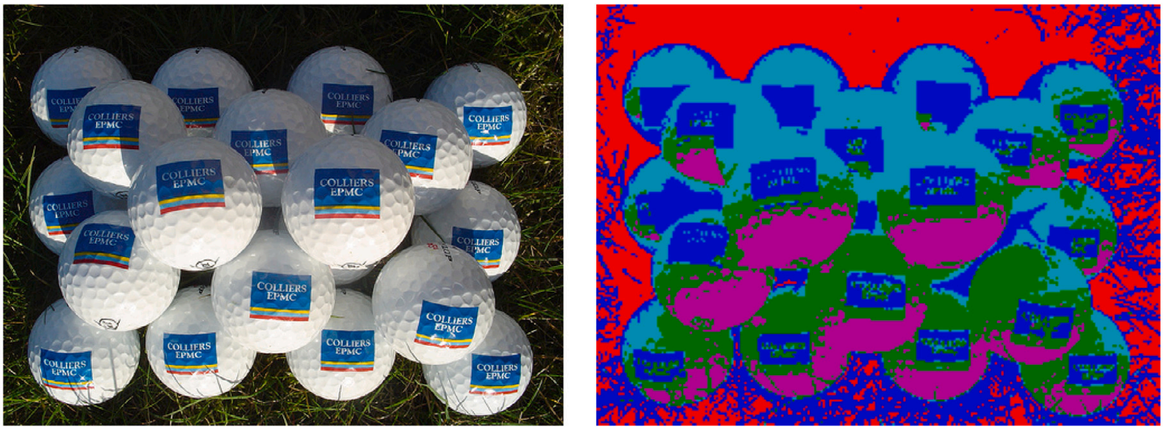

**Fig. 2.** Left: Example of a relatively high-resolution real-world image from ImageNet. ID: n03445777_10762. Right: The matrix $A$ formed by MDL clustering each point in the input image, i.e., by applying the function MDL_Cluster from Algorithm 1; different cluster indices are shown in different colours. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

Entropy of resulting categorical distribution of patch signatures: **7.995**

**Layer 2**: 48450 points to cluster (pixels)

Number of components found by MDL, as per (14): 8

Assign each point a label from $0, \dots, 7$, and form patch signatures as multisets of labels inside all $8 \times 8$ patches, which gives 44954 patches, of 3677 different unique values.

Entropy of resulting categorical distribution of patch signatures: **10.194**

**Layer 3**: 44954 points to cluster (pixels)

Number of components found by MDL, as per (14): 8

Assign each point a label from $0, \dots, 7$, and form patch signatures as multisets of labels inside all $16 \times 16$ patches, which gives 38346 patches, of 7341 different unique values.

Entropy of resulting categorical distribution of patch signatures: **12.772**

**Layer 4**: 38346 points to cluster (pixels)

Number of components found by MDL, as per (14): 7

Assign each point a label from $0, \dots, 6$, and form patch signatures as multisets of labels inside all size $32 \times 32$ patches, which gives 26666 patches, of 5666 different unique values.

Entropy of resulting categorical distribution of patch signatures: **12.353**

Total complexity: $7.995 + 10.194 + 12.772 + 12.753 = \mathbf{43.314}$

## 4. Experimental evaluation

It is difficult to assess the performance of an image complexity measure empirically. Some works gather human subjective judgements on a particular distribution of images (e.g., European renaissance paintings) and report accuracy/correlation, often also training a supervised model on these human judgements [38,57]. Aside from the practical difficulties of running these psychological studies, evaluating a model on a single distribution does not give a rounded indication of its accuracy, it is unclear how such models will perform when presented with a more diverse set of images. Additionally, collecting human judgments of complexity in this way may not be reliable: they are influenced by the presentation of the image as well as cognitive factors such as visual working memory [58], and show high inter-subject variability [59]. There is also EEG evidence suggesting that humans use different cognitive processes to judge an image's complexity depending on its degree of naturalness/familiarity [60]. We instead evaluate this method with a number of different experiments that, together, show that it assigns complexity scores in a coherent and consistent way, and that it accords with our intuitive understanding of complexity.

Firstly, we present the scores produced by our method for a diverse set of images of different types, taken from different datasets, both public and synthetic datasets that we create, and compare these scores to those produced by existing complexity metrics. Comparing sets/types of images, rather than individual images, has the advantage of reducing subjectivity. One can say with reasonable objectivity that ImageNet images are more complex than MNIST images, whereas there is more subjectivity in trying to compare the complexity of two different Renaissance paintings, or even two different ImageNet images. The scores produced by our method match our intuitive notion of complexity on this diverse set of images much more closely than do the scores of existing complexity metrics.

Then, after presenting ablation studies, we investigate the distribution of complexity across different levels of the hierarchy, and show that these agree with the different scales of complexity in the different types of images, e.g., fine-detailed repetitive textures receive high scores on the low levels of the hierarchy but lower scores on the higher levels, compared to globally structured images such as natural scenes from ImageNet.

Next, we show the effect of adding Gaussian noise and of lowering the resolution of images. A small amount of noise or reduction in resolution does not change the content of the image and so should not have a significant effect on the complexity score. For larger reductions in image quality, we would expect a gradual decline in complexity as the information in the image becomes increasingly obscured. This is exactly the case for our method. Its scores are largely unchanged by small quality degradations (addition of noise or reduction in resolution), and then show a steady decline with increasing degradation. As our method so effectively assigns low complexity to white noise images, it is particularly notable that it remains robust to a small/moderate amount of Gaussian noise.

Finally, we present the scores produced by our method on a fractal image, as the fractal dimension is varied. Again, the results are in line with our intuition about the type of complexity expressed by fractal dimension: higher fractal dimensions get a higher complexity score, but this is largely concentrated on the lower, more local levels.

### *4.1. Datasets*

We present the average score of our method on seven different sets of images, four popular image datasets and three synthetic datasets that we created:

1. **ImageNet** is a dataset with high complexity, depicting real-world objects in context.

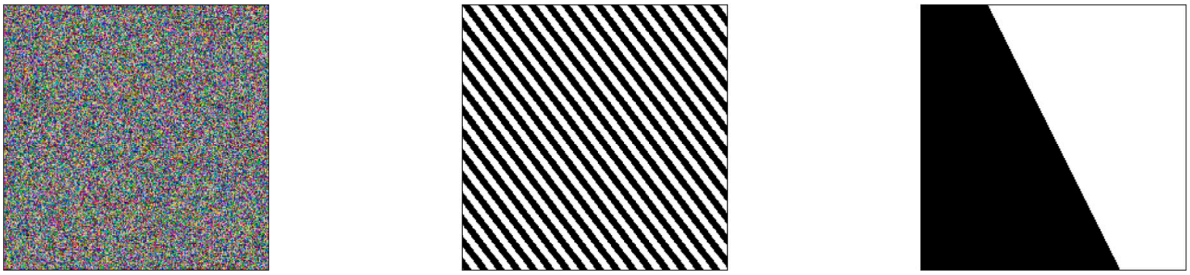

**Fig. 3.** Examples of images from the synthetic datasets we create. Left: Rand dataset; Middle: Stripes dataset; Right: Halves dataset.

2. **CIFAR** also shows real-world objects in context but of a much lower resolution, 32 × 32 vs. approximately 224 × 224 for ImageNet.
3. **MNIST** depicts low-resolution greyscale digits. Its images are simple in that they can be represented exactly with a small number of bits, but still have meaningful semantic content.
4. **DTD2** is a dataset that we created by manually searching through the Describable Textures Dataset [61] for all images of fine-detailed repeating textures.
5. **Stripes** is a synthetic dataset that we created of greyscale images of stripes of varying thickness and orientation. The thickness of the lines, in pixels, is sampled uniformly at random from $[3, 10]$, and the slope of the lines is sampled uniformly at random from $[-0.5, -1.5]$. It is sufficient to consider negative slopes only as our method, and all methods that we compare to, are invariant to reflections, so the striped images with slope in $[0.5, 1.5]$ would receive identical scores to those in $[-0.5, -1.5]$. Note that our method is not necessarily invariant to rotations, because it is based on square, axis-aligned patches of pixels. The same is true of the fractal dimension computed with the Minkowski–Bouligand dimension (i.e., the fractal dimension), as it uses a box-counting method. An example of an image from Stripes images is shown in Fig. 3.
6. **Halves** is a synthetic dataset that we created of greyscale images of half-black and half-white. These images have one half entirely black and the other entirely white, with the dividing line at various angles. As with Stripes, the slope of this dividing line is sampled uniformly at random from $[-0.5, -1.5]$. An example of an image from Halves is shown in Fig. 3.
7. **Rand** is a synthetic dataset that we created of white noise images, i.e., images with independent random pixel values. Their values are sampled uniformly at random from $[0, 1]$, independently for each location and each of three colour channels. Fig. 3 shows an example image.

For DTD2, we find 341 suitable images. For all other datasets, we use 1500 randomly sampled images and report the average for each image type. All images are resized to $224 \times 224$. The GMMs used for clustering are initialized with k-means, use diagonal covariance matrices, have tolerance $1e-3$, and are capped at 100 iterations.

### 4.2. Comparison with existing methods

Table 1 compares our method to seven others: 'khan2021' [41], 'machado2015' [38], and 'redies2012' [39] are as described in Section 2; 'entropy' converts the image to greyscale, discretizes the values into 256 bins, and then computes the Shannon entropy of the bin counts; 'fractal dim.' converts the image to greyscale, then binarizes it to 0 or 1, and computes the fractal dimension of the resulting shape using the box-counting method; 'jpg-ratio' measures the ratio of the JPEG-compressed file size to that of the original; and 'GLCM' computes the average entropy of the grey-level co-occurrence matrix, at offsets 1, 4, 8, 16, and 32 (see Sebastian V. et al. [62] for an account of GLCM in image complexity). All methods are normalized so their maximum score is 1.

The most striking result is that our method assigns zero complexity to white-noise images, while every other method assigns them high complexity, with many assigning maximum complexity. White noise images are not at all meaningful or interesting to humans, and it is a significant finding that our method is the first to reflect this. It suggests that, while existing methods are based only on the variation across the image, our method is able to measure the degree of *meaningful* variation, i.e., it is able to distinguish signal from noise.

The only two existing methods not to measure white noise as maximally complex are 'machado2015' and 'redies2012', though they still give it a high score. Instead, they give their max score to Stripes. This is also undesirable, because the simple repeating black and white stripes are not intuitively complex or meaningful either. These methods are both based on gradients (see Section 2), and the stripes produce a sharp gradient at every transition from black to white, which is likely the reason for these high scores. Stripes is also given a high score by the fractal dimension and JPEG-ratio methods, both assigning it only slightly less than white noise and significantly more than any other dataset, including ImageNet. The method of [41] (denoted 'khan2021') is difficult to interpret at all, because it assigns such a high score to the white noise that, after normalizing, all other datasets end up close to zero, with three being equal to zero. Recall that this method takes the median of the Fourier transform coefficients, so equals zero if over half of the coefficients are zero. Perhaps surprisingly, the relatively simple methods of entropy and GLCM entropy do a reasonable job of distinguishing real-world images from synthetic images and MNIST, compared to the more bespoke methods. However, they cannot detect a significant difference between ImageNet, CIFAR, and DTD, assigning all three very similar scores. In contrast, our method agrees much more closely with the intuitive notion of complexity: it assigns the highest complexity to ImageNet; it puts CIFAR ahead of DTD2 even though the latter is of higher resolution and has a complex texture, which shows that it recognizes CIFAR to have more semantically meaningful content; and it assigns MNIST a reasonably high complexity, despite it being the smallest in terms of file size, again showing that it can recognize global structure. Even aside from the white noise, no method but ours correctly places the remaining six datasets in order of complexity (left-to-right, as they appear in Table 1). This highlights the superior ability of our method to capture meaningful complexity across a variety of image types.

### 4.3. Ablation studies

Table 2 shows the effect of removing two key components of our method. In 'no mdl', rather than selecting the number of clusters $K$ using the minimum description length principle, we fix $K = 5$ for all images. This results in the same problem that existing methods suffer from: white noise is mistaken for high complexity and receives the maximum score. Also, 'no mdl' scores DTD2 too highly, showing that the method is not responding to global structure. In 'no patch', we take the entropy not of patch signatures, but of individual points in

**Table 1**
Comparison of our method with existing methods. The figures for each dataset are the mean across all images from that dataset, with std. dev. from batches of 25 in parentheses. All methods are normalized, so the maximum score that they assign is 1. Ours is the only method that does not assign white noise images high complexity, and gives the most reasonable results on all other datasets.

| | Dataset | | | | | | |
|---|---|---|---|---|---|---|---|
| | ImageNet | CIFAR | DTD2 | MNIST | Stripes | Halves | White-noise |
| Ours | 0.80 (.10) | 0.74 (.06) | 0.62 (.29) | 0.50 (.08) | 0.36 (.11) | 0.26 (.01) | 0.00 (.00) |
| khan2021 | 0.09 (.05) | 0.01 (.01) | 0.07 (.06) | 0.00 (.00) | 0.00 (.00) | 0.00 (.00) | 0.99 (.00) |
| machado2015 | 0.23 (.08) | 0.15 (.02) | 0.38 (.08) | 0.21 (.01) | 0.53 (.02) | 0.06 (.00) | 0.87 (.00) |
| redies2012 | 0.13 (.05) | 0.04 (.01) | 0.21 (.11) | 0.00 (.00) | 0.66 (.34) | 0.01 (.00) | 0.59 (.00) |
| Entropy | 0.89 (.10) | 0.89 (.07) | 0.83 (.13) | 0.30 (.06) | 0.13 (.00) | 0.13 (.00) | 0.96 (.00) |
| Fractal dim. | 0.74 (.09) | 0.61 (.08) | 0.86 (.16) | 0.45 (.06) | 0.98 (.02) | 0.44 (.02) | 1.00 (.00) |
| jpg-ratio | 0.22 (.08) | 0.09 (.0) | 0.29 (.09) | 0.06 (.01) | 0.57 (.01) | 0.06 (.00) | 0.57 (.00) |
| GLCM | 0.84 (.11) | 0.80 (.08) | 0.83 (.14) | 0.27 (.05) | 0.11 (.02) | 0.08 (.00) | 0.98 (.00) |

**Table 2**
Effect of removing two main components of our method. In 'no mdl', clustering is performed without MDL, instead simply fixing the number of clusters to 5 for all images and all scales. In 'no patch', we compute the entropy of the clusters themselves rather than of the patch signatures.

| | Dataset | | | | | | |
|---|---|---|---|---|---|---|---|
| | ImageNet | CIFAR | DTD2 | MNIST | Stripes | Halves | White-noise |
| Main | 0.80 (.10) | 0.74 (.06) | 0.62 (.29) | 0.50 (.08) | 0.36 (.11) | 0.26 (.01) | 0.00 (.00) |
| No mdl | 0.73 (.09) | 0.66 (.06) | 0.90 (.11) | 0.40 (.07) | 0.35 (.13) | 0.27 (.01) | 0.98 (.00) |
| No patch | 0.92 (.09) | 94 (.04) | 0.62 (.28) | 0.61 (.1) | 0.74 (.09) | 0.50 (.01) | 0.00 (.00) |

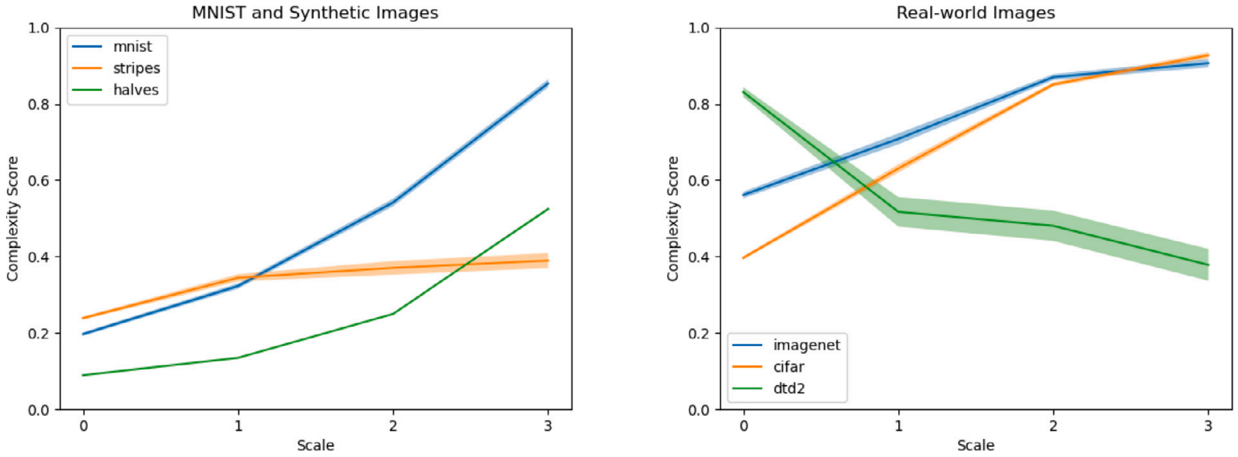

**Fig. 4.** Our complexity measure for different scales. The $x$-axis depicts patch size, on a log scale. Plots show mean score for all images of that type. Shaded regions are std dev from batches of 25 images.

the array, i.e., of $A$ rather than $B$ in the terminology of Section 3.2. (Patch signatures are still used for the iteration step.) This setting still performs reasonably well, but it gives too high a score to Stripes and a higher score to CIFAR than to ImageNet.

### 4.4. Complexity at different scales

The results from Section 4.2 suggest that, unlike existing methods, which focus only on detailed textures, ours is able to recognize complexity at a global level. Fig. 4 provides further support for this claim by showing the breakdown of our complexity measure at the four different scales (that is, four different levels of the hierarchy; see Section 3.2). Smaller scales respond to local complexity, and as the process is iterated to larger scales, global structure can be detected.

The first plot shows MNIST and the synthetic images. While MNIST has a similar local complexity score to Stripes, it has a much higher global complexity score, indicating that the more meaningful global structure in MNIST images can be detected. Halves, which is almost uniform locally but shows some variation globally, is given a very low local complexity but a small amount of global complexity. The second plot compares real-world images. CIFAR has the lowest local complexity, because it is low resolution, because it was resized from 32 × 32, so neighbouring pixels are all similar, but this does not affect its global complexity, which is as high as that of ImageNet. DTD2, on the other hand, has the highest local complexity, because it depicts detailed textures, but the lowest global complexity, because the textures are uniform across different regions of the images.

### 4.5. Effect of adding Gaussian noise

As our method so consistently assigns zero complexity to white noise, one may wonder whether it just searches for randomness in the image, and assigns zero if it finds any. To check this, we progressively add Gaussian noise to the three real-world datasets. The results are shown in Fig. 5. Noise is sampled independently from a standard normal distribution for each pixel, and a fraction of this noise is added to the image. Up until 10%, the scores are largely unchanged (DTD drops slightly), and then the scores for all three datasets steadily decrease with further noise. If the method was simply assigning low complexity in response to any randomness in the image, then we would see a sharp decline as soon as a small amount of noise is added. The results suggest that the method is instead responding to the amount of meaningful content in the image. A gradual decline in complexity is precisely what we would expect as the image quality deteriorates.

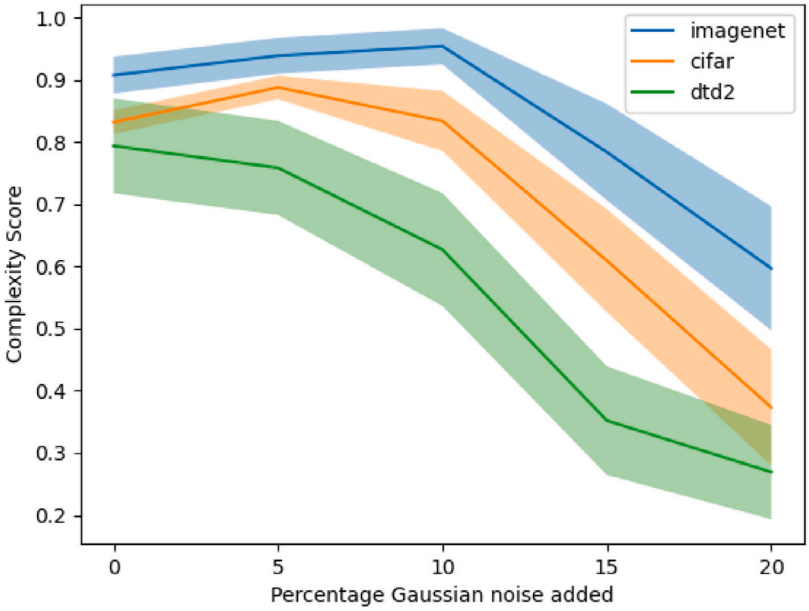

**Fig. 5.** Our complexity measure with different amounts of Gaussian noise added. Shaded regions are std from batches of 25 images. That is, we randomly sample 25 images and compute the mean complexity score, then repeat this for a total of 300 images and report the (unbiased) sample std dev.

**Table 3**
Comparison, on the scores produced by our method, of downsampling ImageNet to 32 × 32. Taken from 300 randomly sampled images of the 1500 used for the main results in Table 1.

| | Level 1 | Level 2 | Level 3 | Level 4 | Total |
|---|---|---|---|---|---|
| Full resolution | 7.70 (1.75) | 9.71 (1.99) | 11.93 (1.75) | 12.43 (1.98) | **41.77** (5.78) |
| Low resolution | 5.60 (0.73) | 9.07 (1.43) | 11.92 (1.14) | 12.72 (0.58) | **39.03** (3.40) |

### 4.6. Effect of changing resolution

To investigate how much our method is affected by the resolution of the input image, we apply it to a downsampled ImageNet. We randomly select 300 of the 1500 ImageNet images used for our main experiment and convert them to resolution 32 × 32. Table 3 shows the results of our method on these downsampled images and compares to the full-sized ImageNet images, which are roughly 256 × 256. There is a slight drop on the lower levels of the hierarchy, which corresponds to the greater uniformity at the local scale in the blurry, low-resolution images. The scores at the higher levels are essentially identical, and overall the scores are almost the same for the downsampled images as for the full-resolution images. This shows our method to be robust to changes in resolution, responding more to the contents of the image than to the resolution it is depicted at.

### 4.7. Scores for varying fractal dimension

Fractal dimension can roughly be defined as the detail in a shape or curve expressed as an exponent of its scale. (See [63] for discussion of different options for a precise definition.) In this section, we test the scores produced by our complexity metric on images of varying fractal dimension, from the dataset “Color Fractal Images with Independent RGB Color Components” [64]. This is a small dataset of nine high-resolution colour images, which are essentially the same except that they differ in fractal dimension. The images are generated using the midpoint displacement algorithm, which iteratively increases the fractal dimension of a piecewise-linear curve (i.e., a joined sequence of straight line segments), by slightly moving the midpoint of each piece. The dataset begins with a straight line and iterates until the fractal dimension is a certain value. The values for the different images range from 1.1 to 1.9 in increments of 0.1. This is repeated independently for each of the three colour channels. The resulting images are shown in Fig. 6.

It is generally thought that a higher fractal dimension indicates greater complexity, and so it is interesting to see whether our method is able to reflect this. Table 4 shows the scores produced by our method for each of the nine images in “Color Fractal Images with Independent RGB Color Components”, averaged over five runs, with the clustering at each level using 10 different GMM initializations and keeping the one with the highest data likelihood. The table shows the total complexity score, and the complexity score for each level. Looking at the total scores, there is a clear trend of increasing complexity scores for increasing fractal dimension, showing that the method can detect the sort of complexity expressed in fractal dimension. Looking at the breakdown of this total across the four levels of the hierarchy, we see that the effect of increased fractal dimension is greater for lower levels. Level 1 increases from 3.79 for fractal dimension 1.1 to 10.75 for fractal dimension 1.9, whereas Level 4 shows no systematic increase at all. The same information is shown graphically in Fig. 7. This reflects the similarity of the images in Fig. 6 at a more global level, e.g., they all have a patch of pink/red in the top-right corner and a patch of purple/blue in the bottom-right corner. It is within each patch, that is at a smaller scale, that the images differ.

This shows that our method is not only able to reliably detect an increase in fractal dimension by assigning a higher complexity score, it also distributes the increase with fractal dimension over the four levels of the hierarchy in the correct way. There is a significant increase at the most local level and progressively smaller increases at higher levels.

## 5. Discussion

### 5.1. Limitations and future work

One drawback of the current version of our method is the time complexity. Most existing image complexity metrics run in $< 0.1s$ per

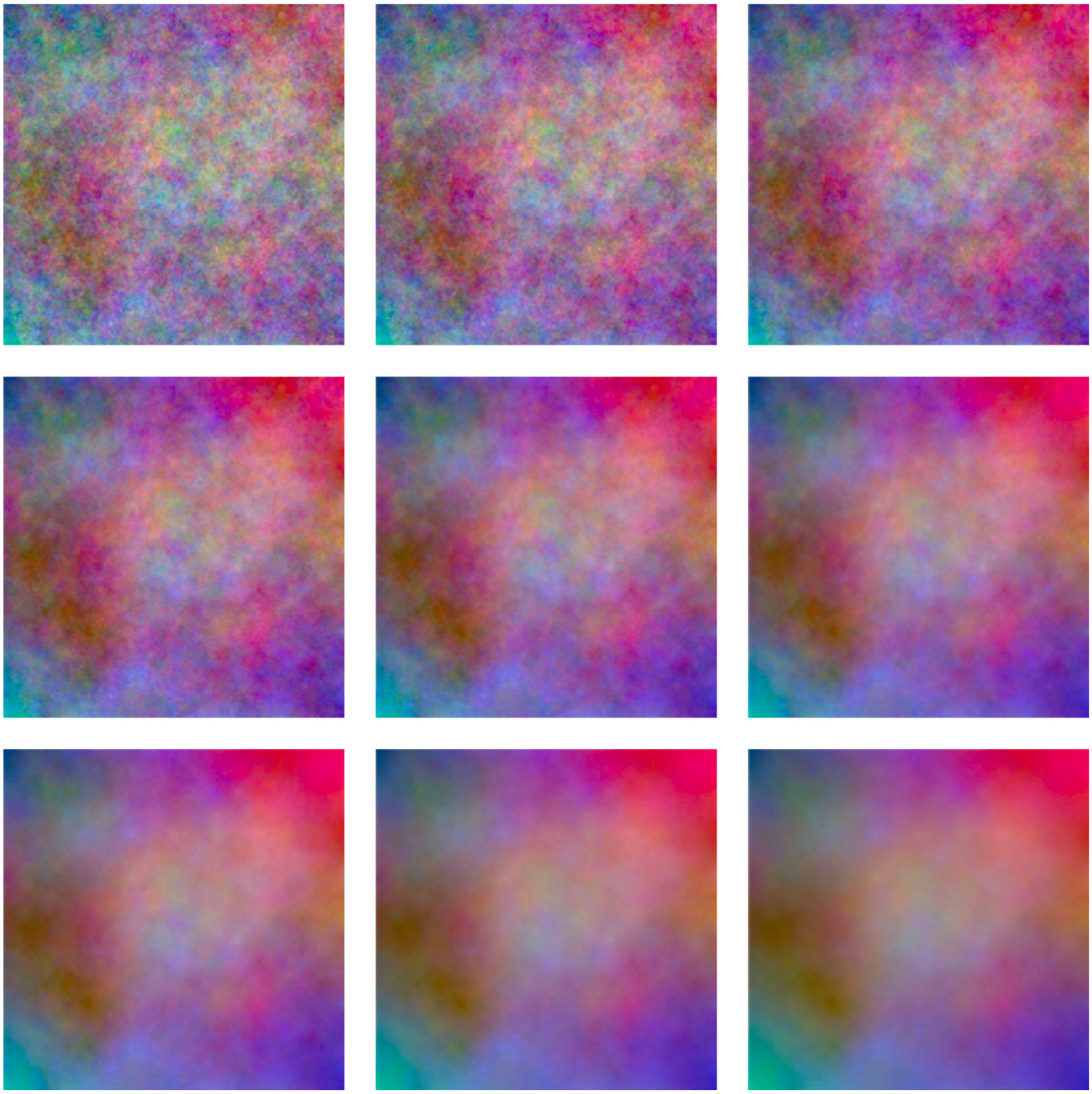

**Fig. 6.** The nine images from the "Color Fractal Images with Independent RGB Color Components" dataset. Top-left images is 1.9, decreases in increments of 0.1 to bottom-right. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

**Table 4**
Scores for increasing fractal dimension. For each fractal dimension, we run our method five times on the single image of that fractal dimension, and compute the (unbiased) sample standard deviation.

| | Total | Level 1 | Level 2 | Level 3 | Level 4 |
|---|---|---|---|---|---|
| Fract-dim 1.1 | 32.96 (0.15) | 3.79 (0.21) | 6.21 (0.27) | 10.42 (0.46) | 12.54 (0.16) |
| Fract-dim 1.2 | 34.39 (0.67) | 4.02 (0.23) | 6.73 (0.33) | 11.22 (0.16) | 12.41 (0.24) |
| Fract-dim 1.3 | 34.95 (0.19) | 4.24 (0.02) | 6.95 (0.21) | 11.06 (0.05) | 12.70 (0.00) |
| Fract-dim 1.4 | 36.02 (0.46) | 4.60 (0.06) | 7.60 (0.55) | 11.58 (0.39) | 12.24 (0.31) |
| Fract-dim 1.5 | 37.67 (0.48) | 5.31 (0.05) | 8.34 (0.39) | 11.63 (0.15) | 12.39 (0.24) |
| Fract-dim 1.6 | 39.64 (0.37) | 6.27 (0.18) | 9.36 (0.27) | 11.56 (0.34) | 12.44 (0.14) |
| Fract-dim 1.7 | 41.45 (0.30) | 7.35 (0.04) | 9.61 (0.18) | 12.10 (0.27) | 12.38 (0.14) |
| Fract-dim 1.8 | 44.94 (0.20) | 8.96 (0.02) | 10.81 (0.15) | 12.59 (0.13) | 12.58 (0.05) |
| Fract-dim 1.9 | 47.75 (0.14) | 10.75 (0.07) | 11.84 (0.25) | 12.75 (0.25) | 12.42 (0.29) |

image, whereas ours takes between $2s$ and $8s$ on average (the simple datasets are faster, ImageNet and CIFAR are the slowest). This can be roughly halved by reducing the range of $K$ explored from 8 to 5, with essentially no change in results. The run time is mostly due to the clustering step on the relatively large number of image patches. One future extension that could significantly improve runtime is, rather than considering all overlapping patches, to determine or approximate an optimal partition of the image into *non-overlapping* patches. This could draw on work in visual tiling [65]. Another future extension is to apply a similar method to other data domains, such as videos, audio, or text.

### 5.2. Conclusion

This paper presented a method for measuring image complexity. This task is inspired by the fact that humans cannot only explicitly recognize patterns in data, but can also detect whether the data contain a complex pattern at all. Our method can assign a complexity score to

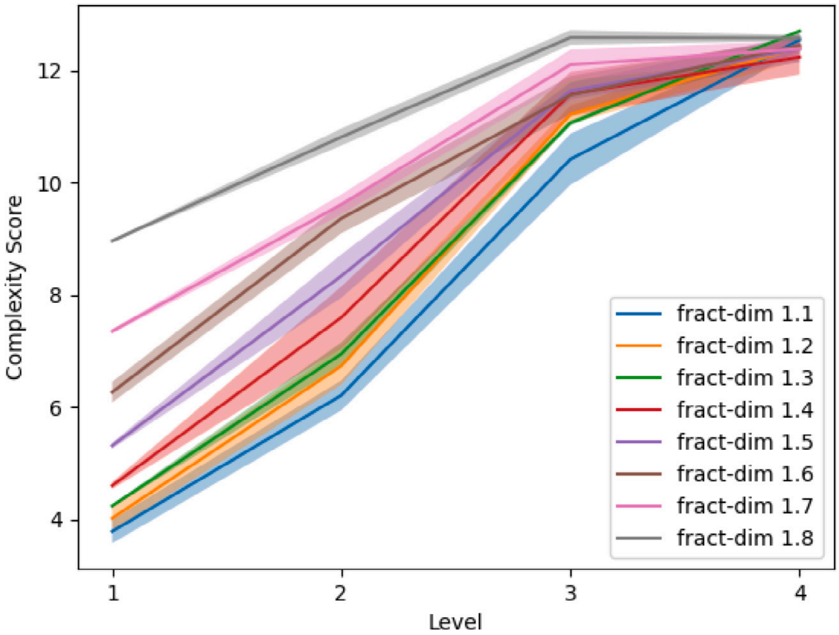

**Fig. 7.** Trend of increasing complexity score for increasing fractal dimension, broken down by level of the hierarchy. Shaded regions are (unbiased) sample std dev, as in Table 4.

data, specifically to images, which quantifies how complex a pattern or structure they contain. Unlike existing ways of quantifying complexity, it is able to capture the amount of meaningful complexity, and does not judge random noise to be complex. It uses clustering to analyse an image as being built out of a hierarchy of patches, with each patch composed of the cluster indices of its sub-patches. Clustering is performed with the minimum description length principle to distinguish signal from noise. We gave a detailed derivation of our method, and then presented an experimental evaluation showing that it performs better than existing measures of image complexity. Most strikingly, it assigns a very low score to white noise, in contrast to existing methods, which all measure white noise as highly complex. This result is also supported theoretically with a proof that white noise contains only one cluster, as judged by MDL, which immediately implies that our method assigns it very low complexity. We then presented ablation studies and a further set of experiments showing that it can accurately capture complexity at different scales, that it is robust to small/moderate degradations in quality, either from the addition of Gaussian noise or from a reduction in resolution, and that it can accurately reflect increasing fractal dimension in fractal images.

## Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

## Data availability

Data is publicly available.

## Acknowledgements

This work was supported by the Alan Turing Institute, UK under the UK EPSRC grant EP/N510129/1 and by the AXA Research Fund, UK.

## Appendix. Proof of correctness on white noise

**Lemma 2.** *When clustering white noise on* $[0,1]^m$*, with a* $k$*-component GMM, the radius* $r$ *of each cluster is approximated by* $\frac{1}{\sqrt{3}\sqrt[m]{k}}\left(\frac{2}{\sqrt{3}}\right)^{1/m}$.

**Proof.** For balls of radius $r$, the distance along each coordinate axis between their centres is $2r$ for one of the dimensions and $2r\frac{\sqrt{3}}{2}$ for all other dimensions. This is because the centres of each 3 touching balls form the vertices of an equilateral triangle with side length $2r$, which then have height $2r\frac{\sqrt{3}}{2}$. Thus, the number of balls of radius $r$ that can fit inside each axis is $\frac{1}{2r}$ for the first axis and $\approx \frac{1}{2r}\frac{2}{\sqrt{3}}$ for all other axes.

**Remark 3.** The approximation is for two reasons. Firstly, the this allows fractions of balls which extend partially, outside the hyperbox, and this is not possible in practice. Secondly, the packing of balls along one axis might interfere with the packing along another axis, e.g., if we pack the balls along axis 1 so that the distance between their centres is $2r\frac{\sqrt{3}}{2}$, then try to do the same along axis 2, we may find that they hit into the balls along axis 1. This would reduce the overall density that could be achieved. Both of these points lead to an overestimation of the number of balls (and hence an underestimation of the radius), but the proof turns out to be simplified by treating the expression in this lemma as an approximation, which we argue is accurate for large box size anyway.

The total number of balls that can fit inside $[0,1]^m$ is, therefore upper-bounded by

$$\frac{1}{(2r)^m}\left(\frac{2}{\sqrt{3}}\right)^{m-1}.$$

Conversely, given that the GMM will have $k$ clusters, we can lower-bound the radius of each cluster using

$$\frac{1}{(2r)^m}\left(\frac{2}{\sqrt{3}}\right)^{m-1} \geq k \iff r \geq \frac{1}{\sqrt{3}\sqrt[m]{k}}\left(\frac{\sqrt{3}}{2}\right)^{1/m}. \quad \square$$

**Proposition 1.** *For uniformly distributed points in an $m$-dimensional hyperball of radius $r$, the pdf of the distance of a point from the centre is given by $p(x) = m\frac{x^{m-1}}{r^m}$.*

**Proof.** The pdf is clearly proportional to $x^{m-1}$. The normalizing constant $c$ can be found by:

$$c\int_0^r x^{m-1}dx = 1 \iff c = \frac{m}{r^m}. \quad \square$$

**Lemma 4.** *When clustering white noise in $[0,1]^m$ dimensions, with a $k$-component GMM, the expected squared distance of a point from the centroid of its cluster, denoted $a$, is given by*

$$a = \frac{m}{3(m+2)}\left(\frac{\sqrt{3}}{2}\right)^{2/m}\frac{1}{k^{2/m}}, \tag{18}$$

**Proof.** Using Proposition 1, the expected squared distance from the centroid can be calculated directly from the pdf as

$$a = \frac{m}{r^m}\int_0^r x^{m+1}dx \iff a = r^2\frac{m}{m+2}.$$

Substituting $r$ from Lemma 2, we get $a \geq \frac{m}{3(m+2)}\left(\frac{\sqrt{3}}{2}\right)^{2/m}\frac{1}{k^{2/m}}$ as desired. □

**Lemma 5.** *When clustering white noise in $[0,1]^m$ dimensions, with a $k$-component GMM, the fit covariance matrix $\Sigma$ is approximated by $\Sigma = \sigma I$, where*

$$\sigma \geq \frac{1}{3(m+2)}\left(\frac{\sqrt{3}}{2}\right)^{2/m}\frac{1}{k^{2/m}}.$$

**Proof.** It is clear that $\Sigma$ is of the form $\sigma I$ for some $\sigma$, as the clusters are all identical hyperballs by symmetry (up to approximation at the boundary of the hyperbox, but this is small for more than a few clusters).

Next, observe that the expected squared distance of a point from the centroid of its cluster is, by linearity of expectation, equal to the sum of the expected squared distances in each coordinate, i.e., $m\sigma$. The result then follows by Lemma 4. □

**Proposition 2.** *The DDL of a point depends only on its distance from the centroid of its cluster.*

**Proof.** The probability density of a point $z$ under a cluster with centroid $\mu$ is

$$p(z) = \frac{1}{\sqrt{2\pi|\Sigma|}}\exp\left(\frac{-1}{2}(z-\mu)^T\Sigma^{-1}(z-\mu)\right) = \frac{1}{\sqrt{2\pi|\Sigma|}}\exp\left(\frac{-|(z-\mu)|^2}{2\sigma}\right).$$

Thus, the DDL under the cluster distribution, which we denote $\bar{D}$, is given by

$$-\ln\left(\frac{1}{\sqrt{2\pi|\Sigma|}}\exp\left(\frac{-|(z-\mu)|^2}{2\sigma}\right)\right) = \frac{1}{2}\ln(2\pi|\Sigma|) + \frac{|(z-\mu)|^2}{2\sigma}.$$

In what follows, we use the function $f(x)$ to denote the DDL under the cluster distribution of a point a distance $x$ from its centroid, where $f(x) = (1/2)\ln(2\pi|\Sigma|) + (x^2)/(2\sigma)$. □

**Definition 6.** Denote as outliers, those points with greater DDL under their cluster than under the prior.

**Lemma 7.** *When clustering white noise, so that the prior distribution is $[0,1]^m$, a point is an outlier if and only if it is greater than a distance $d$ from is centroid, where $d = \sqrt{\sigma\ln\frac{1}{2\pi|\Sigma|}}$.*

**Proof.** The DDL of a point treated as an outlier on the uniform $m$-box $[0,1]^m$ is 0, because treating as an outlier means using the prior distribution, which is uniform $p(x) = 1$, giving DDL of $\ln 1 = 0$. Thus, a point a distance $x$ from its centroid is not an outlier if and only if its DDL under the distribution of its cluster is strictly negative:

$$\frac{1}{2}\ln(2\pi|\Sigma|) + \frac{x^2}{2\sigma} < 0$$
$$\iff x < \sqrt{\sigma\ln\frac{1}{2\pi|\Sigma|}}.$$

We refer to this distance $d$ as the inlier radius. □

**Lemma 8.** *When clustering white noise in $m$ dimensions using a GMM with $k$ components, some points are classed as outliers if and only if $k$ satisfies*

$$k < \frac{2e\sqrt{\pi}}{\sqrt{3}}\left(\frac{e}{3(m+2)}\right)^{m/2}$$

**Proof.** A point is an outlier if and only if it is within the cluster, i.e., with a distance $r$ from its centroid, but outside the inlier radius $d$. This is possible if

$$d < r \iff \sigma\ln\frac{1}{2\pi|\Sigma|} < \frac{1}{3k^{2/m}}\left(\frac{\sqrt{3}}{2}\right)^{2/m}.$$

As $\Sigma = \sigma I$, we can sub in $|\Sigma| = \sigma^m$, and also sub in $\sigma$ from Lemma 5:

$$\frac{1}{3(m+2)}\left(\frac{\sqrt{3}}{2}\right)^{2/m}\frac{1}{k^{2/m}}\ln\frac{1}{2\pi\sigma^m} < \frac{1}{3k^{2/m}}\left(\frac{\sqrt{3}}{2}\right)^{2/m}$$
$$\iff k < \frac{e\sqrt{3\pi}}{2}\left(\frac{e}{3(m+2)}\right)^{m/2}. \quad \square$$

**Lemma 9.** *For a $k$-component GMM fit on $m$-dimensional white noise, the expected DDL of a point is*

$$\frac{m}{r^m}\int_0^x x^{m-1}\min(\{0,f(x)\})dx + \ln k.$$

*with $f(x)$ defined as in Proposition 2.*

**Proof.** The DDL of a point can be decomposed as the number of bits needed to specify which cluster the point belongs to, plus the DDL of the point under that cluster. The former is equal to $\ln k$, because each cluster is, by symmetry, equally sized. The latter, denoted $\bar{D}$, is given by $\int_0^r p(x)\min(\{0,f(x)\})$, where $p(r)$ is the probability density function of the distance of a point from its centroid, and $f(x)$ specifies the DDL of a point in terms of the distance from its centroid.

Substituting for $p(x)$ from Proposition 1 gives the result. □

**Lemma 10.** *When clustering white noise on $[0,1]^m$ with a $k$-component GMM, for $k \leq d$, the expected DDL is an increasing function of $k$.*

**Proof.** When some points are outliers, we have, using Lemma 9 and substituting $f(x)$ from Proposition 2:

$$\bar{D} = \frac{m}{r^m}\int_0^r x^{m-1}\min(\{0,f(x)\})dx$$
$$= \frac{m}{r^m}\int_{\{x\in[0,r]||f(x)<0\}} x^{m-1}f(x)dx = \frac{m}{r^m}\int_0^d x^{m-1}f(x)dx.$$

Integrating, and using the results from the Lemmas 7, 5, 2 and 5, then simplifies to:

$$\bar{D} = \frac{-3}{(m+2)^{\frac{m}{2}+2}}\left(m\ln(3m+6) + 2\ln k + \ln\frac{2}{3\pi}\right)^{\frac{m}{2}+1}.$$

We want to show the derivative of this expression, with respect to $k$, is positive:

$$\frac{d}{dk}\left(\frac{-3}{(m+2)^{\frac{m}{2}+2}}(m\ln(3m+6) + \ln\frac{2}{3\pi} + 2\ln k)^{\frac{m}{2}+1} + \ln k\right) > 0$$

$$\iff \frac{-3}{k(m+2)^{\frac{m}{2}+1}}{}^{\frac{m}{2}+2} \left(m\ln(3m+6) + \ln\tfrac{2}{3\pi} + 2\ln k\right)^{m/2} + \frac{1}{k} > 0$$

$$\iff \frac{3}{(m+2)^{\frac{m}{2}+1}} \left(m\ln(3m+6) + \ln\tfrac{2}{3\pi} + 2\ln k\right)^{m/2} < 1\,.$$

As, by assumption, some points are outliers, we can use Lemma 8 to bound the above LHS:

$$\begin{aligned}
&\leq \frac{3}{(m+2)^{\frac{m}{2}+1}} \left(m\ln(3m+6) + \ln\tfrac{2}{3\pi} + \ln\left(\frac{4\pi e^2}{3}\left(\frac{e}{3(m+2)}\right)^m\right)\right)^{m/2} \\
&= \frac{3}{(m+2)^{\frac{m}{2}+1}} \left(2 + \ln\tfrac{2}{3\pi} + \ln\tfrac{4\pi}{3} + m\right)^{m/2} < \frac{3}{(m+2)^{\frac{m}{2}+1}}(m+2)^{m/2} \\
&= \frac{3}{m+2} \leq 1\,. \quad \square
\end{aligned}$$

**Lemma 11.** *When clustering white noise on* $[0,1]^m$ *with a* $k$*-component GMM, for* $k > d$*, the expected DDL is independent of* $k$*.*

**Proof.** When no points are outliers, $f(x) = g(x)$ for all $x$, so, using Lemma 9,

$$\begin{aligned}
\bar{D} &= \frac{m}{r^m}\int_0^r x^{m-1} f(x)dx = \frac{m}{r^m}\left(\frac{r^m}{2m}\ln(2\pi|\Sigma|) + \frac{r^{m+2}}{2\sigma(m+2)}\right) \\
&= \frac{1}{2}\ln(2\pi|\Sigma|) + \frac{r^2 m}{\sigma(m+2)}.
\end{aligned}$$

Substituting from Lemmas 2 and 5, the full DDL is then

$$\begin{aligned}
\bar{D} + \ln k &= \frac{1}{2}\left(\ln\frac{4\pi}{\sqrt{3}} + m\ln\left(1 + \frac{1}{3m+6}\right) - 2\ln k + m\right) + \ln k \\
&= \frac{1}{2}\left(\ln\frac{4\pi}{\sqrt{3}} + m\ln\left(1 + \frac{1}{3m+6}\right) + m\right). \quad \square
\end{aligned}$$

**Proof.** By Lemma 8, the expected DDL is strictly increasing in k up to k = d. By Lemma 9, the expected DDL is constant in k for k > d. □

## References

[1] Kenneth Falconer, Fractal Geometry: Mathematical Foundations and Applications, John Wiley & Sons, 2004.

[2] W. Sun, G. Xu, P. Gong, S. Liang, Fractal analysis of remotely sensed images: A review of methods and applications, Int. J. Remote Sens. 27 (22) (2006) 4963–4990.

[3] Xiaomei Yang, Chenghu Zhou, Analysis of the complexity of remote sensing image and its role on image classification, in: Proceedings of the IGARSS 2000. IEEE 2000 International Geoscience and Remote Sens. Symposium. Taking the Pulse of the Planet: The Role of Remote Sens. in Managing the Environment. Proc. (Cat. No. 00CH37120), Vol. 5, IEEE, 2000, pp. 2179–2181.

[4] Alex Forsythe, Gerry Mulhern, Martin Sawey, Confounds in pictorial sets: The role of complexity and familiarity in basic-level picture processing, Behav. Res. Methods 40 (1) (2008) 116–129.

[5] Adrian Carballal, Carlos Fernandez-Lozano, Nereida Rodriguez-Fernandez, Iria Santos, Juan Romero, Comparison of outlier-tolerant models for measuring visual complexity, Entropy 22 (4) (2020) 488.

[6] Christian Stickel, Martin Ebner, Andreas Holzinger, The xaos metric: understanding visual complexity as measure of usability, in: Proceedings of the Symposium of the Austrian HCI and Usability Engineering Group, Springer, 2010, pp. 278–290.

[7] Lydia Chioukh, Halim Boutayeb, Dominic Deslandes, Ke Wu, Noise and sensitivity of harmonic radar architecture for remote sensing and detection of vital signs, IEEE Trans. Microw. Theory Tech. 62 (9) (2014) 1847–1855.

[8] Ram M. Narayanan, Sudhir K. Ponnappan, Stephen E. Reichenbach, Effects of noise on the information content of remote sensing images, Geocarto Int. 18 (2) (2003) 15–26.

[9] David A. Landgrebe, Erick Malaret, Noise in remote-sensing systems: the effect on classification error, IEEE Trans. Geosci. Remote Sens. 2 (1) (1986) 294–300.

[10] Yi Chang, Luxin Yan, Tao Wu, Sheng Zhong, Remote sensing image stripe noise removal: From image decomposition perspective, IEEE Trans. Geosci. Remote Sens. 54 (12) (2016) 7018–7031.

[11] Behnood Rasti, Paul Scheunders, Pedram Ghamisi, Giorgio Licciardi, Jocelyn Chanussot, Noise reduction in hyperspectral imagery: Overview and application, Remote Sens. 10 (3) (2018) 482.

[12] Wenzhun Huang, Shanwen Zhang, Harry Haoxiang Wang, Efficient GAN-based remote sensing image change detection under noise conditions, in: Proceedings of the International Conference on Image Processing and Capsule Networks, Springer, 2020, pp. 1–8.

[13] Puhong Duan, Xudong Kang, Shutao Li, Pedram Ghamisi, Noise-robust hyperspectral image classification via multi-scale total variation, IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens. 12 (6) (2019) 1948–1962.

[14] Moshe Koppel, Complexity, depth, and sophistication, Complex Syst. 1 (6) (1987) 1087–1091.

[15] Paul M. Vitányi, Meaningful information, IEEE Trans. Inform. Theory 52 (10) (2006) 4617–4626.

[16] Nihat Ay, Markus Müller, Arleta Szkola, Effective complexity and its relation to logical depth, IEEE Trans. Inform. Theory 56 (9) (2010) 4593–4607.

[17] Murray Gell-Mann, Seth Lloyd, Information measures, effective complexity, and total information, Complex 2 (1) (1996) 44–52.

[18] Leroy Cronin, Natalio Krasnogor, Benjamin G. Davis, Cameron Alexander, Neil Robertson, Joachim H.G. Steinke, Sven L.M. Schroeder, Andrei N. Khlobystov, Geoff Cooper, Paul M. Gardner, et al., The imitation game — A computational chemical approach to recognizing life, Nature Biotechnol. 24 (10) (2006) 1203–1206.

[19] Stuart M. Marshall, Douglas Moore, Alastair R.G. Murray, Sara I. Walker, Leroy Cronin, Quantifying the pathways to life using assembly spaces, 2019, arXiv preprint arXiv:1907.04649.

[20] Stuart M. Marshall, Cole Mathis, Emma Carrick, Graham Keenan, Geoffrey J.T. Cooper, Heather Graham, Matthew Craven, Piotr S. Gromski, Douglas G. Moore, Sara Walker, et al., Identifying molecules as biosignatures with assembly theory and mass spectrometry, Nat. Commun. 12 (1) (2021) 1–9.

[21] Edward W. Schwieterman, Nancy Y. Kiang, Mary N. Parenteau, Chester E. Harman, Shiladitya DasSarma, Theresa M. Fisher, Giada N. Arney, Hilairy E. Hartnett, Christopher T. Reinhard, Stephanie L. Olson, et al., Exoplanet biosignatures: A review of remotely detectable signs of life, Astrobiology 18 (6) (2018) 663–708.

[22] Marc M. Van Hulle, Edgeworth approximation of multivariate differential entropy, Neural Comput. 17 (9) (2005) 1903–1910.

[23] Georg Pichler, Pierre Jean A. Colombo, Malik Boudiaf, Günther Koliander, Pablo Piantanida, A differential entropy estimator for training neural networks, in: Proceedings of the International Conference on Machine Learning, PMLR, 2022, pp. 17691–17715.

[24] Jorma Rissanen, A universal prior for integers and estimation by minimum description length, Ann. Statist. 11 (2) (1983) 416–431.

[25] Trevor Huff, Navid Mahabadi, Prasanna Tadi, Neuroanatomy, visual cortex, in: StatPearls [Internet], StatPearls Publishing, 2021.

[26] Nicolas Passat, Benoıt Naegel, François Rousseau, Mériam Koob, Jean-Louis Dietemann, Interactive segmentation based on component-trees, Pattern Recognit. 44 (10–11) (2011) 2539–2554.

[27] Cong Geng, Xudong Jiang, Face recognition based on the multi-scale local image structures, Pattern Recognit. 44 (10–11) (2011) 2565–2575.

[28] Wendong Zhang, Yunbo Wang, Bingbing Ni, Xiaokang Yang, Fully context-aware image inpainting with a learned semantic pyramid, Pattern Recognit. (2023) 109741.

[29] Nick Chater, A minimum description length principle for perception, Adv. Minim. Descr. Length Theory Appl. (2005) 372–398.

[30] Jacob Feldman, The simplicity principle in perception and cognition, Wiley Interdiscip. Rev. Cogn. Sci. 7 (5) (2016) 330–340.

[31] Chris R. Sims, Rate–distortion theory and human perception, Cognition 152 (2016) 181–198.

[32] Allen Y. Yang, John Wright, Yi Ma, S. Shankar Sastry, Unsupervised segmentation of natural images via lossy data compression, Comput. Vis. Image Underst. 110 (2) (2008) 212–225.

[33] Rhodri H. Davies, Carole J. Twining, Timothy F. Cootes, John C. Waterton, Christopher J. Taylor, A minimum description length approach to statistical shape modeling, IEEE Trans. Med. Imaging 21 (5) (2002) 525–537.

[34] David Gibson, Neill Campbell, Barry Thomas, Visual abstraction of wildlife footage using Gaussian mixture models and the minimum description length criterion, in: Proceedings of the 2002 International Conference on Pattern Recognit., Vol. 2, IEEE, 2002, pp. 814–817.

[35] Nina Siu-Ngan Lam, Hong-lie Qiu, Dale A. Quattrochi, Charles W. Emerson, An evaluation of fractal methods for characterizing image complexity, Cartogr. Geogr. Inf. Sci. 29 (1) (2002) 25–35.

[36] Alex Forsythe, Marcos Nadal, Noel Sheehy, Camilo J. Cela-Conde, Martin Sawey, Predicting beauty: Fractal dimension and visual complexity in art, Br. J. Psychol. 102 (1) (2011) 49–70.

[37] Manuela M. Marin, Helmut Leder, Examining complexity across domains: Relating subjective and objective measures of affective environmental scenes, paintings and music, PLoS One 8 (8) (2013) e72412.

[38] Penousal Machado, Juan Romero, Marcos Nadal, Antonino Santos, João Correia, Adrián Carballal, Computerized measures of visual complexity, Acta Psychol. 160 (2015) 43–57.

[39] Christoph Redies, Seyed Ali Amirshahi, Michael Koch, Joachim Denzler, Phog-derived aesthetic measures applied to color photographs of artworks, natural scenes and objects, in: Proceedings of the European Conference on Computer Vis., Springer, 2012, pp. 522–531.

[40] George David Birkhoff, Aesthetic Measure, Harvard University Press, Cambridge, 1933.

[41] Tariq M. Khan, Syed S. Naqvi, Erik Meijering, Leveraging image complexity in macro-level neural network design for medical image segmentation, Sci. Rep. 12 (1) (2022) 22286.

[42] Bo Peng, Lei Zhang, David Zhang, Jian Yang, Image segmentation by iterated region merging with localized graph cuts, Pattern Recognit. 44 (10–11) (2011) 2527–2538.

[43] Dana E. Ilea, Paul F. Whelan, Image segmentation based on the integration of colour–texture descriptors—A review, Pattern Recognit. 44 (10–11) (2011) 2479–2501.

[44] Frédéric Galland, Nicolas Bertaux, Philippe Réfrégier, Multi-component image segmentation in homogeneous regionsbased on description length minimization: application to speckle, Poisson and Bernoulli noise, Pattern Recognit. 38 (11) (2005) 1926–1936.

[45] Mathilde Caron, Piotr Bojanowski, Armand Joulin, Matthijs Douze, Deep clustering for unsupervised learning of visual features, in: Proceedings of the European Conference on Computer Vision, ECCV, 2018, pp. 132–149.

[46] Louis Mahon, Thomas Lukasiewicz, Selective pseudo-label clustering, in: KI 2021: Advances in Artificial Intelligence: 44th German Conference on AI, Virtual Event, September 27–October 1, 2021, Proceedings 44, Springer, 2021, pp. 158–178.

[47] Uno Fang, Jianxin Li, Xuequan Lu, Ajmal Mian, Zhaoquan Gu, Robust image clustering via context-aware contrastive graph learning, Pattern Recognit. 138 (2023) 109340.

[48] Leland McInnes, John Healy, Steve Astels, HDBSCAN: Hierarchical density based clustering, J. Open Source Softw. 2 (11) (2017) 205.

[49] K. Mahesh Kumar, A. Rama Mohan Reddy, A fast DBSCAN clustering algorithm by accelerating neighbor searching using Groups method, Pattern Recognit. 58 (2016) 39–48.

[50] Kristina P. Sinaga, Miin-Shen Yang, Unsupervised K-means clustering algorithm, IEEE Access 8 (2020) 80716–80727.

[51] Jian Hou, Huaqiang Yuan, Marcello Pelillo, Towards parameter-free clustering for real-world data, Pattern Recognit. 134 (2023) 109062.

[52] Shohei Uchigasaki, Tomo Miyazaki, Shinichiro Omachi, Deep image compression using scene text quality assessment, Pattern Recognit. 142 (2023) 109696.

[53] Dipti Mishra, Satish Kumar Singh, Rajat Kumar Singh, Deep architectures for image compression: A critical review, Signal Process. 191 (2022) 108346.

[54] Zehira Haddad, Azeddine Beghdadi, Amina Serir, Anissa Mokraoui, Wave atoms based compression method for fingerprint images, Pattern Recognit. 46 (9) (2013) 2450–2464.

[55] Louis Mahon, Discrete Representations of Continuous Data Using Deep Learning and Clustering (Ph.D. thesis), University of Oxford, 2022.

[56] M.T.C.A.J. Thomas, A. Thomas Joy, Elements of Information Theory, Wiley-Interscience, 2006.

[57] Fintan Nagle, Nilli Lavie, Predicting human complexity perception of real-world scenes, Royal Soc. Open Sci. 7 (5) (2020) 191487.

[58] Aleksandra Sherman, So Yum Lim, Marcia Grabowecky, Satoru Suzuki, Visual-object working memory affects aesthetic judgments, J. Vis. 13 (9) (2013) 1308.

[59] Luis Madrid-Herrera, Mario I. Chacon-Murguia, Daniel A. Posada-Urrutia, Juan A. Ramirez-Quintana, Human image complexity analysis using a fuzzy inference system, in: Proceedings of the 2019 IEEE International Conference on Fuzzy Systems, FUZZ-IEEE, IEEE, 2019, pp. 1–6.

[60] Irina E. Nicolae, Mihai Ivanovici, Preparatory experiments regarding human brain perception and reasoning of image complexity for synthetic color fractal and natural texture images via EEG, Appl. Sci. 11 (1) (2020) 164.

[61] Mircea Cimpoi, Subhransu Maji, Iasonas Kokkinos, Sammy Mohamed, Andrea Vedaldi, Describing textures in the wild, in: Proceedings of the IEEE Conference on Computer Vis. and Pattern Recognit., 2014, pp. 3606–3613.

[62] Bino Sebastian V., A. Unnikrishnan, Kannan Balakrishnan, Gray level co-occurrence matrices: generalisation and some new features, 2012, arXiv preprint arXiv:1205.4831.

[63] Gerald Edgar, Measure, Topology, and Fractal Geometry, Springer Science & Business Media, 2007.

[64] Mihai Ivanovici, Noël Richard, Fractal dimension of color fractal images, IEEE Trans. Image Process. 20 (1) (2010) 227–235.

[65] Mengbai Xiao, Chao Zhou, Yao Liu, Songqing Chen, Optile: toward optimal tiling in 360-degree video streaming, in: Proceedings of the 25th ACM International Conference on Multimedia, 2017, pp. 708–716.

**Louis Mahon** is a postdoctoral research associate in the School of Informatics, at the University of Edinburgh. He recently completed his Ph.D. at the University of Oxford, under the supervision of Prof. Thomas Lukasiewicz. Currently, he is working on computer vision in conjunction with computational models of child language acquisition.

**Thomas Lukasiewicz** is a professor and head of the Artificial Intelligence Techniques research unit at TU Wien. Until recently, he was a professor of computer science at the University of Oxford.