Noname manuscript No.
(will be inserted by the editor)

Pre-training for Action Recognition with Automatically Generated Fractal Datasets

Davyd Svyezhentsev[1] · **George Retsinas**[2] · **Petros Maragos**[1,2]

Received: date / Accepted: date

Abstract In recent years, interest in synthetic data has grown, particularly in the context of pre-training the image modality to support a range of computer vision tasks, including object classification, medical imaging etc. Previous work has demonstrated that synthetic samples, automatically produced by various generative processes, can replace real counterparts and yield strong visual representations. This approach resolves issues associated with real data such as collection and labeling costs, copyright and privacy.

We extend this trend to the video domain applying it to the task of action recognition. Employing fractal geometry, we present methods to automatically produce large-scale datasets of short synthetic video clips, which can be utilized for pre-training neural models. The generated video clips are characterized by notable variety, stemmed by the innate ability of fractals to generate complex multi-scale structures. To narrow the domain gap, we further identify key properties of real videos and carefully emulate them during pre-training. Through thorough ablations, we determine the attributes that strengthen downstream results and offer general guidelines for pre-training with synthetic videos. The proposed approach is evaluated by fine-tuning pre-trained models on established action recognition datasets HMDB51 and UCF101 as well as four other video benchmarks related to group action recognition, fine-grained action recognition and dynamic scenes. Compared to standard Kinetics pre-training, our reported results come close and are even superior on a portion of downstream datasets. Code and samples of synthetic videos are available at https://github.com/davidsvy/fractal_video.

Keywords Fractal Geometry; Synthetic Data; Action Recognition; Domain Adaptation

Davyd Svyezhentsev (dsvyez@gmail.com)
George Retsinas (gretsinas@central.ntua.gr)
Petros Maragos (maragos@cs.ntua.gr)

[1] School of Electrical and Computer Engineering, National Technical University of Athens, Athens, Greece

[2] Robotics Institute, Athena Research and Innovation Center, Athens, Greece

1 Introduction

Contemporary computer vision models require enormous amounts of data for training. Beginning with ImageNet [63] which consists of 1.4 million labeled images, the scale of vision datasets has been rapidly increasing, reaching tens of millions to a billion of samples [27,77,25]. Regarding such datasets, multiple issues emerge. First, data collection and annotation is arduous and expensive. Second, it has been noted that vision datasets may inherit human biases [66,8,75,79] and contain inappropriate content [7]. Third, the depiction of humans in these datasets poses questions of privacy [4]. Lastly, ownership concerns limit many datasets to noncommercial usage only.

Hence, the computer vision society has recently exhibited growing interest in synthetic datasets that mitigate the aforementioned shortcomings. Amongst them, noteworthy is the seminal work of [42] who proposed to pre-train 2D CNNs with automatically generated images of fractals [6]. Although their results are inferior compared to standard ImageNet pre-training, they significantly surpass training from scratch. Subsequent work has either enhanced their approach [5,2,41], alleviating the gap in downstream results between real and synthetic data or extended it to other domains [78].

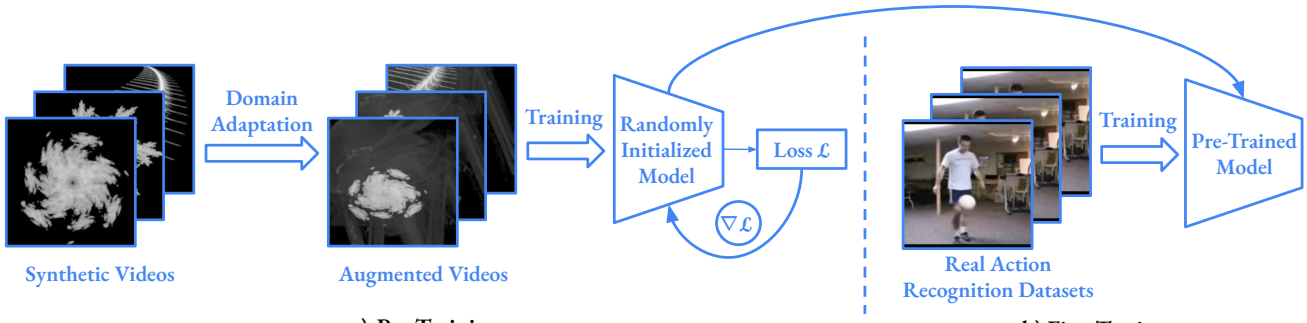


Fig. 1: Overview of the proposed approach. Aiming to pre-train neural models, we utilize fractal geometry and automatically construct large-scale datasets of short synthetic video clips (Sec. 2.2). We additionally narrow the domain gap between real and synthetic videos by identifying key properties of the former and emulating them during pre-training (Sec. 2.3). The transferability of the proposed datasets and transformations is experimentally assessed by fine-tuning the pre-trained models on real action recognition benchmarks (Sec. 3)

Such datasets offer several advantages. They are constructed automatically and thus, do not require a collection or an annotation stage. They are also not bound by copyright limitations. Moreover, there are no questions regarding biases, inappropriate content or privacy, as no human subjects are depicted.

The present work extends the ideas of [42] to the domain of video. Specifically, we seek to automatically produce synthetic datasets that are suitable for pre-training neural networks for the task of action recognition. Automatic action recognition is of paramount importance as it enables accurate detection and interpretation of human actions from video or sensor data. This technology has broad applications across various sectors, including surveillance, healthcare, robotics, sports analysis, and human-computer interaction. The significance of action recognition is additionally outlined by the sheer amount of videos available on the internet. With over 500 hours of video uploaded to YouTube every minute, there is an immediate need for robust algorithms that can help organize, summarize and retrieve this massive amount of data.

Our approach is summarized in Fig. 1, where the two involved stages are depicted: 1) the *pre-training* stage and 2) the *fine-tuning* stage. The former includes the generation of fractal-based videos, followed by their appropriate augmentation to simulate human actions, as well as the training procedure to capture as much relevant information as possible. The latter includes a straightforward fine-tuning pipeline in order to adapt the pre-trained network to real action recognition datasets.

The main contributions of this paper are:

- Using fractal geometry [6], as well as other generative processes, we propose a pipeline that can automatically construct large-scale datasets of short synthetic video clips. These datasets are employed for pre-training neural networks for the task of action recognition instead of the typical large-scale Kinetics [10,43] dataset. Both supervised and self-supervised learning is applicable and explored in the experimental section.
- Starting from the observation of real video samples, we pinpointed their fundamental attributes such a periodic motion, random background, camera displacement etc. These attributes are carefully emulated during pre-training, significantly reducing the domain gap between synthetic and real videos.
- Experimentally, we analyze downstream performance as a function of the training objective and the properties of the synthetic dataset. We determine beneficial attributes and offer general guidelines for pre-training with synthetic videos.
- We conduct error analysis of the pre-trained models' predictions and detect common patterns. As such, we propose tailored modifications to the synthetic data that may further boost downstream results in future work.

2 Proposed Methodology

2.1 Preliminaries: Fractal Images

Before exploring the video modality, it is necessary to first examine the simpler domain of images. Following [42], who produce synthetic image datasets to pre-train 2D CNNs, fractals generated via the Iterated Function Systems (IFS) technique [6] are chosen as the backbone of our work. Specifically, the IFS fractals possess a set of appealing properties, including an easily imple-

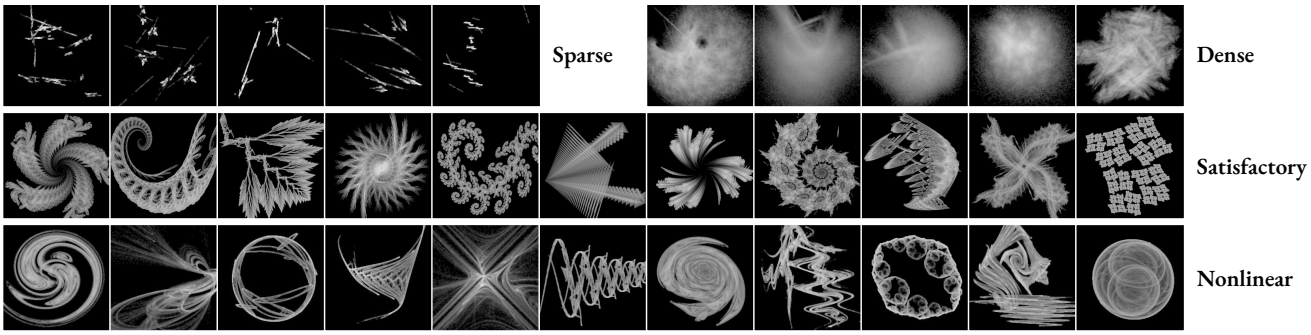


Fig. 2: Examples of rendered 2D IFS attractors. A subset of linear samples exhibits unsatisfactory geometry, being either too sparse or too dense. Adding nonlinearity significantly alters the distribution of produced images and boosts overall diversity.

mentable rendering algorithm as well as the possibility of producing a near-limitless supply of diverse images by randomly sampling parameters.

2.1.1 Classic IFS

Following Barnsley [6], a 2D IFS can be defined as a set of $N > 1$ affine transformations $F_i : \mathbb{R}^2 \mapsto \mathbb{R}^2$. Each F_i is represented by a matrix $A_i \in \mathbb{R}^{2\times 2}$ and a vector $b_i \in \mathbb{R}^2$:

$$F_i(\boldsymbol{x}; A_i, b_i) = A_i\boldsymbol{x} + b_i = \begin{bmatrix} a_i & b_i \\ d_i & e_i \end{bmatrix} \boldsymbol{x} + \begin{bmatrix} c_i \\ f_i \end{bmatrix}$$

The complete parameter matrix will be referred to as $W \in \mathbb{R}^{N\times 6}$. Additionally, it is necessary for each mapping to be contractive with respect to the Euclidean distance $\mathrm{d}(\cdot, \cdot)$:

$$d(F_i(x), F_i(y)) \leq k_i \cdot \mathrm{d}(x, y), 0 < k_i < 1$$

This constraint prevents divergence and ensures that, after successive applications of the mapping, points get progressively closer together. Furthermore, if the affine transformations are extended to be applied to whole subsets of the plane instead of single points, then their union F is a contractive mapping with respect to the Hausdorff distance on the space of nonempty compact sets [6]. If F is iteratively applied, starting from an arbitrary initial set, the iterations will converge to a unique fixed set of points which is referred to as the attractor of the IFS [35]. Since we are working with the 2D Euclidean plane, the resultant attractor is an image.

An approximation of the attractor (Fig. 2) can be rendered with the chaos game algorithm [6]. At first, this algorithm initializes the output image as zeros and samples a starting 2D point. At each iteration, one of the N functions is sampled from the IFS and applied to the said point. The probability of selecting each function is:

$$p_i = \frac{|\det(A_i)|}{\sum_{j=1}^{N} |\det(A_j)|}$$

Given that the coordinates of the point are real numbers, they are quantized to a discrete pixel. The output image value corresponding to this this pixel is then incremented by one. After completing a specified number of iterations, the output, which is a 2D histogram, is normalized, producing a grayscale image.

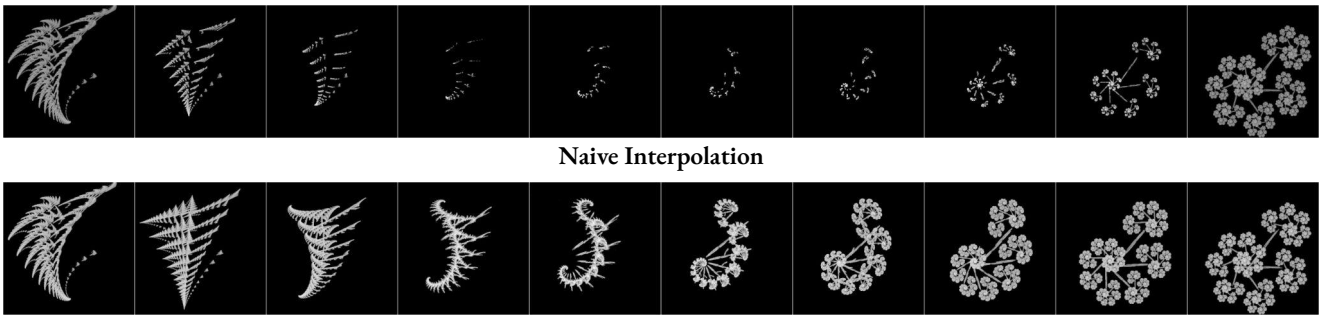
To construct a dataset of fractal images, [42] originally proposed to sample parameters independently from $U(-1, 1)$. However, a subsequent work [2] observed that this strategy often results in images with degenerate geometry, being either too sparse or too dense. As a solution, it is suggested to decompose each weight matrix into $A = R_\theta \Sigma R_\phi D$, omitting the index i for brevity. The decomposed matrices are:

- R_x is a rotation matrix parameterized by angle x.
- Σ is a diagonal matrix containing the singular values σ_1 and σ_2 ordered by decreasing magnitude.
- D is a diagonal matrix with elements $d_1, d_2 \in \{-1, 1\}$, acting as a reflection matrix.

As such, a "well-behaved" A can be sampled by appropriately sampling the decomposed parameters $\{\theta, \phi, \sigma_1, \sigma_2, d_1, d_2\}$. Notably, [2] empirically deduce that the geometry of the resultant attractor depends on the quantity $a = \sum_{i=1}^{N}(\sigma_{i,1} + 2\sigma_{i,2})$, with unsatisfactory behavior being minimized when:

$$a_l = \frac{1}{2}(5 + N) \leq a \leq a_u = \frac{1}{2}(6 + N)$$

Towards satisfying the inequality, the authors of [2] also propose an iterative algorithm for sampling param-



Naive Interpolation

Decomposed Interpolation

Fig. 3: An example of the the proposed animation method compared to naive interpolation (Sec. 2.2). The latter often results in undesired sparseness in the intermediate frames. The former mitigates this issue. More samples are displayed in Fig. 10 and 11 of Appendix B.

eters that fulfill the specified inequality. Their methodology is adopted throughout the present work.

2.1.2 Fractal Flame

The Fractal Flame Algorithm [20] is an extension to the ordinary IFS designed to generate more aesthetically pleasing images, which are often employed as desktop backgrounds. Although multiple modifications to the IFS are introduced, only one is of interest for this work: nonlinearity. In particular, after the application of the affine function F, an additional nonlinear function $G : \mathbb{R}^2 \mapsto \mathbb{R}^2$, can be applied to the coordinates. The latter is referred to as variation. The authors of [20] provide 49 such variations, e.g:

- $G_6(x, y) = r(\sin(\theta + r))$
- $G_{16}(x, y) = \frac{2}{r+1}(y, x)$

Here r and θ are polar coordinates. As seen in Fig. 2, such images differ significantly from the ordinary IFS. Thus, the inclusion of nonlinear functions significantly boosts the overall diversity of generated samples. At the time of writing this document, no other work has explored fractal flames in the context of deep learning.

2.2 Synthetic Videos based on Fractal Geometry

Animation can be simply achieved by sampling parameters for two fractal images, the first and last frame. The only constraint is the number of functions N, which must be shared. To produce motion, parameters of the two images are linearly interpolated. As the order of functions in an IFS is arbitrary, we sort them by their probabilities p_i. The result is a smooth sequence of T IFS images which can be rendered separately to produce a video. Nonetheless, although the beginning and end of the resultant clips are satisfactory, the intermediate attractors are not, often exhibiting evident sparseness (Fig. 3). We consider this behavior inappropriate for producing large-scale synthetic datasets. Therefore, an alternative solution is required.

Instead of directly interpolating IFS parameters, we propose to alleviate detrimental sparseness by employing matrix decomposition (Sec. 2.1.1). Specifically, each parameter matrix A_i can be decomposed into sub-matrices. The approach is to first interpolate each pair of sub-matrices separately and subsequently multiply them to obtain the final parameters of each frame. This method is adapted from [2] and [9] and examples are displayed in Fig. 3. The complete procedure is described in detail in Algorithm 1. Regarding notation, $\texttt{zeros}(x)$ is a matrix of shape x filled with zeros, $\texttt{diag}(x, y)$ is a diagonal $2D$ matrix with x, y as elements, $\texttt{interp}(x, y, z)$ denotes linear interpolation between x and y of length z, $\texttt{rot\_matrix}(x)$ is a $2D$ rotation matrix parameterized by angle x and the symbols : and ... have the same functionality as in `numpy`.

It is noteworthy that fractal geometry can generate videos in an alternative manner: by constructing a point cloud with a 3D IFS. Specifically, by extracting 2D slices at different coordinates within this point cloud, a sequence of images can be produced. However, this approach is not pursued due to its higher computational demands and incompatibility with certain key modifications introduced in Sec. 2.3.

2.3 Domain Adaptation

Previous work on images [5] concludes that downstream performance is boosted if synthetic and real data share structural properties. Likewise, large-scale studies on pre-training [14,45,67] deduce that its

Algorithm 1 `sample-video-decomposed`(): Sample IFS parameters for an animation by interpolating each sub-matrix separately.

Output: Sequence of parameters: $W \in \mathbb{R}^{T\times N\times 6}$

Sample $N \sim U(\{3,\ldots,8\})$ ▷ # functions
Sample $T \sim U(\{18,\ldots,20\})$ ▷ # frames

$D \leftarrow \texttt{zeros}(N,2,2) \in \mathbb{Z}^{N\times 2\times 2}$ ▷ Initialize matrices
$\Sigma \leftarrow \texttt{zeros}(T,N,2,2) \in \mathbb{R}^{T\times N\times 2\times 2}$
$R_\theta \leftarrow \texttt{zeros}(T,N,2,2) \in \mathbb{R}^{T\times N\times 2\times 2}$
$R_\phi \leftarrow \texttt{zeros}(T,N,2,2) \in \mathbb{R}^{T\times N\times 2\times 2}$
$b \leftarrow \texttt{zeros}(T,N,2) \in \mathbb{R}^{T\times N\times 2}$

for $n = 1$ to N **do**
  `Sample` $d_1, d_2 \sim U(\{-1,1\})$
  $D[n,\ldots] \leftarrow \texttt{diag}(d_1,d_2) \in \mathbb{Z}^{2\times 2}$

  `Sample` σ_1^1, σ_2^1 ▷ see Appendix A of [2]
  `Sample` σ_1^T, σ_2^T
  $\Sigma^1, \Sigma^T \leftarrow \texttt{diag}(\sigma_1^1,\sigma_2^1), \texttt{diag}(\sigma_1^T,\sigma_2^T) \in \mathbb{R}^{2\times 2}$
  $\Sigma[:,n,\ldots] \leftarrow \texttt{interp}\left(\Sigma^1,\Sigma^T,T\right) \in \mathbb{R}^{T\times 2\times 2}$

  `Sample` $\theta^1, \theta^T, \phi^1, \phi^T \sim U(0,2\pi)$
  $\theta,\phi \leftarrow \texttt{interp}(\theta^1,\theta^T,T), \texttt{interp}(\phi^1,\phi^T,T)$
  for $t = 1$ to T **do**
    $\mathbb{R}_\theta[t,n,\ldots] \leftarrow \texttt{rot\_matrix}\left(\theta[t]\right) \in \mathbb{R}^{2\times 2}$
    $\mathbb{R}_\phi[t,n,\ldots] \leftarrow \texttt{rot\_matrix}\left(\phi[t]\right) \in \mathbb{R}^{2\times 2}$
  end for

  `Sample` $b^1, b^T \sim U\left(-1,1\right) \in \mathbb{R}^2$
  $b[:,n,:] \leftarrow \texttt{interp}(b^1,b^T,T) \in \mathbb{R}^{T\times 2}$
end for

$A \leftarrow R_\theta \Sigma R_\phi D \in \mathbb{R}^{T\times N\times 2\times 2}$ ▷ Compose A
$W \leftarrow \texttt{reshape}(\texttt{concat}(A, \texttt{expand}(b))) \in \mathbb{R}^{T\times N\times 6}$
return W

effectiveness significantly deteriorates if the source and target domains differ. As such, it can be assumed that obtaining satisfactory video models requires narrowing the domain gap between synthetic videos and samples from action recognition benchmarks. To do so, this section lists manually observed characteristics of real action recognition data [65,46] as well as methods to emulate them within the synthetic pre-training framework (Fig. 5). Amongst these techniques, nonlinear motion and amplified diversity can only be applied offline, requiring the construction of a new video with modifications to Alg. 1. On the contrary, the rest are implemented as online augmentations, further promoting the diversity of the generated videos on-the-fly. It should be noted that in the context of our work, domain adaptation refers to the proposed set of heuristic augmentations which simulate mostly motion-related properties of real videos.

Nonlinear Motion. Our synthesis method produces simple forward motion. However, the real human

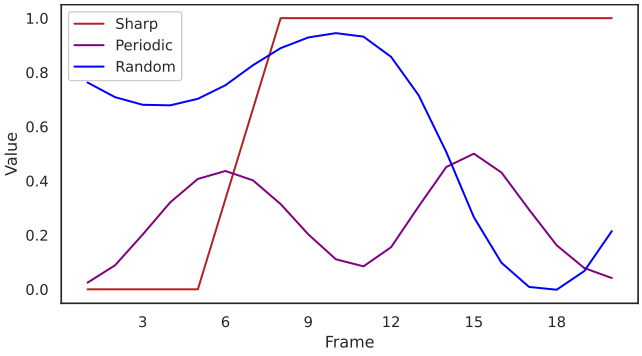


Fig. 4: Examples of the proposed nonlinear interpolation curves. The objective is to approximate the complexity of real human motion.

counterpart is more complex. To this end, more intricate interpolation functions (Fig. 4) can be included in the rendering process (Alg. 1):

- **Sinusoidal Interpolation**. Periodic activity such as exercise can be simulated with a noisy sine function.
- **Sharp Interpolation**. Quick and sudden activity such as boxing can be approximated by a linear interpolant with a significantly sharper slope. The linear interpolant is placed in random timestep while the rest of the curve is padded.
- **Random Interpolation**. Other activity without clear patterns can be simulated with a random waveform. To produce such a waveform, a sequence of real numbers is initially sampled from $U(0,1)$ and then smoothened via 1D quadratic interpolation.

Moreover, human motion is inherently composite, i.e. intricate motions consist of multiple simple ones. For instance, running is comprised of a periodic movement of the legs as well as a different periodic movement of the arms. This can be approximated by assigning a different interpolant to each IFS function pair. As a result, the produced shape will execute multiple motions simultaneously. However, doing so unrestrictedly often results in oscillations that lack the structured flow found in real human motion. To this end, interpolation functions will be sampled under constraints. Initially, the set of chosen interpolants is initialized as the linear interpolant. Next, one or two different nonlinear interpolants are added to the set. Lastly, each of the N IFS function pairs receives a randomly sampled interpolant from the set or a single interpolant is applied to all IFS functions. The N resultant interpolants replace the `interp` operation in Alg. 1. The outcome is a shape that moves in more fluid and constrained manner compared to naive sampling of interpolants.

Diversity. It has been demonstrated that diversity of synthetic images during pre-training leads to stronger visual representations [5]. In the context of fractal animations, this property can be boosted by adding nonlinearity to the image rendering algorithm (Section 2.1.2). In pursuit of reducing the domain gap, only variations $4, 14, 16, 17, 20, 27$ and 29 (Appendix of [20]) are selectively employed. These were chosen due to their distinct and well-defined shape and contours, a characteristic present in real videos.

Random Background. Real videos consist of two essential components: foreground (person performing an action) and background (environment surrounding the person). In its simplest form, the background is completely static. Although in synthetic videos this property is absent, it can be straightforwardly approximated following [72,18]. For each video x_i within a batch, a static frame is sampled from a different video x_j and mixed with every frame of x_i via weighted sum:

$$\tilde{x_i} = (1 - a)x_i + ax_j[f]$$

Here, $f \sim U(\{1, \ldots, T\}$ and $a \sim U(0.25, 0.55)$. In practise, to cover the entirety of the canvas, $N_{back} = 2$ static frames are sampled and are next aggregated with the maximum operation. Furthermore, a random rectangle is cropped from the resultant image and interpolated to input dimensions.

It has been assumed that the background remains motionless. However, real videos often include dynamic elements such as bystanders or water waves. This can be addressed with a modification to the previous approach. Specifically, the given video is mixed not with a single static frame but with a sequence of frames sampled from a different video. Compared to foreground, the magnitude of the background motion should be smaller. Thus, within this sequence, the frame index can either be incremented by one, remain the same, or decrease by one at each timestep. Additionally, the difference between the maximum and minimum frame indices is constrained. For each video in a batch, the type of background is determined by a Bernoulli trial with probability 0.8. Success results in static background, whereas failure in dynamic.

Foreground Scaling and Placement. In synthetic videos, fractal shapes cover the majority of the canvas and are usually positioned around its center. On the contrary, in real videos, position and size of the foreground significantly vary. To address this contradiction, synthetic videos are downsampled in the two spatial dimensions with scales $s_h, s_w \sim U(s_{min}, s_{max})$ and placed in a random position of an empty canvas. We set $s_{min} = 0.3$ and $s_{max} = 1.0$.

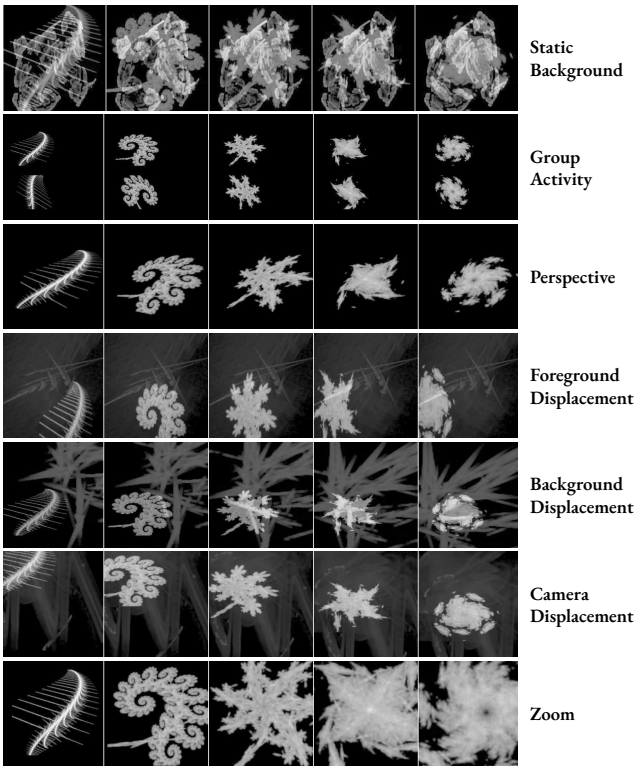


Fig. 5: Examples of the proposed domain adaptation methods. The purpose of these augmentations is to narrow the domain gap between real and synthetic videos.

Group Activity. Videos that display groups performing similar activities (e.g. aerobics) can be approximated with a modification to the previous augmentation. Specifically, after the interpolation step, the synthetic video is copied resulting in $N_{clone} = 2$ clones with each one receiving a different mild augmentation. Augmentations, which render the copies asynchronous, include random rotation, horizontal flipping and temporal offset. As previously, each copy is then placed in a random location of the canvas. We set $s_{min} = 0.2$ and $s_{max} = 0.7$ to reduce overlap between copies.

Perspective. Real cameras record objects from any angle in 3D, whereas fractals are rendered in 2D. As a compromise, minor angle variance can be induced using a random perspective transformation[1]. This augmentation is expected to amplify the model's spatial perception.

Relative Displacement. Aside from the previously mentioned motion, real videos contain additional relative motion between the foreground, the background and the camera:

– **Foreground Displacement**. This occurs when camera is static and individuals perform actions

[1] We employ the `RandomPerspective` transform from `torchvision`.

while simultaneously walking or running. As such, in the captured video, the position of the foreground is shifted while the background remains unaffected.
– **Background Displacement**. This is observed when the human target is displaced and the camera follows it. Consequently, the position of the human subject remains virtually stationary, while the background undergoes an equal displacement in the opposite direction. A notable example is the camera that follows athletes in a running track.
– **Camera Displacement**. In this case, the position of the human target remains unchanged but the focus of the camera is being shifted. In the resultant video, both foreground and background are relatively displaced in the opposite direction of the camera's movement.

Invariance to such movements can be boosted with simple transformations. For background displacement, a static background frame is initially enlarged and then a sequence of crops with dimensions of the original video is created. The sequence is then mixed with the foreground. As the centers of the crops are consecutive points on a 2D line, the result is displacement towards a fixed direction. For camera displacement, the process is similar, with the exception that crops are taken after mixing the foreground with an unaltered background. Lastly, for foreground displacement, the foreground is initially reduced in size and then each frame is placed at a different position inside the background.

Camera Zoom. For implementation, the video is initially interpolated to larger spatial dimensions. This is followed by central cropping which is applied with different scale for each frame. Increasing and decreasing the scale results in zooming out and zooming in respectively. As a final step, each cropped frame is interpolated back to the original spatial dimensions of the video. This implementation differs from the previously proposed camera displacement. The latter alters the position the displayed shapes between consecutive frames. The former alters their size.

Camera Shake. Videos captured by hand-held devices often contain shaking. This phenomenon can be approximately synthesized following [61]. Specifically, the displacement in each dimension is modelled as:

$$d_t = \sum_{i=1}^{n} \frac{1}{i} \sin(2\pi f_i t + \phi) + \eta_t, t = 1, 2, \ldots, T$$

Here, n is the number of components, T is the duration of the video in frames, f_i is the frequency of each component, ϕ denotes the phase and η is the noise component. These parameters are sampled as follows:

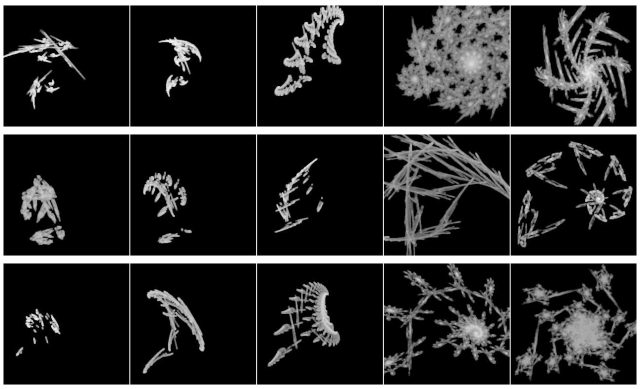


Fig. 6: Examples of the proposed mutation scheme. Each row displays a separate video. All videos belong to the same category, but exhibit differences due to randomly sampled noise, which is injected into the parameters.

$n \sim U(\{2, \ldots 5\}), f_i \sim U(0.1, 1.2), \phi_i \sim U(0, 2\pi), \eta_t \sim U(-0.3, 0.3)$. To apply the shaking effect, two displacement sequences are sampled. Afterwards, the video is enlarged and each frame is cropped. The position of the crop is determined by the two displacement values. This operation can be considered as an extension of the camera displacement augmentation, as it replaces the simple forward camera motion with a complex oscillation.

2.3.1 Automatic Construction of Categories

Algorithm 1 describes a method to sample parameters for a fractal video. Repeatedly executing this algorithm results in a dataset where each video is unique and no information exists about correlation of different samples. Therefore, in the context of machine learning, such a dataset is only suitable for unsupervised learning.

For a supervised classification objective, synthetic videos must be divided into categories (Fig. 6). These can be automatically constructed by adapting the approach of [42]. Specifically, given a predefined number of categories C, we initially sample parameters for C distinct fractal videos. This is achieved by executing Alg. 1 modified with nonlinear motion and diversity domain adaptations (Sec. 2.3). As such, each category c is represented by a parameter matrix $W_c \in \mathbb{R}^{T_c \times N_c \times 6}$ as well as a variation index var_c. Here, T_c is the number of frames in the video, N_c the number of IFS functions and var_c determines the type of nonlinear function utilized in rendering the video (Sec. 2.1.2).

To produce a new video belonging to class c, we mutate the respective parameters W_c with the auxiliary "noise" matrices m_a and m_b. The parameter matrix of the final video is calculated as follows:

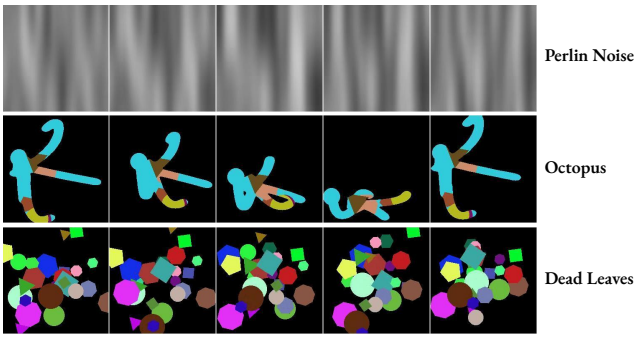


Fig. 7: Examples of alternative synthetic videos. Compared to fractals, these videos are less diverse, whereas perlin noise and dead leaves additionally lack distinct contours.

$$\tilde{W}_c = m_a \odot W_c + m_b$$

Here, $m_a \in \mathbb{R}^{T_c \times 1 \times 6}$ is the scaling component of the noise, which consists of 6 random curves that are sampled in the same manner as the random nonlinear interpolant (Sec. 2.3). The curves are bound between $[-0.35, 0.35]$. On the other hand, $m_b \in \mathbb{R}^{1 \times N_c \times 6}$ is the bias component, which is sampled from $U(-0.2, 0.2)$. Lastly, $\odot$ denotes elementwise multiplication. With this mutation scheme, we achieve variance within the same class and increased difficulty during supervised pre-training. However, it is noteworthy that the produced categories are randomly sampled and therefore, unlike real datasets such as Kinetics [10,43], possess no interpretable information (i.e., each class does not represent an actual human action).

2.4 Alternative Synthetic Videos

Despite their appealing properties (Sec. 2.1), it is not evident that fractal animations are appropriate for training strong visual representations for the task of action recognition. Hence, this section presents alternative generative processes of video (Fig. 7), which will be compared against fractals during experiments. Each generative process produces videos with different characteristics and the objective of the upcoming experiments is to determine which attributes are favorable for downstream results.

Perlin Noise. Adapting the approach of [39], we can generate videos of perlin noise [58,59], a variant of random texture. The label of each video is defined by three frequencies: two spatial and one temporal. These determine the rate of change in their respective dimensions. Unlike fractals, these videos are nebulous, lacking distinct shape and contours. With regard to the proposed domain adaptation techniques, perlin noise is not compatible with diversity and nonlinear motion.

Octopus. By sampling two 1D waveforms, one can construct a random 2D curve. To create an animation, two such curves can be sampled and their coordinates interpolated as has been shown for IFS. This process is repeated N times and the resulting curves are conjoined at a fixed point. As the outcome consists of thin curves, thickness is increased by applying Gaussian blur followed by the operation of morphological closing [71]. These videos will be referred to as "octopus" due to similarity to the mollusk. We additionally apply colorization, interior removal via the morphological gradient [71] and addition of geometrical shapes. Octopus videos resemble fractals due to their distinct contours, but are less diverse. These videos are not suitable for the proposed diversity amplification technique.

Dead Leaves. Dead leaves [62,47] is a simple image model that emulates statistics of natural images, such as the $1/|f|^a$ power spectrum. Such images can be constructed by randomly filling a canvas with geometric shapes, such as circles and regular polygons. More recently, dead Leaves have been employed in deep learning as synthetic images for pre-training [5] and it was deduced that stronger representations are obtained when the said shapes vary in terms of size, color and number edges. To extend this generative process to the video domain, a 2D curve is sampled for each shape. Throughout the video, the shape traverses this curve. Lastly, dead leaves lack distinct contours, while being incompatible with diversity enhancement.

2.5 Training Objective

So far, we have described the proposed method and the respective training procedure. However, the proposed training relies on supervised learning over a pre-defined number of categories, arbitrary generated by our sampling procedure. Despite the non-intuitive formalization of classes (non-intuitive in the sense that the defined distinct categories are arbitrarily selected and do not correspond to a human-related action), such a supervised learning approach is expected to provide good visual embeddings. On the other hand, one may wonder what happens if we do not define such distinct classes and treat our problem as a self-supervised paradigm. To this end, for pre-training, aside from the supervised objective from Sec. 2.3.1, we additionally explore algorithms from the self-supervised learning (SSL) literature, which do not require annotation. During SSL training, a pretext task is designed for a deep learning algorithm to solve and pseudolabels for the pretext task are automatically constructed based on attributes

of the input. Specifically we employ the SSL frameworks SimCLR [12], MoCoV2 [13] and BYOL [29].

SimCLR. Given a batch of size B, SimCLR applies heavy augmentation twice resulting in $2B$ input samples. For each positive pair (i, j) originating from the same sample, the other $2(B-1)$ instances are treated as negatives. The contrastive prediction loss is calculated as:

$$L_{i,j} = -\log \frac{\exp(\text{sim}(z_i, z_j)/T)}{\sum_{\substack{m=1 \\ m \neq i}}^{2B} \exp(\text{sim}(z_i, z_m)/T)}$$

Here, z_i is the model output for the instances i, $\text{sim}(\cdot, \cdot)$ is cosine similarity and T is the temperature hyperparameter. SimCLR requires two forward and two backward passes per training step. Furthermore, large batch size is preferred for training as it leads to increased difficulty and improved downstream results. As such, SimCLR is computationally heavier than supervised classification.

MoCoV2. MoCoV2 similarly employs a contrastive approach resulting in two views per input: q and k_+. However, unlike before, k_+ is produced by a non-differentiable model which is updated as an exponential moving average of the original. Moreover, k_+ is added to circular queue of size K. The loss function maximizes the similarity between q and k_+ while minimizing it between q and all other instances in the queue:

$$L = -\log \frac{\exp(q \cdot k_+/T)}{\sum_{i=1}^{K} \exp(q \cdot k_i/T)}$$

In terms of resources, MoCoV2 requires two forward and one backward pass, while the dependency on large batch size is alleviated due to the queue. Hence MoCoV2 is more lightweight than SimCLR but still more demanding than supervised classification.

BYOL. Lastly, BYOL does not utilize negative pairs. In a manner akin to MoCoV2, BYOL employs two neural networks named online and target, with only the former one being differentiable. As before, two views of the input are constructed: q and k. Both views are fed through both encoders resulting in four output representations in total. The training objective is to simply maximize the cosine similarity between all matching representations:

$$q_o, k_o = \text{encoder\_online}(q), \text{encoder\_online}(k)$$
$$q_t, k_t = \text{encoder\_target}(k), \text{encoder\_target}(q)$$
$$L = 2 - 2 * (\text{sim}(q_o, k_t) + \text{sim}(k_o, q_t))$$

Computationally, each training step requires four forward and two backward passes, rendering it the most demanding framework in the present work. Nonetheless, unlike SimCLR, large batch sizes are not required.

3 Experiments

3.1 Implementation Details

Datasets. The proposed pre-training framework is evaluated by fine-tuning the model on 6 small-scale video classification datasets:

- HMDB51 [46] and UCF101 [65], established action recognition benchmarks.
- DIVING48 [48], a collection of videos from diving competitions.
- EGTEA GAZE+ [49], which consists of first person cooking videos.
- VOLLEYBALL [36], a group action recognition dataset.
- YUP++ [24], which consists of videos depicting dynamic scenes that are not relevant to action recognition.

Indicative examples from these datasets are displayed in Fig. 8. The datasets exhibit significant differences and were chosen to maximize overall diversity. An official train-validation split is provided for each dataset. Thus, after pre-training, we fine-tune models on the training set and report the top-1 accuracy on the validation set of the downstream datasets. Synthetic videos are rendered with spatial resolution of 256×256 and temporal length which is sampled from $\text{U}(\{18, \ldots, 20\})$. For fractals, 50% of videos are rendered with nonlinearities (Sec. 2.1.2).

Model Architecture. Temporal Shift Module (TSM) [50] is employed for all experiments with ResNet-50 [31] utilized as backbone. While preserving the efficiency of 2D CNNs, TSM can achieve results equivalent to 3D CNNs in action recognition tasks. Specifically, TSM efficiently exchanges information between neighboring frames by moving the feature map along the temporal dimension. Despite being computationally cheap, this operation possesses a strong spatio-temporal modeling ability. Solid accuracy can be achieved with as few as 8 input frames. Such short length significantly alleviates both the CPU and GPU bottlenecks. The former is evidenced by reduced dataloading whereas the latter by a reduced amount of computation within the neural network itself. Unless specified otherwise, pre-training is done from scratch and does not utilize any off-the-shelf checkpoints.

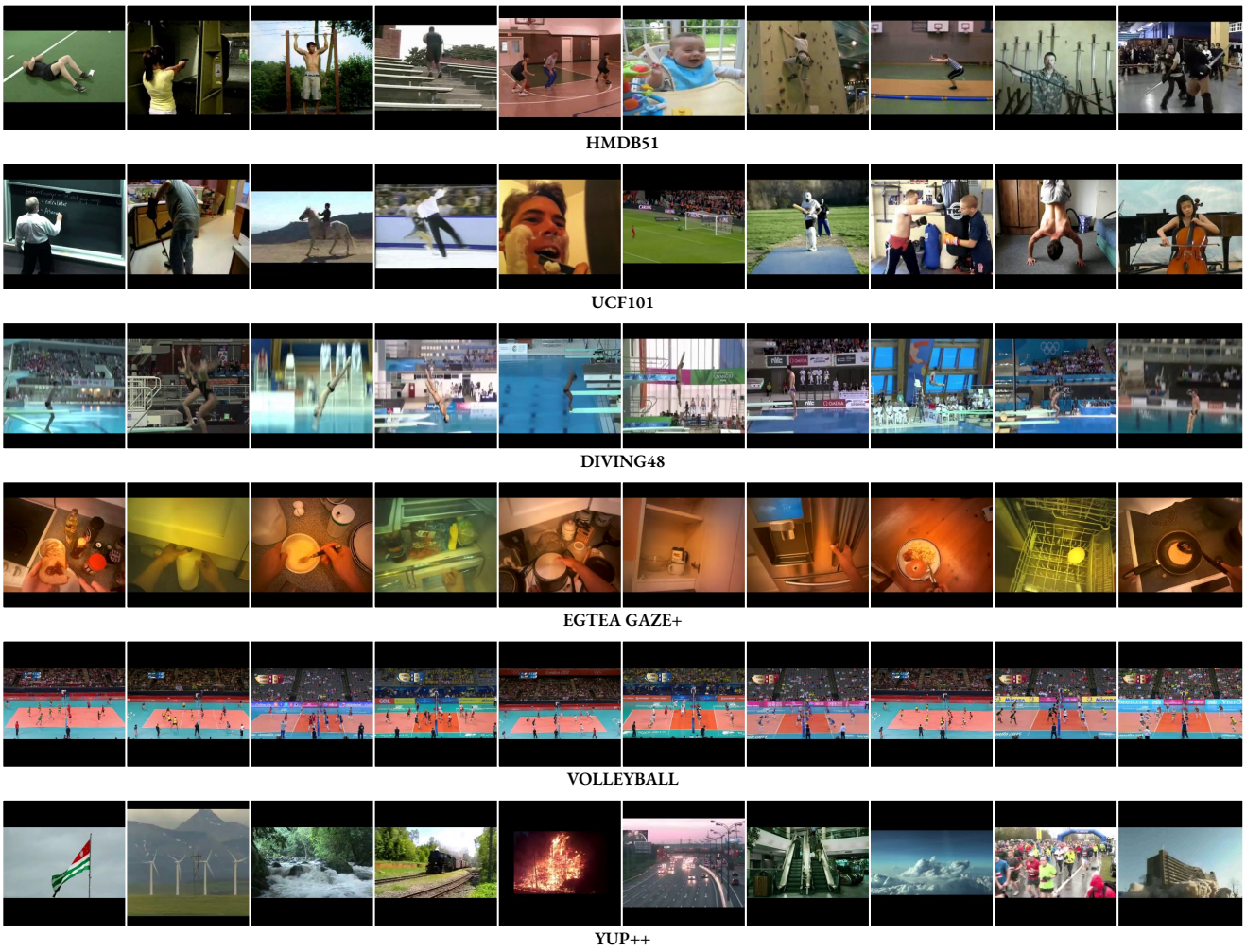


Fig. 8: Frames taken from the evaluated downstream datasets. Note the variance of the displayed background in all datasets aside from VOLLEYBALL and DIVING48.

Training. Only RGB frames are employed as model input in this work. Unless specified otherwise, the spatial resolution of the model input is 112×112. The clip consists of 8 strided frames. This allows us to decode only a fragment of the video file reducing dataloading latency. The stride of the input is 2 frames for pre-training, 4 for fine-tuning VOLLEYBALL and 6 for fine-tuning all other downstream datasets. During each training epoch, one such clip is produced from every video by sampling the index of the first frame from the uniform distribution. During validation, 10 such clips are uniformly selected from a video and separately fed into the model. The final output is the average of softmax scores.

The number of training epochs is 25 and 100 for pre-training and fine-tuning respectively. The number of warmup epochs for the learning rate scheduler is 3 and 10 respectively. The networks are optimized using AdamW [52] with $\beta_1 = 0.9$ and $\beta_2 = 0.999$. Cosine scheduling [51] is employed. The true learning rate is scaled according to the batch size: $lr_{true} = \frac{bs}{bs_{base}} lr$, where the batch size and base batch size are set to 32 and 32 respectively. The default learning rate is initialized at $1 \cdot 10^{-6}$, increases to $8 \cdot 10^{-4}$ during warmup and eventually falls to $1 \cdot 10^{-5}$ at the end of training. The default weight decay is set to $1 \cdot 10^{-2}$. Exceptions are the VOLLEYBALL dataset where the learning rates are $2.5 \cdot 10^{-6}$, $2.5 \cdot 10^{-3}$ and $2.5 \cdot 10^{-5}$ and weight decay is set to $1 \cdot 10^{-1}$ as well as DIVING48 where the learning rates are $1.5 \cdot 10^{-6}$, $1.5 \cdot 10^{-3}$ and $1.5 \cdot 10^{-5}$.

Augmentation. For fine-tuning, augmentation consists of random cropping with bicubic interpolation, horizontal flip, Randaugment [16] and Gaussian blur. For cropping, scale is sampled from $U(0.2, 1.0)$ and ratio from $U(0.75, 1.33)$. Augmentations are applied in the same order as they are listed. As an exception, for VOLLEYBALL we use area interpolation and omit horizontal flip and Gaussian blur.

Method	Kept	HMDB51	UCF101	DIVING48	EGTEA	VOLLEY	YUP
Scratch	-	31.5	70.3	4.7	50.5	53.8	43.7
Initial	-	41.4	72.7	24.3	49.8	80.6	52.3
Background	Yes	47.4	74.2	23.0	47.1	77.6	56.4
Motion	Yes	47.3	75.7	21.7	48.6	79.2	59.5
Diversity	Yes	50.4	77.8	24.9	49.8	80.8	63.1
Scale + Shake	Yes	52.8	78.0	26.0	50.7	80.8	61.9
Shift	Yes	**54.5**	79.6	29.5	50.8	80.9	65.1
Zoom	Yes	54.3	**80.2**	**30.8**	**52.0**	81.4	65.2
Perspective	No	53.3	78.7	25.0	50.0	80.7	62.6
Group	No	53.3	79.7	30.3	51.9	**81.9**	**66.0**

Table 1: Experiment 1 - Effectiveness of domain adaptation. In the majority of cases, emulation of a property boosts accuracy if the downstream dataset includes it (green color) and deteriorates accuracy if it does not (red color).

For pre-training, we retain the same augmentations with the addition of the proposed domain adaptation techniques. Within the aforementioned order, these are applied between horizontal flip and Randaugment. We employ a curriculum and linearly increase the intensity of domain augmentations for the first 5 epochs. All domain augmentations are applied with a probability of 0.3 with the exception of Background, Scale, Perspective and Group whose probabilities are 1.0, 1.0, 0.8 and 0.15 respectively. To accelerate computation, each domain adaptation method is applied in parallel and identically to all of its selected samples inside a batch. Background Randomization, Scale and Group are exceptions that are applied independently for each sample.

3.2 Experimental Results

After each experiment, we retain the configuration with highest accuracy on HMDB51 and UCF101, the focus of our work. Highest accuracy is indicated with **bold** font in the provided tables.

Experiment 1 - Domain Adaptation. Here we evaluate the proposed domain adaptation techniques, which are inserted sequentially into the pre-training framework. A technique will be discarded if improvement is not achieved for HMDB51 and UCF101. All pre-training is done with the MoCoV2 [13] self-supervised framework, due to its computational efficiency. The pre-training dataset consists of 100K unlabeled fractal videos.

From Table 1 it can be observed that, aside from a few exceptions (orange color), emulation of a property is beneficial for datasets that include it (green color) and detrimental for datasets that do not (red color). Hence, it can be inferred that self-supervised pre-training is more effective when the source and target domains are similar. As such, synthetic datasets should be maximally customized to mirror the properties of the target downstream dataset. This observation is in complete agreement with previous work [14, 45, 67]. A notable example is background randomization which increases accuracy for HMDB51 and UCF101, but decreases it for DIVING48 and VOLLEYBALL. For the former, background is not relevant for the category of the video and varies in each sample. However, for the latter, the background in all samples is similar and it can be assumed that neural models take it into account during classification.

Additionally, the only modification that results in non-trivial improvement across all benchmarks is amplified diversity of synthetic videos (blue color). This is again in line with previous work on the image modality [5]. One the contrary, the only modification that results in overall deterioration is random perspective (yellow color). It is noteworthy that compared to training from scratch, considerable improvement in results is attained across all datasets except EGTEA. This shortcoming is further investigated in Sec. 3.3.

Experiment 2 - Alternative Synthetic Data. We investigate pre-training with alternative synthetic datasets from Section 2.3.1. Each dataset exhibits different characteristics and the aim is to determine the ones that are favorable for downstream results.

Based on Table 2, fractal videos lead to superior results across all benchmarks compared to alternatives. Thus it can be deduced that fractals possess more appropriate properties for downstream tasks. Specifically, compared to the octopus dataset, fractals exhibit more diversity. On the other hand, what differentiates fractals from dead leaves and perlin noise is distinct contours. As distinct contours are an attribute of

Dataset	HMDB51	UCF101	DIVING48	EGTEA	VOLLEY	YUP
Octopus	50.9	76.5	25.5	49.1	80.7	58.3
Dead Leaves	39.9	70.2	15.8	47.3	75.7	50.2
Perlin Noise	42.4	72.5	21.4	50.1	76.4	58.2
Fractal	**54.3**	**80.2**	**30.8**	**52.0**	**81.4**	**65.2**

Table 2: Experiment 2 - Evaluation of different synthetic datasets. Fractal videos outperform all alternatives. The characteristics that render them superior are diversity and distinct contours.

Objective	HMDB51	UCF101	DIVING48	EGTEA	VOLLEY	YUP
MoCoV2	54.3	80.2	30.8	52.0	81.4	65.2
SimCLR	57.5	81.3	34.3	54.7	**83.2**	70.2
BYOL	52.0	78.1	26.1	52.0	80.4	66.0
Supervised	**61.5**	**84.9**	**38.5**	**54.8**	82.7	**73.6**

Table 3: Experiment 3 - Exploration of different training objectives. Pre-training through supervised classification yields superior results compared to SSL. A plausible rationale is that supervision is more resistant to the domain gap between real and synthetic videos.

real videos, it is safe to assume that including them in a synthetic dataset bridges the domain gap. The importance of this trait is further highlighted by the fact that octopus videos outperform both perlin noise and dead leaves. As such, the verdict of this experiment is identical to the previous one: to obtain stronger visual representations, synthetic video datasets have to exhibit diversity and mimic characteristics present in target data.

Experiment 3 - Training Objective. So far, MoCoV2 [13] has been exclusively employed for pre-training. This decision was made due to its computational efficiency. Thus, with the intention of maximizing downstream performance, this section explores alternative training objectives: self-supervised frameworks SimCLR [12] and BYOL [29] (Sec. 2.5) as well as a supervised classification objective (Sec. 2.3.1). For the former, the unlabeled fractal dataset from previous experiments is reused. For the latter, a new dataset is constructed with 500 classes and 200 samples per class, resulting in 100K training videos in total.

As seen in Table 3, the supervised objective confidently outperforms alternatives. At first glance this is perhaps surprising as the categories are sampled randomly and therefore lack meaningful information present in real classification datasets. A possible explanation relies on the fact that self-supervised frameworks have been shown to be especially vulnerable to the domain gap between source and target datasets [14,45,67]. Therefore, it can be assumed that the supervised pre-training objective is more resilient to the difference in domains and therefore results in representations that are transferred more robustly to a wide range of downstream tasks.

On a different note, compared to supervised training, self-supervised counterparts require multiple forward passes per step and benefit from larger batch size and increased number of epochs. Hence, it is likely that our SSL models are undertrained and allocating more resources would considerably improve downstream results. Regardless, supervised training performs well under resource constraints and therefore is a more cost-effective solution. A last observation involving the examined SSL frameworks is that SimCLR outperforms MoCoV2, which in turn outperforms BYOL. This is in complete contrast with ImageNet pre-training, where the order is reversed [29].

Experiment 4 - Dataset Size. This segment investigates how transferability to downstream tasks is impacted by two statistical characteristics of the synthetic classification dataset: the number of classes and the number of instances per class. The experiment is divided into two stages. In the first stage, the number of classes is fixed to 500 as in the previous experiment, while the instances are varied at 100, 200, and 400 per class. In the second stage, the number of instances is fixed to the optimal value determined in the first stage, while classes are varied at 250, 500 and 1000. To ensure fairness, each dataset is a superset for all smaller ones. For the largest of the assessed datasets, we additionally evaluate the perspective transform (Sec. 2.3), which previously led to deteriorated results in Experiment 1.

Observing Table 4, it can be deduced that downstream performance is almost a monotonically increas-

#Instance/#Total	HMDB51	UCF101	DIVING48	EGTEA	VOLLEY	YUP
100/50K	56.5	81.8	35.8	52.6	82.6	69.3
200/100K	61.5	84.9	38.5	**54.8**	82.7	73.6
400/200K	**61.5**	**86.0**	**40.8**	53.8	**83.0**	**75.1**

Table 4: Experiment 4A - Impact of the number of instances per category on downstream accuracy.

#Classes/#Total	HMDB51	UCF101	DIVING48	EGTEA	VOLLEY	YUP
250/100K	60.3	85.3	38.1	53.5	84.1	75.6
500/200K	61.5	86.0	40.8	53.8	83.0	75.1
1000/400K	62.4	87.3	**41.2**	**56.1**	**84.0**	**76.0**
1000/400K + Perspective	**65.4**	**87.6**	40.3	56.0	83.1	72.7

Table 5: Experiment 4B - Impact of the number of categories on downstream accuracy.

ing function of the number of instances per class. Judging by Table 5, the same conclusion can be reached for the number of classes. This behavior is not surprising as increasing the number of training videos exposes the model to more spatio-temporal patterns and renders the learnt representations more transferable to new tasks.

Regarding the perspective augmentation, in Table 5 accuracy is boosted for for action recognition datasets HMDB51 and UCF101. This is in contrast with Experiment 1 where performance drops. The difference between these experiments is the training objective: the former employs a supervised objective while the latter utilizes the self-supervised framework MoCoV2 [13]. As such, it is possible that the exact impact of the proposed domain adaptation techniques is dependant on the training objective.

Experiment 5 - Higher Resolution & Larger Scale. Given that all previous experiments have been conducted with a low spatial resolution of 112, we now increase it to 224. Furthermore, to align with the standards set by prior works on synthetic data, we expand the size of our synthetic dataset from 400K to 2M (5K classes and 400 instances per class). These datasets are referred to as Fractal-400K and Fractal-2M respectively. Additionally, we compare the results of fractal pre-training to the Kinetics counterpart [10, 43], which is the established approach for pre-training in action recognition tasks. To this end, we employ an off-the-shelf checkpoint from [50] that has also undergone training with a resolution of 224. Fine-tuning after Kinetics utilizes different hyperparameters which are documented in Appendix A.

As evidenced by Table 6, increasing the resolution of fractals from 112 to 224 leads to nontrivial improvement across all benchmarks. This is reasonable as real videos often contain details such as small objects, which cannot be displayed adequately with lower resolution. Moreover, increasing the size of the synthetic pre-training dataset, leads to considerable accuracy improvements for all small-resource datasets. This suggests that saturation has not yet been reached and further increases in the synthetic dataset size are likely to yield further gains. Additionally, it can be observed that synthetic pre-training surpasses Kinetics on the benchmarks DIVING48 and VOLLEYBALL. As seen in Fig. 8, in the former, all videos display swimming pools and their surroundings such audience seats while the latter is exclusively set in volleyball courts. As such, the attribute that distinguishes the datasets where fractals surpass Kinetics is a small variance of the displayed background.

Nonetheless, synthetic data still lags behind Kinetics on the majority of evaluated benchmarks. However, it has to be reminded that the fractals were constructed automatically and therefore mitigate collection and annotation costs required for Kinetics.

Experiment 6 - Larger Datasets

To provide a more robust analysis and offer greater utility for our proposed approach, we further conduct experiments involving significantly larger benchmarks Something-Something V2 (SSv2) [28] and Mini-Kinetics-200 [76]. Compared to most previously evaluated datasets, SSv2 is more dependant on motion rather than appearance. On the other hand, Mini-Kinetics-200 is a subset of Kinetics-400, containing approximately one third of the data. In the added experiments, we compare pre-training with fractals to training from scratch as well as ImageNet weight initialization, which is the standard approach for these two datasets. Aside from TSM, results are additionally provided for the I3D architecture [11]. I3D differs significantly from TSM, as the former is a 3D CNN while the latter is a 2D one. As such, we believe that

Pre-training	Res	HMDB51	UCF101	DIVING48	EGTEA	VOLLEY	YUP
Scratch	112	31.5	70.3	4.7	50.5	53.8	43.7
Fractal-400K	112	65.4	87.6	40.3	56.0	83.1	72.7
Scratch	224	35.1	76.9	10.6	53.4	55.7	48.3
Fractal-400K	224	66.5	90.8	41.2	59.9	87.6	78.2
Fractal-2M	224	68.2	91.9	**43.7**	60.6	**88.1**	79.6
Kinetics	224	**70.1**	**95.3**	40.9	**64.4**	84.8	**86.9**

Table 6: Experiment 5 - Effectiveness of higher spatial resolution & increased dataset size. Increasing the resolution boosts accuracy across all benchmarks and so does expanding the synthetic dataset. An interesting observation is that DIVING48 and VOLLEYBALL, where synthetic data surpasses Kinetics, exhibit the smallest variance of background among the evaluated datasets.

Model	Pre-training	SSv2	Mini-Kinetics-200
TSM	Scratch	55.8	67.8
TSM	ImageNet	58.8	76.4
TSM	Fractal-400K	59.7	77.0
TSM	Fractal-2M	**59.9**	**77.5**
I3D	Scratch	44.5	54.1
I3D	ImageNet	51.2	72.1
I3D	Fractal-400K	**52.6**	**73.4**

Table 7: Experiment 6 - Impact of pre-training on larger datasets SSv2 and Mini-Kinetics-200. Fractal videos outperform pre-training with static ImageNet images, but the overall benefit of pre-training is limited.

these models should be sufficient to demonstrate the generalizability of our approach.

As shown in Table 7, fractals generally outperform ImageNet across both models. Hence, it can be deduced that, compared to ImageNet, our proposed synthetic data can produce more powerful neural representations for motion-related datasets such as SSv2. A possible explanation is that synthetic videos contain temporal patterns, while ImageNet is composed of static images. However, the accuracy improvement after pre-training is significantly less perceptible compared to previous experiments. This can be justified by the size of these two datasets, which is clearly larger than all the previous datasets. As the size of the downstream dataset increases, the models can develop more robust and generalized features without the need for pre-training. This behavior is similarly confirmed by the difference between HMDB51 & UCF101 in previous experiments. While both contain videos of similar nature, the latter is twice as large but achieves half the accuracy increase from pre-training compared to the former ($\sim$15 & 30%). Finally, it is evident that increasing the quantity of synthetic videos has a significantly smaller impact on larger datasets compared to low-resource datasets. This observation aligns with our earlier explanations.

3.3 Manual Error Analysis

Upon manual inspection of miss-classified samples from downstream datasets, a recurring characteristic can be distinguished. In particular, models pre-trained with fractals struggle with videos whose label is dictated not by global information which covers the entire screen but by small details which occupy only a few pixels. A notable instance is small objects that are handled by humans. Examples are categories “Throw”, “Swing Baseball”, “Brush Teeth” and “Hammering” from datasets HMDB51 and UCF101. Likewise, the same holds for the entirety of EGTEA which displays cooking tools and ingredients. This justifies the ineffectiveness of synthetic pre-training: only a 6% accuracy gain is achieved compared to training from scratch. Another case involves facial expressions and motion of the mouth, as evidenced by categories “Smoke”, “Eat”, “Drink”, “Smile” and “Laugh” from HMDB51. Similarly, models are unsuccessful with limb movement. This is demonstrated again by EGTEA as well as the HMDB51 classes “Clap”, “Wave”, “Punch” and “Kick”. Frames from the aforementioned examples are displayed in Appendix Fig. 9.

This deficiency in not unexpected. In our proposed synthetic datasets, the label, which is necessary for the training objective, depends exclusively on the overall displayed fractal formation. Cases where the label is de-

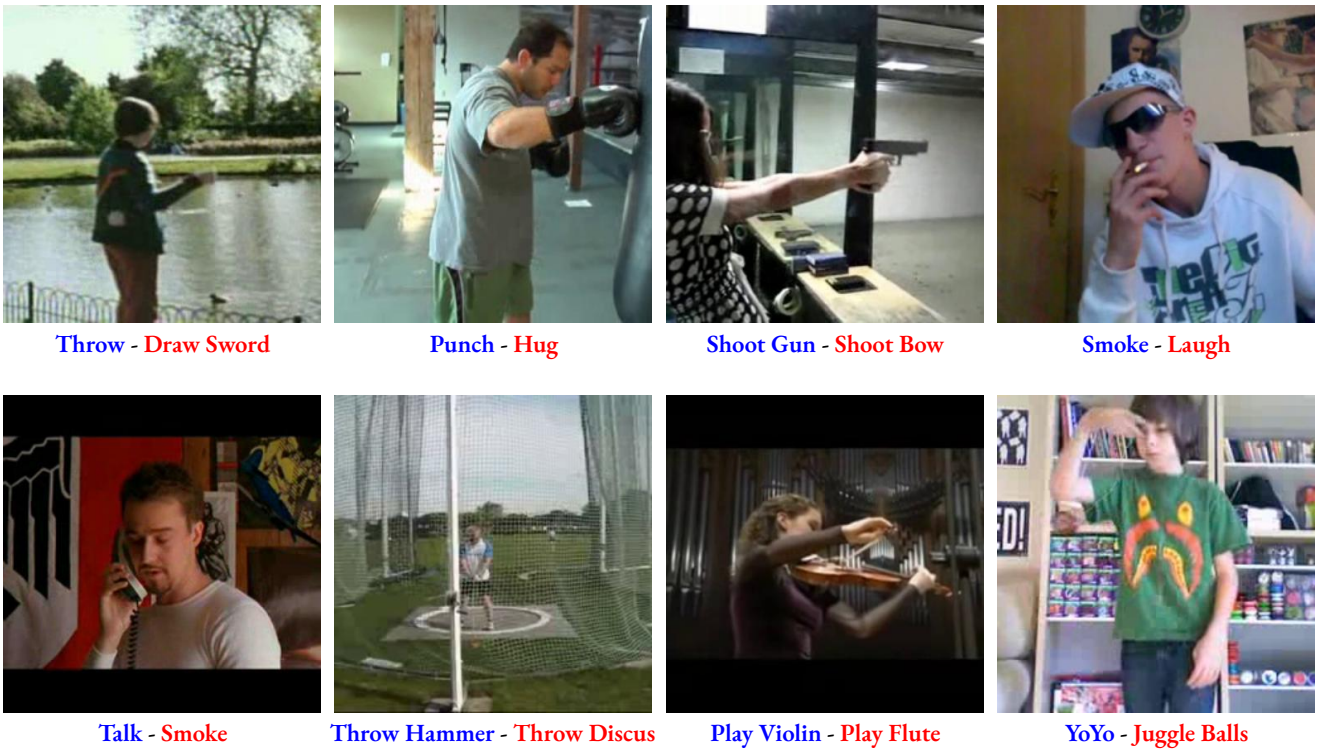


Fig. 9: Frames from misclassified videos. Blue color indicates ground truth, whereas red color indicates the model's incorrect prediction. In such videos the label is often determined by subtle details that cover a small percentage of the overall pixels. As such, the model fails to differentiate between similar categories.

termined by local details are nonexistent. Consequently, the proposed pre-training does not adequately prepare CNNs for the aforementioned real-world instances. This shortfall should be addressed by designing alternative generative processes in future work. Specifically, mirroring the conditions observed in real-world data, the constructed synthetic datasets should incorporate videos where the label is conditioned on a small percentage of pixels.

4 Related Work

Formula-driven synthetic data in computer vision. The work of [42] employed fractal geometry to automatically generate large-scale labeled image datasets to pre-train image models. Although their reported metrics are inferior to ImageNet pre-training [63], they evidently surpass training from scratch. Subsequent work has proposed a more intuitive augmentation policy [2] and demonstrated that the framework is compatible with different neural architectures [57]. More recently, [40] designed an alternative image synthesis method which results in representations that exceed ImageNet pre-training on specific model architectures. As an orthogonal approach, it has been demonstrated [5] that synthetic data results in strong representations only when certain conditions are met, such as replicating the attributes of real data. Thus, the design of synthetic data should be approached with meticulous care.

Synthetic data has also been utilized in other tasks. In particular, [78] pre-trained neural networks for point clouds with a synthetic datasets of 3D fractals. In the context of action recognition, [39] pre-trained neural models with 3D perlin noise [58,59]. Such videos are relevant to our work and thus serve as a baseline in later experiments. Additionally, [80] construct synthetic images of palm prints with the help of Bezier curves [22].

Lastly, on a relevant note, synthetic data can alternatively be created with neural models such as Generative Adversarial Networks [26] and Diffusion Models [33]. Indeed, it has been demonstrated that datasets of such synthetic images paired with unsupervised training objectives can produce impressive results that are on par or even superior to pre-training with real data [21,68]. This methodology have also been shown to benefit greatly from scaling the synthetic data. However, compared to the formula driven counterpart, this approach is computationally more expensive and requires

large amounts of real-world samples, suffering from the same limitations pertaining to real world datasets.

Video Diffusion Models. Diffusion models [33] are an emerging approach that has demonstrated impressive results in generation of images. More recently, diffusion has been extended to the domain of video [34], exhibiting very promising results and great potential in many fields. This methodology bears similarity to the present work as it can also produce large datasets of synthetic videos.

However, despite the remarkable amount of new research, video diffusion models are not yet sufficiently mature [34,55], posing multiple challenges that still need to be fully overcome. In the case of text-to-video models, a notable difficulty is the maintenance of temporal consistency. It has been observed that both the appearance and position of objects change wildly between video frames. This shortcoming has been constrained thanks to recent work, but not yet eliminated. Additionally, compared to our video synthesis methods, diffusion is typically significantly more demanding both in terms of resources and real-world training data. As such, video diffusion will not be investigated in the present work but remains an exciting prospect for future research.

Action Recognition. Contemporary action recognition models are first pre-trained on large-scale generic curated video datasets such as Kinetics [10,43], Moments in Time [56] or YouTube 8M [1]. This process utilizes a supervised training objective. Alternatively, strong representations can also be obtained by utilizing the vast amount of available unlabelled videos and employing an unsupervised training objective [23]. Subsequently, transfer learning is employed and the models are fine-tuned on smaller specialized datasets such as UCF101 [65], ActivityNet [32] and HMDB51 [46]. Omitting the first step significantly deteriorates downstream results. As such, this paper seeks to replace real large-scale video datasets with synthetic and automatically generated ones. Regarding neural architecture, the most common approaches are 2D [64,74] and 3D [11, 30,70] CNNs. However, recently, Vision Transformers [69,73] have also begun to receive increased popularity.

Applications of Fractals. Owing to their aesthetic qualities, 2D fractals have been utilized in art [20,19]. Furthermore, fractals are prominent in the field of image compression [38]. It is noteworthy that fractal geometry plays a significant role in domains outside of imaging including signal analysis [53,44,17], speech recognition [54,60,81] and telecommunications [3]. Lastly, due to their resemblance to biological structures, fractals have been employed in medical simulation [15,37].

5 Discussion and Conclusion

The present work automatically constructs synthetic datasets of short video clips. Such videos can be utilized for pre-training neural networks for the task of action recognition. Compared to real data, this approach eliminates the necessity for manual dataset collection and annotation. Additionally, in pursuit of minimizing the domain gap between real and synthetic videos, we introduce a set of heuristic domain adaptation techniques which mimic characteristics present in real data. The overall objective of our work is to determine properties of synthetic data as well as general guidelines that strengthen downstream performance. Observing experimental results, the following conclusions are reached:

- Diversity of synthetic videos is a key factor for obtaining stronger visual representations and can boost results regardless of the characteristics of the downstream dataset. This is in agreement with previous work [5].
- Reducing the domain gap between real and synthetic videos also strengths downstream results. This can be achieved by emulating structural and motion-related properties of former during pre-training. Strict realism is not necessary and rough approximations of the said properties are sufficient. A few examples are background randomization, periodic motion and camera shaking.
- Supervised pre-training is a more cost-effective solution compared to self-supervised counterparts, achieving superior results under limited resources.
- Increasing the dataset size or spatial resolution consistently improves transferability.

It has to be noted that video action recognition is a field where access to video data on the scale of millions is currently relatively straightforward. In this regard, synthetic videos are unnecessary. However, we firmly believe that our findings are generalizable and can be transferred to other domains and tasks where available samples are scarce. A notable example could be medical video understanding, which could greatly benefit from synthetic data since obtaining real-world counterparts can be very expensive. Nonetheless, the proposed methodology suffers from multiple limitations and improvements are expected to be added in future work. Specifically:

- On the majority of evaluated benchmarks, synthetic pre-training lags behind Kinetics. We hypothesize that the gap in performance can be reduced, but not eliminated, by further increasing the quantity of synthetic training samples.

– Models pre-trained with synthetic data underperform in the detection of details, such as tools or facial expressions. As such, our proposed approach is not effective for datasets such as EGTEA GAZE+. Future work should mitigate this by constructing synthetic categories that are conditioned on a local cues and not global information, mirroring real data.
– All synthetic videos produced in this work are of short length and depict a single motion. Thus, our framework is not suitable for applications involving longer videos, where the ability to model contextual relation between distant frames is required.
– Another limitation of our study is the lack of benchmarking against established works such as [23], something crucial for situating our contributions within the existing literature. However, it is important to consider that pre-training with synthetic video is a nascent research direction. Our study is among the earliest investigations. Consequently, it is reasonable to expect that our synthetic data will underperform compared to [23] or similar works using real-world data. Therefore, at this stage, we believe it is more appropriate to benchmark our work against other studies on synthetic videos rather than real-world datasets. To this end, we conduct experiments with data from [39], which is the most relevant study to ours. Furthermore, focusing on understanding the properties of synthetic videos that enhance downstream performance is more practical at this point. By doing so, we hope to provide guidelines for future research to create more effective synthetic videos for training neural networks.

References

1. Abu-El-Haija, S., Kothari, N., Lee, J., Natsev, P., Toderici, G., Varadarajan, B., Vijayanarasimhan, S.: Youtube-8m: A large-scale video classification benchmark. In: arXiv preprint arXiv:1609.08675 (2016) 16
2. Anderson, C., Farrell, R.: Improving fractal pre-training. In: Proceedings of the IEEE Winter Conference on Applications of Computer Vision (WACV) (2022) 1, 3, 4, 5, 15
3. Anguera, J., Andújar, A., Jayasinghe, J., Chakravarthy, V.V.S.S.S., Chowdary, P.S.R., Pijoan, J.L., Ali, T., Cattani, C.: Fractal antennas: An historical perspective. In: Fractal and Fractional, vol. 4, no. 1 (2020) 16
4. Asano, Y., Rupprecht, C., Zisserman, A., Vedaldi, A.: Pass: An imagenet replacement for self-supervised pretraining without humans. In: Proceedings of the International Conference on Neural Information Processing Systems (NeurIPS) Datasets and Benchmarks Track (2021) 1
5. Baradad Jurjo, M., Wulff, J., Wang, T., Isola, P., Torralba, A.: Learning to see by looking at noise. In: Proceedings of the International Conference on Neural Information Processing Systems (NeurIPS) (2021) 1, 4, 6, 8, 11, 15, 16
6. Barnsley, M.F.: Fractals Everywhere. Morgan Kaufmann (1993) 1, 2, 3
7. Birhane, A., Prabhu, V.U.: Large image datasets: A pyrrhic win for computer vision? In: Proceedings of the IEEE Winter Conference on Applications of Computer Vision (WACV) (2021) 1
8. Buolamwini, J., Gebru, T.: Gender shades: Intersectional accuracy disparities in commercial gender classification. In: Proceedings of the Conference on Fairness, Accountability and Transparency (2018) 1
9. Burch, B., Hart, J.C.: Linear fractal shape interpolation. In: Proceedings of the Graphics Interface Conference (1997) 4
10. Carreira, J., Noland, E., Hillier, C., Zisserman, A.: A short note on the kinetics-700 human action dataset. In: arXiv preprint arXiv:1907.06987 (2019) 2, 8, 13, 16
11. Carreira, J., Zisserman, A.: Quo vadis, action recognition? a new model and the kinetics dataset. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2017) 13, 16
12. Chen, T., Kornblith, S., Norouzi, M., Hinton, G.: A simple framework for contrastive learning of visual representations. In: Proceedings of the International Conference on Machine Learning (ICML) (2020) 9, 12
13. Chen, X., Fan, H., Girshick, R., He, K.: Improved baselines with momentum contrastive learning. In: arXiv preprint arXiv:2003.04297 (2020) 9, 11, 12, 13
14. Cole, E., Yang, X., Wilber, K., Mac Aodha, O., Belongie, S.: When does contrastive visual representation learning work? In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2022) 4, 11, 12
15. Costabal, F., Hurtado, D., Kuhl, E.: Generating purkinje networks in the human heart. In: Journal of Biomechanics, vol. 49, pp. 2455–2465 (2015) 16
16. Cubuk, E.D., Zoph, B., Shlens, J., Le, Q.: Randaugment: Practical automated data augmentation with a reduced search space. In: Proceedings of the International Conference on Neural Information Processing Systems (NeurIPS) (2020) 10
17. Dimakis, A., Maragos, P.: Phase-modulated resonances modeled as self-similar processes with application to turbulent sounds. In: IEEE Transactions on Signal Processing, vol. 53, no. 11, pp. 4261–4272 (2005) 16
18. Ding, S., Li, M., Yang, T., Qian, R., Xu, H., Chen, Q., Wang, J., Xiong, H.: Motion-aware contrastive video representation learning via foreground-background merging. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2022) 6
19. Draves, S.: The electric sheep screen-saver: A case study in aesthetic evolution. In: Proceedings of the European conference on Applications of Evolutionary Computing (2005) 16
20. Draves, S., Reckase, E.: The fractal flame algorithm. (2008) (2008) 4, 6, 16
21. Fan, L., Chen, K., Krishnan, D., Katabi, D., Isola, P., Tian, Y.: Scaling laws of synthetic images for model training ... for now. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2024) 15
22. Farin, G.: Curves and Surfaces for Computer Aided Geometric Design: A Practical Guide. Academic Press Professional, Inc. (1988) 15

23. Feichtenhofer, C., Fan, H., Xiong, B., Girshick, R., He, K.: A large-scale study on unsupervised spatiotemporal representation learning. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2021) 16, 17
24. Feichtenhofer, C., Pinz, A., Wildes, R.P.: Temporal residual networks for dynamic scene recognition. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2017) 9
25. Ghadiyaram, D., Tran, D., Mahajan, D.: Large-scale weakly-supervised pre-training for video action recognition. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2019) 1
26. Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., Bengio, Y.: Generative adversarial nets. In: Proceedings of the International Conference on Neural Information Processing Systems (NeurIPS) (2014) 15
27. Goyal, P., Caron, M., Lefaudeux, B., Xu, M., Wang, P., Pai, V., Singh, M., Liptchinsky, V., Misra, I., Joulin, A., Bojanowski, P.: Self-supervised pretraining of visual features in the wild. In: arXiv preprint arXiv:2103.01988 (2021) 1
28. Goyal, R., Ebrahimi Kahou, S., Michalski, V., Materzynska, J., Westphal, S., Kim, H., Haenel, V., Fruend, I., Yianilos, P., Mueller-Freitag, M., Hoppe, F., Thurau, C., Bax, I., Memisevic, R.: The "something something" video database for learning and evaluating visual common sense. In: Proceedings of the IEEE International Conference on Computer Vision (ICCV) (2017) 13
29. Grill, J.B., Strub, F., Altché, F., Tallec, C., Richemond, P., Buchatskaya, E., Doersch, C., Avila Pires, B., Guo, Z., Gheshlaghi Azar, M., Piot, B., kavukcuoglu, k., Munos, R., Valko, M.: Bootstrap your own latent - a new approach to self-supervised learning. In: Proceedings of the International Conference on Neural Information Processing Systems (NeurIPS) (2020) 9, 12
30. Hara, K., Kataoka, H., Satoh, Y.: Can spatiotemporal 3d cnns retrace the history of 2d cnns and imagenet? In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2018) 16
31. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2016) 9
32. Heilbron, F.C., Escorcia, V., Ghanem, B., Niebles, J.C.: Activitynet: A large-scale video benchmark for human activity understanding. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2015) 16
33. Ho, J., Jain, A., Abbeel, P.: Denoising diffusion probabilistic models. In: Proceedings of the International Conference on Neural Information Processing Systems (NeurIPS) (2020) 15, 16
34. Ho, J., Salimans, T., Gritsenko, A., Chan, W., Norouzi, M., Fleet, D.J.: Video diffusion models. In: Proceedings of the International Conference on Neural Information Processing Systems (NeurIPS) (2022) 16
35. HUTCHINSON, J.E.: Fractals and self similarity. In: Indiana University Mathematics Journal, vol. 30, no. 5, pp. 713–747 (1981) 3
36. Ibrahim, M.S., Muralidharan, S., Deng, Z., Vahdat, A., Mori, G.: A hierarchical deep temporal model for group activity recognition. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2016) 9
37. Ionescu, C., Oustaloup, A., Levron, F., Melchior, P., Sabatier, J., De Keyser, R.: A model of the lungs based on fractal geometrical and structural properties. In: IFAC Proceedings Volumes, vol. 42, no. 10, pp. 994–999 (2009) 16
38. Jacquin, A.E.: Image coding based on a fractal theory of iterated contractive image transformations. In: IEEE transactions on image processing, vol. 1 1, pp. 18–30 (1992) 16
39. Kataoka, H., Hara, K., Hayashi, R., Yamagata, E., Inoue, N.: Spatiotemporal initialization for 3d cnns with generated motion patterns. In: Proceedings of the IEEE Winter Conference on Applications of Computer Vision (WACV) (2022) 8, 15, 17
40. Kataoka, H., Hayamizu, R., Yamada, R., Nakashima, K., Takashima, S., Zhang, X., Martinez-Noriega, E.J., Inoue, N., Yokota, R.: Replacing labeled real-image datasets with auto-generated contours. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2022) 15
41. Kataoka, H., Matsumoto, A., Yamada, R., Satoh, Y., Yamagata, E., Inoue, N.: Formula-driven supervised learning with recursive tiling patterns. In: Proceedings of the IEEE International Conference on Computer Vision (ICCV) Workshops (2021) 1
42. Kataoka, H., Okayasu, K., Matsumoto, A., Yamagata, E., Yamada, R., Inoue, N., Nakamura, A., Satoh, Y.: Pre-training without natural images. In: Proceedings of the Asian Conference on Computer Vision (ACCV) (2020) 1, 2, 3, 7, 15
43. Kay, W., Carreira, J., Simonyan, K., Zhang, B., Hillier, C., Vijayanarasimhan, S., Viola, F., Green, T., Back, T., Natsev, P., Suleyman, M., Zisserman, A.: The kinetics human action video dataset. In: arXiv preprint arXiv:1705.06950 (2017) 2, 8, 13, 16
44. Kokkinos, I., Maragos, P.: Nonlinear speech analysis using models for chaotic systems. In: IEEE Transactions on Speech and Audio Processing, vol. 13, no. 6, pp. 1098–1109 (2005) 16
45. Kotar, K., Ilharco, G., Schmidt, L., Ehsani, K., Mottaghi, R.: Contrasting contrastive self-supervised representation learning pipelines. In: Proceedings of the IEEE International Conference on Computer Vision (ICCV) (2021) 4, 11, 12
46. Kuehne, H., Jhuang, H., Garrote, E., Poggio, T., Serre, T.: Hmdb: A large video database for human motion recognition. In: Proceedings of the IEEE International Conference on Computer Vision (ICCV) (2011) 5, 9, 16
47. Lee, A.B., Mumford, D., Huang, J.: Occlusion models for natural images: A statistical study of a scale-invariant dead leaves model. In: International Journal of Computer Vision (IJCV), vol. 41, pp. 35–59 (2004) 8
48. Li, Y., Li, Y., Vasconcelos, N.: Resound: Towards action recognition without representation bias. In: Proceedings of the European Conference on Computer Vision (ECCV) (2018) 9
49. Li, Y., Liu, M., Rehg, J.M.: In the eye of beholder: Joint learning of gaze and actions in first person video. In: Proceedings of the European Conference on Computer Vision (ECCV) (2018) 9
50. Lin, J., Gan, C., Han, S.: Tsm: Temporal shift module for efficient video understanding. In: Proceedings of the IEEE International Conference on Computer Vision (ICCV) (2019) 9, 13, 20
51. Loshchilov, I., Hutter, F.: SGDR: Stochastic gradient descent with warm restarts. In: Proceedings of the Interna-

tional Conference on Learning Representations (ICLR) (2017) 10
52. Loshchilov, I., Hutter, F.: Decoupled weight decay regularization. In: Proceedings of the International Conference on Learning Representations (ICLR) (2019) 10
53. Maragos, P.: Fractal signal analysis using mathematical morphology. In: Advances in Electronics and Electron Physics, vol. 88, pp. 199–246 (1994) 16
54. Maragos, P., Potamianos, A.: Fractal dimensions of speech sounds: Computation and application to automatic speech recognition. In: The Journal of the Acoustical Society of America, vol. 105, pp. 1925–32 (1999) 16
55. Melnik, A., Ljubljanac, M., Lu, C., Yan, Q., Ren, W., Ritter, H.: Video diffusion models: A survey. In: arXiv preprint arXiv:2405.03150 (2024) 16
56. Monfort, M., Andonian, A., Zhou, B., Ramakrishnan, K., Bargal, S.A., Yan, T., Brown, L., Fan, Q., Gutfreund, D., Vondrick, C., Oliva, A.: Moments in time dataset: One million videos for event understanding. In: IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 42, pp. 502–508 (2020) 16
57. Nakashima, K., Kataoka, H., Matsumoto, A., Iwata, K., Inoue, N., Satoh, Y.: Can vision transformers learn without natural images? In: Proceedings of the AAAI Conference on Artificial Intelligence (2022) 15
58. Perlin, K.: An image synthesizer. In: Proceedings of the Annual Conference on Computer Graphics and Interactive Techniques (SIGGRAPH) (1985) 8, 15
59. Perlin, K.: Improving noise. In: Proceedings of the Annual Conference on Computer Graphics and Interactive Techniques (SIGGRAPH) (2002) 8, 15
60. Pitsikalis, V., Maragos, P.: Analysis and classification of speech signals by generalized fractal dimension features. In: Speech Communication, vol. 51, no. 12, pp. 1206–1223 (2009) 16
61. Qu, H., Song, L., Xue, G.: Shaking video synthesis for video stabilization performance assessment. In: Proceedings of the Visual Communications and Image Processing (VCIP) (2013) 7
62. Ruderman, D.L.: Origins of scaling in natural images. In: Vision Research, vol. 37, no. 23, pp. 3385–3398 (1997) 8
63. Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., Huang, Z., Karpathy, A., Khosla, A., Bernstein, M., Berg, A.C., Fei-Fei, L.: Imagenet large scale visual recognition challenge. In: International Journal of Computer Vision (IJCV), vol. 115, no. 3, pp. 1573–1405 (2015) 1, 15
64. Simonyan, K., Zisserman, A.: Two-stream convolutional networks for action recognition in videos. In: Proceedings of the International Conference on Neural Information Processing Systems (NeurIPS) (2014) 16
65. Soomro, K., Zamir, A.R., Shah, M.: Ucf101: A dataset of 101 human actions classes from videos in the wild. In: arXiv preprint arXiv:1212.0402 (2012) 5, 9, 16
66. Steed, R., Caliskan, A.: Image representations learned with unsupervised pre-training contain human-like biases. In: Proceedings of the ACM Conference on Fairness, Accountability, and Transparency (2021) 1
67. Thoker, F.M., Doughty, H., Bagad, P., Snoek, C.G.M.: How severe is benchmark-sensitivity in video self-supervised learning? In: Proceedings of the European Conference on Computer Vision (ECCV) (2022) 4, 11, 12
68. Tian, Y., Fan, L., Isola, P., Chang, H., Krishnan, D.: Stablerep: Synthetic images from text-to-image models make strong visual representation learners. In: Proceedings of the International Conference on Neural Information Processing Systems (NeurIPS) (2023) 15
69. Tong, Z., Song, Y., Wang, J., Wang, L.: VideoMAE: Masked autoencoders are data-efficient learners for self-supervised video pre-training. In: Proceedings of the International Conference on Neural Information Processing Systems (NeurIPS) (2022) 16
70. Tran, D., Bourdev, L., Fergus, R., Torresani, L., Paluri, M.: Learning spatiotemporal features with 3d convolutional networks. In: Proceedings of the IEEE International Conference on Computer Vision (ICCV) (2015) 16
71. Vincent, L.: Morphological transformations of binary images with arbitrary structuring elements. In: Signal Processing, vol. 22, no. 1, pp. 3–23 (1991) 8
72. Wang, J., Gao, Y., Li, K., Lin, Y., Ma, A.J., Cheng, H., Peng, P., Huang, F., Ji, R., Sun, X.: Removing the background by adding the background: Towards background robust self-supervised video representation learning. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2021) 6
73. Wang, L., Huang, B., Zhao, Z., Tong, Z., He, Y., Wang, Y., Wang, Y., Qiao, Y.: Videomae v2: Scaling video masked autoencoders with dual masking. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2023) 16
74. Wang, L., Xiong, Y., Wang, Z., Qiao, Y., Lin, D., Tang, X., Van Gool, L.: Temporal segment networks: Towards good practices for deep action recognition. In: Proceedings of the European Conference on Computer Vision (ECCV) (2016) 16
75. Wilson, B., Hoffman, J., Morgenstern, J.: Predictive inequity in object detection. In: arXiv preprint arXiv:1902.11097 (2019) 1
76. Xie, S., Sun, C., Huang, J., Tu, Z., Murphy, K.: Rethinking spatiotemporal feature learning: Speed-accuracy trade-offs in video classification. In: Proceedings of the European Conference on Computer Vision (ECCV) (2018) 13
77. Yalniz, I.Z., Jégou, H., Chen, K., Paluri, M., Mahajan, D.: Billion-scale semi-supervised learning for image classification. In: arXiv preprint arXiv:1905.00546 (2019) 1
78. Yamada, R., Kataoka, H., Chiba, N., Domae, Y., Ogata, T.: Point cloud pre-training with natural 3d structures. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2022) 1, 15
79. Zhao, J., Wang, T., Yatskar, M., Ordonez, V., Chang, K.W.: Men also like shopping: Reducing gender bias amplification using corpus-level constraints. In: Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP) (2017) 1
80. Zhao, K., Shen, L., Zhang, Y., Zhou, C., Wang, T., Zhang, R., Ding, S., Jia, W., Shen, W.: Bézierpalm: A free lunch for palmprint recognition. In: Proceedings of the European Conference on Computer Vision (ECCV) (2022) 15
81. Zlatintsi, N., Maragos, P.: Multiscale fractal analysis of musical instrument signals with application to recognition. In: Audio, Speech, and Language Processing, IEEE Transactions on, vol. 21, pp. 737–748 (2013) 16

Hyperparameter	HMDB51	UCF101	DIVING48	EGTEA	VOLLEY	YUP
Weight Decay	$1 \cdot 10^{-1}$	$1 \cdot 10^{-1}$	$4 \cdot 10^{-1}$	$4 \cdot 10^{-1}$	$4 \cdot 10^{-1}$	$4 \cdot 10^{-1}$
LR-Init	$1 \cdot 10^{-8}$	$1 \cdot 10^{-8}$	$2 \cdot 10^{-7}$	$2.5 \cdot 10^{-8}$	$5 \cdot 10^{-8}$	$5 \cdot 10^{-8}$
LR-Peak	$1 \cdot 10^{-5}$	$1 \cdot 10^{-5}$	$2 \cdot 10^{-4}$	$2.5 \cdot 10^{-5}$	$5 \cdot 10^{-5}$	$5 \cdot 10^{-5}$
LR-Final	$1 \cdot 10^{-7}$	$1 \cdot 10^{-7}$	$2 \cdot 10^{-6}$	$2.5 \cdot 10^{-7}$	$5 \cdot 10^{-7}$	$5 \cdot 10^{-7}$

Table 8: Hyperparameters used for fine-tuning the Kinetics checkpoint.

Appendices

A Kinetics Fine-tuning Hyperparameters

Table 8 lists the hyperparameters that were used to fine-tune the Kinetics checkpoint from [50] in the final experiment.

B Supplementary Visual Material

This section features additional visualizations of concepts described in the main document. Specifically:

- Fig. 10 contains examples of synthetic videos produced with Algorithm 1 using standard IFS.
- Fig. 11 contains examples of synthetic videos produced with Algorithm 1 using nonlinear IFS.

Acknowledgements

This work has received funding from the European Union's Horizon Europe research and innovation programme under grant agreement no. 101070381 (project: PILLAR-Robots).

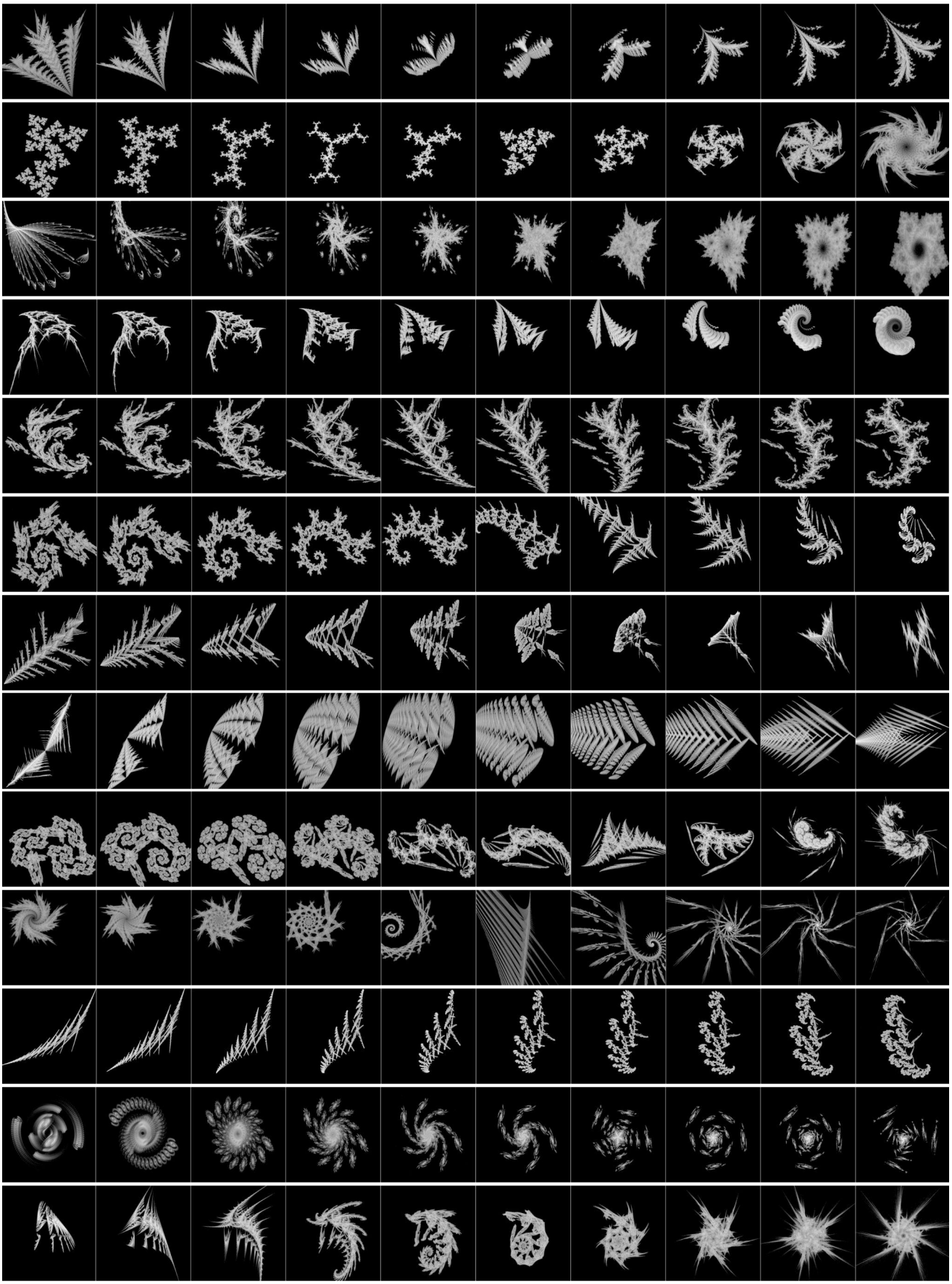


Fig. 10: Examples of videos produced with Algorithm 1 using standard IFS. Each row displays frames from a different video.

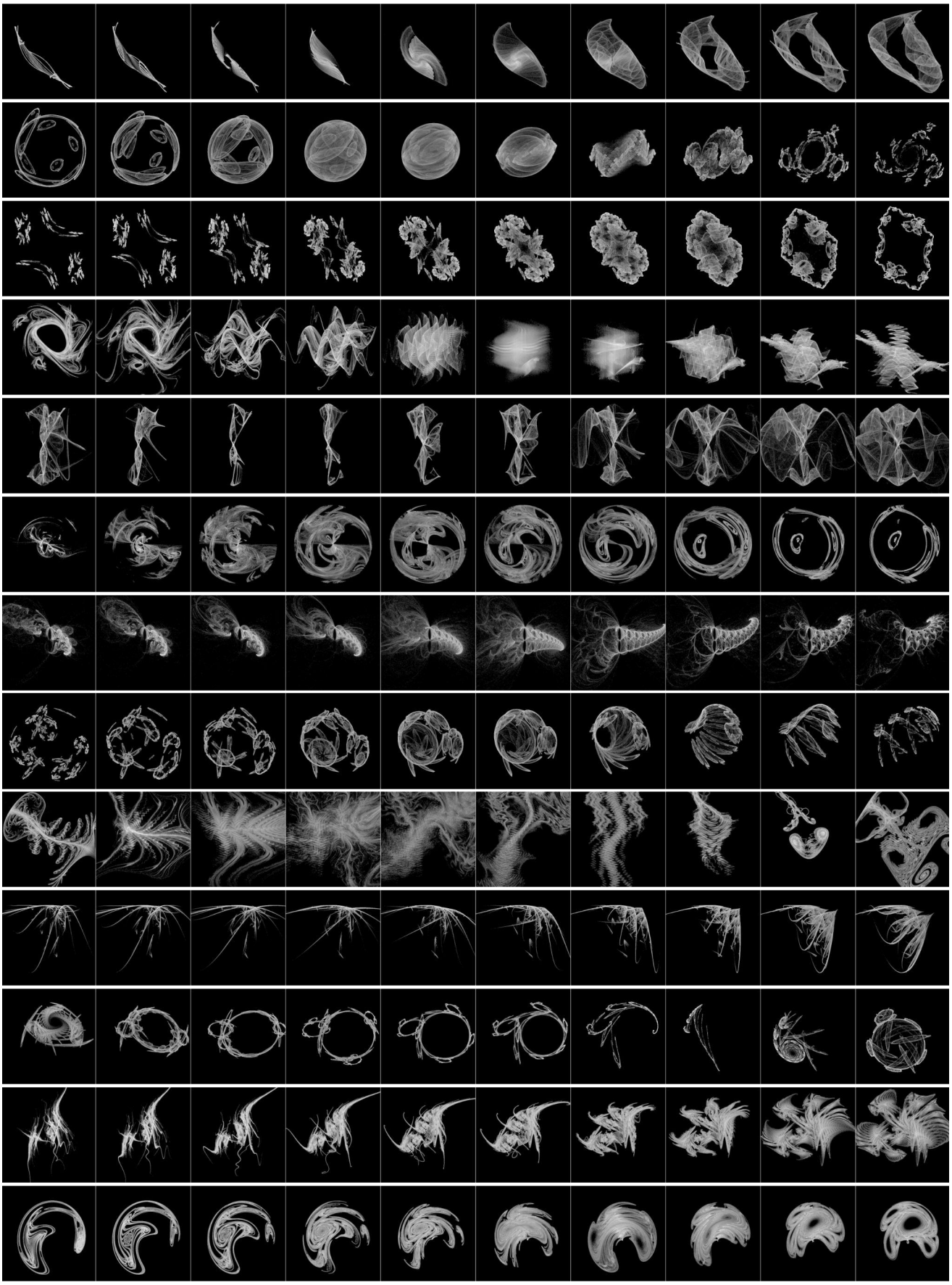


Fig. 11: Examples of videos produced with Algorithm 1 using nonlinear IFS. Each row displays frames from a different video. Note the differences between Fig. 10.