Taylor Videos for Action Recognition

Lei Wang [*1,2] **Xiuyuan Yuan** [*1] **Tom Gedeon** [3] **Liang Zheng** [1]

Abstract

Effectively extracting motions from video is a critical and long-standing problem for action recognition. This problem is very challenging because motions (i) do not have an explicit form, (ii) have various concepts such as displacement, velocity, and acceleration, and (iii) often contain noise caused by unstable pixels. Addressing these challenges, we propose the Taylor video, a new video format that highlights the dominant motions (*e.g.*, a waving hand) in each of its frames named the Taylor frame. Taylor video is named after Taylor series, which approximates a function at a given point using important terms. In the scenario of videos, we define an implicit motion-extraction function which aims to extract motions from video temporal blocks. In these blocks, using the frames, the difference frames, and higher-order difference frames, we perform Taylor expansion to approximate this function at the starting frame. We show the summation of the higher-order terms in the Taylor series gives us dominant motion patterns, where static objects, small and unstable motions are removed. Experimentally, we show that Taylor videos are effective inputs to popular architectures including 2D CNNs, 3D CNNs, and transformers. When used individually, Taylor videos yield competitive action recognition accuracy compared to RGB videos and optical flow. When fused with RGB or optical flow videos, further accuracy improvement is achieved.

1. Introduction

Extracting motions and thus recognizing actions from videos are important problems. Extensive efforts have been made in improving video inputs (Kim et al., 2022; Bilen et al., 2016; Wang, 2017; Bilen et al., 2018; Wang et al., 2019a; Wang & Koniusz, 2024), optimizing networks (Carreira & Zisserman, 2018; Wang et al., 2018; 2023b; Lin et al., 2019), and training skills (Wang et al., 2019b; Mandal et al., 2019; Sabater et al., 2021; Perrett et al., 2021; Wang et al., 2023c). This paper focuses on improving the input, while leaving the rest unchanged.

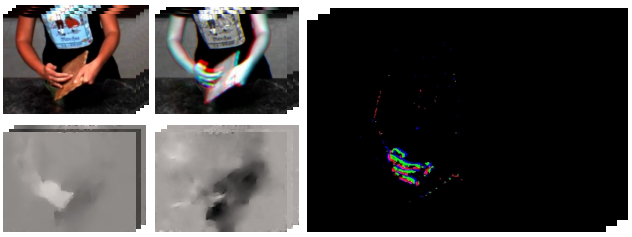


Figure 1. Visualizing different video formats. (Top left): RGB video and time-color reordering frames (Kim et al., 2022). (Bottom left): $\boldsymbol{U}$ and $\boldsymbol{V}$ components of optical flow. (Right): proposed Taylor video frames. Taylor frames clearly (i) remove static objects and unstable motions and (ii) highlight motions.

Motion is an abstract and general concept. By 'abstract', motion does not have an *explicit* form, so we ask whether we can use a function to *implicitly* capture motions in different granularity levels. In comparison, existing methods *explicitly* define motions in different formats, such as optical flow that tracks pixels and objects. By 'general', there are various motion concepts, such as displacement, velocity, and acceleration. While they look challenging to compute, they have clear physical relationships, *e.g.*, velocity is the derivative of displacement with respect to time. In action recognition, this has not been explicitly considered, to the best of our knowledge.

We consider the above points and propose to model motion using an implicit function that outputs motion in some unknown format. Instead of seeking an exact function output which is intractable (motion is very abstract), we propose to approximate this output, especially the dominant motions, using video frames, their differences, and higher-order differences. We find that Taylor series is a great tool for this, allowing us to use similar approximation formulas to compute displacement, velocity, and acceleration.

[*]Equal contribution [1]School of Computing, Australian National University, Canberra, Australia [2]Data61/CSIRO, Canberra, Australia [3]School of Electrical Engineering, Computing and Mathematical Sciences, Curtin University, Perth, Australia. Correspondence to: Lei Wang <lei.w@anu.edu.au>.

Proceedings of the 41st International Conference on Machine Learning, Vienna, Austria. PMLR 235, 2024.

In this paper, we propose a video format named Taylor video, which is composed of Taylor frames, to extract motions for human action recognition, as an alternative to RGB videos and optical flow. Fig. 1 shows a comparison. Taylor videos are directly converted from RGB videos. Each Taylor frame has three channels, representing displacement, velocity, and acceleration of motion, respectively.

To compute a Taylor frame from a temporal block, we first define an implicit motion-extraction function that uses the last frame as input and outputs the encoded motion. We then perform Taylor expansion of this implicit function, and sum up the important terms comprising of derivatives of the motion-extraction function. Here, derivatives are differences and higher-order differences of the frames in the temporal block. By defining the motion-extraction functions to extract displacement, velocity, and acceleration, the difference frames are used as different orders in Taylor series, thus producing the displacement, velocity, and acceleration channels, comprising a Taylor frame. Multiple Taylor frames constitute a Taylor video. Fig. 2 shows an overview of forming a single Taylor frame.

Methodologically, Taylor videos have a few important advantages. First, compared with optical flow, it is much faster to compute due to its in-place matrix operations, and is not subject to computing errors due to its mathematical nature. Second, compared with RGB images, it gets rid of redundant static content, tiny and unimportant motions. Third, Taylor expansion allows for controllable motion capture: if fewer derivative terms are used, we capture dominant motions; if more terms are summed, more motion details are included. Lastly, Taylor videos benefit from the high resolution of RGB images while event cameras do not.

Experimentally, Taylor videos are very competitive inputs compared with RGB and optical flow videos, when applied as the sole input. If combined, further improvements are observed, demonstrating their complementary nature. This is confirmed using various existing action recognition networks, including 2D CNNs, 3D CNNs, and transformers. Further, because Taylor frames contain a lot of zeros, we observe reduction in storage, training and inference cost. The main points of this paper are summarized below.

i. We introduce Taylor videos as an alternative to RGB videos and optical flow to extract motions for action recognition. Its computation comes from a decent application of Taylor series to videos.

ii. Taylor videos are quick to compute from RGB videos, can be of high resolution, and can dynamically capture different levels of dominant motions.

iii. We demonstrate Taylor videos are competitive with and also complementary to RGB videos and optical flow.

2. Related Work

Conventional videos and relevant networks. RGB videos are mostly commonly used, with each pixel having three color channels. They do not encode much temporal dynamics in their individual frames, necessitating architectures for temporal modeling and reasoning. Popular models for this purpose include IDT (Wang et al., 2013), two-stream networks (Simonyan & Zisserman, 2014), 3D spatio-temporal features (Tran et al., 2015), and spatio-temporal ResNet models (Feichtenhofer et al., 2016). Recent advanced models include (Wang et al., 2019c; Wang & Koniusz, 2021; Wang et al., 2021a; Bertasius et al., 2021; Wang & Koniusz, 2022a; Qin et al., 2022; Koniusz et al., 2020; Wang & Koniusz, 2022b; Liu et al., 2022; Radford et al., 2021; Ni et al., 2022; Wu et al., 2023; Wang & Koniusz, 2023; Wang, 2023; Wang et al., 2024b;a).

A few closely related works are (Wang et al., 2018; 2021b). To reduce video redundancy, they compute differential images by subtracting pairs of consecutive frames, providing the network with a more explicit representation of high-frequency temporal changes. But they suffer from spatial shifting, such as camera jitter. In comparison, Taylor frames reduce those jittering motions and retain the dominant ones, and effectively improve action recognition performance.

Optical flow and relevant networks. Optical flow is computed between consecutive video frames and a widely used secondary input for action recognition (Simonyan & Zisserman, 2014; Carreira & Zisserman, 2018; Wang & Koniusz, 2024). It contains both the direction and magnitude of motion at each pixel. Common methods for computing optical flow in action recognition include (Zach et al., 2007; Brox & Malik, 2011; Weinzaepfel et al., 2013; Revaud et al., 2015). Because it is computationally expensive, recent works (Zhang et al., 2016; Wu et al., 2018) use motion vectors from compressed videos, *e.g.*, H.264, to avoid high computational cost. In action recognition (Simonyan & Zisserman, 2014; Carreira & Zisserman, 2018), optical flow is usually implemented in a separate stream, commonly referred to as the temporal stream, in addition to the appearance stream (*a.k.a.* RGB stream).

The Taylor video is very different from optical flow. While optical flow focuses more on pixel displacement[1], Taylor frames records more motion concepts including displacement, velocity and acceleration. Taylor frames can even account for higher order motion concepts such as jerk (derivative of acceleration), snap, crackle, *etc*. Taylor videos are also much faster to compute.

Extracting temporal dynamics in a single frame. Bilen et al. (2016) propose dynamic images that record spatial and temporal information in a single frame, allowing image

[1]Its computation method is very different from ours.

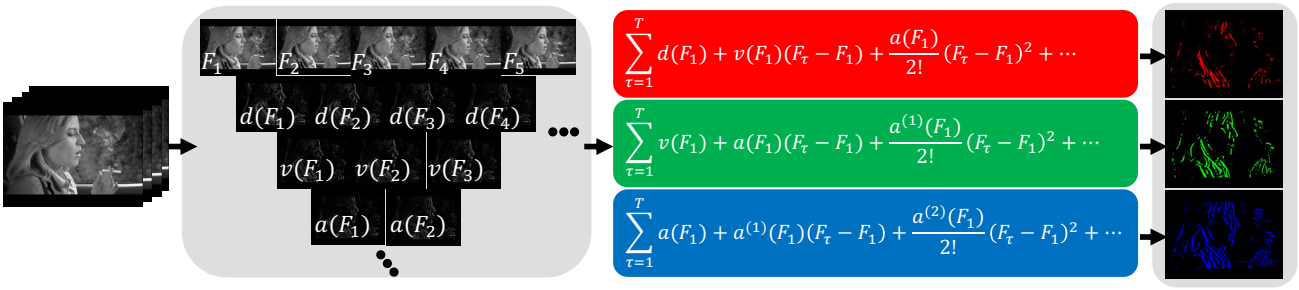


Figure 2. Computing a single Taylor frame from a grayscale video temporal block $\mathbf{F} = [\boldsymbol{F}_1, \boldsymbol{F}_2, \cdots, \boldsymbol{F}_\tau, \cdots, \boldsymbol{F}_T], \tau = 1, 2, \cdots, T$. We calculate the difference map between each two consecutive frames: $d(\boldsymbol{F}_i) = \boldsymbol{F}_{i+1} - \boldsymbol{F}_i, i = 1, 2, \cdots, T$. We then calculate the higher-order differences, *e.g.*, velocity maps using $v(\boldsymbol{F}_i) = d(\boldsymbol{F}_{i+1}) - d(\boldsymbol{F}_i)$, acceleration maps using $a(\boldsymbol{F}_i) = v(\boldsymbol{F}_{i+1}) - v(\boldsymbol{F}_i)$, jerk maps, *etc.*, in the temporal block. We compute three channels of a Taylor frame by Eq. (4), (5), and (6), visualized in red, green, and blue, respectively.

classification networks to be used for action classification. Recently, Kim et al. (2022) introduced a channel sampling method that puts together R (G, or B) channels of consecutive frames into a single frame, allowing 2D CNNs to better capture motion. On the down side, these alternative video representations still contain lots of redundancy inherited from RGB videos. Moreover, they lack flexibility to extract various orders of motions. In comparison, the Taylor video is backed by well-founded mathematical concepts, reduces redundancy significantly, and is very flexible.

3. Preliminaries

Notations. Scalars are written in regular fonts; vectors are denoted by lowercase boldface letters, *e.g.*, $\boldsymbol{x}$; matrices by uppercase boldface, *e.g.*, $\boldsymbol{X}$; tensors by calligraphic letters, *e.g.*, $\mathbf{X}$. Let $\mathbf{X} \in \mathbb{R}^{d_1 \times d_2 \times d_3}$ denote a third-order tensor. Using the Matlab convention, we refer to its k-th slice as $\mathbf{X}_{:,:,k}$, which is a $d_1 \times d_2$ matrix.

Taylor series revisit. Taylor series locally approximates non-linear functions. It is an infinite sum of terms expressed in terms of the function's derivatives at a single point:

$$f(x) = \sum_{k=0}^{\infty} \frac{f^{(k)}(a)}{k!}(x-a)^k, \tag{1}$$

where $k!$ denotes the factorial of k. $f^{(k)}(a)$ denotes the kth derivative of f evaluated at point a. The 0th-order derivative of f is defined as f itself, and $(x-a)^0$ and $0!$ are both equal to 1. Theoretically, the first few terms of the series can reconstruct most of $f(x)$. This forms the foundation of our use: the first few terms encode dominant motions.

4. Proposed Approach

4.1. Taylor videos: General formulations

Taylor series to compute motion within a temporal block. For an RGB video, we first convert each frame to grayscale. We are given a $\mathcal{T}$-frame grayscale video $\mathbf{F} = [\boldsymbol{F}_1, \boldsymbol{F}_2, \cdots, \boldsymbol{F}_\mathcal{T}] \in \mathbb{R}^{H \times W \times \mathcal{T}}$, where H and W denote respectively the height and width. We use a temporal sliding window sized T with step size 1 to produce individual subsequences (*a.k.a.* video temporal blocks), resulting in N temporal blocks denoted as $\{\mathbf{F}^1, \mathbf{F}^2, \cdots, \mathbf{F}^N\}$, where $\mathbf{F}^i = [\boldsymbol{F}_1^i, \boldsymbol{F}_2^i, \cdots, \boldsymbol{F}_T^i] \in \mathbb{R}^{H \times W \times T}$. For simplicity we drop superscript i in what follows unless otherwise stated.

We define a motion extraction function $f : \boldsymbol{F} \in \mathbb{R}^{H \times W} \mapsto \boldsymbol{M} \in \mathbb{R}^{H \times W}$, where the function input argument $\boldsymbol{F}$ is a gray-scale frame, and the output $\boldsymbol{M}$, presenting pixel-level motion, is the motion map encoded in the temporal block ending at $\boldsymbol{F}$. Given a temporal block, we aim to reveal its motion dynamics. To achieve this, we propose to use Taylor series with Eq. (1):

$$f(\boldsymbol{F}_T) = \sum_{k=0}^{\infty} \frac{f^{(k)}(\boldsymbol{F}_1)}{k!} \odot (\boldsymbol{F}_T - \boldsymbol{F}_1)^{\circ k}, \tag{2}$$

where $\odot$ and $^{\circ k}$ denote the Hadamard (element-wise) product and Hadamard (element-wise) power, respectively. $f^{(k)}(\boldsymbol{F}_1)$ denotes the kth derivative of f evaluated at $\boldsymbol{F}_1$.

Combining short-term and long-term motions in a temporal block. Eq. (2) computes motions covering the entire temporal block with length T, seen through the term $\boldsymbol{F}_T - \boldsymbol{F}_1$. In order to consider shorter-term motions in a temporal block, we also compute motions between frame $\boldsymbol{F}_\tau$ ($\tau < T$) and frame $\boldsymbol{F}_1$ using Eq. (2), and then average them. This process can be described as:

$$\boldsymbol{M}_f = \frac{1}{T} \sum_{\tau=1}^{T} f(\boldsymbol{F}_\tau). \tag{3}$$

where $f(\boldsymbol{F}_\tau) = \sum_{k=0}^{\infty} \frac{f^{(k)}(\boldsymbol{F}_1)}{k!} \odot (\boldsymbol{F}_\tau - \boldsymbol{F}_1)^{\circ k}$, and $\tau = 1, 2, \cdots, T$. Here, $\boldsymbol{M}_f$ represents the motions encoded in the temporal block ending with $\boldsymbol{F}_\tau$. Here subscript f is used to denote extracting a certain motion concept, which can be one of displacement, velocity, and acceleration.

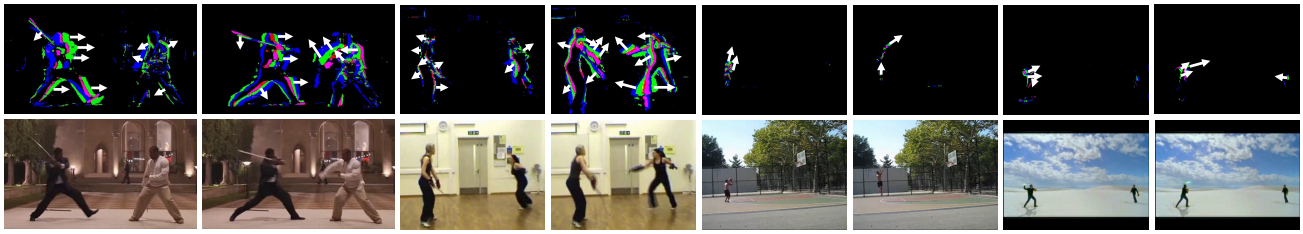


Figure 3. Taylor frames indicate motion strengths and directions. (Top): Taylor frames. (Bottom): original RGB frames. All videos are from HMDB-51. Red, green, and blue represent displacement, velocity, and acceleration, respectively. Bolder colors indicate greater strength. We depict motion directions with white arrows: if green (velocity) is to the right of red (displacement), the object is moving rightwards *in the next frame*; if blue (acceleration) is to the left of red (displacement), the object is moving leftwards *in the frame after*.

4.2. Computing three motion concepts in Taylor videos

By defining different motion extraction functions, we use Eq (3) to compute three critical motion concepts: displacement, velocity, and acceleration within a temporal block.

Displacement refers to the change in position of a pixel, and it has both magnitude and direction. To approximate displacement, we define displacement-extraction function f_d, which takes a frame as input and outputs displacement within a temporal block. After aggregating the short-term and long-term motions using Eq. (3), the displacement motion map can be approximated as:

$$\boldsymbol{M}_d = \frac{1}{T}\sum_{\tau=1}^{T}\sum_{k=0}^{\infty}\frac{f_d^{(k)}(\boldsymbol{F}_1)}{k!}\odot(\boldsymbol{F}_\tau - \boldsymbol{F}_1)^{\circ k}. \tag{4}$$

Here $f_d^{(0)} = d(\boldsymbol{F}_1) = \boldsymbol{F}_2 - \boldsymbol{F}_1$, $f_d^{(1)} = v(\boldsymbol{F}_1) = d(\boldsymbol{F}_2) - d(\boldsymbol{F}_1)$, and $f_d^{(2)} = a(\boldsymbol{F}_1) = v(\boldsymbol{F}_2) - v(\boldsymbol{F}_1)$ (assume the time step is 1). We take $v(\boldsymbol{F}_1)$ as an example to illustrate the intuition behind these equations. Because f_d characterizes the displacement of the temporal block, its first-order derivative, or $\frac{d_{f_d}}{d_{\boldsymbol{F}}}$ means the displacement change between two consecutive frames, which should be intuitively computed as the second-order difference between two gray-scale frames.

Velocity describes the rate of change of displacement, and includes both speed and direction. Similar to the computation of displacement, we form velocity motion map as:

$$\boldsymbol{M}_v = \frac{1}{T}\sum_{\tau=1}^{T}\sum_{k=0}^{\infty}\frac{f_v^{(k)}(\boldsymbol{F}_1)}{k!}\odot(\boldsymbol{F}_\tau - \boldsymbol{F}_1)^{\circ k}. \tag{5}$$

Similarly, $f_v^{(0)} = v(\boldsymbol{F}_1) = d(\boldsymbol{F}_2) - d(\boldsymbol{F}_1)$, $f_v^{(1)} = a(\boldsymbol{F}_1) = v(\boldsymbol{F}_2) - v(\boldsymbol{F}_1)$, and $f_v^{(2)} = j(\boldsymbol{F}_1) = a(\boldsymbol{F}_2) - a(\boldsymbol{F}_1)$. Jerk $j(\boldsymbol{F}_1)$ is the rate of change of acceleration at frame $\boldsymbol{F}_1$.

Acceleration describes the rate of change of velocity. Similar to displacement and velocity, the acceleration map of a given temporal block can be approximated as:

$$\boldsymbol{M}_a = \frac{1}{T}\sum_{\tau=1}^{T}\sum_{k=0}^{\infty}\frac{f_a^{(k)}(\boldsymbol{F}_1)}{k!}\odot(\boldsymbol{F}_\tau - \boldsymbol{F}_1)^{\circ k}. \tag{6}$$

Here $f_a^{(0)} = a(\boldsymbol{F}_1) = v(\boldsymbol{F}_2) - v(\boldsymbol{F}_1)$, and $f_a^{(1)} = j(\boldsymbol{F}_1) = a(\boldsymbol{F}_2) - a(\boldsymbol{F}_1)$. Given a temporal block, we compute three motion maps $\boldsymbol{M}_d$, $\boldsymbol{M}_v$, and $\boldsymbol{M}_a$. These maps are stacked into an image, forming a **Taylor frame**: $\boldsymbol{M} \in \mathbb{R}^{H\times W\times 3}$. Given an RGB video that has N temperal blocks: $\{\mathbf{F}^1, \mathbf{F}^2, \cdots, \mathbf{F}^N\}$, we compute N Taylor frames $\boldsymbol{M}^1, \boldsymbol{M}^2, \cdots, \boldsymbol{M}^N$. These Taylor frames form the **Taylor video**: $\mathbf{M} = [\boldsymbol{M}^1, \boldsymbol{M}^2, \cdots, \boldsymbol{M}^N] \in \mathbb{R}^{H\times W\times 3\times N}$. Fig. 2 illustrates the computation of a single Taylor frame. In Fig. 3, we show that Taylor frames allow us to visually perceive the strength and direction of motion.

4.3. Efficient computation of Taylor videos

Given a temporal block $\mathbf{F}\in\mathbb{R}^{H\times W\times T}$ in the form of a tensor, we create an inflated tensor (mimicking a static temporal block) $\widetilde{\mathbf{F}} \in \mathbb{R}^{H\times W\times T}$ by duplicating the first frame of the temporal block T times. Instead of using f that uses a frame as input, we use function t that takes a video temporal block, or a tensor as input, and outputs a single three-channel Taylor frame that summarises the motions. The proposed tensor representation is given below:

$$t(\mathbf{F}) = \sum_{k=0}^{\infty}\frac{t^{(k)}(\widetilde{\mathbf{F}})}{k!}\odot\frac{1}{T}\sum_{\tau=1}^{T}(\mathbf{F} - \widetilde{\mathbf{F}})^{\circ k}_{:,:,\tau}. \tag{7}$$

Eq. (7) is equivalent to Eq. (3) giving the same Taylor frame output, and is more efficient to compute because of the use of tensors. A proof of the equivalence is presented in Section A of Appendix. Algorithm 1 in Section B of Appendix shows the efficient implementation of Taylor video.

4.4. Discussion

Can we sum up an infinite number of terms in Eq. (2)? No. An infinite number of terms require an infinite number of frames in a temporal block, which is infeasible.

Can we compute more motion concepts in Taylor video? Technically yes, by extending Eq. (3) into jerk, *etc*. However, doing this has two down sides. First, computing higher-order motions requires heavier computation, because we

Figure 4. Taylor videos remove redundancy, such as static backgrounds, unstable pixels, watermarks, and captions. This, together with its ability to highlight motion including strengths and directions, is beneficial for action recognition. Videos are from HMDB-51.

need to compute more frame differencing operations. Second, if we include more channels (>3) for Taylor videos, we will have to modify model architecture, and it is non-trivial when using pre-trained models. In fact, Taylor videos with three channels can easily replace RGB video inputs and reuse existing pre-trained models. Taylor videos have superior performance on pre-trained models compared with being trained from scratch.

How to choose the number of frames in a temporal block and the number of terms in Taylor series? Because of the discrete nature of frames, and the fact that higher-order image differences are fewer than low-order ones, we need at least 4 frames per temporal block to compute a 3-channel Taylor frame. Any temporal block with fewer than 4 terms will result in insufficient data for Taylor frame computation. If a temporal block has $T > 4$ frames, we recommend to use $T-3$ terms to fully exploit data in the block. If we use fewer than $T-3$ terms, some frames will not be used; but it is our future work to explore how to sample frames so that we can use few terms to reduce computation cost. Empirical analysis of the impact of T is provided in Section 5.3.

Connection with the event camera. Event cameras continuously capture differences in pixel brightness. From a visualization standpoint, the displacement channel M_d of the Taylor video appears similar to event camera data. However, utilizing event camera data for action recognition necessitates distinct temporal preprocessing due to its high temporal resolution in capturing fine-grained temporal dynamics. This characteristic differs from traditional frame-based cameras, which may overlook rapid changes. Taylor videos are computed from conventional videos, the most popular video format, focusing on extracting dominant motions using frame differencing as the measure of time concept, *e.g.*, $\boldsymbol{F}_\tau - \boldsymbol{F}_1$, $\boldsymbol{F}_{\tau-1} - \boldsymbol{F}_1$, *etc.*, instead of using discrete time steps, *e.g.*, $\tau-1$, $\tau-2$, *etc.*

Taylor skeleton sequences. Note that Taylor videos can be naturally applied to human skeleton sequences. Similar to the RGB modality, we extract dominant motions for each human body joint within the video temporal block. By applying Taylor series to skeleton sequences, we obtain Taylor skeleton sequences.

Limitation. Taylor video obviously does not encode much static texture pattern. It means Taylor videos at their current form are not an ideal option for tasks like video captioning and anomaly detection. A potential solution is to stack some RGB / grayscale frames to Taylor frames so that static objects and backgrounds can be considered for more general video understanding, such as large video language models. However, the Taylor video effectively removes backgrounds, watermarks, captions, *etc.*, while retaining significant motions (refer to samples from HMDB-51 in Fig. 4).

Other potential applications. Because Taylor videos capture dominant motions, it may be used in tasks like video-based face anti-spoofing. Taylor videos also present motion strength and direction[2] (see Fig. 3), they can potentially be used for player performance analysis. Moreover, Taylor videos are valuable in video analysis for predicting the next frames or future motion trajectories, aiding tasks such as object tracking and scene understanding, as they encode both short-term and long-term motions per Taylor frame.

5. Experiments

5.1. Datasets and implementation details

Datasets. First, we use three small-scale datasets: HMDB-51 (Kuehne et al., 2011), MPII Cooking Activities (Rohrbach et al., 2012), and CATER (Girdhar & Ramanan, 2020). HMDB-51 has 51 human action categories and 6766 videos. It is challenging due to its significant camera and background motion. The MPII dataset has 64 distinct activities from 3748 clips include coarse actions such as *opening refrigerator*, and fine-grained actions such as *peel*, *slice*, and *cut apart*. While its cameras are fixed, the human actions usually occupy very small areas with relatively subtle movements. CATER is a synthetic action

[2]Each channel of the Taylor frame represents a motion concept with positive and negative values indicating motion directions (0 for static pixels). Velocity and acceleration channels are computed per video temporal block, capturing relative motion directions from the initial frame.

	Model	Pretrain	Input	HMDB-51	CATER		MPII
					static	moving	
2D CNNs	TSN	ImageNet	RGB	54.9	49.6	51.6	38.4
			Taylor	56.4	73.8	62.7	42.2
	TSM	ImageNet	RGB	-	79.9	65.8	46.7
			GrayST	-	82.2	74.7	48.7
			Taylor	-	83.1	75.5	50.1
3D CNNs	I3D	ImageNet	RGB	49.8	73.5	57.7	42.8
			Taylor	65.2	74.7	60.5	43.0
		Kinetics	RGB	74.3	75.4	61.9	48.7
			OPT	77.3	78.5	66.3	51.0
			Taylor	78.1	80.2	69.8	52.3
	R(2+1)D	Sports1M	RGB	66.6	-	-	-
			Taylor	67.4	-	-	-
Transf.	TimeSformer	Kinetics	RGB	71.7	69.9	57.6	41.0
			Taylor	72.1	71.2	58.2	42.8
	Swin Transformer	Kinetics	RGB	72.9	72.2	63.5	46.6
			Taylor	73.5	73.0	64.7	47.0

Table 1. Comparing the Taylor video with other input modalities on three datasets with various action recognition models and pre-training datesets.

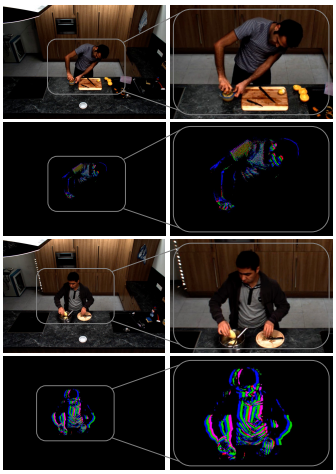


Figure 5. Taylor frame captures subtle motions on MPII. (*Top 4 images*) show *squeeze* and (*Bottom 4 images*) show *put in pan/pot*. In each set, the left motion region is zoomed in on the right to enhance visualization. Better view in color.

	TSN	TSM	C3D	I3D	R(2+1)D
RGB	49.6	40.3	24.3	43.9	26.0
Taylor	49.8	41.3	25.2	44.4	26.9

Table 2. Comparing RGB videos and Taylor videos under the training-from-scratch setup. Various action recognition models are used on HMDB-51. Top-1 accuracy (%) is reported.

recognition dataset involving long-term temporal reasoning. This provides the biggest challenge for action recognition methods that only focus on short-term clips. It has two versions of the videos, with and without camera motion, and we experiment with both of them. Because of the diverse aspects focused by the datasets, we can evaluate Taylor videos w.r.t. its effectiveness for static and moving cameras, complex backgrounds, subtle human actions, and non-human movements. We then evaluate Taylor videos on large-scale Kinetics (K400 / K600) (Kay et al., 2017) and Something-Something v2 (SSv2) (Mahdisoltani et al., 2018). We also evaluate the effectiveness of Taylor skeleton sequences using NTU-60 (Shahroudy et al., 2016), NTU-120 (Liu et al., 2019), and Kinetics-Skeleton (K-Skel) (Yan et al., 2018).

Evaluation protocols. For HMDB-51, we use the standard 3 train/test splits and report the mean accuracy across 3 splits. For MPII, we use the mean Average Precision (mAP) over 7-fold cross validation. For CATER, we also report the mAP on both static and moving camera setups. For large-scale video and skeleton datasets, we adhere to standard evaluation protocols and report Top-1 accuracy.

Implementation details. Using Taylor videos as input, we either train action recognition models from scratch or fine-tune them on top of models pretrained with RGB image or videos. The model architectures cover popular 2D and 3D CNNs, such as TSM (Lin et al., 2019), TSN (Wang et al., 2018), TSN (Wang et al., 2018), SlowFast (Feichtenhofer et al., 2019), R(2+1)D (Tran et al., 2018), C3D (Tran et al., 2015), and I3D (Carreira & Zisserman, 2018), as well as recent transformer-based models such as Swin Transformer (Liu et al., 2021) and TimeSformer (Bertasius et al., 2021). We use pre-computed optical flow provided by the dataset websites. When implementing Taylor videos for transformer architectures, we add the grayscale frame to each of the displacement, velocity, and acceleration maps. Without this strategy, some patch embeddings of transformers would only be computed on patches that are almost all zeros, compromising transformer performance[3]. We follow the default settings from the original papers, where we replicate their performance using RGB videos and/or optical flow as input. Hyperparameters such as the number of epochs for training/fine-tuning are determined on the validation sets.

We present train-from-scratch results using either RGB or Taylor videos under the TSM, I3D, TimeSformer (L), Swin Transformer, and VideoMAE v2 (Wang et al., 2023a) backbones on large-scale video datasets. We also present experiments using ST-GCN (Yan et al., 2018), InfoGCN (Chi et al., 2022), AGE-Ens (Qin et al., 2022), and 3Mformer (Wang

[3]In fact, existing transformer architectures are predominately designed for RGB videos. Adapting them to Taylor videos is an interesting future direction.

Model	Input	K400	K600	SSv2
TSM	RGB	76.3	-	63.4
	Taylor	77.6	-	65.1
I3D	RGB	77.7	-	-
	Taylor	79.3	-	-
TimeSformer	RGB	80.7	82.2	62.5
	Taylor	81.5	83.1	63.7
VideoMAE	RGB	79.8	-	69.3
	Taylor	80.4	-	70.0
Swin Transformer	RGB	-	-	69.6
	Taylor	-	-	71.1

Table 3. Evaluations of Taylor videos on large-scale Kinetics (K400 / K600) and Something-Something v2 (SSv2).

Model	Input	NTU-60		NTU-120		K-Skel
		X-Sub	X-View	X-Sub	X-Set	Top-1
ST-GCN	Skeleton	81.5	88.3	70.7	73.2	30.7
	Taylor	85.4	93.0	78.5	80.1	35.1
InfoGCN	Skeleton	93.0	97.1	89.8	91.2	-
	Taylor	94.6	98.5	91.6	93.7	-
AGE-Ens	Skeleton	91.0	96.1	87.6	88.8	-
	Taylor	95.0	98.3	91.8	92.5	-
3Mformer	Skeleton	94.8	98.7	92.0	93.8	48.3
	Taylor	95.3	98.8	92.6	94.7	49.2

Table 4. Comparing Taylor-transformed skeletons with original skeletons on NTU-60, NTU-120 and Kinetics-Skeleton (K-Skel).

& Koniusz, 2023) backbones on both original and Taylor skeleton sequences. We simply use the displacement concept with 1 term (4 frames per temporal block with a step size of 1) to compute the Taylor skeleton sequences.

5.2. Main evaluation

Comparing Taylor videos with RGB videos and optical flow. In this experiment, we use only one modality as network input and compare the results. Various networks are used, including 2D CNNs, 3D CNNs, and transformers. We always use models pre-trained on various datasets before fine-tuning with the compared inputs. Results are summarized in Table 1. We have two main observations.

First, on all the three datasets, using Taylor videos as input is very competitive compared with RGB videos, GrayST (Kim et al., 2022), and optical flow. For example, under the I3D model with Kinetics pre-training, we achieve top-1 accuracy of 78.1%, 80.2%, 69.8%, and 52.3% on HMDB-51, CATER-static, CATER-moving, and MPII, respectively. Compared with RGB videos, the improvements are +3.8%, +4.8%, +7.9%, and +3.6% on the four test setups, respectively; compared with optical flow, the improvements are +0.8%, +1.7%, +3.5%, and +1.3%, respectively.

	RGB	*Diff.*	*Displ.*	*Veloc.*	*Accel.*	Taylor
Acc(%)	59.1	53.9	61.9	62.1	59.9	65.1

Table 5. Evaluations on different combinations of motion concepts on Something-Something v2 with TSM backbone.

Second, the competitiveness of Taylor videos can be observed under various models. For example, on the CATER-moving dataset, if we compare Taylor videos with RGB videos, the improvements are +11.1%, +9.7%, +2.8%, +0.6%, +1.2% under TSN, TSM, I3D (ImageNet pretrained), TimeSformer, Swin-T models, respectively. This demonstrates the model-generic use of Taylor videos.

These results indicate that Taylor video can deal with complex backgrounds with moving/static cameras (HMDB-51), long-term temporal reasoning (CATER), and fine-grained motions that only take up a small area (MPII, see Fig. 5).

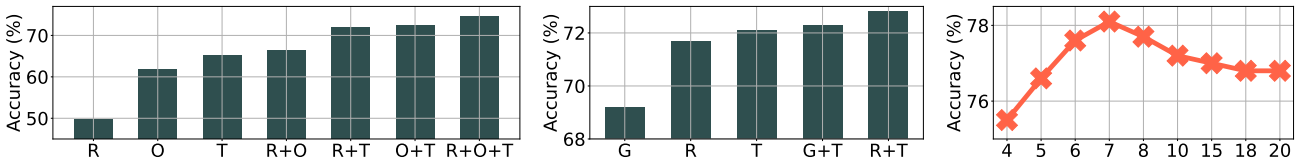
Taylor video complements RGB videos and optical flow. We now evaluate different combinations of these input modalities, including RGB, RGB + Taylor, RGB + optical flow, and RGB + optical flow + Taylor. Results are shown in Fig. 6(a) and 6(b). We have the following observations.

First, under the I3D model, combining Taylor with RGB or optical flow yields higher accuracy compared with using them individually. For example, using RGB+Taylor produces +22.1% and +6.7% improvement over using only RGB or Taylor, respectively. If we combine all the three modalities, further improvement is observed.

Second, under TimeSformer, we show performance of Taylor videos with either RGB (R) or gray-scale (G) frames, and such combinations are again superior to individual modalities. On the HMDB-51 dataset, improvement of RGB+Taylor is +1.1% and +0.7% over RGB or Taylor alone, respectively. These experiments demonstrate that the Taylor video is a different but complementary type of input.

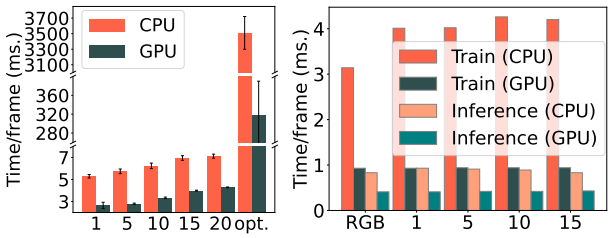
Performance of Taylor videos and Taylor skeleton sequences under training from scratch. We first evaluate Taylor video by training networks from scratch on HMDB-51. Results using various 2D/3D models are shown in Table 2. First, compared with fine-tuning with Taylor videos (Table 1), training from scratch with Taylor videos yields much lower accuracy. Second, under the training-from-scratch setup, Taylor video has similar accuracy compared with RGB videos. These results highlight the importance of pre-training for Taylor videos to give superior performance.

We then present train-from-scratch results on large-scale video datasets and skeleton sequences in Table 3 and 4, respectively. As shown in Table 3, models trained with Tay-



(a) I3D (b) TimeSformer (c) Impact of T

Figure 6. Comparison of different input modalities and their combinations using the (a) I3D and (b) TimeSformer models on HMDB-51. 'R', 'O' 'G', and 'T' denote RGB, optical flow, gray-scale, and Taylor videos, respectively. (c) Impact of the length T of video temporal blocks for computing Taylor videos on the HMDB-51 dataset. Top-1 accuracy is reported.



(a) Computing a frame (b) Training/Inference

Figure 7. Time cost (milliseconds, ms.) of (a) computing a single Taylor frame (with 1, 5, 10, 15, 20 terms) and optical flow (opt.) and (b) training and testing an RGB frame and Taylor frame (with 1, 5, 10, 15 terms). We use videos from HMDB-51 dataset.

lor videos consistently outperform those trained with RGB videos on all three datasets. We also observe that Taylor-transformed skeleton sequences consistently outperform the original skeleton sequences across various backbones on three large-scale benchmarks (Table 4).

5.3. Further analysis

Computational cost of producing a Taylor frame. In this and following two experiments, we use 1 NVIDIA Tesla V100 GPU (with 12 CPUs). We use time per frame[4] as the evaluation metric. In Fig. 7(a), we compare the average time cost of computing a Taylor frame and that of computing an optical flow frame on HMDB-51, where the former cost depends on the length of the temporal block. We demonstrate that the calculation of Taylor frames requires significantly less time than TVL1 optical flow (Zach et al., 2007).

Computational cost of training with Taylor videos. We assume Taylor videos, RGB videos, and optical flow are all pre-computed. We fine-tuning an I3D model pretrained on Kinetics. From Fig. 7(b), we observe the training time for a single Taylor frame is very similar to a single RGB frame, under GPU. On the other hand, Training with taylor videos usually has a cold start: loss drops much more slowly in the beginning (around 20 epochs) but catches up later. A possible reason is that existing architectures are not efficient in reading motion captured by the new Taylor video modality. This cold start affects training speed on small datasets: Taylor videos take around 20 more epochs to converge than RGB videos. But on large datasets such as Kinetics-600, the cold start does not matter due to the overall very long training time: Taylor videos have similar convergence speed with RGB videos, both using about 1200 epochs.

Computational cost of inference with Taylor videos. We also assume pre-computed inputs. We find that Taylor videos take similar time for network processing compared with RGB videos. We also note that the efficiency of the Taylor video can be further improved through developing a particular neural network layer dedicated for its computation. This is particularly feasible because the computing of Taylor videos is all composed of in-place matrix operations, which can be easily parallelized.

Evaluations on frame differencing and different combinations of motion concepts. The comparisons using (i) frame differencing maps and (ii) different combinations of motion concepts on Something-Something v2 with TSM are shown in Table 5. We clearly see that frame differencing (*Diff.*) is even worse the RGB video. This is because the difference frames contain lots of noise that negatively affects the network. We observe that incorporating all three motion concepts (Taylor) achieves the best performance compared to using only displacement (*Displ.*), velocity (*Veloc.*), or acceleration (*Accel.*) individually.

How many RGB frames are required to compute Taylor video and whether comparing with RGB is fair. The video trunks used to compute Taylor frames have significant overlaps to ensure motion continuity between Taylor frames. For example, to compute a 16-frame Taylor video, we need 19 and 20 RGB frames if #terms = 1 and 2, respectively. Note that 2 terms is a good hyperparameter for action recognition. So we are not using much more RGB frames.

To compute a 16-frame Taylor video, a few more RGB frames, *e.g.*, 19 frames, are needed. This is similar to optical flow: to compute a 16-frame optical flow, 17 RGB frames

[4]The average time taken to process a single frame: Time per Frame = Total Processing Time / Number of Frames

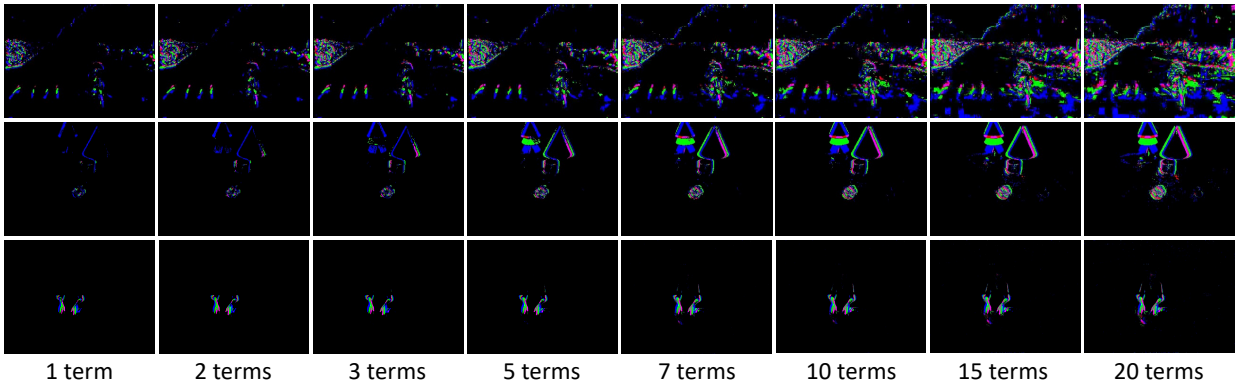


Figure 8. Qualitative impact of the number of terms used in Taylor series on the action *ride bike* in HMDB-51 (top row), a *synthetic action* video of CATER (middle row), and the fine-grained action *place* in MPII Cooking Activity (bottom row). We observe that as the number of terms increases, more motion patterns are captured. In scenarios with a moving camera, such as *ride bike* in the top row, using more terms leads to the inclusion of more background details, such as crosswalk and crowds. For quantitative analysis see Fig. 6(c).

	Dataset-level	Action-level
HMDB-51	0.87	0.68 – 1.23
MPII Cooking	3.04	0.93 – 10.07

Table 6. Compression ratio of Taylor videos on HMDB-51 and MPII on the dataset level and action class level. The number of Taylor terms is 5. A ratio greater than 1 means reduced video sizes, while ratio lower than 1 means increased video sizes.

are needed in the optimal case. In fact, to encode motion, it is necessary to use more RGB frames. We compare models trained with 32-frame Taylor videos and RGB videos of 32 and 64 frames, on HMDB-51 under I3D backbone. The top-1 accuracies are: 65.2%, 53.4%, and 49.8%, respectively, demonstrating the superiority of our method.

Impact of the length of temporal blocks. For the Taylor video defined in this work, the number of frames T in a temporal block is the only hyperparameter, which also determines the degree of Taylor series to be $T-3$. As discussed before, a degree lower than $T-3$ will result in under-exploitation of the frames. The evaluation of the impact of T on the HMDB-51 dataset is shown in Fig. 6(c).

Interestingly, we observe that the effectiveness of Taylor videos remains stable as the number of frames increases. This suggests a trade-off between capturing long-term motion and introducing noise (see Fig. 8). To be specific, when using fewer frames, *e.g.*, 7 frames and 4 Taylor terms, only the highly dominant motions are encoded, meaning less noisy motions; but to the down side, longer-term motions are not captured. In comparison, if we use 15 or more frames, while long-term motions are computed, using more Taylor terms introduce undesirable noisy motions.

Does Taylor videos compress datasets or action classes? We calculate the compression ratio of RGB videos to Taylor videos (using $\frac{\text{Uncompressed size}}{\text{Compressed size}}$) in different action classes ('action-level') and datasets ('dataset-level') on both HMDB-51 and MPII. Results are shown in Fig. 9 and Fig. 10 of Appendix and Table 6. Interestingly, we find compression ratio relatively high for static cameras and 'big' motions such as *hug* and *take & put in oven*, where there exist high background redundancy and dominant motions. In comparison, compression ratio is low for very slow actions, facial movements such as *laugh*, *kiss*, *smile* and *smoke*, and moving cameras. Consequently, converting RGB videos into Taylor videos achieves the highest compression ratio on MPII (static backgrounds and small motion areas). On the contrary, the dataset size even increases for HMDB-51, because of its moving cameras and complex backgrounds. Despite different compression effects on different datasets, our method has consistent accuracy improvement.

6. Conclusion

In this paper, we introduce Taylor videos, a new video format for action recognition. Computed through the use of Taylor series for videos, each Taylor video frame captures diverse motion concepts through motion dynamics distillation, effectively eliminating redundancy such as static backgrounds, watermarks, and text captions, while preserving dominant motions for downstream tasks. Taylor videos can be seamlessly integrated into various existing pre-trained RGB-based models, including 2D CNNs, 3D CNNs, and transformer-based architectures, for fine-tuning. We show that the Taylor video yields very competitive accuracy compared with the conventional RGB videos, and optical flow, and once combined, produces even better performance. Additionally, we demonstrate that Taylor videos on large-scale datasets and Taylor-transformed skeleton sequences outperform the use of original RGB and skeletons, respectively.

Potential Broader Impact

From a societal perspective, Taylor videos can be adapted to a broad range of areas, such as biology (time-lapse microscopy), medical analysis (endoscopy videos), sports analysis (match footage), physics (motion model), security (surveillance) and many more.

As Taylor videos focus on motions rather than spatial features such as color, our method can provide a higher degree of privacy and security compared to RGB videos. For example, Taylor videos can remove the distinct facial features of individuals within RGB videos (see Sec. D of supplementary material), allowing data collection and processing to have improved privacy.

Acknowledgements

Xiuyuan Yuan conducted this research under the supervision of Lei Wang in the ANU Summer Scholars Program. Lei Wang focused on mathematical analysis and modeling, while Xiuyuan Yuan implemented the code and conducted experiments. Xiuyuan Yuan is supported by a Summer Research Internship provided by the ANU School of Computing. This work is also supported by the NCI Adapter Scheme, with computational resources provided by NCI Australia, an NCRIS-enabled capability supported by the Australian Government.

References

Bertasius, G., Wang, H., and Torresani, L. Is space-time attention all you need for video understanding? In *ICML*, volume 2, pp. 4, 2021.

Bilen, H., Fernando, B., Gavves, E., Vedaldi, A., and Gould, S. Dynamic image networks for action recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016.

Bilen, H., Fernando, B., Gavves, E., and Vedaldi, A. Action recognition with dynamic image networks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 40(12):2799–2813, 2018. doi: 10.1109/TPAMI.2017.2769085.

Brox, T. and Malik, J. Large displacement optical flow: Descriptor matching in variational motion estimation. *TPAMI*, 33(3):500–513, March 2011. doi: 10.1109/TPAMI.2010.143.

Carreira, J. and Zisserman, A. Quo Vadis, Action Recognition? A New Model and the Kinetics Dataset. *CVPR*, pp. 1–10, 2018.

Chi, H.-G., Ha, M. H., Chi, S., Lee, S. W., Huang, Q., and Ramani, K. Infogcn: Representation learning for human skeleton-based action recognition. In *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 20154–20164, 2022. doi: 10.1109/CVPR52688.2022.01955.

Feichtenhofer, C., Pinz, A., and Wildes, R. P. Spatiotemporal residual networks for video action recognition. In *NIPS*, pp. 3468–3476, 2016.

Feichtenhofer, C., Fan, H., Malik, J., and He, K. Slowfast networks for video recognition. In *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 6202–6211, 2019.

Girdhar, R. and Ramanan, D. CATER: A diagnostic dataset for Compositional Actions and TEmporal Reasoning. In *ICLR*, 2020.

Kay, W., Carreira, J., Simonyan, K., Zhang, B., Hillier, C., Vijayanarasimhan, S., Viola, F., Green, T., Back, T., Natsev, P., Suleyman, M., and Zisserman, A. The kinetics human action video dataset. *arXiv*, 2017.

Kim, K., Gowda, S. N., Aodha, O. M., and Sevilla-Lara, L. Capturing temporal information in a single frame: Channel sampling strategies for action recognition. In *33rd British Machine Vision Conference 2022, BMVC 2022, London, UK, November 21-24, 2022*. BMVA Press, 2022. URL https://bmvc2022.mpi-inf.mpg.de/0355.pdf.

Koniusz, P., Wang, L., and Cherian, A. Tensor representations for action recognition. *TPAMI*, 2020.

Kuehne, H., Jhuang, H., Garrote, E., Poggio, T., and Serre, T. HMDB: A large video database for human motion recognition. In *ICCV*, pp. 2556–2563, 2011.

Li, Y., Sun, P., Qi, H., and Lyu, S. Celeb-DF: A Large-scale Challenging Dataset for DeepFake Forensics. In *IEEE Conference on Computer Vision and Patten Recognition (CVPR)*, Seattle, WA, United States, 2020.

Lin, J., Gan, C., and Han, S. Tsm: Temporal shift module for efficient video understanding, 2019.

Liu, J., Shahroudy, A., Perez, M., Wang, G., Duan, L.-Y., and Kot, A. C. Ntu rgb+d 120: A large-scale benchmark for 3d human activity understanding. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 2019.

Liu, Z., Lin, Y., Cao, Y., Hu, H., Wei, Y., Zhang, Z., Lin, S., and Guo, B. Swin transformer: Hierarchical vision transformer using shifted windows. In *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 10012–10022, 2021.

Liu, Z., Ning, J., Cao, Y., Wei, Y., Zhang, Z., Lin, S., and Hu, H. Video swin transformer. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 3202–3211, 2022.

Mahdisoltani, F., Berger, G., Gharbieh, W., Fleet, D., and Memisevic, R. On the effectiveness of task granularity for transfer learning. *arXiv preprint arXiv:1804.09235*, 2018.

Mandal, D., Narayan, S., Dwivedi, S. K., Gupta, V., Ahmed, S., Khan, F. S., and Shao, L. Out-of-distribution detection for generalized zero-shot action recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019.

Ni, B., Peng, H., Chen, M., Zhang, S., Meng, G., Fu, J., Xiang, S., and Ling, H. Expanding language-image pretrained models for general video recognition. In *Computer Vision–ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part IV*, pp. 1–18. Springer, 2022.

Perrett, T., Masullo, A., Burghardt, T., Mirmehdi, M., and Damen, D. Temporal-relational crosstransformers for few-shot action recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 475–484, June 2021.

Qin, Z., Liu, Y., Ji, P., Kim, D., Wang, L., McKay, R., Anwar, S., and Gedeon, T. Fusing higher-order features in graph neural networks for skeleton-based action recognition. *IEEE Transactions on Neural Networks and Learning Systems*, 2022.

Radford, A., Kim, J. W., Hallacy, C., Ramesh, A., Goh, G., Agarwal, S., Sastry, G., Askell, A., Mishkin, P., Clark, J., et al. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pp. 8748–8763. PMLR, 2021.

Revaud, J., Weinzaepfel, P., Harchaoui, Z., and Schmid, C. Epicflow: Edge-preserving interpolation of correspondences for optical flow. In *CVPR*, pp. 1164–1172, 2015. doi: 10.1109/CVPR.2015.7298720.

Rohrbach, M., Amin, S., Andriluka, M., and Schiele, B. A database for fine grained activity detection of cooking activities. In *CVPR*, 2012.

Sabater, A., Santos, L., Santos-Victor, J., Bernardino, A., Montesano, L., and Murillo, A. C. One-shot action recognition in challenging therapy scenarios. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, pp. 2777–2785, June 2021.

Shahroudy, A., Liu, J., Ng, T.-T., and Wang, G. Ntu rgb+d: A large scale dataset for 3d human activity analysis. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

Simonyan, K. and Zisserman, A. Two-stream convolutional networks for action recognition in videos. In *NIPS*, pp. 568–576, 2014.

Tran, D., Bourdev, L., Fergus, R., Torresani, L., and Paluri, M. Learning Spatiotemporal Features with 3D Convolutional Networks. *ICCV*, pp. 4489–4497, 2015.

Tran, D., Wang, H., Torresani, L., Ray, J., LeCun, Y., and Paluri, M. A closer look at spatiotemporal convolutions for action recognition. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, pp. 6450–6459, 2018.

Wang, H., Kläser, A., Schmid, C., and Liu, C.-L. Dense Trajectories and Motion Boundary Descriptors for Action Recognition. *IJCV*, 2013.

Wang, L. Analysis and evaluation of Kinect-based action recognition algorithms. Master's thesis, School of the Computer Science and Software Engineering, The University of Western Australia, Nov 2017.

Wang, L. *Robust human action modelling*. PhD thesis, The Australian National University (Australia), 2023.

Wang, L. and Koniusz, P. Self-supervising action recognition by statistical moment and subspace descriptors. In *Proceedings of the 29th ACM international conference on multimedia*, pp. 4324–4333, 2021.

Wang, L. and Koniusz, P. Temporal-viewpoint transportation plan for skeletal few-shot action recognition. In *Proceedings of the Asian Conference on Computer Vision*, pp. 4176–4193, 2022a.

Wang, L. and Koniusz, P. Uncertainty-dtw for time series and sequences. In *European Conference on Computer Vision*, pp. 176–195. Springer, 2022b.

Wang, L. and Koniusz, P. 3mformer: Multi-order multimode transformer for skeletal action recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 5620–5631, 2023.

Wang, L. and Koniusz, P. Flow dynamics correction for action recognition. In *ICASSP 2024-2024 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 3795–3799. IEEE, 2024.

Wang, L., Xiong, Y., Wang, Z., Qiao, Y., Lin, D., Tang, X., and Van Gool, L. Temporal segment networks for action recognition in videos. *IEEE transactions on pattern*

analysis and machine intelligence, 41(11):2740–2755, 2018.

Wang, L., Huynh, D. Q., and Koniusz, P. A comparative review of recent kinect-based action recognition algorithms. *TIP*, 2019a. ISSN 1057-7149. doi: 10.1109/TIP.2019.2925285.

Wang, L., Huynh, D. Q., and Mansour, M. R. Loss switching fusion with similarity search for video classification. *ICIP*, 2019b.

Wang, L., Koniusz, P., and Huynh, D. Q. Hallucinating idt descriptors and i3d optical flow features for action recognition with cnns. In *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 8698–8708, 2019c.

Wang, L., Liu, J., and Koniusz, P. 3d skeleton-based few-shot action recognition with jeanie is not so na\" ive. *arXiv preprint arXiv:2112.12668*, 2021a.

Wang, L., Tong, Z., Ji, B., and Wu, G. Tdn: Temporal difference networks for efficient action recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 1895–1904, 2021b.

Wang, L., Huang, B., Zhao, Z., Tong, Z., He, Y., Wang, Y., Wang, Y., and Qiao, Y. Videomae v2: Scaling video masked autoencoders with dual masking. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 14549–14560, 2023a.

Wang, L., Huang, B., Zhao, Z., Tong, Z., He, Y., Wang, Y., Wang, Y., and Qiao, Y. Videomae v2: Scaling video masked autoencoders with dual masking. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 14549–14560, June 2023b.

Wang, L., Koniusz, P., Gedeon, T., and Zheng, L. Adaptive multi-head contrastive learning. *arXiv preprint arXiv:2310.05615*, 2023c.

Wang, L., Liu, J., Zheng, L., Gedeon, T., and Koniusz, P. Meet jeanie: a similarity measure for 3d skeleton sequences via temporal-viewpoint alignment. *International Journal of Computer Vision (IJCV)*, 2024a.

Wang, L., Sun, K., and Koniusz, P. High-order tensor pooling with attention for action recognition. In *ICASSP 2024-2024 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 3885–3889. IEEE, 2024b.

Weinzaepfel, P., Revaud, J., Harchaoui, Z., and Schmid, C. Deepflow: Large displacement optical flow with deep matching. In *ICCV*, pp. 1385–1392, 2013. doi: 10.1109/ICCV.2013.175.

Wolf, L., Hassner, T., and Maoz, I. Face recognition in unconstrained videos with matched background similarity. In *CVPR 2011*, pp. 529–534, 2011. doi: 10.1109/CVPR.2011.5995566.

Wu, C.-Y., Zaheer, M., Hu, H., Manmatha, R., Smola, A. J., and Krähenbühl, P. Compressed video action recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 6026–6035, 2018.

Wu, W., Sun, Z., and Ouyang, W. Revisiting classifier: Transferring vision-language models for video recognition. 2023.

Yan, S., Xiong, Y., and Lin, D. Spatial Temporal Graph Convolutional Networks for Skeleton-Based Action Recognition. In *Proceedings of the AAAI Conference on Artificial Intelligence (AAAI)*, 2018.

Zach, C., Pock, T., and Bischof, H. A duality based approach for realtime tv-l1 optical flow. In Hamprecht, F. A., Schnörr, C., and Jähne, B. (eds.), *Pattern Recognition*, pp. 214–223, Berlin, Heidelberg, 2007. Springer Berlin Heidelberg. ISBN 978-3-540-74936-3.

Zhang, B., Wang, L., Wang, Z., Qiao, Y., and Wang, H. Real-time action recognition with enhanced motion vector cnns. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 2718–2726, 2016.

A. Proof of Equivalence

Below we provide the proof that Eq.(7) is equivalent to Eq.(3):

$$
\begin{aligned}
t(\mathbf{F}) &= \sum_{k=0}^{\infty} \frac{t^{(k)}(\widetilde{\mathbf{F}})}{k!} \odot \frac{1}{T}\sum_{\tau=1}^{T} (\mathbf{F} - \widetilde{\mathbf{F}})^{\circ k}_{:,:,\tau} = \frac{1}{T}\sum_{\tau=1}^{T}\sum_{k=0}^{\infty} \frac{t^{(k)}(\widetilde{\mathbf{F}})}{k!} \odot (\mathbf{F} - \widetilde{\mathbf{F}})^{\circ k}_{:,:,\tau} \\
&= \frac{1}{T}\sum_{\tau=1}^{T}\sum_{k=0}^{\infty} \frac{t^{(k)}(\widetilde{\mathbf{F}})}{k!} \odot ([\boldsymbol{F}_1 - \boldsymbol{F}_1, \boldsymbol{F}_2 - \boldsymbol{F}_1, \cdots, \boldsymbol{F}_\tau - \boldsymbol{F}_1, \cdots, \boldsymbol{F}_T - \boldsymbol{F}_1])^{\circ k}_{:,:,\tau} \\
&= \frac{1}{T}\sum_{\tau=1}^{T}\sum_{k=0}^{\infty} \frac{t^{(k)}(\widetilde{\mathbf{F}})}{k!} \odot [(\boldsymbol{F}_1 - \boldsymbol{F}_1)^{\circ k}, (\boldsymbol{F}_2 - \boldsymbol{F}_1)^{\circ k}, \cdots, \\
&\qquad\qquad (\boldsymbol{F}_\tau - \boldsymbol{F}_1)^{\circ k}, \cdots, (\boldsymbol{F}_T - \boldsymbol{F}_1)^{\circ k}]_{:,:,\tau} \\
&= \frac{1}{T}\sum_{\tau=1}^{T}\sum_{k=0}^{\infty} \Bigg[\frac{t^{(k)}(\widetilde{\mathbf{F}})}{k!} \odot (\boldsymbol{F}_1 - \boldsymbol{F}_1)^{\circ k}, \frac{t^{(k)}(\widetilde{\mathbf{F}})}{k!} \odot (\boldsymbol{F}_2 - \boldsymbol{F}_1)^{\circ k}, \cdots, \\
&\qquad\qquad \frac{t^{(k)}(\widetilde{\mathbf{F}})}{k!} \odot (\boldsymbol{F}_\tau - \boldsymbol{F}_1)^{\circ k}, \cdots, \frac{t^{(k)}(\widetilde{\mathbf{F}})}{k!} \odot (\boldsymbol{F}_T - \boldsymbol{F}_1)^{\circ k} \Bigg]_{:,:,\tau} \\
&= \frac{1}{T}\sum_{\tau=1}^{T}\sum_{k=0}^{\infty} \frac{t^{(k)}(\widetilde{\mathbf{F}})}{k!} \odot (\boldsymbol{F}_\tau - \boldsymbol{F}_1)^{\circ k}.
\end{aligned} \tag{8}
$$

Note that Eq.(8) is equivalent to Eq.(3) under the following conditions: (i) $\boldsymbol{F}_1$ is the first frame of temporal block $\mathbf{F} = [\boldsymbol{F}_1, \boldsymbol{F}_2, \cdots, \boldsymbol{F}_T] \in \mathbb{R}^{H \times W \times T}$, and $\widetilde{\mathbf{F}} = [\boldsymbol{F}_1, \boldsymbol{F}_1, \cdots, \boldsymbol{F}_1] \in \mathbb{R}^{H \times W \times T}$ denotes the generated dummy temporal block with no motion, (ii) both $f^{(k)}(\boldsymbol{F}_1)$ and $t^{(k)}(\widetilde{\mathbf{F}})$ captures the motions and dynamics within the temporal block. It is important to note that while the inputs to f and t differ, both functions capture the same motions. Specifically, f captures motions until frame $\boldsymbol{F}_\tau$, whereas t captures motions within the given temporal block.

B. Algorithm for Efficient Implementation

Algorithm 1 shows the efficient implementation of the Taylor video.

C. Action-level Compression

Below we provide visualisations on action-level compression ratio on both HMDB-51 and MPII Cooking Activity.

Fig. 9 shows the results for HMDB-51. It is noteworthy that, on average, the action *catch* exhibits the highest compression ratio compared to other actions. Interestingly, actions such as *smile*, *kiss*, and *laugh* demonstrate the lowest compression ratios. This can be attributed to the fact that these actions primarily occur around facial regions, making it challenging to capture their dominant motions compared to actions like *catch*, *hug* and *kick*.

Another potential factor contributing to this observation is the lower resolution and inherent noise present in the HMDB-51 dataset. Extracting dominant motions from videos with these characteristics proves to be a challenging task.

Fig. 10 shows the results for MPII Cooking Activity. As shown in the figure, most actions have more than $2\times$ compression ratio on average, it shows that fine-grained action recognition dataset indeed has much redundancy. We also notice that some actions such as *change temperature* and *take out from spice holder* have lower compression ratio. The possible reason behind this phenomenon is that these actions are too tiny and sometimes even smaller than the background noisy patterns, hence these noises become the dominant motions.

D. Taylor Frames for Face Videos

We use videos from the following two datasets to compute the Taylor videos.

Celeb-DF (v2) (Li et al., 2020) dataset contains real and DeepFake synthesized videos having similar visual quality on par

Algorithm 1 Efficient implementation of Taylor video

Input: Grayscale video $\mathbf{F} \in \mathbb{R}^{H \times W \times \mathcal{T}}$, total number of terms K, temporal sliding window size T (step size is 1)
if $T-3 < K$ **then**
  Print 'The given temporal block length T is not enough to compute K terms.'
else
  Create $\mathbf{M} \in \mathbb{R}^{H \times W \times 3 \times N}$ ($N = \mathcal{T} - T + 1$) for storing Taylor video
  for $i = 1$ **to** $\mathcal{T} - T + 1$ **do**
    Get the ith video temporal block $\mathbf{F}^i \in \mathbb{R}^{H \times W \times T}$
    Create a temporary copy: $\mathbf{\Delta}^i_{\text{temp}} = \mathbf{F}^i$
    Create a static video temporal block $\widetilde{\mathbf{F}}^i \in \mathbb{R}^{H \times W \times T}$ by duplicating the first frame of $\mathbf{F}^i$
    Create $\mathbf{D}^i$ to store different orders of frame differencing maps
    for $j = 1$ **to** $K+2$ **do**
      Compute frame differencing maps: $\mathbf{\Delta}^i_j = \mathbf{\Delta}^i_{\text{temp}}[:, :, 1:] - \mathbf{\Delta}^i_{\text{temp}}[:, :, :-1]$
      Save the 1st frame differencing map: $\mathbf{D}^i[:, :, j-1] = \mathbf{\Delta}^i_j[:, :, 0]$
      Save temporary frame differencing maps: $\mathbf{\Delta}^i_{\text{temp}} = \mathbf{\Delta}^i_j \in \mathbb{R}^{H \times W \times (T-j)}$
    end for
    Initialise $\mathbf{M}^i_d, \mathbf{M}^i_v, \mathbf{M}^i_a = \mathbf{0} \in \mathbb{R}^{H \times W \times T}$
    Compute $\mathbf{X}^i = \mathbf{F}^i - \widetilde{\mathbf{F}}^i$
    for $k = 0$ **to** K **do**
      $\mathbf{M}^i_d = \mathbf{M}^i_d + \frac{\mathbf{D}^i[:,:,k]}{k!} \odot (\mathbf{X}^i)^{\circ k}$
      $\mathbf{M}^i_v = \mathbf{M}^i_v + \frac{\mathbf{D}^i[:,:,k+1]}{k!} \odot (\mathbf{X}^i)^{\circ k}$
      $\mathbf{M}^i_a = \mathbf{M}^i_a + \frac{\mathbf{D}^i[:,:,k+2]}{k!} \odot (\mathbf{X}^i)^{\circ k}$
    end for
    Get the ith Taylor frame: $\boldsymbol{M}^i = [\mathbf{M}^i_d.\text{mean}(2); \mathbf{M}^i_v.\text{mean}(2); \mathbf{M}^i_a.\text{mean}(2)] \in \mathbb{R}^{H \times W \times 3}$
    Form a Taylor video: $\mathbf{M}[:, :, :, i] = \boldsymbol{M}^i$
  end for
end if
Output: Taylor video $\mathbf{M} \in \mathbb{R}^{H \times W \times 3 \times N}$

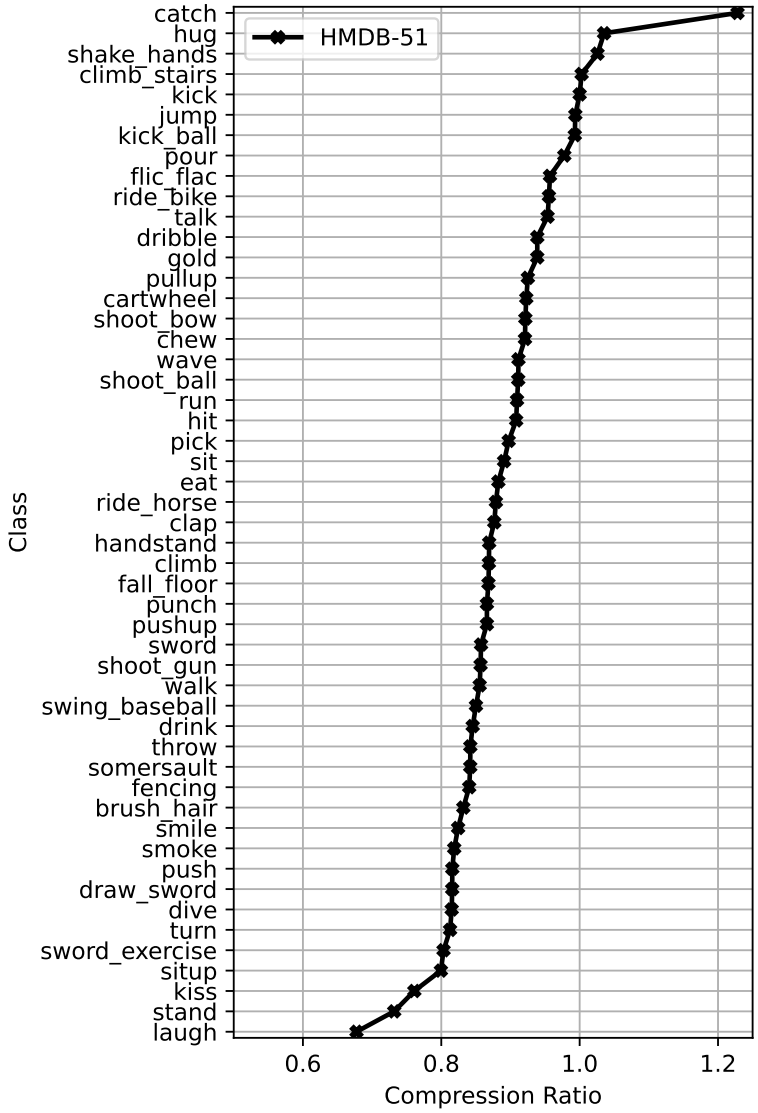


Figure 9. Compression ratio per action on HMDB-51.

Figure 10. Compression ratio per action on MPII Cooking Activities.

Figure 11. We use videos from (*Top two rows*) Celeb-DF (v2) and (*Bottom two rows*) YouTube Faces to show that Taylor videos are able to remove distinct facial features of individuals compared to RGB videos. This allows the data collection and processing to have improved privacy.

with those circulated online. The Celeb-DF (v2) dataset is greatly extended from the previous Celeb-DF (v1), which only contains 795 DeepFake videos. To date, Celeb-DF includes 590 original videos collected from YouTube with subjects of different ages, ethic groups and genders, and 5639 corresponding DeepFake videos.

YouTube Faces (Wolf et al., 2011) is a database of face videos designed for studying the problem of unconstrained face recognition in videos. The data set contains 3,425 videos of 1,595 different people. All the videos were downloaded from YouTube. An average of 2.15 videos are available for each subject. The shortest clip duration is 48 frames, the longest clip is 6,070 frames, and the average length of a video clip is 181.3 frames.

Fig. 11 shows that Taylor videos can remove the distinct facial features of individuals within RGB videos. This allows the data collection and processing to have improved privacy.