A New Large Scale Dynamic Texture Dataset with Application to ConvNet Understanding

Isma Hadji and Richard P. Wildes

York University, Toronto, Ontario, Canada
{hadjisma,wildes}@cse.yorku.ca

Abstract. We introduce a new large scale dynamic texture dataset. With over 10,000 videos, our Dynamic Texture DataBase (DTDB) is two orders of magnitude larger than any previously available dynamic texture dataset. DTDB comes with two complementary organizations, one based on dynamics independent of spatial appearance and one based on spatial appearance independent of dynamics. The complementary organizations allow for uniquely insightful experiments regarding the abilities of major classes of spatiotemporal ConvNet architectures to exploit appearance vs. dynamic information. We also present a new two-stream ConvNet that provides an alternative to the standard optical-flow-based motion stream to broaden the range of dynamic patterns that can be encompassed. The resulting motion stream is shown to outperform the traditional optical flow stream by considerable margins. Finally, the utility of DTDB as a pretraining substrate is demonstrated via transfer learning on a different dynamic texture dataset as well as the companion task of dynamic scene recognition resulting in a new state-of-the-art.

1 Introduction

Visual texture, be it static or dynamic, is an important scene characteristic that provides vital information for segmentation into coherent regions and identification of material properties. Moreover, it can support subsequent operations involving background modeling, change detection and indexing. Correspondingly, much research has addressed static texture analysis for single images (*e.g.* [21, 6, 5, 36, 35]). In comparison, research concerned with dynamic texture analysis from temporal image streams (*e.g.* video) has been limited (*e.g.* [15, 26, 38, 27]).

The relative state of dynamic vs. static texture research is unsatisfying because the former is as prevalent in the real world as the latter and it provides similar descriptive power. Many commonly encountered patterns are better described by global dynamics of the signal rather than individual constituent elements. For example, it is more perspicuous to describe the global motion of the leaves on a tree as windblown foliage rather than in terms of individual leaf motion. Further, given the onslaught of video available via on-line and other sources, applications of dynamic texture analysis may eclipse those of static texture.

Dynamic texture research is hindered by a number of factors. A major issue is lack of clarity on what constitutes a dynamic texture. Typically, dynamic textures are defined as temporal sequences exhibiting certain temporal statistics or stationary properties in time [30]. In practice, however, the term dynamic texture is usually used to describe the case of image sequences exhibiting stochastic dynamics (*e.g.* turbulent water and windblown vegetation). This observation is evidenced by the dominance of such textures in the UCLA [30] and DynTex [24] datasets. A more compelling definition describes dynamic texture as any temporal sequence that can be characterized by the same aggregate dynamic properties across its support region [8]. Hence, the dominant dynamic textures in UCLA and DynTex are the subclass of textures that exhibit stochastic motion. Another concern with definitions applied in extant datasets is that the classes are usually determined by appearance, which defeats the purpose of studying the *dynamics* of these textures. The only dataset that stands out in this regard is YUVL [8], wherein classes were defined explicitly in terms of pattern dynamics.

The other major limiting factors in the study of dynamic textures are lack of size and diversity in extant datasets. Table 1 documents the benchmarks used in dynamic texture recognition. It is apparent that these datasets are small compared to what is available for static texture (*e.g.* [5, 7, 23]). Further, limited diversity is apparent, *e.g.* in cases where the number of sequences is greater than the number videos, multiple sequences were generated as clips from single videos. Diversity also is limited by different classes sometimes being derived from slightly different views of the same physical phenomenon. Moreover, diversity is limited in variations that have a small number of classes. Finally, it is notable that all current dynamic texture datasets are performance saturated [15].

Table 1. Comparison of the new DTDB dataset with other dynamic texture datasets

Dataset	DynTex [24]					UCLA [30]					YUVL [8]			**DTDB (Ours)**	
Dataset Variations	Alpha [11]	Beta [11]	Gamma [11]	35 [40]	++ [14]	50 [30]	9 [14]	8 [28]	7 [9]	SIR [9]	1 [8]	2 [8]	3 [15]	Appearance	Dynamics
#Videos	60	162	264	35	345	50	50	50	50	50	610	509	610	**>9K**	**>10K**
#Sequences	60	162	264	350	3600	200	200	92	400	400	610	509	610	**>9K**	**>10K**
#Frames	>140K	>397K	>553K	>8K	>17K	15K	15K	>6K	15K	15K	>65K	>55K	>65K	**>3.1 million**	**>3.4 million**
#Classes	3	10	10	35	36	50	9	8	7	50	5	6	8	**45**	**18**

Over the past few years, increasingly larger sized datasets (*e.g.* [29, 41, 18]) have driven progress in computer vision, especially as they support training of powerful ConvNets (*e.g.* [19, 32, 16]). For video based recognition, action recognition is the most heavily researched task and the availability of large scale datasets (*e.g.* UCF-101 [33] and the more recent Kinetics [3]) play a significant role in the progress being made. Therefore, large scale dynamic texture datasets are of particular interest to support use of ConvNets in this domain.

In response to the above noted state of affairs, we make the following contributions. 1) We present a new large scale dynamic texture dataset that is two orders of magnitude larger than any available. At over 10,000 videos, it is comparable in size to UCF-101 that has played a major role in advances to action recognition. 2) We provide two complementary organizations of the dataset. The first groups videos based on their dynamics irrespective of their static (single frame) appearance. The second groups videos purely based on their visual appearance. For example, in addition to describing a sequence as containing car traffic, we complement the description with dynamic information that allows making the distinction between smooth and chaotic car traffic. Figure 1 shows frames from the large spectrum of videos present in the dataset and illustrates how videos are assigned to different classes depending on the grouping criterion (*i.e.* dynamics vs. appearance). 3) We use the new dataset to explore the representational power of different spatiotemporal ConvNet architectures. In particular, we examine the relative abilities of architectures that directly apply 3D filtering to input videos [34, 15] vs. two-stream architectures that explicitly separate appearance and motion information [31, 12]. The two complementary organizations of the same dataset allow for uniquely insightful experiments regarding the capabilities of the algorithms to exploit appearance vs. dynamic information. 4) We propose a novel two-stream architecture that yields superior performance to more standard two-stream approaches on the dynamic texture recognition task. 5) We demonstrate that our new dataset is rich enough to support transfer learning to a different dynamic texture dataset, YUVL [8], and to a different task, dynamic scene recognition [13], where we establish a new state-of-the-art. Our novel Dynamic Texture DataBase (DTDB) is available at http://vision.eecs.yorku.ca/research/dtdb/.

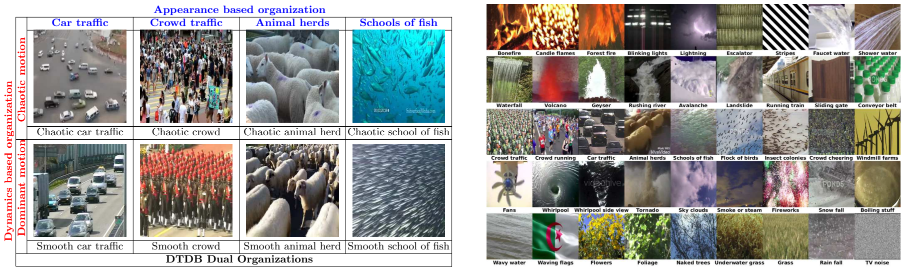


Fig. 1. (Left) Sample frames from the proposed Dynamic Texture DataBase (DTDB) and their assigned categories in both the dynamics and appearance based organizations. **(Right)** Thumbnail examples of the different appearance based dynamic textures present in the new DTDB dataset. See supplemental material for videos.

2 Dynamic Texture DataBase (DTDB)

The new dataset, Dynamic Texture DataBase (DTDB), constitutes the largest dynamic texture dataset available with $>$ *10,000 videos* and $\approx$ *3.5 million frames.* As noted above, the dataset is organized in two different ways with 18 dynamics based categories and 45 appearance based categories. Table 1 compares our dataset with previous dynamic texture benchmarks showing the significant improvements compared to alternatives. The videos are collected from various sources, including the web and various handheld cameras that we employed, which helps ensure diversity and large intra-class variations. Figure 1 provides thumbnail examples from the entire dataset. Corresponding videos and descriptions are provided in the supplemental material.

Dynamic Category Specification. The dataset was created with the main goal of building a true *dynamic* texture dataset where sequences exhibiting similar dynamic behaviors are grouped together irrespective of their appearance. Previous work provided a principled approach to defining five coarse dynamic texture categories based on the number of spatiotemporal orientations present in a sequence [8], as given in the left column of Table 2. We use that enumeration as a point departure, but subdivide the original categories to yield a much larger set of 18 categories, as given in the middle column of Table 2. Note that the original categories are subdivided in a way that accounts for increased variance about the prescribed orientation distributions in the original classes. For example, patterns falling under *dominant orientation* (*i.e.* sequences dominated by a single spacetime orientation) were split into five sub-categories: (1) Single Rigid Objects, (2) Multiple Rigid Objects, (3) Smooth Non-Rigid Objects, (4) Turbulent Non-Rigid Objects and (5) Pluming Non-Rigid Objects, all exhibiting motion along a dominant direction, albeit with increasing variance (*c.f.* [20]); see Fig. 2. At an extreme, the original category *Isotropic* does not permit further subdivision based on increased variance about its defining orientations, because although it may have significant spatiotemporal contrast, it lacks in discernable orientation(s), *i.e.* it exhibits isotropic pattern structure. See supplemental material for video examples of all categories, with accompanying discussion.

Table 2. Dynamics based categories in the DTDB dataset. A total of 18 different categories are defined by making finer distinctions in the spectrum of dynamic textures proposed originally in [8]. Subdivisions of the original categories occur according to increased variance (indicated by arrow directions) about the orientations specified to define the original categories; see text for details. The supplement provides videos.

Original YUVL categories		**DTDB categories**	
Name/Description		**Name/Description**	**Example sources**
Underconstrained spacetime orientation	↓	Aperture Problem	conveyor belt, barber pole
		Blinking	blinking lights, lightning
		Flicker	fire, shimmering steam
Dominant spacetime orientation	↓	Single Rigid Object	train, plane
		Multiple Rigid Objects	smooth traffic, smooth crowd
		Smooth Non-Rigid Objects	faucet water, shower water
		Turbulent Non-Rigid Objects	geyser, fountain
		Pluming Non-Rigid Objects	avalanche, landslide
Multi-dominant spacetime orientation	↓	Rotary Top-View	fan, whirlpool from top
		Rotary Side-View	tornado, whirlpool from side
		Transparency	translucent surfaces, chain link fence vs. background
		Pluming	smoke, clouds
		Explosion	fireworks, bombs
		Chaotic	swarming insects, chaotic traffic
Heterogeneous spacetime orientation	↓	Waves	wavy water, waving flags
		Turbulence	boiling liquid, bubbles
		Stochastic	windblown leaves, flowers
Isotropic	↓	Scintillation	TV noise, scintillating water

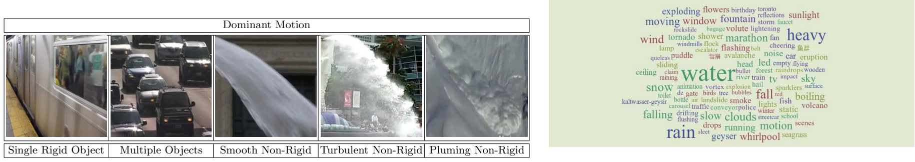

Dominant Motion
Single Rigid Object
Multiple Objects
Smooth Non-Rigid
Turbulent Non-Rigid
Pluming Non-Rigid
exploding flowers birthday toronto
reflections sunlight
moving window fountain storm faucet
rockslide bagage volute lightening
wind tornado shower marathon fan heavy
windmills flock cheering
lamp escalator flashing belt 鱼群
queleas puddle 雪崩 avalanche noise car eruption
sliding head led empty flying
ceiling claim water bullet forest raindrops wooden
raining river train tv impact sky
snow animation vortex explosion hail sparklers surface
toilet de gate birds tree bubbles fall red boiling
kaltwasser-geysir bottle air landslide smoke fish
carousel traffic conveyor police lights volcano
falling drifting winter static
flushing slow clouds streetcar school
drops running motion scenes
rain sleet geyser whirlpool seagrass

Fig. 2. (Left) Example of the finer distinctions we make within dynamic textures falling under the broad dominant motion category. Note the increased level of complexity in the dynamics from left to right. (**Right**) Keywords wordle. Bigger font size of a word indicates higher frequency of the keyword resulting in videos in the dataset.

Keywords and Appearance Categories. For each category, we brainstormed a list of scenes, objects and natural phenomena that could contain or exhibit the desired dynamic behavior and used their names as keywords for subsequent web search. To obtain a large scale dataset, an extensive list of English keywords were generated and augmented with their translations to various languages: Russian, French, German and Mandarin. A visualization of the generated keywords and their frequency of occurrence across all categories is represented as a wordle [2] in Fig. 2. To specify appearance catergories, we selected 45 of the keywords, which taken together covered all the dynamics categories. This approach was possible, since on-line tags for videos are largely based on appearance. The resulting appearance categories are given as sub-captions in Fig. 1.

Video Collection. The generated keywords were used to crawl videos from YouTube [39], Pond5 [25] and VideoHive [37]. In doing so, it was useful to specifically crawl playlists. Since playlists are created by human users or generated by machine learning algorithms, their videos share similar tags and topics; therefore, the videos crawled from playlists were typically highly correlated and had a high probability of containing the dynamic texture of interest. Finally, the links (URLs) gathered using the keywords were cleaned to remove duplicates.

Annotation. Annotation served to verify via human inspection the categories present in each crawled video link. This task was the main bottleneck of the collection process and required multiple annotators for good results. Since the annotation required labeling the videos according to dynamics while ignoring appearance and vice versa, it demanded specialist background and did not lend itself well to tools such as Mechanical Turk [1]. Therefore, two annotators with computer vision background were hired and trained for this task.

Annotation employed a custom web-based tool allowing the user to view each video according to its web link and assign it the following attributes: a dynamics-based label (according to the 18 categories defined in Table 2), an appearance-based label (according to the 45 categories defined in Fig. 1) and start/end times of the pattern in the video. Each video was separately reviewed by both annotators. When the two main annotators disagreed, a third annotator (also with computer vision background) attempted to resolve matters with consensus and if that was not possible the link was deleted. Following the annotations, the specified portions of all videos were downloaded with their labels.

Dataset Cleaning. For a clean dynamic texture dataset, we chose that the target texture should occupy at least 90% of the spatial support of the video and all of the temporal support. Since such requirements are hard to meet with videos acquired in the wild and posted on the web, annotators were instructed to accept videos even if they did not strictly meet this requirement. In a subsequent step, the downloaded videos were visually inspected again and spatially cropped so that the resulting sequences had at least 90% of their spatial support occupied by the target dynamic texture. To ensure the cropping did not severely compromise the overall size of the texture sample, any video whose cropped spatial dimensions were less than 224×224 was deleted from the dataset. The individuals who did the initial annotations also did the cleaning.

This final cleaning process resulted in slightly over 9000 clean sequences. To obtain an even larger dataset, it was augmented in two ways. First, relevant videos from the earlier DynTex [24] and UCLA [30] datasets were selected (but none from YUVL [8]), while avoiding duplicates; second, several volunteers contributed videos that they recorded (*e.g.* with handheld cameras). These additions resulted in the final dataset containing 10,020 sequences with various spatial supports and temporal durations (5-10 seconds).

Dynamics and Appearance Based Organization. All the 10,020 sequences were used in the dynamics based organization with an average number of videos per category of 556 ± 153. However, because the main focus during data collection was dynamics, it was noticed that not all appearance based video tags generated enough appearance based sequences. Therefore, to keep the dataset balanced in the appearance organization as well, any category containing less than 100 sequences was ignored in the appearance based organization. This process led to an appearance based dataset containing a total 9206 videos divided into 45 different classes with an average number of videos per category of 205±95.

3 Spatiotemporal ConvNets

There are largely two complementary approaches to realizing spatiotemporal ConvNets. The first works directly with input temporal image streams (*i.e.* video), *e.g.* [17, 18, 34]. The second takes a two-stream approach, wherein the image information is processed in parallel pathways, one for appearance (RGB images) and one for motion (optical flow), *e.g.* [31, 22, 12]. For the sake of our comparisons, we consider a straightforward exemplar of each class that previously has shown strong performance in spatiotemporal image understanding. In particular, we use C3D [34] as an example of working directly with input video and Simonyan and Zisserman Two-Stream [31] as an example of splitting appearance and motion at the input. We also consider two additional networks: A novel two-stream architecture that is designed to overcome limitations of optical flow in capturing dynamic textures and a learning-free architecture that works directly on video input and recently has shown state-of-the-art performance on dynamic texture recognition with previously available datasets [15]. Importantly, in selecting this set of four ConvNet architectures to compare, we are not seeking to compare details of the wide variety of instantiations of the two broad classes considered, but more fundamentally to understand the relative power of the single and two-stream approaches. In the remainder of this section we briefly outline each algorithm compared; additional details are in the supplemental material.

C3D. C3D [34] works with temporal streams of RGB images. It operates on these images via multilayer application of learned 3D, (x, y, t), convolutional filters. It thereby provides a fairly straightforward generalization of standard 2D ConvNet processing to image spacetime. This generalization entails a great increase in the number of parameters to be learned, which is compensated for by using very limited spacetime support at all layers ($3 \times 3 \times 3$ convolutions). Consideration of this type of ConvNet allows for evaluation of the ability of integrated spacetime filtering to capture both appearance and dynamics information.

Two-stream. The standard Two-Stream architecture [31] operates in two parallel pathways, one for processing appearance and the other for motion. Input to the appearance pathway are RGB images; input to the motion path are stacks of optical flow fields. Essentially, each stream is processed separately with fairly standard 2D ConvNet architectures. Separate classification is performed by each pathway, with late fusion used to achieve the final result. Consideration of this type of ConvNet allows evaluation of the two streams to separate appearance and dynamics information for understanding spatiotemporal content.

MSOE-two-stream. Optical flow is known to be a poor representation for many dynamic textures, especially those exhibiting decidedly non-smooth and/or stochastic characteristics [10, 8]. Such textures are hard for optical flow to capture as they violate the assumptions of brightness constancy and local smoothness that are inherent in most flow estimators. Examples include common real-world patterns shown by wind blown foliage, turbulent flow and complex lighting effects (*e.g.* specularities on water). Thus, various alternative approaches have been used for dynamic texture analysis in lieu of optical flow [4].

A particularly interesting alternative to optical flow in the present context is appearance Marginalized Spatiotemporal Oriented Energy (MSOE) filtering [8]. This approach applies 3D, (x, y, t), oriented filters to a video stream and thereby fits naturally in a convolutional architecture. Also, its appearance marginalization abstracts from purely spatial appearance to dynamic information in its output and thereby provides a natural input to a motion-based pathway. Correspondingly, as a novel two-stream architecture, we replace input optical flow stacks in the motion stream with stacks of MSOE filtering results. Otherwise, the two-stream architecture is the same, including use of RGB frames to capture appearance. Our hypothesis is that the resulting architecture, MSOE-two-stream, will be able to capture a wider range of dynamics in comparison to what can be captured by optical flow, while maintaining the ability to capture appearance.

SOE-Net. SOE-Net [15] is a learning-free spatiotemporal ConvNet that operates by applying 3D oriented filtering directly to input temporal image sequences. It relies on a vocabulary of theoretically motivated, analytically defined filtering operations that are cascaded across the network layers via a recurrent connection to yield a hierarchical representation of input data. Previously, this network was applied to dynamic texture recognition with success. This network allows for consideration of a complimentary approach to that of C3D in the study of how direct 3D spatiotemporal filtering can serve to jointly capture appearance and dynamics. Also, it serves to judge the level of challenge given by the new DTDB dataset in the face of a known strong approach to dynamic texture.

4 Empirical Evaluation

The goals of the proposed dataset in its two organizations are two fold. First, it can be used to help better understand strengths and weaknesses of learning based spatiotemporal ConvNets and thereby guide decisions in the choice of architecture depending on the task at hand. Second, it can serve as a training substrate to advance research on dynamic texture recognition, in particular, and an initialization for other related tasks, in general. Correspondingly, from an algorithmic perspective, our empirical evaluation aims at answering the following questions: **1)** Are spatiotemporal ConvNets able to disentangle appearance and dynamics information? **2)** What are the relative strengths and weaknesses of popular architectures in doing so? **3)** What representations of the input data are best suited for learning strong representations of image dynamics? In complement, we also address questions from the dataset's perspective. **1)** Does the new dataset provide sufficient challenges to drive future developments in spatiotemporal image analysis? **2)** Can the dataset be beneficial for transfer learning to related tasks? And if so: **3)** What organization of the dataset is more suitable in transfer learning? **4)** Can finetuning on our dataset boost the state-of-the-art on related tasks even while using standard spatiotemporal ConvNet architectures?

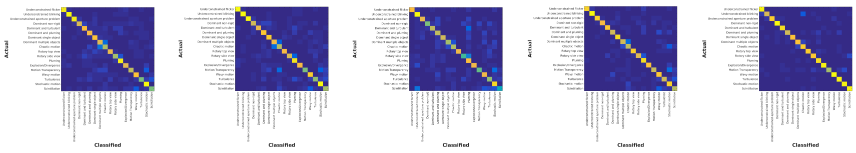
4.1 What Are Spatiotemporal ConvNets Better at Learning? Appearance vs. Dynamics

Experimental Protocol. For training purposes each organization of the dataset is split randomly into training and test sets with 70% of the videos from each category used for training and the rest for testing. The C3D [34] and standard two-stream [31] architectures are trained following the protocols given in their original papers. The novel MSOE-two-stream is trained analogously to the standard two-stream, taking into account the changes in the motion stream input (*i.e.* MSOE rather than optical flow). For a fair comparison of the relative capabilities of spatiotemporal ConvNets in capitalizing on both motion and appearance, all networks are trained from scratch on DTDB to avoid any counfounding variables (*e.g.* as would arise from using the available models of C3D and two-stream as pretrained on different datasets). Training details can be found in the supplemental material. No training is associated with SOE-Net, as all its parameters are specified by design. At test time, the held out test set is used and the reported results are obtained from the softmax scores of each network. Note that we compare recognition performance for each organization separately; it does not make sense in the present context to train on one organization and test on the other since the categories are different. (We do however report related transfer learning experiments in Secs. 4.2 and 4.3. The experiments of Sec. 4.3 also consider pretrained versions of the C3D and two-stream architectures.)

Table 3. Recognition accuracy of all the evaluated networks using both organizations of the new Dynamic Texture DataBase

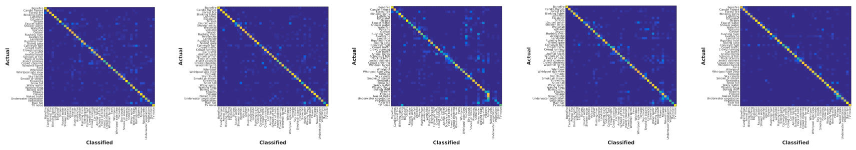
	DTDB-Dynamics	**DTDB-Appearance**
C3D [34]	74.9	75.5
RGB Stream [31]	76.4	**76.1**
Flow Stream [31]	72.6	64.8
MSOE Stream	**80.1**	72.2
MSOE-two-stream	**84.0**	**80.0**
SOE-Net [15]	**86.8**	**79.0**

Results. Table 3 provides a detailed comparison of all the evaluated Networks. To begin, we consider the relative performance of the various architectures on the dynamics-based organization. Of the learning-based approaches (*i.e.* all but SOE-Net), it is striking that RGB stream outperforms the Flow stream as well as C3D, even though the latter two are designed to capitalize on motion information. A close inspection of the confusion matrices (Fig. 3) sheds light on this situation. It is seen that the networks are particularly hampered when similar appearances are present across different dynamics categories as evidenced by the two most confused classes (*i.e.* Chaotic motion and Dominant Multiple Rigid Objects). These two categories were specifically constructed to have this potential source of appearance-based confusion to investigate an algorithm's ability to abstract from appearance to model dynamics; see Fig. 1 and accompanying videos in the supplemental material. Also of note is performance on the categories that are most strongly defined in terms of their dynamics and show little distinctive structure in single frames (*e.g.* Scintillation and motion Transparency). The confusions experienced by C3D and the Flow stream indicate that those approaches have poor ability to learn the appropriate abstractions. Indeed, the performance of the Flow stream is seen to be the weakest of all. The likely reason for the poor Flow stream performance is that its input, optical flow, is not able to capture the underlying dynamics in the videos because they violate standard optical flow assumptions of brightness constancy and local smoothness.



C3D **RGB Stream** **Flow Stream** **MSOE Stream** **SOE-Net**

Fig. 3. Confusion matrices of all the compared ConvNet architectures on the *dynamics* based organization of the new DTDB



C3D **RGB Stream** **Flow Stream** **MSOE Stream** **SOE-Net**

Fig. 4. Confusion matrices of all compared ConvNet architectures on the *appearance* based organization of the new DTDB

These points are underlined by noting that MSOE stream has the best performance compared to the other individual streams, with increased performance margin ranging from ≈4-8%. Based on this result, to judge the two-stream benefit we fuse the appearance (RGB) stream with MSOE stream to yield MSOE-two-stream as the overall top performer among the learning-based approaches. Importantly, recall that the MSOE input representation was defined to overcome the limitations of optical flow as a general purpose input representation for learning dynamics. These results speak decisively in favour of MSOE filtering as a powerful input to dynamics-based learning: It leads to performance that is as good as optical flow for categories that adhere to optical flow assumptions, but extends performance to cases where optical flow fails. Finally, it is interesting to note that the previous top dynamic texture recognition algorithm, hand-crafted SOE-Net, is the best overall performer on the dynamics organization, showing that there remains discriminatory information to be learned from this dataset.

Turning attention to the appearance based results reveals the complementarity between the proposed dynamics and appearance based organizations. In this case, since the dataset is dominated by appearance, the best performer is the RGB stream that is designed to learn appearance information. Interestingly, C3D's performance, similar to the RGB stream, is on par for the two organizations although C3D performs slightly better on the appearance organization. This result suggests that C3D's recognition is mainly driven by similarities in appearance in both organizations and it appears relatively weak at capturing dynamics. This limitation may be attributed to the extremely small support of C3D's kernels (*i.e.* $3 \times 3 \times 3$). Also, as expected, the performance of the Flow and MSOE streams degrade on the appearance based organization, as they are designed to capture dynamics-based features. However, even on the appearance based organization, MSOE stream outperforms its Flow counterpart by a sizable margin. Here inspection of the confusion matrices (Fig. 4), reveals that C3D and the RGB stream tend to make similar confusions, which confirms the tendency of C3D to capitalize on appearance. Also, it is seen that the Flow and MSOE streams tend to confuse categories that exhibit the same dynamics (*e.g.* classes with stochastic motion such as Flower, Foliage and Naked trees), which explains the degraded performance of these two streams. Notably, MSOE streams incurs less confusions, which demonstrates the ability of MSOE filters to better capture fine grained differences. Also, once again MSOE-two-stream is the best performer among the learning based approaches and in this case it is better than SOE-Net.

Conclusions. Overall, the results on both organizations of the dataset lead to two main conclusions. First, comparison of the different architectures reveal that two-stream networks are better able to disentangle motion from appearance information for the learning-based architectures. This fact is particularly clear from the inversion of performance between the RGB and MSOE streams depending on whether the networks are trained to recognize dynamics or appearance, as well as the degraded performance of both the Flow and MSOE streams when asked to recognize sequences based on their appearance. Second, closer inspection of the confusion matrices show that optical flow fails on most categories where the sequences break the fundamental optical flow assumptions of brightness constancy and local smoothness (*e.g.* Turbulent motion, Transparency and Scintillation). In contrast, the MSOE stream performs well on such categories as well as others that are relatively easy for the Flow stream. The overall superiority of MSOE reflects in its higher performance, compared to flow, on both organizations of the dataset. These results challenge the common practice of using flow as the default representation of input data for motion stream training and should be taken into account in design of future spatiotemporal ConvNets.

Additionally, it is significant to note that a ConvNet that does not rely on learning, SOE-Net, has the best performance on the dynamics organization and is approximately tied for best on the appearance organization. These results suggests the continued value of DTDB, as there is more for future learning-based approaches to glean from its data.

4.2 Which Organization of DTDB Is Suitable in Transfer Learning?

Experimental Protocol. Transfer learning is considered with respect to a different dynamic texture dataset and a different task, dynamic scene recognition. The YUVL dataset [8] is used for the dynamic texture experiment. Before the new DTDB, YUVL was the largest dynamic texture dataset with a total of 610 sequences and it is chosen as a representative of a dataset with categories mostly dominated by the dynamics of its sequences. It provides 3 different dynamics based organizations, YUVL-1, YUVL-2 and YUVL-3 with 5, 6 and 8 classes (resp.) that make various dynamics based distinctions; see [8, 15]. For the dynamic scene experiment, we use the YUP++ dataset [13]. YUP++ is the largest dynamic scenes dataset with 1200 sequences in total divided into 20 classes; however, in this case the categories are mostly dominated by differences in appearance. Notably, YUP++ provides a balanced distribution of sequences with and without camera motion, which allows for an evaluation of the various trained networks in terms of their ability to abstract scene dynamics from camera motion. Once again, for fair comparison, the various architectures trained from scratch on DTDB are used in this experiment because the goal is not to establish new state-of-the-art on either YUVL or YUP++. Instead, the goal is to show the value of the two organizations of the dataset and highlight the importance of adapting the training data to the application. The conclusions of this experiment are used next, in Sec 4.3, as a basis to finetune the architectures under considerations using the appropriate version of DTDB.

For both the dynamic texture and dynamic scenes cases, we consider the relative benefits of training on the appearance vs. dynamics organizations of DTDB. We also compare to training using UCF-101 as a representative of a similar scale dataset but that is designed for the rather different task of action recognition. Since the evaluation datasets (*i.e.* YUVL and YUP++) are too small to support finetuning, we instead extract features from the last layers of the networks as trained under DTDB or UCF-101 and use those features for recognition (as done previously under similar constraints of small target datasets, *e.g.* [34]). A preliminary evaluation comparing the features extracted from the last pooling layer, fc6 and fc7, of the various networks used, showed that there is always a decrement in performance going from fc6 to fc7 on both datasets and out of 48 comparison points the performance of features extracted from the last pooling layer was better 75% of the time. Hence, results reported in the following rely on features extracted from the last pool layer of all used networks.

For recognition, extracted features are used with a linear SVM classifier using the standard leave-one-out protocol usually used with these datasets [8, 27, 15].

Results. We begin by considering results of transfer learning applied to the YUVL dataset, summarized in Table 4 (Left). Here, it is important to emphasize that YUVL categories are defined in terms of texture dynamics, rather than appearance. Correspondingly, we find that for every architecture the best performance is attained via pretraining on the DTDB dynamics-based organization as opposed to the appearance-based organization or UCF-101 pretraining. These results clearly support the importance of training for a dynamics-based task on dynamics-based data. Notably, MSOE stream, and its complementary MSOE-two-stream approach, with dynamics training show the strongest performance on this task, which provides further support for MSOE filtering as the basis for input to the motion stream of a two-stream architecture.

Table 4. Performance of spatiotemporal ConvNets, *trained* using both organizations of DTDB, **(Left)** on the various breakdowns of the YUVL dataset [8] and **(Right)** on the **S**tatic and **M**oving camera portions of YUP++ and the entire YUP++ [13]

		YUVL-1	**YUVL-2**	**YUVL-3**
UCF-101 based training	**C3D**	61.4	65.4	55.7
	RGB Stream	63.6	72.8	60.0
	Flow Stream	84.8	87.3	81.7
	MSOE Stream	80.0	80.2	74.4
	MSOE-two-stream	80.8	84.5	78.8
Dynamics based training	**C3D**	83.3	86.4	83.4
	RGB Stream	68.1	75.4	65.0
	Flow Stream	87.7	86.9	83.1
	MSOE Stream	**89.2**	**89.3**	**84.8**
	MSOE-two-stream	**90.7**	**91.4**	**87.6**
Appearance based training	**C3D**	82.2	85.4	80.9
	RGB Stream	67.6	72.8	64.3
	Flow Stream	86.7	85.7	81.3
	MSOE Stream	87.7	87.3	83.6
	MSOE-two-stream	89.8	90.2	86.7

		YUP++(S)	**YUP++(M)**	**YUP++**
UCF-101 based training	**C3D**	62.5	55.8	58.3
	RGB Stream	64.9	54.4	63.5
	Flow Stream	83.6	51.9	68.9
	MSOE Stream	74.3	52.7	62.0
	MSOE-two-stream	80.1	66.6	74.6
Dynamics based training	**C3D**	84.3	71.8	76.5
	RGB Stream	81.8	73.7	78.3
	Flow Stream	89.3	64.7	76.8
	MSOE Stream	90.0	67.5	78.4
	MSOE-two-stream	93.3	81.5	87.7
Appearance based training	**C3D**	85.0	73.7	78.1
	RGB Stream	82.0	**76.2**	**79.9**
	Flow Stream	90.6	65.8	77.0
	MSOE Stream	**91.0**	69.5	79.1
	MSOE-two-stream	**94.7**	**83.2**	**89.6**

Comparison is now made on the closely related task of dynamic scene recognition. As previously mentioned, although YUP++ is a dynamic scenes datasets its various classes are still largely dominated by differences in appearance. This dominance of appearance is well reflected in the results shown in Table 4 (Right). As opposed to the observations made on the previous task, here networks benefited more from an appearance-based training to various extents with the advantage over UCF-101 pretraining being particularly striking. In agreement with findings on the YUVL dataset and in Section 4.1, the RGB stream trained on appearance is the overall best performing individual stream on this appearance dominated dataset. Comparatively, MSOE stream performed surprisingly well on the static camera portion of the dataset, where it even outperformed RGB stream. This result suggests that the MSOE stream is able to capitalize on both dynamics and appearance information in absence of distracting camera motion. In complement, MSOE-two-stream trained on appearance gives the overall best performance and even outperforms previous state-of-the-art on YUP++ [13].

Notably, all networks incur a non-negligible performance decrement in the presence of camera motion, with RGB being strongest in the presence of camera motion and Flow suffering the most. Apparently, the image dynamics resulting from camera motion dominate those from the scene intrinsics and in such cases it is best to concentrate the representation on the appearance.

Conclusions. The evaluation in this section proved the expected benefits of the proposed dataset over reliance on other available large scale datasets that are not necessarily related to the end application (*e.g.* use of action recognition datasets, *i.e.* UCF-101 [33] for pretraining, when the target task is dynamic scene recognition, as done in [13]). More importantly, the benefits and complementarity of the proposed two organizations were clearly demonstrated. Reflecting back on the question posed in the beginning of this section, the results shown here suggest that none of the organizations is better than another in considerations of transfer learning. Instead, they are complementary and can be used judiciously depending on the specifics of the end application.

4.3 Finetuning on DTDB to Establish New State-of-the-art

Experimental Protocol. In this experiment we evaluate the ability of the architectures considered in this study to compete with the state-of-the-art on YUVL for dynamic textures and YUP++ for dynamic scenes when finetuned on DTDB. The the goal is to further emphasize the benefits of DTDB when used to improve on pretrained models. In particular, we use the C3D and two-stream models that were previously pretrained on Sports-1M [18] and ImageNet [29], respectively, then finetune those models using both versions of DTDB. Finetuning details are provided in the supplemental material.

Results. We first consider the results on the YUVL dataset, shown in Table 5 (Left). Here, it is seen that finetuning the pretrained models using either the dynamics or appearance organizations of DTDB improves the results of both C3D and MSOE-two-stream compared to the results in Table 4 (Left). Notably, the boost in performance is especially significant for C3D. This can be largely attributed to the fact that C3D is pretrained on a large video dataset (*i.e.* Sports-1M), while in the original two-stream architecture only the RGB stream is pretrained on ImageNet and the motion stream is trained from scratch. Notably, MSOE-two-stream finetuned on DTDB-dynamics still outperforms C3D and either exceeds or is on-par with previous results on YUVL using SOE-Net.

Turning attention to results obtained on YUP++, summarized in Table 5 (Right), further emphasizes the benefits of finetuning on the proper data. Similar to observations made on YUVL, the boost in performance is once again especially notable on C3D. Importantly, finetuning MSOE-two-stream on DTDB-appearance yields the overall best results and considerably outperforms previous state-of-the-art, which relied on a more complex architecture [13].

Table 5. Performance of spatiotemporal ConvNets, *finetuned* using both organizations of DTDB, **(Left)** on the various breakdowns of the YUVL dataset [8] and **(Right)** on the **S**tatic and **M**oving camera portions of YUP++ and the entire YUP++ [13]

		YUVL-1	YUVL-2	YUVL-3
State-of-the-art	SOE-Net [15]	**95.6**	91.7	**91.0**
Dynamics based fine-tuning	C3D	89.1	90.0	89.5
	MSOE-two-stream	91.1	**92.7**	90.0
Appearance based fine-tuning	C3D	88.8	87.4	85.4
	MSOE-two-stream	90.2	91.2	87.8

		YUP++(S)	YUP++(M)	YUP++
State-of-the-art	T-ResNet [13]	92.4	81.5	89.0
Dynamics based fine-tuning	C3D	89.4	80.8	85.5
	MSOE-two-stream	95.9	84.5	90.4
Appearance based fine-tuning	C3D	90.0	82.7	86.3
	MSOE-two-stream	**97.0**	**87.0**	**91.8**

Interestingly, results of finetuning using either version of DTDB also outperform previously reported results using C3D or two-stream architectures, on both YUVL and YUP++, with sizable margins [15, 13]. Additional one-to-one comparisons are provided in the supplemental material.

Conclusions. The experiments in this section further highlighted the added value of the proposed dual organization of DTDB in two ways. First, on YUVL, finetuning standard architectures led to a notable boost in performance, competitive with or exceeding previous state-of-the-art that relied on SOE-Net, which was specifically hand-crafted for dynamic texture recognition. Hence, an interesting way forward, would be to finetune SOE-Net on DTDB to further benefit this network from the availability of a large scale dynamic texture dataset. Second, on YUP++, it was shown that standard spatiotemporal architectures, trained on the right data, could yield new state-of-the-art results, even while compared to more complex architectures (*e.g.* T-ResNet [13]). Once again, the availability of a dataset like DTDB could allow for even greater improvements using more complex architectures provided with data adapted to the target application.

5 Summary and Discussion

The new DTDB dataset has allowed for a systematic comparison of the learning abilities of broad classes of spatiotemporal ConvNets. In particular, it allowed for an exploration of the abilities of such networks to represent dynamics vs. appearance information. Such a systematic and direct comparison was not possible with previous datasets, as they lacked the necessary complementary organizations. The results especially show the power of two-stream networks that separate appearance and motion at their input for corresponding recognition. Moreover, the introduction of a novel MSOE-based motion stream was shown to improve performance over the traditional optical flow stream. This result has potential for important impact on the field, given the success and popularity of two-stream architectures. Also, it opens up new avenues to explore, *e.g.* using MSOE filtering to design better performing motion streams (and spatiotemporal ConvNets in general) for additional video analysis tasks, *e.g.* action recognition. Still, a learning free ConvNet, SOE-Net, yielded best overall performance on DTDB, which further underlines the room for further development with learning based approaches. An interesting way forward is to train the analytically defined SOE-Net on DTDB and evaluate the potential benefit it can gain from the availability of suitable training data.

From the dataset perspective, DTDB not only has supported experiments that tease apart appearance vs. dynamics, but also shown adequate size and diversity to support transfer learning to related tasks, thereby reaching or exceeding state-of-the-art even while using standard spatiotemporal ConvNets. Moving forward, DTDB can be a valuable tool to further research on spacetime image analysis. For example, training additional state-of-the-art spatiotemporal ConvNets using DTDB can be used to further boost performance on both dynamic texture and scene recognition. Also, the complementarity between the two organizations can be further exploited for attribute-based dynamic scene and texture description. For example, the various categories proposed here can be used as attributes to provide more complete dynamic texture and scene descriptions beyond traditional categorical labels (*e.g.* pluming vs. boiling volcano or turbulent vs. wavy water flow). Finally, DTDB can be used to explore other related areas, including dynamic texture synthesis, dynamic scene segmentation as well as development of video-based recognition algorithms beyond ConvNets.

References

1. Amazon Mechanical Turk: www.mturk.com
2. Beautiful word clouds: www.wordle.net
3. Carreira, J., Zisserman, A.: Quo vadis, action recognition? a new model and the kinetics dataset. In: CVPR (2017)
4. Chetverikov, D., Peteri, R.: A brief survey of dynamic texture description and recognition. In: CORES (2005)
5. Cimpoi, M., Maji, S., Kokkinos, I., Mohamed, S., , Vedaldi, A.: Describing textures in the wild. In: CVPR (2014)
6. Cimpoi, M., Maji, S., Vedaldi, A.: Deep filter banks for texture recognition and segmentation. In: CVPR (2015)
7. Dai, D., Riemenschneider, H., Gool, L.: The synthesizability of texture examples. In: CVPR (2014)
8. Derpanis, K., Wildes, R.P.: Spacetime texture representation and recognition based on spatiotemporal orientation analysis. PAMI **34**, 1193–1205 (2012)
9. Derpanis, K.G., Wildes, R.P.: Dynamic texture recognition based on distributions of spacetime oriented structure. In: Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on. pp. 191–198 (June 2010)
10. Doretto, G., Chiuso, A., Wu, Y., Soatto, S.: Dynamic textures. IJCV **51**, 91–109 (2003)
11. Dubois, S., Peteri, R., Michel, M.: Characterization and recognition of dynamic textures based on the 2D+T curvelet. Sig. Im. & Vid. Proc. **9**, 819–830 (2013)
12. Feichtenhofer, C., Pinz, A., Wildes., R.P.: Spatiotemporal residual networks for video action recognition. In: NIPS (2016)
13. Feichtenhofer, C., Pinz, A., Wildes., R.P.: Temporal residual networks for dynamic scene recognition. In: CVPR (2017)
14. Ghanem, B., Narendra, A.: Max margin distance learning for dynamic texture. In: ECCV (2010)
15. Hadji, I., Wildes, R.P.: A spatiotemporal oriented energy network for dynamic texture recognition. In: ICCV (2017)
16. He, K., Zhang, X., Ren, S., Sun., J.: Deep residual learning for image recognition. In: CVPR (2016)
17. Ji, S., Xu, W., Yang, M., Yu, K.: 3D convolutional neural networks for human action recognition. PAMI **35**, 1915–1929 (2013)
18. Karpathy, A., Toderici, G., Shetty, S., Leung, T., Sukthankar, R., Fei-Fei, L.: Large-scale video classification with convolutional neural networks. In: CVPR (2014)
19. Krizhevsky, A., Sutskever, I., Hinton, G.E.: Imagenet classification with deep convolutional neural networks. In: NIPS (2012)
20. Langer, M., Mann, R.: Optical snow. IJCV **55**, 55–71 (2003)
21. Lin, T.Y., Maji, S.: Visualizing and understanding deep texture representations. In: CVPR (2016)
22. Ng, J., Hausknecht, M., Vijayanarasimhan, S., Vinyals, O., Monga, R., Toderici., G.: Beyond short snippets: Deep networks for video classification. In: CVPR (2015)
23. Oxholm, G., Bariya, P., Nishino, K.: The scale of geometric texture. In: ECCV (2012)
24. Peteri, R., Sandor, F., Huiskes, M.: DynTex: A comprehensive database of dynamic textures. PRL **31**, 1627–1632 (2010)
25. Pond5: www.pond5.com

26. Quan, Y., Bao, C., Ji, H.: Equiangular kernel dicitionary learning with applications to dynamic textures analysis. In: CVPR (2016)
27. Quan, Y., Huang, Y., Ji, H.: Dynamic texture recognition via orthogonal tensor dictionary learning. In: ICCV (2015)
28. Ravichandran, A., Chaudhry, R., R. Vidal, R.: View-invariant dynamic texture recognition using a bag of dynamical systems. In: CVPR (2009)
29. 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. IJCV **115**(3), 211–252 (2015)
30. Saisan, P., Doretto, G., Wu, Y., Soatto, S.: Dynamic texture recognition. In: CVPR (2001)
31. Simonyan, K., Zisserman, A.: Two-stream convolutional networks for action recognition in videos. In: NIPS (2014)
32. Simonyan, K., Zisserman, A.: Very deep convolutional networks for large-scale image recognition. In: ICLR (2015)
33. Soomro, K., Zamir, A.R., Shah, M.: UCF101: A dataset of 101 human actions classes from videos in the wild. Tech. Rep. CRCV-TR-12-01, University of Central Florida (2012)
34. Tran, D., Bourdev, L., Fergus, R., Torresani, L., Paluri, M.: Learning spatiotemporal features with 3D convolutional networks. In: ICCV (2015)
35. Varma, M., Zisserman, A.: Texture classification: Are filter banks necessary? In: CVPR (2003)
36. Varma, M., Zisserman, A.: A statistical approach to texture classification from single images. IJCV **62**, 61–81 (2005)
37. VideoHive: www.videohive.net
38. Yang, F., Xia, G., Liu, G., Zhang, L., Huang, X.: Dynamic texture recognition by aggregating spatial and temporal features via SVMs. Neurocomp. **173**, 1310 – 1321 (2016)
39. YouTube: www.youtube.com
40. Zhao, G., Pietikainen, M.: Dynamic texture recognition using volume local binary patterns. In: ECCV (2006)
41. Zhou, B., Lapedriza, A., Xiao, J., Torralba, A., Oliva, A.: Learning deep features for scene recognition using places database. In: NIPS (2014)