AWSD: Adaptive Weighted Spatiotemporal Distillation for Video Representation

Mohammad Tavakolian
University of Oulu

Hamed R. Tavakoli
Aalto University & Nokia Technologies

Abdenour Hadid
Univesity of Oulu

Abstract

We propose an Adaptive Weighted Spatiotemporal Distillation (AWSD) technique for video representation by encoding the appearance and dynamics of the videos into a single RGB image map. This is obtained by adaptively dividing the videos into small segments and comparing two consecutive segments. This allows using pre-trained models on still images for video classification while successfully capturing the spatiotemporal variations in the videos. The adaptive segment selection enables effective encoding of the essential discriminative information of untrimmed videos. Based on Gaussian Scale Mixture, we compute the weights by extracting the mutual information between two consecutive segments. Unlike pooling-based methods, our AWSD gives more importance to the frames that characterize actions or events thanks to its adaptive segment length selection. We conducted extensive experimental analysis to evaluate the effectiveness of our proposed method and compared our results against those of recent state-of-the-art methods on four benchmark datatsets, including UCF101, HMDB51, ActivityNet v1.3, and Maryland. The obtained results on these benchmark datatsets showed that our method significantly outperforms earlier works and sets the new state-of-the-art performance in video classification. Code is available at the project webpage: https://mohammadt68.github.io/AWSD/

1. Introduction

Video understanding is a challenging task especially for untrimmed videos, where several events happen during one video. In this annals, the preliminary works treated videos as either sequence of still images or volumetric objects, and applied handcrafted local descriptors on a stack of images [33, 38, 32]. With the advent of representation learning and wave of deep neural networks in image understanding tasks, *e.g.* image classification [16], object, scene, and face recognition [41, 9, 26], video understanding with neural networks has been receiving a plethora of attention in recent years [23, 29, 15, 13].

Most of the existing deep models extend convolutional or recurrent neural networks to learn representations on short interval of videos [23, 29]. This strategy limits the application of such models for capturing dynamics of the video because they capture information of a short interval that can lead to loss of critical statistics. Scaling an image-based convolutional neural network (CNN) for videos often adds another dimension of complexity as the number of parameters grows significantly. Note withstanding, training such an architecture requires large volume of training data and computational resources.

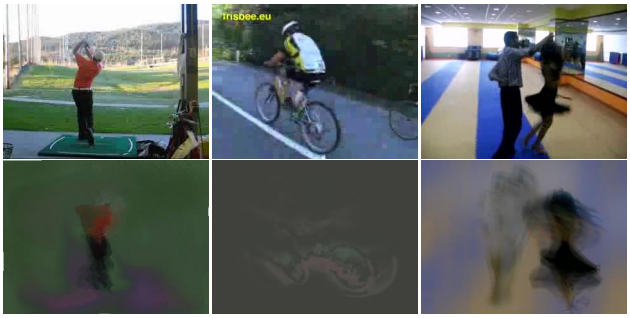

Figure 1: A visualization of Adaptive Weighted Spatiotemporal Distillation (AWSD) applied on RGB frames of videos. Our AWSD captures the appearance and dynamic information of videos and encodes them into one image, which can be used as the input of deep models pre-trained on still images.

Recently, to circumvent the deficiencies caused by processing video on short intervals and/or avoid scaling deep neural networks for temporal processing, a wave of methods has started proposing learning an intermediate representation instead of a video volume prior to using a neural network for obtaining a final neural representation of a video, *e.g.* [2, 37, 35]. A caveat to such approaches is the requirement for learning the intermediate representation, which adversely affects their generalization and efficient handling of untrimmed videos.

We propose Adaptive Weighted Spatiotemporal Distillation (AWSD) for video representation. In contrast to existing approaches, our proposed model is free from learning the intermediate representations and can handle untrimmed videos effectively. The intermediate representation is di-

rectly inferred from the video statistics. AWSD relies on the underlying statistical information of videos to obtain an image map that can be used as input to image-based CNNs. Figure 1 illustrates the visualization of encoded still images using AWSD. The proposed method encodes the statistical information of a video into an image map and successfully handles untrimmed videos.

In a nutshell, our proposed AWSD method offers several benefits: (1) it encodes dynamics and appearance of a video of arbitrary length into a single image map using statistical information of the video, (2) it does not require any training process. In consequence, it is computationally efficient and generalizes to other sequence type easily, (3) the adaptive nature of the method enables it to handle the untrimmed videos effectively. To demonstrate these properties, we extensively conduct experiments on four benchmark video datasets including, UCF101 [24], HMDB51 [17], and ActivityNet v1.3 [3] for action classification, and Maryland [22] for dynamic scene classification. The results of our experiments show that our proposed AWSD is applicable to different video understanding tasks.

2. Related Work

In the early days, videos were treated as a sequence of still images or as a smooth evolution of consecutive frames. By considering a video as a stack of still frames, several spatiotemporal feature extraction methods have been proposed [33, 38, 32]. These methods define a local spatiotemporal neighborhood around each point of interest and a histogram descriptor is extracted to capture the spatial and temporal information. Then, some aggregation approaches generate holistic representation from the local descriptors. Although such handcrafted features are effective for video representation, they lose discriminative capacity in the presence of camera motion and some other variations.

Recently, Convolutional Neural Networks (CNNs) have been employed for video understanding tasks. To capture the appearance and dynamics of the video, CNNs have been extended to temporal domain by adding another dimension. Tran *et al*. [29] studied 3D CNN [15] on realistic (captured in the wild) and large-scale video databases. Their C3D model learns both the spatial and temporal information in a short segment of the video using 3D convolution operations. Carreira *et al*. [5] proposed a two-stream inflated 3D CNN (I3D) by converting vanilla Inception-V1 architecture to a 3D model. They replaced 2D kernels of Inception-V1 [14] to 3D kernel in which the model can use the knowledge of pre-trained 2D model on ImageNet database [21]. Qiu *et al*. [20] developed a Pseudo-3D Residual Network by applying a spatiotemporal factorization on a residual learning module. Diba *et al*. [6] embedded a temporal transition layer in the DenseNet [12] architecture and replaced 2D convolutional filters and pooling layers with their 3D counterparts. In spite the fact that 3D CNN-based architectures perform reasonably well in capturing spatiotemporal information, they usually need a lot of training data to achieve a good representation of the video due to their huge number of parameters.

The aforementioned methods only capture local spatiotemporal information within a small time window. Hence, they are not capable of capturing long-range dynamics. Recently, Wang *et al*. [37] proposed a temporal segment network to model long-range temporal structure of actions within a video. The authors randomly selected snippets of the video and extracted optical flow and RGB differences from frames that are fed to CNN models for feature extraction. This method achieves a global representation of the video using a segmental consensus function to aggregate the information from different snippets of the video. Bilen *et al*. [2] introduced dynamic image by employing a rank pooling technique to capture the temporal evolution of actions and representing the video as one RGB image. They distill the appearance and dynamics of a scene into one single image, which is fed to 2D CNN models for action classification. Pooling techniques consolidate data into compact representations. These techniques also impose equal importance on all frames, which is not favorable. Wang *et al*. [35] proposed SVM pooling for video summarization. They reformulated the pooling problem as a multiple instance learning context and learned useful decision boundaries on the frame level features from each video against background features. Methods in [2, 35] have shown good performances on trimmed videos for action classification. However, their performance on untrimmed videos has not been explored. These methods involve a parameter learning process to achieve video representations for action classification. Hence, they do not generalize efficiently to other video representation tasks. Our proposed AWSD method follows a similar approach to [2, 35] but without requirement of learning any parameter.

3. Video Representation

This section presents our proposed Adaptive Weighted Spatiotemporal Distillation (AWSD). We first discuss the motivations behind Weighted Spatiotemporal Distillation (WSD). Then, we discuss the adaptive temporal window size selection technique, which controls the length of consecutive segments for untrimmed videos.

3.1. Weighted Spatiotemporal Distillation

Visual attention is usually given to the regions that have more descriptive information. Inspired by information theory, the local information of an image can be quantified in terms of sequences of bits [11]. We extend this notion to the temporal dimension in order to capture the discriminative spatiotemporal information. To this end, under a Marko-

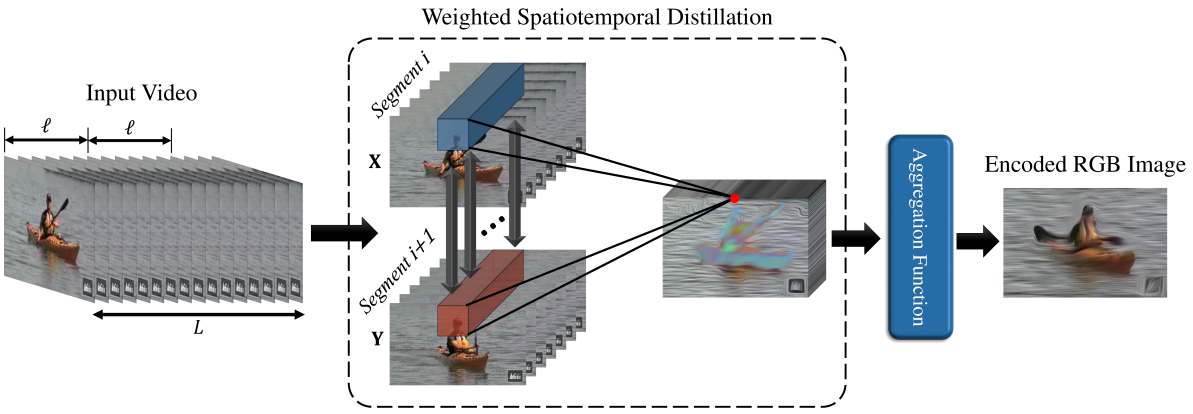

Figure 2: The outline of our proposed WSD for video representation. Given a video of length L, we divide it into segments of smaller length ℓ. By comparing two consecutive segments, WSD generates multiple image maps, which encode appearance and dynamic variations of the scene. We further employ a weighted summation technique for aggregating the obtained image maps into one single RGB frame, which can be used as the input of deep models pre-trained on still images.

vian assumption, we delineate a video as an image map by devising a statistical model for neighboring group of pixels using Gaussian Scale Mixture (GSM).

We aim to capture the mutual information between frames of two time instances to model the variations in dynamics and appearance as a set of weighted points. Hence, we encode the spatiotemporal variations of two consecutive segments of videos into one image map (Figure 2). In our framework, different regions of frames, $\mathbf{x}$, are characterized by a Gaussian noise, $\mathbf{n}_1$, added to a zero-mean Gaussian vector of points intensities $\mathbf{u}$.

$$\mathbf{p} = \mathbf{x} + \mathbf{n}_1 = \alpha\mathbf{u} + \mathbf{n}_1 \tag{1}$$

where α is a mixing multiplier whose value varies over space and time. Intuitively, each region of frames is deformed as the result of spatiotemporal variations $\mathbf{v}$.

$$\mathbf{q} = \mathbf{y} + \mathbf{n}_2 = g\alpha\mathbf{u} + \mathbf{v} + \mathbf{n}_2 \tag{2}$$

where $\mathbf{y}$ represents deformation of region $\mathbf{x}$, g is a gain factor, and $\mathbf{n}_2$ denotes Gaussian noise. In our model, $\mathbf{n}_1$ and $\mathbf{n}_2$ are independent Gaussian noise with covariance matrices $\mathbf{C}_{\mathbf{n}_1} = \mathbf{C}_{\mathbf{n}_2} = \sigma_n^2\mathbf{I}$. The parameter σ_n^2 is the uncertainty of noisy observations. So, we can derive the covariance matrices of $\mathbf{p}$ and $\mathbf{q}$ as:

$$\mathbf{C}_\mathbf{p} = \alpha^2\mathbf{C}_\mathbf{u} + \sigma_n^2\mathbf{I} \tag{3}$$

$$\mathbf{C}_\mathbf{q} = g^2\alpha^2\mathbf{C}_\mathbf{u} + \sigma_v^2\mathbf{I} + \sigma_n^2\mathbf{I} \tag{4}$$

where $\mathbf{C}_\mathbf{u}$ is the covariance matrix of $\mathbf{u}$.

At each point, the information of the reference and deformed frames is obtained by the mutual information $\mathcal{I}(\mathbf{x}|\mathbf{p})$ and $\mathcal{I}(\mathbf{y}|\mathbf{q})$, respectively. We aim to approximate the perceptual information content from both frames. To be specific, we subtract the common information shared between $\mathbf{p}$ and $\mathbf{q}$ from $\mathcal{I}(\mathbf{x}|\mathbf{p})$ and $\mathcal{I}(\mathbf{y}|\mathbf{q})$. So, we define a weight based on the mutual information as:

$$w = \mathcal{I}(\mathbf{x}|\mathbf{p}) + \mathcal{I}(\mathbf{y}|\mathbf{q}) - \mathcal{I}(\mathbf{p}|\mathbf{q}) \tag{5}$$

In Eq. (5), $\mathbf{x}$, $\mathbf{y}$, $\mathbf{p}$, and $\mathbf{q}$ are all Gaussian for a given α. Therefore, the mutual information approximation can be achieved using the determinants of convariances due to the independency of $\mathbf{u}$ and noise $\mathbf{n}_1$ and $\mathbf{n}_2$.

$$w = \frac{1}{2}\log\left[\frac{|\mathbf{C}_{(\mathbf{p},\mathbf{q})}|}{\sigma_n^{4K}}\right] \tag{6}$$

where K is the total number of points in each region of frames and

$$|\mathbf{C}_{(\mathbf{p},\mathbf{q})}| = |\left(\left(\sigma_v^2 + \sigma_n^2\right)\alpha^2 + \sigma_n^2 g^2\alpha^2\right)\mathbf{C}_\mathbf{u} + \sigma_n^2\left(\sigma_v^2 + \sigma_n^2\right)\mathbf{I}| \tag{7}$$

Applying an eigenvalue decomposition to the covariance matrix $\mathbf{C}_\mathbf{u} = \mathbf{O}\Lambda\mathbf{O}^T$, where $\mathbf{O}$ is an orthogonal matrix and Λ is a diagonal matrix with eigenvalues λ_k for $k = 1, \ldots, K$ along its diagonal entries, we can compute $|\mathbf{C}_{(\mathbf{p},\mathbf{q})}|$.

$$|\mathbf{C}_{(\mathbf{p},\mathbf{q})}| = |\mathbf{O}\{\left(\sigma_v^2 + (1+g^2)\sigma_n^2\right)\alpha^2\Lambda + \sigma_n^2(\sigma_v^2 + \sigma_n^2)\mathbf{I}\}\mathbf{O}^T| \tag{8}$$

Due to the orthogonal property of $\mathbf{O}$ and the expression between $\mathbf{O}$ and $\mathbf{O}^T$ in Eq. (8), $|\mathbf{C}_{(\mathbf{p},\mathbf{q})}|$ is obtained as a closed-form equation.

$$|\mathbf{C}_{(\mathbf{p},\mathbf{q})}| = \prod_{k=1}^{K}\{\left(\sigma_v^2 + (1+g^2)\sigma_n^2\right)\alpha^2\lambda_k + \sigma_n^2(\sigma_v^2 + \sigma_n^2)\} \tag{9}$$

Hence, Eq. (6) can be expressed as

$$w = \frac{1}{2}\sum_{k=1}^{K}\log\left(1+\frac{\sigma_v^2}{\sigma_n^2}+\left(\frac{\sigma_v^2}{\sigma_n^4}+\frac{1+g^2}{\sigma_n^2}\right)\alpha^2\lambda_k\right) \tag{10}$$

The obtained weight function shows an interesting connection with the local deformation within frames of video. According to the deformation model in Eq. (2), the variations from $\mathbf{x}$ to $\mathbf{y}$ are characterized by the gain factor g and the random deformation σ_v^2. As g is a scale factor along the frame evolution, it does not cause any changes in the structure of the image. Thus, the structural deformations are captured by σ_v^2. Our weight function increases monotonically with σ_v^2. This demonstrates that more weights are cast to the areas that have larger variations.

We still need to approximate a set of parameters, *i.e.* $\mathbf{C_u}$, α^2, g, and σ_v^2, to use the weight function of Eq. (10). We estimate $\mathbf{C_u}$ as

$$\hat{\mathbf{C}}_\mathbf{u} = \frac{1}{N}\sum_{i=1}^{N}\mathbf{x}_i\mathbf{x}_i^T \tag{11}$$

where N is the number of evaluation widows and $\mathbf{x}_i$ is the i-th neighborhood vector. The multiplier α is spatially varying and can be approximated using a maximum likelihood estimator.

$$\hat{\alpha}^2 = \frac{1}{K}\mathbf{x}^T\mathbf{C}_\mathbf{u}^{-1}\mathbf{x} \tag{12}$$

We can also obtain the deformation parameters g and σ_v^2 by optimizing the following least square regression problem.

$$\hat{g} = \arg\min_g \|\mathbf{y} - g\mathbf{x}\|_2^2 \tag{13}$$

By taking the first-order derivative from Eq. (13), we have:

$$\hat{g} = \frac{\mathbf{x}^T\mathbf{y}}{\mathbf{x}^T\mathbf{x}} \tag{14}$$

Putting this into Eq. (2), we can compute σ_v^2 using $\mathbf{v}^T\mathbf{v}/K$, which results in:

$$\hat{\sigma}_v^2 = \frac{1}{K}\left(\mathbf{y}^T\mathbf{y} - \hat{g}\mathbf{x}^T\mathbf{y}\right) \tag{15}$$

For each color channel, we compute a set of weights by moving a sliding window across frames of two consecutive frames (see Figure 2), where the window covers an $H \times W$ spatial neighborhood at each location. This process results in an image map for each two overlapping segments of the video. Let x_i and y_i be the i-th points in the reference frame $\mathbf{X}$ and the deformed frame $\mathbf{Y}$, respectively. The Mean Square Error (MSE) between two frames is given by

$$\text{MSE} = \frac{1}{P}\sum_{i=1}^{P}(x_i - y_i)^2 \tag{16}$$

where P is the total number of points in the frame. We define a weighted MSE for the corresponding location of the central point in the spatial neighborhood using Eq. (10). Assuming $x_{j,i}$ and $y_{j,i}$ are the i-th points at the j-th frame and $w_{j,i}$ be the weight computed at the corresponding location, we derive Weighted Spatiotemporal Distillation (WSD) as:

$$\text{WSD}(\mathbf{x},\mathbf{y}) = \prod_{j=1}^{\ell}\left(\frac{\sum_i w_{j,i}\left(x_{j,i}-y_{j,i}\right)^2}{\sum_i w_{j,i}}\right) \tag{17}$$

where ℓ is the length of each segment of the video. Repeating this process for all two consecutive segments, we obtain $L/\ell \; - 1$ single images per channel (where L is the video length), which encode the appearance and dynamic variations within the whole video. This distilled information can not be used as the input of pre-trained CNN models due to multiple channels. To tackle this issue, we use a weighted aggregation technique to generate a single RGB image from the obtained $L/\ell \; - 1$ channel image maps. This aggregation technique computes a weighted sum of each point in the images. The weights are calculated as:

$$\beta_i = \frac{\exp(e_i)}{\sum_i \exp(e_i)} \tag{18}$$

where e_i is the i-th point of the image map. Hence, we compute the weighted sum of points for each channel separately to generate the RGB distilled representation of the video.

$$\mathcal{S}_j = \sum_{j=1}^{L/\ell-1}\beta_j e_j \tag{19}$$

where $\mathcal{S}_j$ denotes the j-th point from one channel of the obtained representation. We calculate Eq. (19) for each channel of our data.

3.2. Adaptive Segment Length Selection

The temporal size of segments affects the quality of distilled information and choosing fixed-length equal-size segments can compromise the quality of the RGB output image. Enlarging the window size increases the spatiotemporal information to noise ratio, while decreasing the window size limits WSD computation to only a local, likely irrelevant portion of the video. Hence, determining the optimal length for two consecutive segments is of high importance. Given a video and a window size, there are two factors that affect computing WSD: (1) spatiotemporal variation and (2) disparity variation within the window. Spatiotemporal variation should be large enough relative to noise, which is easily measurable from the input. On the other hand, the disparity variation is difficult to measure because it should be chosen to enhance the comparison of information between two segments. To this end, we propose an adaptive temporal

window size selection method to change the length of segments based on the amount of discriminative information required for optimal representations of videos by exploiting local intensity and disparity patterns.

Let $f_1(t)$ and $f_2(t)$ be two consecutive segments with disparity d.

$$f_1(t) = f(t) + n_1(t) \tag{20}$$
$$f_2(t) = f(t-d) + n_2(t) \tag{21}$$

where $n_1(t)$ and $n_2(t)$ are zero-mean Gaussian noise, *i.e.* $n_1(t), n_2(t) \sim N(0, \sigma^2)$. A direct matching between two segments gives us

$$f_1(t) - f_2(t+d) = n_1(t) - n_2(t+d) \equiv n(t) \tag{22}$$

where $n(t) \sim N(0, 2\sigma^2)$ is Gaussian noise. If d_0 is an initial estimate of the disparity, we can use the Taylor expansion, $f_2(t+d) \approx f_2(t+d_0) + \Delta d f_2'(t+d_0)$, where $\Delta d = d - d_0$.

$$f_1(t) - f_2(t+d_0) - \Delta d f_2'(t+d_0) = n(t) \tag{23}$$

We select N frames with equal interval from the segments, *i.e.* $t_0, t_1, \ldots, t_{N-1}$, and compute the distribution function of $n(t_i)$ for them.

$$\rho(n(t_i)|\Delta d) = \frac{1}{2\sqrt{\pi}\sigma_n} \exp\left(-\frac{(f_1(t) - f_2(t+d_0) - \Delta d f_2'(t+d_0))^2}{4\sigma_n^2}\right) \tag{24}$$

Since $n(t)$ is Gaussian, $n(t_i)$'s are independent from each other. So, they can be expressed as

$$\rho(n(t_0), \ldots, n(t_{N-1})|\Delta d) = \prod_{i=0}^{N-1} \rho(n(t_i)|\Delta d) \tag{25}$$

Due to low variations of $\rho(\Delta d)$, we can approximate the conditional probability of the disparity variations [1] based on the Bayes theorem as

$$\rho(\Delta d|n(t_0), \ldots, n(t_{N-1})) = \frac{\prod_{i=0}^{N-1} \rho(n(t_i)|\Delta d)}{\int_{-\infty}^{\infty} \prod_{i=0}^{N-1} \rho(n(t_i)|\Delta d)\, d(\Delta d)} \tag{26}$$

By substituting Eq. (24) into Eq. (26), we obtain:

$$\rho(\Delta d|n(t_0), \ldots, n(t_{N-1})) = \frac{1}{\sqrt{2\pi}\sigma_{\Delta d}} \exp\left(-\frac{(\Delta d - \Delta^* d)^2}{2\sigma_{\Delta d}^2}\right) \tag{27}$$

where

$$\Delta^* d = \frac{\sum_{i=0}^{N-1} (f_1(t_i) - f_2(t_i + d_0)) f_2'(t_i + d_0)}{\sum_{i=0}^{N-1} (f_2'(t_i + d_0))^2} \tag{28}$$
$$\sigma_{\Delta d}^2 = \frac{2\sigma_n^2}{\sum_{i=0}^{N-1} (f_2'(t_i + d_0))^2} \tag{29}$$

Intuitively, the conditional probability density function of Δd becomes a Gaussian distribution with mean $\Delta^* d$ and variance $\sigma_{\Delta d}^2$. By letting $\Delta t = \frac{\ell}{N}$ be the sampling interval, where ℓ is the size of the window, we multiply the numerator and the denominator of $\Delta^* d$ and $\sigma_{\Delta d}^2$ and sample all frames within a segment, *i.e.* $N \to \infty$.

$$\Delta^* d = \frac{\int_0^{\ell} (f_1(t) - f_2(t + d_0)) f_2'(t + d_0)\, dt}{\int_0^{\ell} (f_2'(t + d_0))^2\, dt} \tag{30}$$
$$\sigma_{\Delta d}^2 = \frac{2\sigma_n^2 \Delta t}{\int_0^{\ell} (f_2'(t + d_0))^2\, dt} \to 0 \tag{31}$$

This implies that the variance of the estimated Δd becomes small by dense sampling. In other words, the intra-segment spatiotemporal variation is proportional to the length of the segment. Hence, we are able to measure the disparity within segments to determine the optimal length for each two consecutive segments. In other words, we consider a direct relationship between the length of the segment and the disparity of information within the segment. We initialize the disparity variations with a small value and compute a correction $\Delta^* d$ and an uncertainty of the correction $\sigma_{\Delta d}^2$ for a segment length ℓ. Repeating this process for different segment lengths enables us to achieve the lowest uncertainty. The value of $\sigma_{\Delta d}^2$ indicates two characteristics. First, the larger absolute value of the first-order derivative makes the uncertainty smaller. Second, the larger is the segment length, the smaller is the uncertainty. The former is intuitive, *i.e.* the more variation in the intensity pattern, the more possibility for existing actions in the scene. The latter characteristic is also understandable, since a large segment can average out the effect of noise. Hence, we update Δd by the amount of $\Delta^* d$.

4. Experiments

We evaluated the performance of our proposed method by conducting extensive experiments on four video classification benchmarks, including, UCF101 [24], HMDB51 [17], ActivityNet v1.3 [3], and Maryland [22]. Table 1 summarizes the content of these datasets.

4.1. Experimental Setup

In our experiments, we used four deep architectures that are pre-trained on ImageNet dataset [21]. We employed

Table 1: Characterstics of the considered datasets for video representation. Three action recognition datasets and one dynamic scene classification dataset are used for evaluating the proposed method.

Dataset	Videos	Classes	Catergory	Trimmed/ Untrimmed
UCF101 [24]	13,320	101	Human Actions	Trimmed
HMDB51 [17]	6,766	51	Human Actions	Trimmed
ActivityNet [3]	19,994	200	Human Actions	Untrimmed
Maryland [22]	130	10	Dynamic Scenes	Trimmed

Tensorflow implementations of AlexNet [16], Inception-V1 [14], ResNet-50 [10], and ResNet-101 [10]. All of these deep models are fine-tuned by using stochastic gradient descent with the momentum of 0.9 and an annealed learning rate, starting from 3×10^{-3} and multiplied by a factor of 0.2 per epoch. During training, we randomly performed size jittering, cropping, flipping, and rescaling on images. We also applied our AWSD on optical flow data. For the computation of the optical flow, we used TLV1 optical flow algorithm [40], which is implemented in OpenCV with CUDA.

4.2. WSD vs. AWSD

In this section, to further explain the importance for AWSD, we analyzed the effect of segment's length on the performance of WSD. Then, we made a comparative study between WSD and AWSD. The segment's length determines the amount of information that WSD summarizes. First, we considered WSD and analyzed its performance by varying the length of video segment from 10 to 60 frames, *i.e*. a video is divided into fixed-length non-overlapping segments.

Figure 3 shows the results of our experiments using both trimmed and untrimmed videos. As depicted, for trimmed videos, the classification accuracy increases as the number of frames per segment increases. However, after a certain number of frames, we do not observe any significant improvement in the accuracy. This means that small segments are sufficient to achieve good performance using WSD on trimmed videos. In contrast, for the untrimmed videos (ActivityNet), the accuracy drops with the increase of the segment's length from 57.8% to 28.1%, The deterioration in performance is likely due to capturing too much content within segments, which fades the discriminative information in the distilled image maps.

We also compared the effect of adaptive selection of video segments. Table 2 reports the results, indicating a significant improvement using adaptive segment length selection technique, *i.e*. 9.6%, 7.7%, 32.3%, and 23.9% improvement in UCF101 [24], HMDB51 [17], ActivityNet [3], and Maryland [22] datasets, respectively. This improvement is more significant in ActivityNet and Maryland datasets, where there are complex dynamic scene and/or untrimmed action videos.

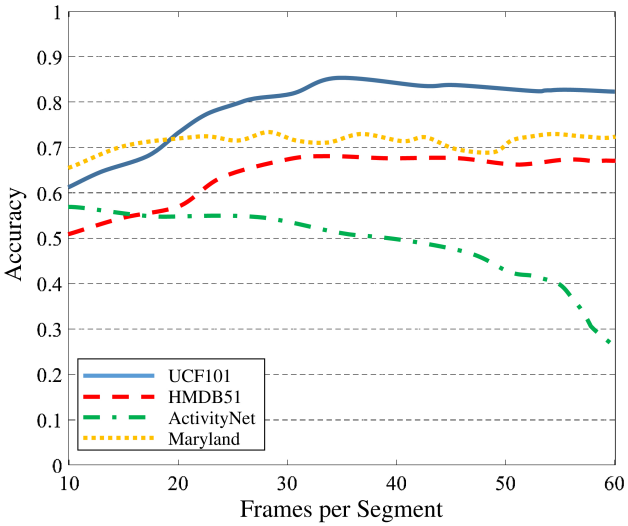

Figure 3: Accuracy of the proposed WSD when changing the segment length using ResNet-50. The performance drops significantly on untrimmed videos.

Table 2: The accuracy (%) of the proposed method with and without using adaptive segment length selection. Maximum accuracy is reported for each dataset using the fixed-length segments.

	UCF101	HMDB51	ActivityNet	Maryland
WSD	86.4	67.9	58.1	73.6
AWSD	96.0	75.6	90.4	97.5

Table 3: The accuracy (%) of the proposed method using different aggregation functions with ResNet-50 [10]. The weighted sum of image maps achieves the highest accuracy compared to the average and max aggregation functions.

	UCF101	HMDB51	ActivityNet	Maryland
Average	91.5	69.7	83.9	87.7
Max	94.8	72.1	85.6	92.4
Weighted Sum	96.0	75.6	90.4	97.5

4.3. Analysis of Aggregation Functions

We evaluated the performance of our proposed AWSD employing different aggregation functions for converting the calculated image maps into a three channel image map. Table 3 summarizes the results of average, max, and weighted sum aggregation. The weighted sum gives higher importance to the regions that effectively represent the events in the scene.

Table 4: The accuracy (%) of our proposed method versus the accuracy of representations from optical flow and RGB data using ResNet-50 [10].

	UCF101	HMDB51	ActivityNet	Maryland
RGB	89.3	69.7	83.9	87.7
OF	90.1	70.2	67.3	89.0
AWSD (OF)	94.8	72.1	85.6	92.4
AWSD (RGB)	96.0	75.6	90.4	97.5

Table 5: AWSD performance analysis (%) using different 2D CNN models pre-trained on ImageNet dataset [21].

	UCF101	HMDB51	ActivityNet	Maryland
AlexNet [16]	91.2	69.8	81.6	90.8
Inception-V1 [14]	95.3	73.0	88.5	94.3
ResNet-50 [10]	96.0	75.6	90.4	97.5
ResNet-101 [10]	97.6	79.3	93.1	98.1

4.4. Comparisons against Frame-Based Baseline Models

We compared our proposed AWSD against two frame-based baseline models. The first model operates on RGB images and the second model works on optical flow. We employed ReseNet-50 on RGB frames along with the image maps obtained from applying AWSD on the optical flow and video sequence data.The results are summarized in Table 4. The proposed AWSD has 5.1% improvement over optical flow, indicating the superiority of AWSD in capturing appearance and dynamic of information from video data.

4.5. AWSD and Deep Architectures

We employed four deep architectures that are pre-trained on ImageNet dataset [21] (namely AlexNet [16], Inception-V1 [14], ResNet-50 [10], and ResNet-101 [10]) to the representation obtained by AWSD in order to investigate the usefulness of various 2D CNN models. The results are summarized in Table 5. As expected, the best performances are given by deeper networks. It is notwithstanding that given the significant reduction in the large amount of annotated videos for training a 3D model, AWSD is an effective alternative for enabling usage of 2D CNNs.

4.6. Cross Databases Analysis

To show the generalization of distilled representations obtained by our AWSD, we conducted cross-dataset experiments. In these experiments, we fine-tuned a pre-trained ResNet-50 [10] on ImageNet [21] using the distilled representations of training samples from one dataset and used the distilled representations of the test set of other datasets. Table 6 summarizes the results. Although the accuracy drops in the cross-dataset setting, the AWSD method still shows a good performance, demonstrating the high capability to represent video sequences discriminatively.

Table 6: The accuracy (%) of our proposed ASWD method in the cross-dataset experimental setting using ResNet-50 [10]. The model is fine-tuned in one dataset and is tested on another dataset.

		Test on		
		UCF101	HMDB51	ActivityNet
Train on	UCF101	**96.0**	71.0	85.6
	HMDB51	88.6	**75.6**	82.7
	ActivityNet	91.3	72.5	**90.4**

4.7. Comparison against the State-of-the-Art

We compared AWSD against the state-of-the-art methods on four video-based benchmarks. Table 7 summarizes the comparative results on UCF101 [24] and HMDB51 [17] datasets. We compared our method with both traditional methods, such as Improved Dense Trajectory (iDT) [34] and MoFAP [36], and deep learning based methods, such as 3D Convolutional Neural Networks (C3D) [29], Temporal Segment Network (TSN) [37], and Long Term Convolutional Network (LTC) [31]. Among the compared methods, Dynamic Image (DI) [2] and SVM Pooled descriptor (SVMP) [35] are the closest to our work. We encoded the RGB frames using our AWSD. The obtained representations are fed to ResNet-50 and ResNet-101. From Table 7, we observe that DI achieves 95.5% and 72.5% using a four stream network (still images, dynamic images, optical flow, and dynamic optical flow) and ResNext-101 on UCF101 and HMDB51, respectively. Our method improves DI by 0.5% and 4.3% using ResNet-50 on UCF101 and HMDB51, respectively. It is worthnoting that ResNet-50 has a relatively lower performance in comparison to ResNext-101 on images.

We also compared the performance of our method on untrimmed videos of ActivityNet v1.3 [3]. We ran the original implementation of algorithms and reported their best performance. The results are summarized in Table 8. As depicted, our proposed AWSD achieves 97.6% mAP accuracy using ResNet-101 and improves the highest performance by 4.9%. The improvement on untrimmed videos stresses the efficient selection of temporal length for capturing the spatiotemporal information by AWSD.

We further employed AWSD to dynamic scene classification on Maryland dataset [22] and draw a comparison between our proposed method and the state-of-the-arts. Table 9 reports the results, showing that AWSD achieves 97.5% and 98.1% classification accuracy using ResNet-50 and ResNet-101, respectively. For example, our method using ResNet-101 outperforms LSTF [13] by 3.1%.

Figure 4 shows a visualization of our AWSD and

Table 7: Comparison of classification accuracy (%) of the proposed approach against those of state-of-the-art methods on UCF101 [24] and HMDB51 [17] datasets.

Method	UCF101	HMDB51
iDT+FV [34]	85.9	57.2
DT+MVSV [4]	83.5	55.9
iDT+HSV [18]	87.9	61.1
MoFAP [36]	88.3	61.7
Two-Stream [23]	88.0	59.4
C3D (3 nets) [29]	85.2	51.6
Res3D [30]	95.6	54.9
I3D [5]	95.6	74.8
FSTCN [25]	88.1	59.1
LTC [31]	91.7	64.8
KVMF [42]	93.1	63.3
TSN (7 seg) [37]	94.9	71.0
DI (4 stream) [2]	95.5	72.5
SVMP [35]	-	71.0
S3D-G [39]	96.8	75.9
AWSD (ResNet-50)	**96.0**	**75.6**
AWSD (ResNet-101)	**97.6**	**79.3**

Table 8: Comparison of our proposed method's performance in terms of classification accuracy (%) against the state-of-the-art methods on ActivityNet dataset [3].

Method	Accuracy (mAP)
iDF+FV [34]	64.3
Two-Stream [23]	69.1
C3D [29]	73.4
Res3D [30]	74.3
I3D [5]	88.6
TSN (7 seg) [37]	86.5
DI (4 stream) [2]	88.2
SVMP [35]	86.9
AWSD (ResNet-50)	**90.4**
AWSD (ResNet-101)	**93.1**

Dynamic Image [2] intermediate representations for untrimmed videos. The comparison of the two visualizations suggests that AWSD captures the essence of action dynamics in more details and preserves the appearance information better. We assert that this efficient representation is due to adaptive selection of segment length, which is crucial for untrimmed videos. From Figure 4, it is also obvious that the information of different parts of the video are mixed for DI, while AWSD seems to encode more discriminative information.

5. Conclusion

We presented an Adaptive Weighted Spatiotemporal Distillation (AWSD) for capturing and encoding the appearance and dynamics of the video into one single image, which can be processed by deep models pre-trained on still images. This technique tackles the problem of tuning huge number of parameters in deep models for videos. Our AWSD divides the given video into smaller segments and compares two consecutive segments to generate multiple image maps. The obtained image maps are then aggregated to produce one single RGB image. The adaptive nature of video segment selection enables the method to efficiently capture the information of untrimmed videos. Moreover, representing videos by AWSD is easy and straightforward as it does not involve any parameter learning nor optimization process. We evaluated the effectiveness of the proposed method on four benchmark datasts, namely UCF101, HMDB51, ActivityNet v1.3, and Maryland, for video representation and classification. The experimental results demonstrated the superior performance of our proposed method.

Acknowledgements The support of Academy of Finland, Infotech Oulu, Nokia, Tauno Tönning, and KAUTE Foundations is acknowledged.

Table 9: Comparison of the accuracy (%) of our proposed method against the state-of-the-art for dynamic scene classification on Maryland dataset [22].

Method	Accuracy
CSO [7]	67.7
SFA [28]	60.0
SOE [8]	43.1
BoSE [8]	77.7
C3D [29]	87.7
st-TCoF [19]	88.4
DDM+SCSP [27]	90.3
LSTF [13]	95.0
DI (4 stream) [2]	92.5
SVMP [35]	90.1
AWSD (ResNet-50)	**97.5**
AWSD (ResNet-101)	**98.1**

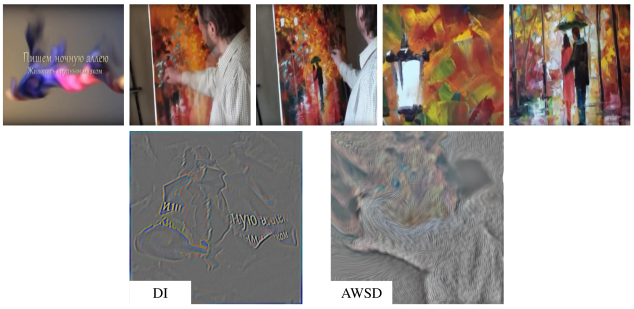

Figure 4: A visualization of AWSD applied on RGB frames of an untrimmed video. **Top:** sample frames from different time instances of the untrimmed video. **Bottom:** distilled information of the video in RGB format. Our representation captures more details of the scene and contains the gist of the video.

References

[1] James O Berger. *Statistical decision theory and Bayesian analysis*. Springer-Verlag, 1985. 5

[2] Hakan Bilen, Basura Fernando, Efstratios Gavves, Andrea Vedaldi, and Stephen Gould. Dynamic image networks for action recognition. In *CVPR*, pages 3034–3042, 2016. 1, 2, 7, 8

[3] Fabian Caba Heilbron, Victor Escorcia, Bernard Ghanem, and Juan Carlos Niebles. Activitynet: A large-scale video benchmark for human activity understanding. In *CVPR*, pages 961–970, 2015. 2, 5, 6, 7, 8

[4] Zhuowei Cai, Limin Wang, Xiaojiang Peng, and Yu Qiao. Multi-view super vector for action recognition. In *CVPR*, pages 596–603, 2014. 8

[5] Joao Carreira and Andrew Zisserman. Quo vadis, action recognition? a new model and the kinetics dataset. In *CVPR*, pages 6299–6308, 2017. 2, 8

[6] Ali Diba, Mohsen Fayyaz, Vivek Sharma, A Hossein Karami, M Mahdi Arzani, Rahman Yousefzadeh, and Luc Van Gool. Temporal 3d convnets using temporal transition layer. In *CVPR Workshops*, pages 1117–1121, 2018. 2

[7] Christoph Feichtenhofer, Axel Pinz, and Richard P Wildes. Spacetime forests with complementary features for dynamic scene recognition. In *BMVC*, 2013. 8

[8] Christoph Feichtenhofer, Axel Pinz, and Richard P Wildes. Bags of spacetime energies for dynamic scene recognition. In *CVPR*, pages 2681–2688, 2014. 8

[9] Ross Girshick, Jeff Donahue, Trevor Darrell, and Jitendra Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. In *CVPR*, pages 580–587, 2014. 1

[10] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, pages 770–778, 2016. 6, 7

[11] Sayed Hossein Khatoonabadi, Nuno Vasconcelos, Ivan V Bajic, and Yufeng Shan. How many bits does it take for a stimulus to be salient? In *CVPR*, pages 5501–5510, 2015. 2

[12] Gao Huang, Zhuang Liu, Laurens Van Der Maaten, and Kilian Q Weinberger. Densely connected convolutional networks. In *CVPR*, pages 4700–4708, 2017. 2

[13] Yuanjun Huang, Xianbin Cao, Qi Wang, Baochang Zhang, Xiantong Zhen, and Xuelong Li. Long-short term features for dynamic scene classification. *IEEE Trans. Circuits and Systems for Video Technology*, 2018. 1, 7, 8

[14] Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. *arXiv preprint arXiv:1502.03167*, 2015. 2, 6, 7

[15] Shuiwang Ji, Wei Xu, Ming Yang, and Kai Yu. 3D convolutional neural networks for human action recognition. *IEEE Trans. PAMI*, 35(1):221–231, 2013. 1, 2

[16] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In *NIPS*, pages 1097–1105, 2012. 1, 6, 7

[17] Hildegard Kuehne, Hueihan Jhuang, Estíbaliz Garrote, Tomaso Poggio, and Thomas Serre. HMDB: a large video database for human motion recognition. In *ICCV*, pages 2556–2563, 2011. 2, 5, 6, 7, 8

[18] Xiaojiang Peng, Limin Wang, Xingxing Wang, and Yu Qiao. Bag of visual words and fusion methods for action recognition: Comprehensive study and good practice. *CVIU*, 150:109–125, 2016. 8

[19] Xianbiao Qi, Chun-Guang Li, Guoying Zhao, Xiaopeng Hong, and Matti Pietikäinen. Dynamic texture and scene classification by transferring deep image features. *Neurocomputing*, 171:1230–1241, 2016. 8

[20] Zhaofan Qiu, Ting Yao, and Tao Mei. Learning spatio-temporal representation with pseudo-3d residual networks. In *CVPR*, pages 5533–5541, 2017. 2

[21] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, et al. Imagenet large scale visual recognition challenge. *IJCV*, 115(3):211–252, 2015. 2, 5, 7

[22] Nitesh Shroff, Pavan Turaga, and Rama Chellappa. Moving vistas: Exploiting motion for describing scenes. In *CVPR*, pages 1911–1918, 2010. 2, 5, 6, 7, 8

[23] Karen Simonyan and Andrew Zisserman. Two-stream convolutional networks for action recognition in videos. In *NIPS*, pages 568–576, 2014. 1, 8

[24] Khurram Soomro, Amir Roshan Zamir, and Mubarak Shah. UCF101: A dataset of 101 human actions classes from videos in the wild. *arXiv preprint arXiv:1212.0402*, 2012. 2, 5, 6, 7, 8

[25] Lin Sun, Kui Jia, Dit-Yan Yeung, and Bertram E Shi. Human action recognition using factorized spatio-temporal convolutional networks. In *ICCV*, pages 4597–4605, 2015. 8

[26] Yaniv Taigman, Ming Yang, Marc'Aurelio Ranzato, and Lior Wolf. Deepface: Closing the gap to human-level performance in face verification. In *CVPR*, pages 1701–1708, 2014. 1

[27] Mohammad Tavakolian and Abdenour Hadid. Deep discriminative model for video classification. In *ECCV*, pages 382–398, 2018. 8

[28] Christian Theriault, Nicolas Thome, and Matthieu Cord. Dynamic scene classification: Learning motion descriptors with slow features analysis. In *CVPR*, pages 2603–2610, 2013. 8

[29] Du Tran, Lubomir Bourdev, Rob Fergus, Lorenzo Torresani, and Manohar Paluri. Learning spatiotemporal features with 3d convolutional networks. In *ICCV*, pages 4489–4497, 2015. 1, 2, 7, 8

[30] Du Tran, Jamie Ray, Zheng Shou, Shih-Fu Chang, and Manohar Paluri. Convnet architecture search for spatiotemporal feature learning. *arXiv preprint arXiv:1708.05038*, 2017. 8

[31] Gül Varol, Ivan Laptev, and Cordelia Schmid. Long-term temporal convolutions for action recognition. *IEEE Trans. PAMI*, 40(6):1510–1517, 2018. 7, 8

[32] Heng Wang, Alexander Kläser, Cordelia Schmid, and Liu Cheng-Lin. Action recognition by dense trajectories. In *CVPR*, pages 3169–3176, 2011. 1, 2

[33] Heng Wang and Cordelia Schmid. Action recognition with improved trajectories. In *CVPR*, pages 3551–3558, 2013. 1, 2

[34] Heng Wang and Cordelia Schmid. Action recognition with improved trajectories. In *ICCV*, pages 3551–3558, 2013. 7, 8

[35] Jue Wang, Anoop Cherian, Fatih Porikli, and Stephen Gould. Video representation learning using discriminative pooling. In *CVPR*, pages 1149–1158, 2018. 1, 2, 7, 8

[36] Limin Wang, Yu Qiao, and Xiaoou Tang. MoFAP: A multi-level representation for action recognition. *IJCV*, 119(3):254–271, 2016. 7, 8

[37] Limin Wang, Yuanjun Xiong, Zhe Wang, Yu Qiao, Dahua Lin, Xiaoou Tang, and Luc Van Gool. Temporal segment networks for action recognition in videos. *IEEE Trans. PAMI*, 2018. 1, 2, 7, 8

[38] Geert Willems, Tinne Tuytelaars, and Luc Van Gool. An efficient dense and scale-invariant spatio-temporal interest point detector. In *ECCV*, pages 650–663, 2008. 1, 2

[39] Saining Xie, Chen Sun, Jonathan Huang, Zhuowen Tu, and Kevin Murphy. Rethinking spatiotemporal feature learning: Speed-accuracy trade-offs in video classification. In *ECCV*, pages 305–321, 2018. 8

[40] Christopher Zach, Thomas Pock, and Horst Bischof. A duality based approach for realtime tv-l 1 optical flow. In *Joint pattern recognition symposium*, pages 214–223, 2007. 6

[41] Bolei Zhou, Agata Lapedriza, Jianxiong Xiao, Antonio Torralba, and Aude Oliva. Learning deep features for scene recognition using places database. In *NIPS*, pages 487–495, 2014. 1

[42] Wangjiang Zhu, Jie Hu, Gang Sun, Xudong Cao, and Yu Qiao. A key volume mining deep framework for action recognition. In *CVPR*, pages 1991–1999, 2016. 8