Human Action Recognition Using Convolutional Neural Networks with Symmetric Time Extension of Visual Rhythms

Hemerson Tacon[1], André S. Brito[1], Hugo L. Chaves[1], Marcelo Bernardes Vieira[1] ✉, Saulo Moraes Villela[1], Helena de Almeida Maia[2], Darwin Ttito Concha[2], and Helio Pedrini[2] ⋆

[1] Department of Computer Science, Universidade Federal de Juiz de Fora, Juiz de Fora, Brazil
{hemerson, andre.brito, hugo.chaves}@ice.ufjf.br
{marcelo.bernardes, saulo.moraes}@ufjf.edu.br
[2] Institute of Computing, University of Campinas, Campinas, Brazil
{helena.maia, darwin.ttito}@liv.ic.unicamp.br
helio@ic.unicamp.br

Abstract. Despite the expressive progress of deep learning models on the image classification task, they still need enhancement for efficient human action recognition. One way to achieve such gain is to augment the existing datasets. With this goal, we propose the usage of multiple Visual Rhythm crops, symmetrically extended in time and separated by a fixed stride. The symmetric extension preserves the video frame rate, which is crucial to not distort actions. The crops provide a 2D representation of the video volume matching the fixed input size of the 2D Convolutional Neural Network (CNN) employed. In addition, multiple crops with stride guarantee coverage of the entire video. Aiming to evaluate our method, a multi-stream strategy combining RGB and Optical Flow information is extended to include the Visual Rhythm. Accuracy rates fairly close to the state-of-the-art were obtained from the experiments with our method on the challenging UCF101 and HMDB51 datasets.

Keywords: Deep Learning · Action Recognition · Data Augmentation · Video Analysis · Visual Rhythm.

1 Introduction

In the last years, much progress has been made in the field of image classification. This success is the result of the combination of large image datasets, such as ImageNet [1], and the creation of new CNN approaches [2,3]. A natural consequence of this success was the exploitation of these advances in the field of

⋆ All authors thank CAPES, FAPEMIG (grant CEX-APQ-01744-15), FAPESP (grants #2017/09160-1 and #2017/12646-3), CNPq (grant #305169/2015-7) for the financial support, and NVIDIA for the grant of two GPUs (GPU Grant Program).

video classification. In this domain, one problem consists in recognizing the main action executed by a person along a video. A solution to this problem is crucial to automate many tasks and it has outstanding applications: video retrieval, intelligent surveillance and autonomous driving [4–6]. This specific problem is called human action recognition and it is the subject of this paper.

The time dimension that is presented in videos produces a significant data increase if compared to images. Although some works have used 3D CNNs [5, 7], the additional data of time dimension makes it prohibitive to use them without any previous polling step [8, 9]. Most of recent works have used 2D CNNs for action recognition and this choice requires a video volume representation in a 2D space [9–11]. Such representation also needs to match the input size of the employed neural network which is commonly fixed. Another problem related to the data is the lack of massive labeled datasets. The existing ones [12, 13] tend to be poorly annotated [6]. A workaround is to augment some well established datasets [14, 15]. However, once their video lengths vary between samples, the time dimension manipulation is not simple and special cautions are required when performing the augmentation. For instance, keeping the original video frame rate is critical for the action recognition problem. Any variation in the frame rate could alter the action speed and distort it. When classifying a video with "walking" action, for example, this could be easily confused with the "running" action if a video with the first action had its frame rate increased compared to a video containing the second action.

In previous works, the usage of Visual Rhythms (VRs) [16–19] was proposed to address the issues imposed by time dimension handling. The VR is a 2D video representation with combined 1D RGB information varying in time. In this work, we propose a data augmentation for the VR by extending it symmetrically in time. This augmentation is an improvement of our previous work [19]. It is assumed that most actions presented from back to front in time can be properly classified. Furthermore, abrupt brightness changes are not introduced such as the periodic extension used in [19] . The symmetric extension in time also allows the extraction of multiple VR crops without deformations in frame rate. In addition, the crop dimensions can also be set to match any required input size of the employed neural network. All of these characteristics together make the symmetric extension a proper method to augment video datasets.

Our experiments are performed on two well-known challenging datasets, HMDB51 [15] and UCF101 [14]. A modified version of the widely known InceptionV3 network [2] is used. When combined with other features in a multi-stream architecture, the VR provides complementary information, which is crucial to achieve accuracy rates close to state-of-the-art methods. It is used the multi-stream architecture presented in [19]. This architecture takes RGB, Optical Flow and symmetrically extended VR images as input. It is empirically showed that symmetrically extended VRs can improve the final classification accuracy.

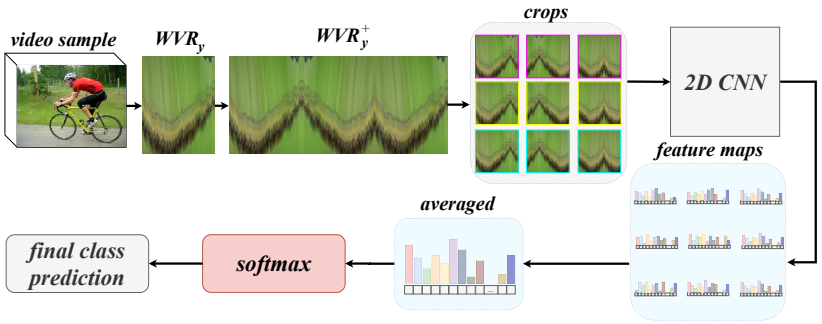


Fig. 1: Overview of the Visual Rhythm stream. After symmetric extension, n_w crops apart from each other by a stride s are extracted in the center (yellow). Depending on the dataset, extra crops aligned with the image top (magenta) and bottom (cyan) are extracted. All crops are applied to the CNN. The resulting features are averaged and the final class is predicted through a softmax layer.

2 Related Work

The VR is a spatio-temporal slice of a video, i.e., a predefined set of pixels forming an arbitrary 2D surface embedded in a 3D volume of a video. Despite it has been first employed to detect camera transitions (cut, wipe and dissolve) in videos [16, 17], the term VR was just mentioned a couple of years later by Kim et al. [20]. The first employment of VRs in the human action recognition problem was accomplished by Torres and Pedrini [21]. They utilized high-pass filters to obtain regions of interest (ROI) in VRs of videos. It is argued that the action patterns are present in only some parts of the VR.

By extracting the VR from videos, we attempt to reduce the human action recognition problem to image classification. There are highly successful convolutional neural networks [2, 3] for this problem. Aiming to take advantage of such CNNs, many works have proposed to combine distinct 2D representations of the videos. The RGB information is a basic feature for this purpose. But even multiple image frames are not able to capture movement correlations along time and fail to distinguish similar actions [22]. In order to complement RGB based CNNs, many works have employed Optical Flow sequences as temporal features to supply the correlations along time [23–25]. Thus, a two-stream model was proposed to exploit and merge these two features [26, 27]. This method showed to be successful and other extensions emerged combining more than two streams [9–11].

Despite the success of multi-stream methods, they do not allow communication between streams [23, 25–27]. This lack of interaction hinders the models from learning spatio-temporal features [6]. An attempt to address this problem was proposed by Feichtenhofer et al. [10] with an architecture that provided a multiplicative interaction between spatial and temporal features. Another way to address this issue is to merge the spatial and temporal information into a single feature. To represent such spatio-temporal features, it is necessary to ap-

ply a pooling method to the video. Wang et al. [9] propose a pooling descriptor, based on SVM, to obtain a compact video representation. The pooling scheme is coupled into a CNN model and trained end-to-end. Similarly, the VR is also a kind of spatio-temporal feature. In the VR, a spatial dimension (X or Y axis) and the temporal dimension are aggregated into a 2D feature.

As shown in our previous work [19], the VR combined with a multi-stream model makes it possible to explore time and space interactions in videos to improve action recognition. The main interest was to introduce the VR as a spatio-temporal feature and show its contribution to a well-known architecture. A contribution was a method to detect the better direction (horizontal or vertical) to extract the VR. The criterion was to use the VR of the direction with more movement. Typical data augmentation techniques were also applied to the VRs. Such techniques significantly increased the classification accuracy. Motivated by this, we propose the use of multiple VR crops, symmetrically extended and separated by a fixed stride. This method consists of a proper data augmentation for the VR. Furthermore, some parameters related to the VR are explored aiming to extract more relevant information from video sequences.

3 Proposed Method

An overview of the VR stream is depicted in Figure 1. It consists of a classification protocol using a version of the InceptionV3 network with VRs. A VR is computed for each video and its data augmentation is driven by symmetric extension. Multiple crops with fixed stride are extracted from the symmetric extension. The final class prediction is the averaged prediction of all crops.

3.1 Visual Rhythm

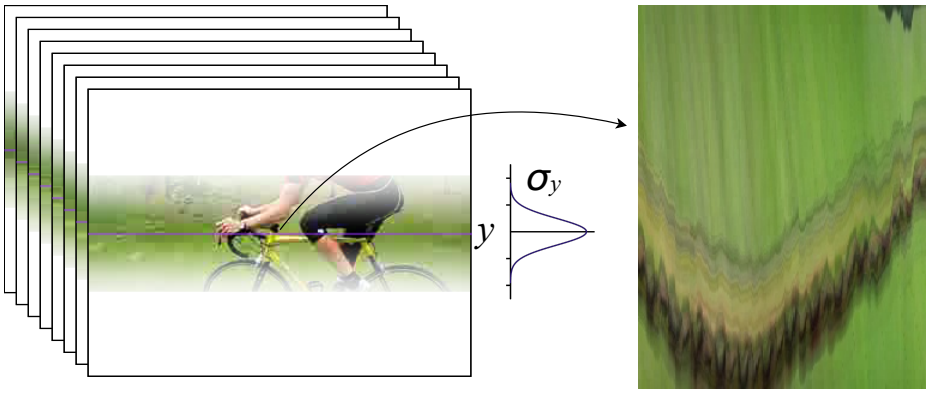


Fig. 2: Horizontal weighted rhythm: y is the middle row in this example.

In most cases, the trajectory that is formed by the points in P is compact and thus the VR represents a 2D subspace embedded in video volume XYT. For instance, if P is the set of points of a single frame row, the resulting VR is a plane parallel to XT. Analogously, setting P as a single frame column results in a plane that is parallel to YT.

The proposal of [18, 19], that takes the mean VR formed by all rows, was adapted. The reason is that the underlying moving object in a video is more likely to be observed far from the frame borders. By weighting the VRs far to the main object's location as the closest ones, one might hinder the motion representation. Instead, we propose to weight less as the VRs get farther from a reference row or column. Let $P_r = \{(r,1),(r,2),\cdots,(r,w)\}$ be the set of points forming the row r. We define the horizontal weighted VR as:

$$\mathrm{WVR}_y = \sum_{r=1}^{h} VR_{P_r} \cdot g(r-y,\sigma_y) \cdot \left[\sum_{r=1}^{h} g(r-y,\sigma_y)\right]^{-1} \tag{1}$$

where y is the reference row of the horizontal VR, and $g(s,\sigma) = e^{-\frac{s^2}{\sigma^2}}$ is the weighting function that decays as the other VRs get farther from the reference y. Thus, the horizontal VR used in this work is defined by two parameters: the reference row y and standard-deviation σ_y. Figure 2 depicts a video of the *Biking* class of UCF101 (240 frames with 320×240 pixels), forming a VR of 320×240 elements. In practice, an interval $y \pm d_y$, is defined from σ_y such that outer rows have zero weight. In practice, to make the parameter y invariant to video height h, we define a factor f_y such that $y = \alpha_y \cdot h$.

3.2 Symmetric Extension with Fixed Stride Crops

The symmetric extension of a VR, named WVR_y, is

$$\mathrm{WVR}_y^+(i,k) = \begin{cases} \mathrm{WVR}_y(i, f-m), & \text{for } \lfloor k/f \rfloor \text{ odd} \\ \mathrm{WVR}_y(i, m+1), & \text{otherwise} \end{cases} \tag{2}$$

where $1 \leq i \leq w$, m is the remainder of the integer division of k by f and $k \in \mathbb{Z}$. Thus, the WVR is composed of several copies of the VR concatenated several times along the temporal dimension with the even occurrences being horizontally flipped. Figure 3 shows a video of the *Biking* class of UCF101 (Figure 2) extended three times. The premise is as follows: the action performed backwards in time also represents the class and can be used to reinforce the NN training.

The VR is extracted from each video in the dataset. Our proposal is to use multiple crops from each extended VR as a data augmentation process. Each crop is formed by the image constrained in a $w_{CNN} \times h_{CNN}$ window (matching the CNN's input). A crop with lower left coordinates x and t is defined as:

$$\mathrm{C}_{xt}^+(a,b) = \mathrm{WVR}_y^+(x+a, t+b), \tag{3}$$

with $x \leq a < x+h_{CNN}$ and $t \leq b < t+w_{CNN}$. The VR is extended symmetrically until n_w crops are extracted using a stride s, i.e., the first crop is taken at $t = 0$ and all subsequent $n_w - 1$ crops are taken s frames ahead the previous one. The resulting set of crops for a fixed row x is $\{C^+_{xt} \mid t = js\}$, for $j \in \{0, 1, ..., n_w - 1\}$.

If h_{CNN} is smaller than w, i.e., the video frame width is greater than the corresponding dimension of the CNN, the crops are centered in X as depicted in Figure 3. This approach assumes that the main action motion is mostly performed in this region. Notice that the top and bottom sides are not reached by the crops. In order to include these regions, extra n_w crops keeping the stride s from each other are obtained, aligned with the top and bottom borders. Thus, up to $3 \cdot n_w$ crops can be obtained depending on the application. This is useful to get all information in X and for most videos reinforce the central information. The mean and standard-deviation are computed to normalize each RGB channel of all crops.

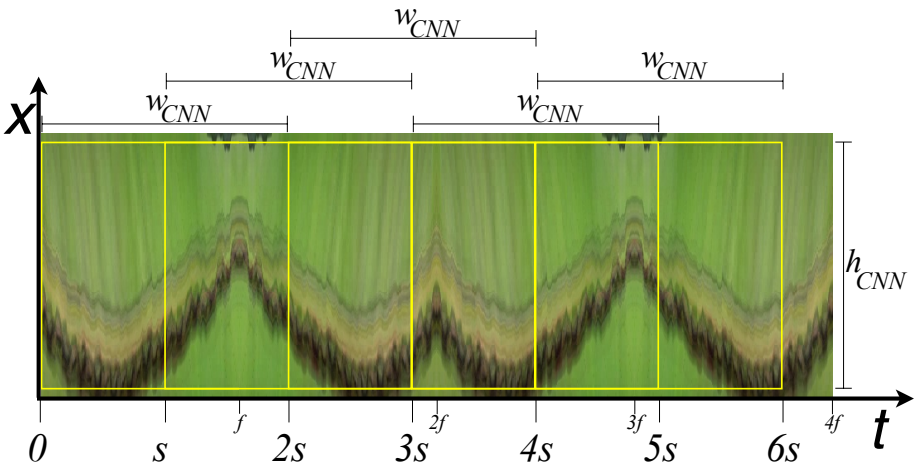


Fig. 3: Symmetric extension of a VR covering five squared crops: the frame width is $w = 320$ pixels, the corresponding video length is $f = 240$ frames, the stride between crops is $s = 150$ pixels and the crop dimensions are $w_{CNN} = h_{CNN} = 299$. The central area in X is selected in this example.

3.3 Video Classification Protocol

At inference time and for video classification, all the augmented crops are applied to the CNN and their last layer feature maps are extracted (just before softmax activation) and averaged. The softmax activation is applied to this average feature maps and then used to predict the sample class. We argue that this process might yield better class predictions based on the assumption that

multiple crops taken at different time positions are representative of distinct portion of the underlying action in the video. The whole process is depicted in Figure 1. In training stage, however, each crop is processed as a distinct sample and separately classified, i.e. the average is not taken into account.

4 Experimental Results

Datasets. The proposed method was evaluated through experiments performed on two challenging video action datasets: UCF101 [14] and HMDB51 [15]. The UCF101 dataset contains 13320 videos. All videos have fixed frame rate and resolution of 25 FPS and 320×240 pixels, respectively. This dataset covers a broad scope of actions from the simplest to the most complex ones. An example of the latter is playing some sport or playing some instrument. These videos were collected from Youtube and divided into 101 classes. Since they were uploaded by multiple users, there is a great diversity in terms of variations in camera motion, object appearance and pose, object scale and viewpoint. This diversity is essential to replicate the variety of actions that a more realistic scenario could have. HMDB51 is an action recognition dataset containing 6766 videos from 51 different action classes. The HMDB51 includes a wide variety of samples collected from various sources, including blurred videos, or with lower quality, and actions performed in distinct viewpoints. The evaluation protocol used for both datasets is the same. The average accuracy of the three training/testing splits available for both datasets is reported as the final result.
Implementation Details. The Keras framework [28] was used for all experiments. A slightly modified version of the InceptionV3 network [2] initialized with ImageNet [1] weights was used in the experiments of the Visual Rhythm Parameterization section. The InceptionV3 was modified to have an additional fully connected layer with 1024 neurons and 60% of dropout. The softmax classifier was adapted to match the number of classes in each dataset. It was used in the experiments that explored the variation of VR parameters.

All training parameters were kept the same for both datasets. Some Keras random data augmentation approaches (horizontal flip, vertical flip and zoom in the range of 0.8 to 1.2) were applied to the VRs. The network was trained with the following parameters: learning rate of $1e^{-3}$, batch size of 16, Stochastic Gradient Descent (SGD) optimizer with momentum of 0.9 and categorical cross entropy loss function. The early stopping training strategy is adopted with patience of 6 epochs. The learning rate was also scaled down by a factor of 10 after 3 epochs without any improvement in the loss function. The learning rate decrease was limited to $1e^{-6}$.

The representation of the video through the weighted VR depends on the choice of two parameters: a reference row (or column) α_y and the standard-deviation σ_y. The impacts of these parameters are explored in the first two experiments. The results show that the right choice of these parameters can help to improve the accuracy in both datasets. These results also provide evidence that the main action of the videos tends to focus around a certain region of

the frames. We also perform experiments varying the symmetrical extension parameters aiming to achieve the better settings for it. The results corroborate the assumption that a data augmentation method is essential for increasing the accuracy rates.

4.1 Visual Rhythm Parameterization

A sequence of experiments was performed to discover the best set of parameters for WVR$^+$. The WVR approach is used as baseline comparison method in order to assess the performance gain along these experiments. The mean accuracy reached with WVR is 64.84% and 34.34% in UCF101 and HMDB51 datasets, respectively. All the evaluations used horizontal VRs. This is justified by its superior accuracy in contrast to the vertical one [19]. Throughout the experiments, the best parameters are employed in the subsequent executions. Initially, the parameters used for VR and the symmetric extension are: $\alpha_y = 0.5$ (middle row), $n_w = 1$ and only the central crop in X is extracted. Since InceptionV3 expects input images of 299×299 pixels, w_{CNN} and h_{CNN} are both set to 299. The method depicted in Figure 1 is employed in all experiments.

In the first experiment, we compare the impact of the variation of the σ_y parameter. The following values for σ_y were tested: 7, 15, 33, 49 and 65. These values were chosen with the purpose of verifying if the region in which the action is performed is concentrated in a small area or it is more vertically spread. The results are shown in Table 1. The better standard-deviation for UCF101 was 33 and for HMDB51 it was 15. This indicates that actions on UFC101 tend to occur in a more spread region compared to HMDB51, since a smaller standard-deviation means more concentrated Gaussian weighting around the middle row.

Table 1: Comparison of accuracy rates (%) for UCF101 and HDMB51 varying the σ_y parameter

σ_y	UCF101 (%)	HMDB51 (%)
7	63.29	33.66
15	63.85	**33.99**
33	**65.26**	33.40
49	64.62	32.24
65	63.00	31.46

In the second experiment, we show the influence of the reference row on the accuracy rate. As mentioned earlier, a factor α_y is used instead the parameter y. The values chosen for the factor were: 0.40, 0.45, 0.50, 0.55 and 0.60. Values above 0.5 indicate the lower part of the image. Because the samples in both datasets come from multiple sources, the main action in each video may not happen exactly in the center of the video. This is the case of the UCF101. It was

empirically observed that the better results were obtained when the reference row is located just below the center of the video. The mean action position in this dataset tends to be shifted around 5% below the middle of the video (Table 2).

Table 2: Comparison of mean accuracy rates (%) of UCF101 and HMDB51 varying the α_y factor

α_y	UCF101 (%)	HMDB51 (%)
0.40	62.82	30.06
0.45	64.83	31.00
0.50	65.26	**33.99**
0.55	**65.32**	33.35
0.60	65.24	33.48

In the next experiment, the number of windows n_w is increased in order to check if the accuracy rate also increases. The premise is that with more windows it is possible to cover the entire temporal extension of the video present in WVR$^+$. It is expected that the additional windows incorporates more discriminant aspects of the video. The n_w values used are: 1, 2, 3 and 4. In this experiment the stride s between the windows is fixed to 299 matching the w_{CNN} size. Thus, consecutive and non-overlapping crops are obtained. Table 3 show the results of this experiment. The expected correlation between n_w and the accuracy rate can be endorsed by the results.

Table 3: Comparison of accuracy rates (%) of UCF101 and HMDB51 datasets varying n_w parameter

n_w	UCF101 (%)	HMDB51 (%)
1	65.32	33.99
2	65.64	34.42
3	66.19	34.03
4	**67.70**	**34.99**

The fourth experiment consists of using windows that overlaps each other along the time dimension. When the extended VR completes a cycle it begins to repeat its temporal patterns as shown in Figure 3. Crops can be extracted along the extended time with or without direct overlapping, consecutively or not. Consecutive and non-overlapping neighbor crops are obtained with $s = w_{CNN}$. Gaps between the crops are obtained by using stride $s > w_{CNN}$. Overlaps between consecutive crops occur when using $s < w_{CNN}$. In this work, we investigate the

cases having $0 < s \leq w_{CNN}$. Notice that multiple parts of the VR, forward or backward in time, will be repeated unless $w_{CNN} + (n_w - 1) \cdot s < f$.

We used the strides 13, 25, 274, 286 and 299 that have a direct relation with video frame rate. Since all videos have 25 FPS, each 25 columns of the VR represents one second of the video. With a stride of 25, for instance, two consecutive crops overlap each other along their entire length except for the first second of the current crop and the last second of the next crop. On the other hand, with a stride of 274 the overlap occurs only between the last second of a crop and the first second of the following crop. Table 4 shows the results of this experiment. Notice that $s = 299$ provided the best accuracy for both datasets. This is exactly the same width of the CNN input. Further experiments are necessary to check if there is some relation between the stride s and the architecture input size.

Table 4: Comparison of accuracy rates (%) for UCF101 and HMDB51 varying the stride s parameter

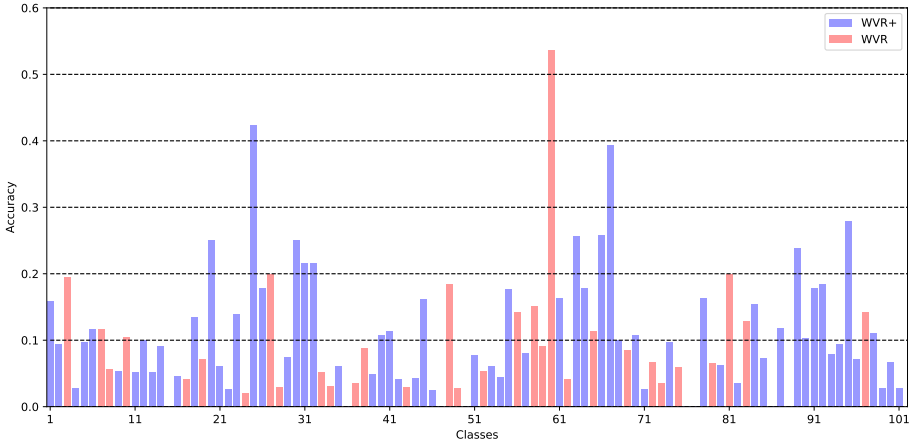
s	UCF101 (%)	HMDB51 (%)
13	66.26	34.10
25	65.60	33.75
274	66.56	33.86
286	66.18	34.09
299	**67.70**	**34.99**

We also used the top and bottom regions in X direction. Therefore, each video is covered by 12 windows. The results are presented in Table 5, using the best parameters found in previous experiments: $n_w = 4$ and $s = 299$ for both datasets, $\alpha_y = 0.55$ and $\sigma_y = 0.33$ for UCF101 and $\alpha_y = 0.5$ and $\sigma_y = 0.15$ for HMDB51. The extra 8 crops helped to increase the accuracy rate in both datasets. Similar to the previous experiment, the use of the extra regions produced an overlap between the crops along the spatial dimension. However, more experiments need to be performed to assess how the overlap in X can be explored for data augmentation.

Table 5: Comparison of accuracy rates (%) for UCF101 and HMDB51 when extra crops are used

Regions	UCF101 (%)	HMDB51 (%)
Central	67.70	34.99
Central + Top + Bottom	**68.01**	**35.29**

Figure 4 show for UFC101 and HMDB51 the accuracy difference between the best result of WVR$^+$ (Table 5) and the baseline method WVR. Only differences for the split 1 of both datasets are shown. Blue bars mean the WVR$^+$ were better by the given amount. Conversely, red bars favor the WVR. For the UCF101, WVR$^+$ performs better in 62 classes, worse in 31 classes, with even results in 8 classes. For the HMDB51, WVR$^+$ performs better in 24 classes, worse in 18 classes, with even results in 9 classes. The classes which demonstrated improvement for the proposed method seem to share some characteristics among each other. They often present actions with certain cyclic movements (e.g., *Brushing Teeth, Playing Violin, Typing*). This kind of action takes full advantage of the symmetrical extension of VR. Since the reverse movement generates patterns very similar to the one generated by the original movement, the multiple crops reinforce this kind of action and increase accuracy of them.



(a) UCF101

Fig. 4: Accuracy difference for each class between WVR$^+$ (blue) and WVR (red) for split 1 of UCF101.

4.2 Multi-stream Classification using Visual Rhythms

Our goal in this section is to show that our method can complement multi-stream architectures to get more competitive accuracy rates. The results of individual streams are shown in Table 6. The first three approaches, RGB*, Horizontal-mean and Adaptive Visual Rhythm (AVR), are contributions of our previous work [19]. Similar to other multi-stream networks [26, 29], the Optical Flow performs better in both datasets. So, the other streams are crucial to complement the Optical Flow and to improve accuracy when combined. In order to achieve competitive results, experiments were performed merging the three streams: our

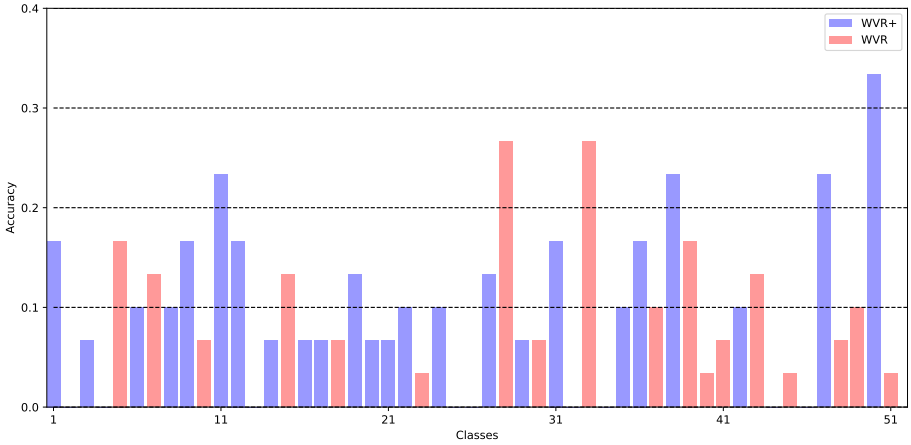


Fig. 5: Accuracy difference for each class between WVR$^+$ (blue) and WVR (red) for split 1 of HMDB51.

best WVR$^+$ setup, the RGB* and the Optical Flow. The multi-stream approach of our previous work [19] was adopted to accomplish this purpose.

Table 6: Results for single-stream features.

Single-Stream	UCF101	HMDB51
RGB* images [19]	86.61	51.77
Horizontal - mean [19]	62.37	35.57
AVR [19]	64.74	39.63
Optical Flow [19]	**86.95**	**59.91**
Our method WVR$^+$	68.01	35.29

Table 7 presents the results of our method combined with RGB* and Optical Flow features through multi-stream late fusion. More specifically, at testing stage, three weights were evaluated through a grid search strategy. For each weight, we tested every value from 0 to 10 with a 0.5 step. It was observed that a higher accuracy is reached when the combination is done with the feature maps before the softmax normalization. The best combination found for UCF101 was 7.5, 6.0 and 1.0, respectively for Optical Flow, RGB* and WVR$^+$. And the best combination found for HMDB51 was 3.5, 1.5 and 0.5, respectively for Optical Flow, RGB* and WVR$^+$. We obtained 93.8% for UCF101 and 65.7% for HMDB51. Although WVR$^+$ by itself is not able to achieve accuracy rates comparable to the state-of-the-art (Table 7), our multi-stream method achieve fairly competitive accuracy rates. It is overcame only by the state-of-the-art work

presented in [7] and the others that were also pre-trained with the Kinetics [30] dataset. Considering the UCF101, our method outperforms the proposal of [19], using the InceptionV3. Our approach is not better than the ResNet152 result for the UCF101. Due to the differences between InceptionV3 and ResNet152, further investigation is needed. The HMDB51 accuracy rate is lower than previous methods due to the lack of vertical VR information. The possible fusion with vertical VRs, however, makes our method promising.

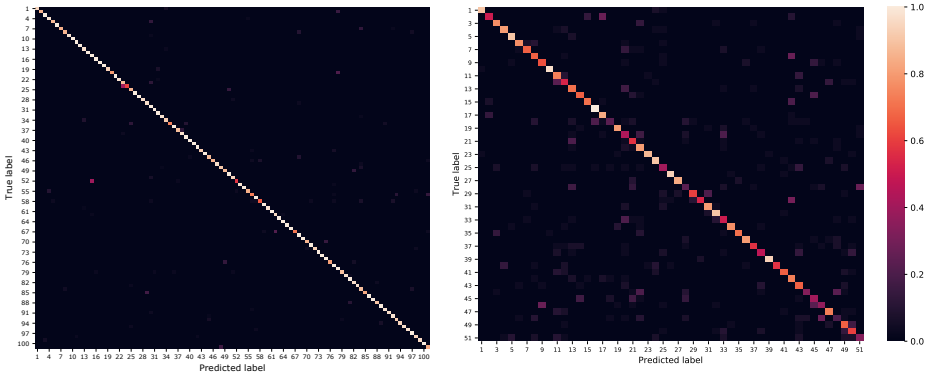
Table 7: Comparison of accuracy rates (%) for UCF101 and HMDB51 datasets

Method	UCF101 (%)	HMDB51 (%)
iDT + HSV [31]	87.9	61.1
Two-Stream [26]	88.0	59.4
Two-Stream TSN [25]	94.0	68.5
Three-Stream TSN [25]	94.2	69.4
Three-Stream [32]	94.1	70.4
Two-Stream I3D [7]	98.0	80.7
I3D + PoTion [11]	**98.2**	80.9
SVMP+I3D [9]	-	81.3
DTPP (Kinetics pre-training) [22]	98.0	**82.1**
TDD+iDT [33]	91.5	65.9
LTC+iDT [34]	91.7	64.8
KVMDF [24]	93.1	63.3
STP [35]	94.6	68.9
L^2STM [36]	93.6	66.2
Multi-Stream + ResNet152 [19]	94.3	68.3
Multi-Stream + InceptionV3 [19]	93.7	69.9
Our method	93.8	67.1

The confusion matrices of our multi-stream method applied for UCF101 and HMDB51, respectively, are shown on Figures 6a and 6b. On UCF101 is possible to notice a reasonable misclassification between *Body Weight Squats* and *Lunges* classes (indexes 15 and 52 respectively) because their similar motion aspect.

5 Conclusions and Future Work

In this work, we proposed an approach to deal with video classification using a 2D representation of videos. The method consists of symmetrically extending the temporal dimension of the VR and taking crops apart by a stride. This method maintains the video frame rate and allows multiple samples of the underlying motion pattern to be obtained. It also provides data augmentation which is valuable for training 2D CNNs with small datasets. Furthermore, we explore the



(a) (b)

Fig. 6: Confusion matrix of the final multi-stream method for split 3: (a) UCF101. (b) HMDB51.

parameters of our method and verified that each dataset requires different settings to achieve better performance. Experimental results show that our method improves accuracy rates if compared to the resized horizontal VR. Results for HMDB51, which is more challenging, show that the information of the vertical rhythm can be valuable to improve the method efficiency. We also showed that our method achieves fairly competitive results compared to state-of-the-art approaches when combined with other features in a multi-stream architecture. As future work, it is worthy to investigate how multiple directions of VRs can be used for a single video. Vertical VRs, for instance, may improve recognition rates. More experiments are needed to check the relationship between the stride s and the network input size. It is also important to test our method with other 2D CNNs, such as ResNet152.

References

1. J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. ImageNet: A Large-Scale Hierarchical Image Database. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2009.
2. Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jon Shlens, and Zbigniew Wojna. Rethinking the Inception Architecture for Computer Vision. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 2818–2826, 2016.
3. Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep Residual Learning for Image Recognition. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 770–778, 2016.
4. Arridhana Ciptadi, Matthew S Goodwin, and James M Rehg. Movement pattern histogram for action recognition and retrieval. In *European Conference on Computer Vision*, pages 695–710. Springer, 2014.

5. Shuiwang Ji, Wei Xu, Ming Yang, and Kai Yu. 3D Convolutional Neural Networks for Human Action Recognition. *in IEEE Transactions on Pattern Analysis and Machine Intelligence*, 35(1):221–231, 2013.
6. Yu Kong and Yun Fu. Human action recognition and prediction: A survey. *arXiv preprint arXiv:1806.11230*, 2018.
7. Joao Carreira and Andrew Zisserman. Quo Vadis, Action Recognition? A New Model and the Kinetics Dataset. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 4724–4733. IEEE, 2017.
8. Hakan Bilen, Basura Fernando, Efstratios Gavves, Andrea Vedaldi, and Stephen Gould. Dynamic image networks for action recognition. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 3034–3042, 2016.
9. Jue Wang, Anoop Cherian, Fatih Porikli, and Stephen Gould. Video representation learning using discriminative pooling. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 1149–1158, 2018.
10. Christoph Feichtenhofer, Axel Pinz, and Richard P Wildes. Spatiotemporal multiplier networks for video action recognition. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 7445–7454. IEEE, 2017.
11. Vasileios Choutas, Philippe Weinzaepfel, Jérôme Revaud, and Cordelia Schmid. Potion: Pose motion representation for action recognition. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2018.
12. Sami Abu-El-Haija, Nisarg Kothari, Joonseok Lee, Paul Natsev, George Toderici, Balakrishnan Varadarajan, and Sudheendra Vijayanarasimhan. Youtube-8M: A large-scale video classification benchmark. *arXiv preprint arXiv:1609.08675*, 2016.
13. Andrej Karpathy, George Toderici, Sanketh Shetty, Thomas Leung, Rahul Sukthankar, and Li Fei-Fei. Large-Scale Video Classification with Convolutional Neural Networks. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 1725–1732, 2014.
14. 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.
15. Hilde Kuehne, Hueihan Jhuang, Rainer Stiefelhagen, and Thomas Serre. HMDB51: A Large Video Database for Human Motion Recognition. In *High Performance Computing in Science and Engineering*, pages 571–582. Springer, 2013.
16. Chong-Wah Ngo, Ting-Chuen Pong, and Roland T Chin. Camera Break Detection by Partitioning of 2D Spatio-Temporal Images in MPEG Domain. In *IEEE International Conference on Multimedia Computing and Systems*, volume 1, pages 750–755. IEEE, 1999.
17. Chong-Wah Ngo, Ting-Chuen Pong, and Roland T Chin. Detection of Gradual Transitions through Temporal Slice Analysis. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, volume 1, pages 36–41. IEEE, 1999.
18. Marcos Roberto Souza. Digital Video Stabilization: Algorithms and Evaluation. Master's thesis, Institute of Computing, University of Campinas, Campinas, Brazil, 2018.
19. Darwin Ttito Concha, Helena de Almeida Maia, Helio Pedrini, Hemerson Tacon, André de Souza Brito, Hugo de Lima Chaves, and Marcelo Bernardes Vieira. Multi-Stream Convolutional Neural Networks for Action Recognition in Video Sequences Based on Adaptive Visual Rhythms. In *IEEE International Conference on Machine Learning and Applications*. IEEE, 2018.

20. Hyeokman Kim, Jinho Lee, Jae-Heon Yang, Sanghoon Sull, Woonkyung M Kim, and S Moon-Ho Song. Visual rhythm and shot verification. *Multimedia Tools and Applications*, 15(3):227–245, 2001.
21. Berthin S Torres and Helio Pedrini. Detection of Complex Video Events through Visual Rhythm. *The Visual Computer*, pages 1–21, 2016.
22. Jiagang Zhu, Zheng Zhu, and Wei Zou. End-to-end video-level representation learning for action recognition. In *2018 24th International Conference on Pattern Recognition (ICPR)*, pages 645–650. IEEE, 2018.
23. Joe Yue-Hei Ng, Mattew Hausknecht, Sudheendra Vijayanarasimhan, Oriol Vinyals, Rajat Monga, and George Toderici. Beyond Short Snippets: Deep Networks for Video Classification. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 4694–4702, 2015.
24. Wangjiang Zhu, Jie Hu, Gang Sun, Xudong Cao, and Yu Qiao. A Key Volume Mining Deep Framework for Action Recognition. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 1991–1999. IEEE, 2016.
25. Limin Wang, Yuanjun Xiong, Zhe Wang, Yu Qiao, Dahua Lin, Xiaoou Tang, and Luc Van Gool. Temporal Segment Networks: Towards Good Practices for Deep Action Recognition. In *European Conference on Computer Vision*, pages 20–36. Springer, 2016.
26. Karen Simonyan and Andrew Zisserman. Two-Stream Convolutional Networks for Action Recognition in Videos. In *Advances in Neural Information Processing Systems*, pages 568–576, 2014.
27. Karen Simonyan and Andrew Zisserman. Very Deep Convolutional Networks for Large-Scale Image Recognition. *arXiv preprint arXiv:1409.1556*, 2014.
28. François Chollet et al. Keras. https://keras.io, 2015.
29. Limin Wang, Yuanjun Xiong, Zhe Wang, and Yu Qiao. Towards Good Practices for very Deep Two-Stream Convnets. *arXiv preprint arXiv:1507.02159*, 2015.
30. Will Kay, Joao Carreira, Karen Simonyan, Brian Zhang, Chloe Hillier, Sudheendra Vijayanarasimhan, Fabio Viola, Tim Green, Trevor Back, Paul Natsev, Mustafa Suleyman, and Andrew Zisserman. The Kinetics Human Action Video Dataset. *arXiv preprint arXiv:1705.06950*, 2017.
31. Xiaojiang Peng, Limin Wang, Xingxing Wang, and Yu Qiao. Bag of Visual Words and Fusion Methods for Action Recognition: Comprehensive Study and Good Practice. *Computer Vision and Image Understanding*, 150:109–125, 2016.
32. Hao Wang, Yanhua Yang, Erkun Yang, and Cheng Deng. Exploring Hybrid Spatio-Temporal Convolutional Networks for Human Action Recognition. *Multimedia Tools and Applications*, 76(13):15065–15081, 2017.
33. Limin Wang, Yu Qiao, and Xiaoou Tang. Action Recognition with Trajectory-Pooled Deep-Convolutional Descriptors. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 4305–4314, 2015.
34. Gül Varol, Ivan Laptev, and Cordelia Schmid. Long-Term Temporal Convolutions for Action Recognition. *arXiv preprint arXiv:1604.04494*, 2016.
35. Yunbo Wang, Mingsheng Long, Jianmin Wang, and Philip S Yu. Spatiotemporal Pyramid Network for Video Action Recognition. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 2097–2106. IEEE, 2017.
36. Lin Sun, Kui Jia, Kevin Chen, Dit Yan Yeung, Bertram E Shi, and Silvio Savarese. Lattice Long Short-Term Memory for Human Action Recognition. *arXiv preprint arXiv:1708.03958*, 2017.