Spatio-temporal Video Autoencoder for Human Action Recognition

Anderson Carlos Sousa e Santos and Helio Pedrini
Institute of Computing, University of Campinas, Campinas-SP, 13083-852, Brazil

Keywords: Action Recognition, Multi-stream Neural Network, Video Representation, Autoencoder.

Abstract: The demand for automatic systems for action recognition has increased significantly due to the development of surveillance cameras with high sampling rates, low cost, small size and high resolution. These systems can effectively support human operators to detect events of interest in video sequences, reducing failures and improving recognition results. In this work, we develop and analyze a method to learn two-dimensional (2D) representations from videos through an autoencoder framework. A multi-stream network is used to incorporate spatial and temporal information for action recognition purposes. Experiments conducted on the challenging UCF101 and HMDB51 data sets indicate that our representation is capable of achieving competitive accuracy rates compared to the literature approaches.

1 INTRODUCTION

Due to the large availability of digital content captured by cameras in different environments, the recognition of events in video sequences is a very challenging task. Several problems have benefited from these recognition systems (Cornejo et al., 2015; Gori et al., 2016; Ji et al., 2013; Ryoo and Matthies, 2016), such as health monitoring, surveillance, entertainment and forensics.

Typically, visual inspection is performed by a human operator to identify events of interest in video sequences. However, this process is time consuming and susceptible to failure under fatigue or stress. Therefore, the massive amount of data involved in real-world scenarios makes the event recognition impracticable, such that automatic systems are crucial in monitoring tasks in real-world scenarios.

Human action recognition (Alcantara et al., 2013, 2016, 2017a,b; Concha et al., 2018; Moreira et al., 2017) is addressed in this work, whose purpose is to identify activities performed by a number of agents from observations acquired by a video camera. Although several approaches have been proposed in the literature, many questions remain open because of the challenges associated with the problem, such as lack of scalability, spatial and temporal relations, complex interactions among objects and people, as well as complexity of the scenes due to lighting conditions, occlusions, background clutter, camera motion.

Most of the approaches available in the literature can be classified into two categories: (i) traditional shallow methods and (ii) deep learning methods.

In the first group, shallow hand-crafted features are extracted to describe regions of the video and combined into a video level description (Baumann et al., 2014; Maia et al., 2015; Peng et al., 2016; Perez et al., 2012; Phan et al., 2016; Torres and Pedrini, 2016; Wang et al., 2011; Yeffet and Wolf, 2009). A popular feature representation is known as bag of visual words. A conventional classifier, such as support vector machine or random forest, is trained on the feature representation to produce the final action prediction.

In the second group, deep learning techniques based on convolutional neural networks and recurrent neural networks have automatically learned features from the raw sensor data (Ji et al., 2013; Kahani et al., 2017; Karpathy et al., 2014; Ng et al., 2015; Ravanbakhsh et al., 2015; Simonyan and Zisserman, 2014a).

Although there is a significant growth of approaches in the second category, recent deep learning strategies have explored information from both to preprocess and combine the data (Kahani et al., 2017; Karpathy et al., 2014; Ng et al., 2015; Ravanbakhsh et al., 2015; Simonyan and Zisserman, 2014a). Spatial and temporal information can be incorporated through a two-dimensional (2D) representation in contrast to a three-dimensional (3D) scheme. Some advantages of modeling videos as images instead of volumes is the use of pre-trained image networks, reduction of training cost, and availability of large image data sets.

Santos, A. and Pedrini, H.
Spatio-temporal Video Autoencoder for Human Action Recognition.
DOI: 10.5220/0007409401140123
In *Proceedings of the 14th International Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications (VISIGRAPP 2019)*, pages 114-123
ISBN: 978-989-758-354-4

In a pioneering work, Simonyan and Zisserman (2014a) proposed a two-stream architecture based on convolutional networks to recognize action in videos. Each stream explored a different type of features, more specifically, spatial and temporal information. Inspired by their satisfactory results, several authors have developed networks based on multiple streams to explore complementary information (Gammulle et al., 2017; Khaire et al., 2018; Tran and Cheong, 2017; Wang et al., 2017a,b, 2016a).

We propose a spatio-temporal 2D video representation learned by a video autoencoder, whose encoder transforms a set of frames to a single image and then the decoder transforms it back to the set of frames. As a compact representation of the video content, this learned encoder serves as a stream in our multi-stream proposal.

Experiments conducted on two well-known challenging data sets, HMDB51 (Kuehne et al., 2013) and UCF101 (Soomro et al., 2012a), achieved accuracy rates comparable to state-of-the-art approaches, which demonstrates the effectiveness of our video encoding as a spatio-temporal stream to a convolutional neural network (CNN) in order to improve action recognition performance.

This paper is organized as follows. In Section 2, we briefly describe relevant related work. In Section 3, we present our proposed multi-stream architecture for action recognition. In Section 4, experimental results achieved with the proposed method are presented and discussed. Finally, we present some concluding remarks and directions for future work in Section 5.

2 RELATED WORK

The first convolutional neural networks (CNNs) proposed for action recognition used 3D convolutions to capture spatio-temporal features (Ji et al., 2013). Karpathy et al. (2014) trained 3D networks from scratch using the Sports-1M, a data set with more than 1 million videos. However, it does not outperform traditional methods in terms of accuracy due to the difficulty in representing motion.

To overcome this problem, Simonyan and Zisserman (2014a) proposed a two-stream method in which motion is represented by pre-computed optical flows that are encoded with a 2D CNN. Later, Wang et al. (2015b) further improved the method, especially using more recent deeper architectures for 2D CNN and taking advantage of the pre-trained weights for the temporal stream. Based on this two-stream framework, Carreira and Zisserman (2017) proposed a 3D CNN which is an inflated version of a 2D CNN and also uses the pre-trained weights, in addition to training the network with a huge database of action and achieving significant higher accuracies. These improvements show the importance of using well-established 2D deep CNN architectures and their pre-trained weights from ImageNet (Russakovsky et al., 2015).

Despite the advantage of the two-stream approach, it still fails to capture long-term relationships. There are several approaches that attempt to tackle this problem. There are two primary strategies for this problem: work on the CNN output by searching for a way to aggregate the features from frames or snippets (Diba et al., 2017; Donahue et al., 2015; Ma et al., 2018; Ng et al., 2015; Varol et al., 2016; Wang et al., 2016a) or introduce a different temporal representation (Bilen et al., 2017; Hommos et al., 2018; Wang et al., 2017b, 2016b). Our work fits into this latter type of approach.

Wang et al. (2016b) used a siamese network to model the action as a transformation from a preconditioned state to an effect. Wang et al. (2017b) used a handcrafted representation called Motion Stacked Difference Image that is inspired by Motion Energy Image (MEI) (Ahad et al., 2012) as a third stream. Hommos et al. (2018) introduced an Eulerian phase-based motion representation that can be learned end-to-end, but it is showed as an alternative for optical flow and does not improve further on the two-stream framework. The same can be said in the work by Zhu et al. (2017) that introduced a network that computes an optical flow representation that can be learned in an end-to-end fashion.

The work that most resembles ours is based on Dynamic Images (Bilen et al., 2017), which is a 2D image representation that summarizes a video and is easily added as a stream for action recognition. However, this representation constitutes the parameters of a ranking function that is learned for each video and, although it can be presented as a layer and trained together with the 2D CNN, this layer works similarly to a temporal pooling and the representation is not adjusted. On the other hand, our method learns a video-to-image mapping with an autoencoder and is incorporated into the 2D CNN, allowing full end-to-end learning.

Autoencoders are not a new idea for dimensionality reduction that more recently gained attention as a generative model. It is trained in order to copy the input to the output, but with constraints that hopefully will reveal useful properties in data (Goodfellow et al., 2016).

Video autoencoders are used for anomaly detection in video sequences, where the reconstruction error threshold given training with normal samples indicates the abnormality (Kiran et al., 2018). The architectures vary from stack of frames submitted to 2D (Hasan et al., 2016) or 3D convolutions (Zhao et al., 2017b) and Convolutional LSTMs (Chong and Tay, 2017). In addition, the intermediate representation is not the ultimate goal and presents low spatial resolution and high depth, which is not useful for classification with the target CNNs for action. To the best of our knowledge, there are no approaches that, similar to ours, use a video autoencoder to map a video into a 2D image representation that maintains the spatial size of the frames.

3 PROPOSED METHOD

In this section, we describe the proposed action representation based on a video autoencoder that produces an image representation for a set of video frames. This representation can be learned for end-to-end classification. Furthermore, we coupled it with a different stream in a multi-stream framework for action recognition.

3.1 Video Autoencoder

An autoencoder is an unsupervised learning approach that aims to learn an identity function, that is, the input and the expected output are equal. The goal is to reveal interesting structures in the data by placing constraints in the learning process.

Figure 1 shows our proposed architecture for a video autoencoder, in which a set of N grayscale frames is arranged as an N dimensional image for input. This image is passed through an encoder, whose output is a three-dimensional image. This image is then passed to the decoder, where the output is again N dimensional and represents the reconstructed video. The purpose of this autoencoder is to shrink the video to a single image representation by learning how to reconstruct a set of frames using only a 3-channel tensor.

The main advantage of our video representation as an image is that it can be used in any of the many well-established 2D CNN architectures with pre-trained weights from the ImageNet competition. The use of these deep convolution networks achieved state-of-the-art results in many computer vision tasks.

Unlike some other handcrafted representations, ours provides end-to-end learning. It can be easily directly linked to any 2D CNN, where the encoder will have its weights updated with respect to loss of action classification, which would make the representation specifically improve for the desired problem and this is, in fact, what we observe in the experiments described in Section 4.

3.1.1 Encoder

The encoder is a simple block with a 3×3 convolution layer with 3 filters followed by a batch normalization and a hyperbolic tangent activation function. In order to maintain the image size, zero padding is applied and no strides are used.

The choice of an activation function was guided by the goal to easily link the encoder to a 2D CNN. Using the hyperbolic tangent imposes an output in the range of $[-1,1]$, which is the standard input normalization for CNNs pretrained in ImageNet such as Inception (Szegedy et al., 2016).

3.1.2 Decoder

The decoder is even simpler, consisting of only a 3×3 convolution layer with linear activation and N filters, where N corresponds to the same number as the frames in the input. The lack of batch normalization and a non-linear activation forces the model to concentrate most of the reconstruction capacity on the intermediate representation.

Maintaining a simple decoder causes the encoder output to be encapsulated more clearly the structures of the input data, so the generated images still make sense and resemble the original video frames, as illustrated in Figure 2.

3.1.3 Loss

We analyze two types of losses for the autoencoder. They are defined for images and we extend it to video by computing the average of all frames.

The most common loss function is the mean square error (MSE) expressed in Equation 1.

$$\text{MSE} = \sum_i \sum_j (f(i,j) - g(i,j))^2 \qquad (1)$$

where f and g are the images and i and j the vertical and horizontal coordinates respectively. It corresponds to the L2 norm and is the standard for image reconstruction problems (Zhao et al., 2017a).

The second loss function tested is based on the structural similarity index metric (SSIM) (Wang et al., 2004). It is a quality measure that computes on two image windows $\mathbf{x}$ and $\mathbf{y}$ of the same size, each in a different image $\mathbf{f}$ and $\mathbf{g}$. Equation 2 expresses the SSIM metric.

$$\text{SSIM}(\mathbf{x},\mathbf{y}) = \frac{(2\mu_x\mu_y + C_1)(2\sigma_{xy} + C_2)}{(\mu_x^2 + \mu_y^2 + C_1)(\sigma_x^2 + \sigma_y^2 + C_2)} \qquad (2)$$

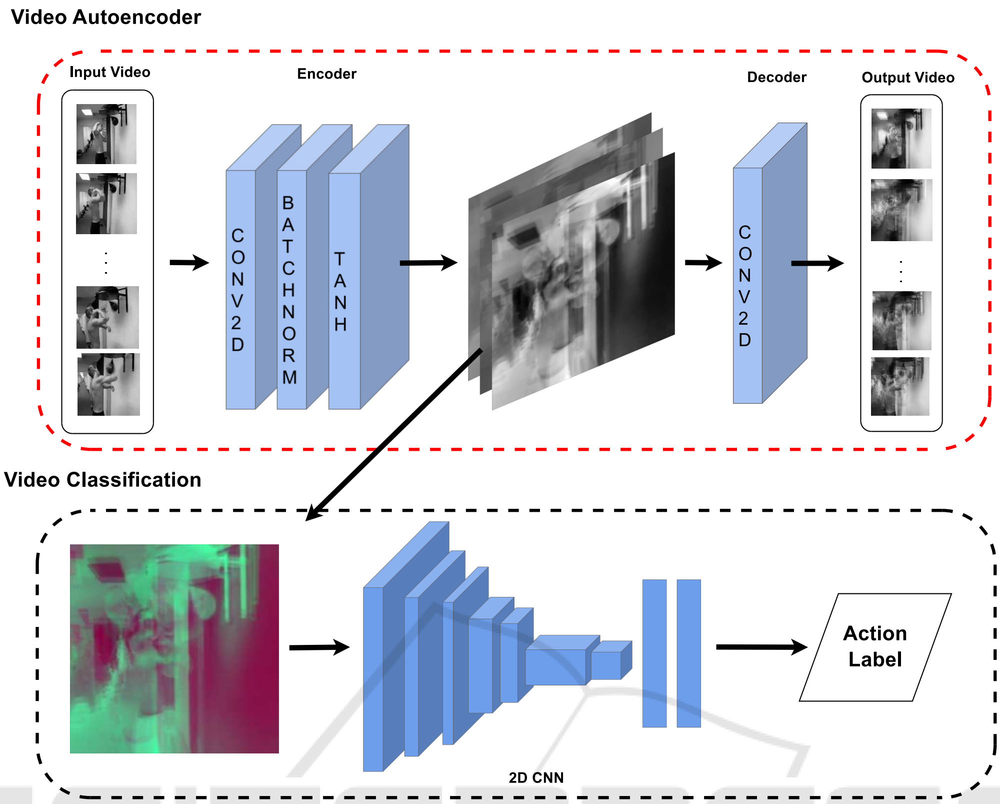


Figure 1: Video autoencoder architecture and its use in action classification.

where μ_x is the mean of $\mathbf{x}$, μ_y is the mean of $\mathbf{y}$, σ_x^2 is the variance of $\mathbf{x}$, σ_y^2 is the variance of $\mathbf{y}$, and σ_{xy} is the covariance of $\mathbf{x}$ and $\mathbf{y}$, C_1 and C_2 are constants that stabilize the equation ($C_1 = 0.01 * 255^2$ and $C_2 = 0.03 * 255^2$).

The final index between f and g is the average of all windows for each pixel. Since we need a loss function, the DSSIM is simply $\dfrac{(1 - \text{SSIM})}{2}$.

The DSSIM loss corresponds more to a perceived human difference than the MSE. The latter will further penalize the differences in contrast and brightness, whereas the first will focus on the structure of the image, which is more interesting to our problem. A comparative analysis is shown in Section 4.

3.2 Multi-stream Architecture

We propose to add our representation as a third stream in the common two-stream architecture for action recognition (Gammulle et al., 2017; Simonyan and Zisserman, 2014a; Wang et al., 2015b). It is also composed of a spatial stream (formed by a single RGB image) and a temporal stream (formed by a stack of optical flow images).

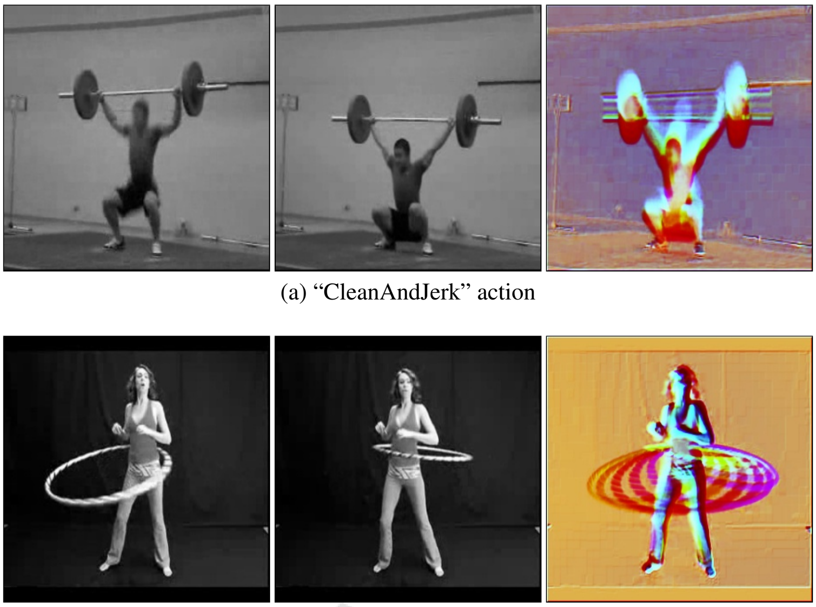
Our stream can be thought as a spatio-temporal encoding since it encapsulates the contextual information and also temporal differences. Figure 3 shows our framework with multiple streams, whose final result is a weighted average among the softmax predictions.

The 2D CNN is basically the same in all streams, the main difference relies on the inputs. The spatial CNN receives a 3-channel image, whereas the temporal CNN receives as input a stack of 20 optical flow images, 10 for each direction. Our proposed spatio-temporal stream receives 10 grayscale images that are passed through the encoder outputting a 3-channel tensor which in turn is passed to a spatio-temporal CNN similar to the spatial.

Each CNN is trained separately and the streams are combined only for the classification in which they generate the predictive confidences for each class. A weighted average produces a final prediction, where the action label is the one with the highest confidence.

4 EXPERIMENTAL RESULTS

In order to evaluate our proposed method, experiments were conducted on two challenging UCF101 and HMDB51 data sets. In this section, we describe



(a) “CleanAndJerk” action

(b) “HulaHoop” action

Figure 2: Examples of the images generated by the encoder of our video autoencoder network.

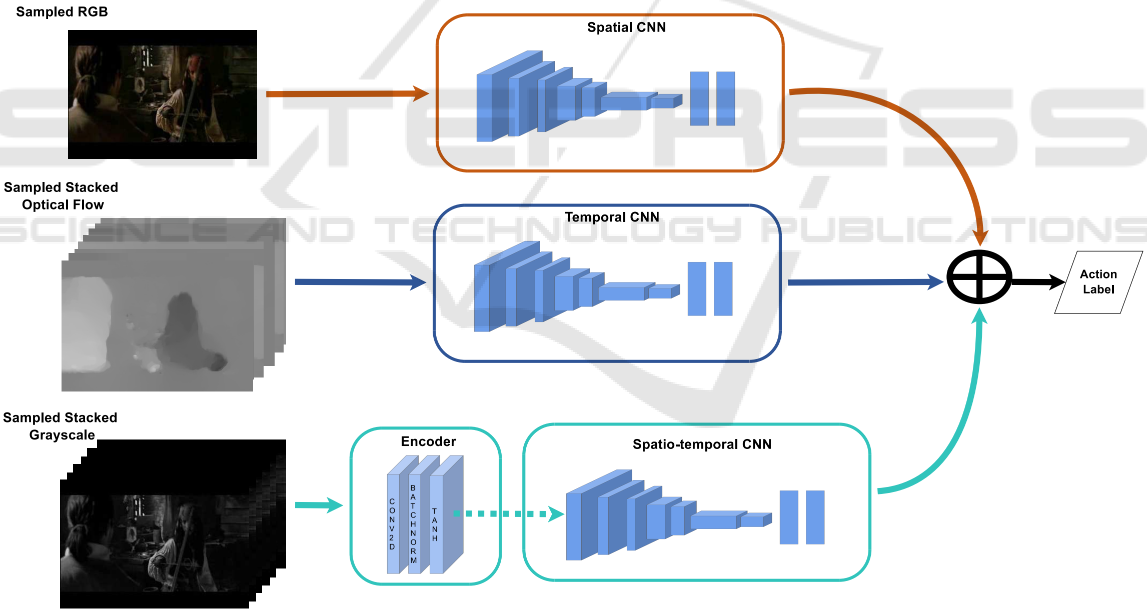


Figure 3: Our action recognition architecture composed of spatial, temporal and spatio-temporal streams.

the data sets used in the experiments, relevant implementation details, results for different configurations of our method and a comparison with some approaches available in the literature.

4.1 Data Sets

The UCF101 (Soomro et al., 2012b) data set contains 13,320 video clips collected from YouTube, with 101 action classes. The videos are grouped into 25 categories. The sequences have a fixed resolution of 320 × 240 pixels, a frame rate of 25 fps and different lengths. The protocol provides three splits into approximately 70% of samples for training and 30% of samples for testing.

The HMDB51 (Kuehne et al., 2013) data set is composed of 6,766 sequences extracted from various sources, mostly from movies, with 51 classes. It presents a variety of video sequences, including blurred videos or with lower quality and actions from different points of views. The protocol provides three splits of the samples, where each split contains 70% of samples for training and 30% for testing for each action class.

4.2 Implementation Details

The Inception V3 (Szegedy et al., 2016) network was the 2D CNN selected to use in our experiments. It achieved state-of-the-art results in the ImageNet competition, such that we started with trained weights from it in all cases.

We fixed the autoencoder input in 10 consecutive frames ($N = 10$). It was trained using Adadelta (Zeiler, 2012) optimizer with the default configuration, zero as initial decay and initial learning rate equal one ($lr = 1.0$).

Data augmentation was applied using random crop and random horizontal flip. The random crop scheme is the same as in the work by Wang et al. (2015b) that uses multi-scale crops of the 4 corners and the center. The complete autoencoder was trained using only the first split of UCF101 with a maximum of 300 epochs saving the weights with the best validation loss.

The multi-stream approach is inspired by the practices described by Wang et al. (2015b). The data augmentation is the same as for the autoencoder. The spatial stream uses a 0.8 dropout before the softmax layer and 250 epochs, whereas the temporal stream uses a 0.7 dropout and 350 epochs. Finally, the proposed spatio-temporal stream uses a 0.7 dropout and 250 epochs. In all of them, the stochastic gradient descent optimizer is used with decay zero, Nesterov momentum equal to 0.9. For all tests, the used batch size is 32 and the learning rate starts at 0.001 and drops by a factor of 0.1 – until the bottom limit of 1^{-10} – if the validation loss does not improve in more than 20 epochs.

The final classification of each testing video is an average of the predictions for 25 frames considering the augmented version – four corners, the center and the horizontal flip – adding up to 10 predictions per frame.

The weights for the fusion between streams follows 8 for temporal, 3 for spatial and 3 for our third stream.

The method was implemented in Python 3 programming language using Keras library. All experiments were performed on a machine with an Intel® Core$^{\text{TM}}$ i7-3770K 3.50GHz processor, 32GB of memory, an NVIDIA GeForce R GTX 1080 GPU and Ubuntu 16.04.

4.3 Results

Initially, we analyze the effects of the loss functions applied to the encoder with respect to the action classification. It can be pre-trained in the autoencoder with MSE or DSSIM functions and additionally be re-trained when linked to the 2D CNN (end-to-end) using the classification loss. One last possibility is to train the encoder only for classification, what discards the autoencoder training. Table 1 shows the accuracy for action classification in the first split of HMDB51 considering these different scenarios.

Table 1: Action classification accuracy for HMDB51 (Split 01).

Loss	Accuracy (%)
MSE	46.80
DSSIM	48.89
Only classification loss	50.52
MSE + classification loss	47.19
DSSIM + classification loss	**51.18**

It is noticeable that further training the encoder along with the Inception V3 considering the classification loss is beneficial, as well as training with the autoencoder previously. The best result was obtained with DSSIM on the autoencoder with a considerable difference from MSE, this highlights the importance of the loss function. From now on, the results of our method refer to the training first with DSSIM and later with the action classification loss.

Table 2 reports the results for the separated streams, the two-stream baseline that combines spatial and temporal information and, finally, the results for our multi-stream architecture including our spatio-temporal encoder.

Individually, the temporal stream obtains the best results and our proposed spatio-temporal approach has similar results from the spatial only. Nonetheless, the fusion of the three streams offers the highest accuracies improving from the traditional two-stream fusion. This shows that our method adds important information, for action recognition, that is not captured by RGB or optical flow images.

Figures 4 and 5 show the accuracy rates per class for our stream, what allows us to investigate the types of actions where it performs better or worse. For the UCF101 data set, “nunchucks” (56) and “JumpingJack” (47) classes present the worst results, whereas “pick” (25), “shoot_ball” (35), “cartwheel” (2) and “swing_baseball” (44) classes achieve the lowest

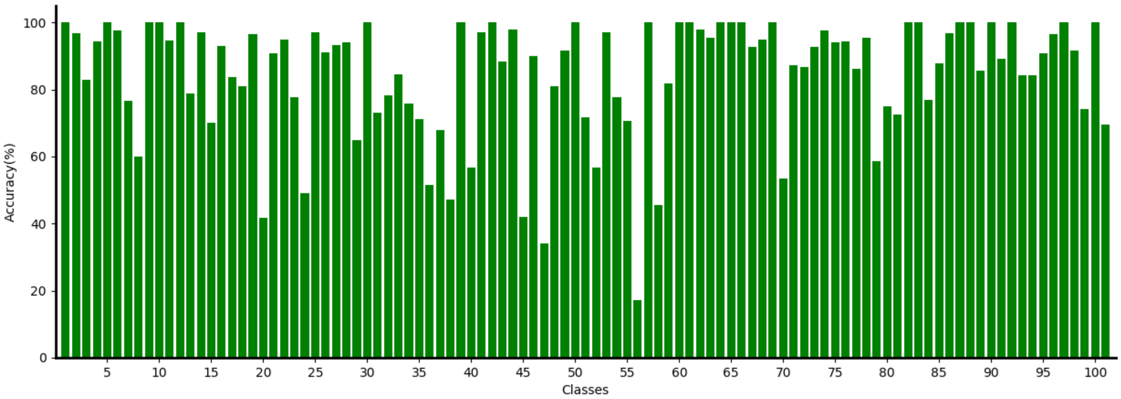


Figure 4: Class-wise accuracy for UCF101 data set (Split 01).

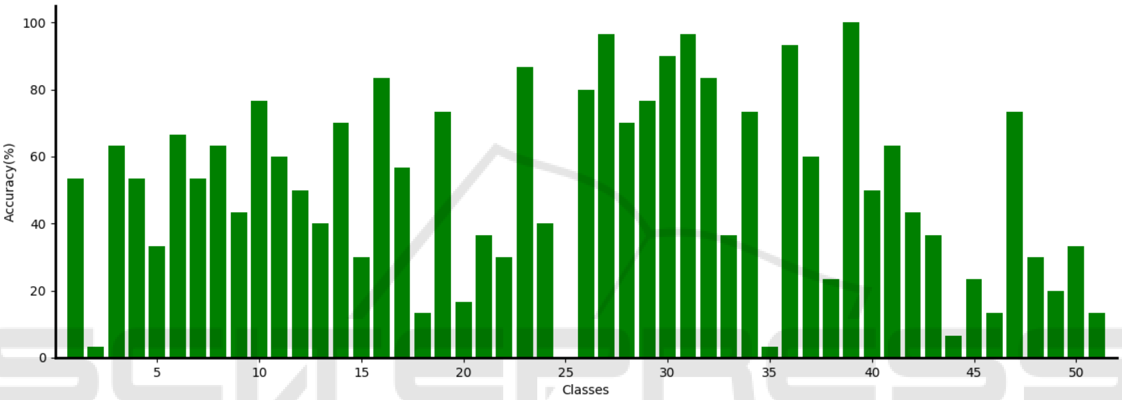


Figure 5: Class-wise accuracy for HMDB51 data set (Split 01).

accuracies in the HMDB51 data set. As common characteristics, these categories have their independence from context, differing mainly in motion. Our method still captures too much information from the scene background, which can be a disadvantage in these cases.

In order to validate our action recognition method, comparative results with state-of-the-art approaches are presented for HMDB51 and UCF101 data sets in Table 3.

The method that presents higher accuracies make use of different strategies for better sampling and late fusion of features and/or streams or perform training with larger data sets. All of which our method can benefit and further improve. In comparison to similar methods that employ an action representation, our approach is competitive.

5 CONCLUSIONS

This work presented and analyzed a proposal to learn 2D representations from videos using an autoencoder framework, where the encoder reduces the video to a 3-channel image and the decoder uses it to reconstruct the original video. Thus, the encoder learns a mapping that compresses video information into an image, which allows input to 2D CNN, providing end-to-end learning for video action recognition with recent deep convolutional neural networks.

Experiments conducted on two well-known challenging data sets, HMDB51 (Kuehne et al., 2013) and UCF101 (Soomro et al., 2012b), demonstrated the importance of prior training of the autoencoder with a proper loss function. The use of the structural dissimilarity index for the autoencoder and subsequent training of the encoder for action classification presented the best results. We included our representation as a third stream and compared it with a strong two-stream baseline architecture that it reveals to add complementary information. This multi-stream network achieved competitive results compared to approaches available in the literature.

Future directions include the investigation of deeper autoencoders that could make use of 3D or LSTM convolutions. Recurrent frameworks are interesting as they allow inputs of variable size for the prediction without the need for retraining and changing the net-

Table 3: Accuracy results for different approaches on HMDB51 and UCF101 data sets.

Method	Accuracy (%)	Accuracy (%)
Liu et al. (2016)	48.40	–
Jain et al. (2013)	52.10	–
Wang and Schmid (2013)	57.20	–
Simonyan and Zisserman (2014b)	59.40	88.00
Peng et al. (2016)	61.10	87.90
Fernando et al. (2015)	61.80	–
Wang et al. (2016b)	62.00	92.40
Shi et al. (2015)	63.20	86.60
Lan et al. (2015)	65.10	89.10
Wang et al. (2015a)	65.90	91.05
Carreira and Zisserman (2017)	66.40	93.40
Peng et al. (2014)	66.79	–
Zhu et al. (2017)	66.80	93.10
Wang et al. (2017c)	68.30	93.40
Feichtenhofer et al. (2017)	68.90	94.20
Bilen et al. (2017)	72.50	95.50
Carreira and Zisserman (2017) (additional training data)	80.70	98.00
Proposed method	64.51	92.56

Table 2: Performance of multi-stream network on three different splits for the UCF101 and HMDB51 data sets

Data set	Accuracy (%)			
	Split 1	Split 2	Split 3	Average
Spatial Stream				
UCF101	85.57	83.64	85.25	84.82
HMDB51	48.69	49.87	50.26	49.61
Temporal Stream				
UCF101	86.17	88.56	87.88	87.54
HMDB51	57.97	59.08	58.43	58.50
Spatio-Temporal Stream				
UCF101	84.88	85.22	84.63	84.91
HMDB51	51.18	49.54	50.07	50.26
Two Streams				
UCF101	91.65	92.07	92.59	92.10
HMDB51	63.59	65.29	64.12	64.34
Three Streams				
UCF101	92.07	92.82	92.78	92.56
HMDB51	64.12	65.03	64.38	64.51

work architecture. The main challenge is to maintain the spatial size of the frames as the deep learning literature goes in the opposite direction, increasing the depth and decreasing the image size.

ACKNOWLEDGMENTS

We thank FAPESP (grant #2017/12646-3), CNPq (grant #305169/2015-7) and CAPES for the financial support. We are also grateful to NVIDIA for the donation of a GPU as part of the GPU Grant Program.

REFERENCES

Ahad, M. A. R., Tan, J. K., Kim, H., and Ishikawa, S. (2012). Motion History Image: Its Variants and Applications. *Machine Vision and Applications*, 23(2):255–281.

Alcantara, M. F., Moreira, T. P., and Pedrini, H. (2013). Motion Silhouette-based Real Time Action Recognition. In *Iberoamerican Congress on Pattern Recognition*, pages 471–478. Springer.

Alcantara, M. F., Moreira, T. P., and Pedrini, H. (2016). Real-Time Action Recognition using a Multilayer Descriptor with Variable Size. *Journal of Electronic Imaging*, 25(1):013020–013020.

Alcantara, M. F., Moreira, T. P., Pedrini, H., and Flórez-Revuelta, F. (2017a). Action Identification using a Descriptor with Autonomous Fragments in a Multilevel Prediction Scheme. *Signal, Image and Video Processing*, 11(2):325–332.

Alcantara, M. F., Pedrini, H., and Cao, Y. (2017b). Human Action Classification based on Silhouette Indexed Interest Points for Multiple Domains. *International Journal of Image and Graphics*, 17(3):1750018_1–1750018_27.

Baumann, F., Lao, J., Ehlers, A., and Rosenhahn, B. (2014). Motion Binary Patterns for Action Recognition. In *International Conference on Pattern Recognition Applications and Methods*, pages 385–392.

Bilen, H., Fernando, B., Gavves, E., and Vedaldi, A. (2017). Action Recognition with Dynamic Image Networks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*.

Carreira, J. and Zisserman, A. (2017). Quo Vadis, Action Recognition? A New Model and the Kinetics Dataset. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 4724–4733. IEEE.

Chong, Y. S. and Tay, Y. H. (2017). Abnormal Event Detection in Videos using Spatiotemporal Autoencoder. In *International Symposium on Neural Networks*, pages 189–196. Springer.

Concha, D., Maia, H., Pedrini, H., Tacon, H., Brito, A., Chaves, H., and Vieira, M. (2018). Multi-Stream Convolutional Neural Networks for Action Recognition in Video Sequences Based on Adaptive Visual Rhythms. In *17th IEEE International Conference on Machine Learning and Applications*.

Cornejo, J. Y. R., Pedrini, H., and Flórez-Revuelta, F. (2015). Facial Expression Recognition with Occlusions based on Geometric Representation. In *Iberoamerican Congress on Pattern Recognition*, pages 263–270. Springer.

Diba, A., Sharma, V., and Van Gool, L. (2017). Deep Temporal Linear Encoding Networks. In *IEEE Conference on Computer Vision and Pattern Recognition*, volume 1.

Donahue, J., Anne Hendricks, L., Guadarrama, S., Rohrbach, M., Venugopalan, S., Saenko, K., and Darrell, T. (2015). Long-Term Recurrent Convolutional Networks for Visual Recognition and Description. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 2625–2634.

Feichtenhofer, C., Pinz, A., and Wildes, R. P. (2017). Spatiotemporal Multiplier Networks for Video Action Recognition. In *Computer Vision and Pattern Recognition*, pages 4768–4777.

Fernando, B., Gavves, E., Oramas, M. J., Ghodrati, A., and Tuytelaars, T. (2015). Modeling Video Evolution for Action Recognition. In *Computer Vision and Pattern Recognition*, pages 5378–5387.

Gammulle, H., Denman, S., Sridharan, S., and Fookes, C. (2017). Two Stream LSTM: A Deep Fusion Framework for Human Action Recognition. In *IEEE Winter Conference on Applications of Computer Vision*, pages 177–186. IEEE.

Goodfellow, I., Bengio, Y., Courville, A., and Bengio, Y. (2016). *Deep Learning*, volume 1. MIT Press Cambridge.

Gori, I., Aggarwal, J. K., Matthies, L., and Ryoo, M. S. (2016). Multitype Activity Recognition in Robot-Centric Scenarios. *IEEE Robotics and Automation Letters*, 1(1):593–600.

Hasan, M., Choi, J., Neumann, J., Roy-Chowdhury, A. K., and Davis, L. S. (2016). Learning Temporal Regularity in Video Sequences. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 733–742.

Hommos, O., Pintea, S. L., Mettes, P. S., and van Gemert, J. C. (2018). Using Phase Instead of Optical Flow for Action Recognition. *arXiv preprint arXiv:1809.03258*.

Jain, M., Jégou, H., and Bouthemy, P. (2013). Better Exploiting Motion for Better Action Recognition. In *Computer Vision and Pattern Recognition*, pages 2555–2562.

Ji, S., Xu, W., Yang, M., and Yu, K. (2013). 3D Convolutional Neural Networks for Human Action Recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 35(1):221–231.

Kahani, R., Talebpour, A., and Mahmoudi-Aznaveh, A. (2017). A Correlation Based Feature Representation for First-Person Activity Recognition. *arXiv preprint arXiv:1711.05523*.

Karpathy, A., Toderici, G., Shetty, S., Leung, T., Sukthankar, R., and Fei-Fei, L. (2014). Large-Scale Video Classification with Convolutional Neural Networks. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 1725–1732.

Khaire, P., Kumar, P., and Imran, J. (2018). Combining CNN Streams of RGB-D and Skeletal Data for Human Activity Recognition. *Pattern Recognition Letters*.

Kiran, B. R., Thomas, D. M., and Parakkal, R. (2018). An Overview of Deep Learning based Methods for Unsupervised and Semi-supervised Anomaly Detection in Videos. *Journal of Imaging*, 4(2):36.

Kuehne, H., Jhuang, H., Stiefelhagen, R., and Serre, T. (2013). HMDB51: A Large Video Database for Human Motion Recognition. In *High Performance Computing in Science and Engineering*, pages 571–582. Springer.

Lan, Z.-Z., Lin, M., Li, X., Hauptmann, A. G., and Raj, B. (2015). Beyond Gaussian Pyramid: Multi-Skip Feature Stacking for Action Recognition. In *Computer Vision and Pattern Recognition*, pages 204–212. IEEE Computer Society.

Liu, L., Shao, L., Li, X., and Lu, K. (2016). Learning Spatio-Temporal Representations for Action Recognition: A Genetic Programming Approach. *IEEE Transactions on Cybernetics*, 46(1):158–170.

Ma, C.-Y., Chen, M.-H., Kira, Z., and AlRegib, G. (2018). TS-LSTM and Temporal-Inception: Exploiting Spatiotemporal Dynamics for Activity Recognition. *Signal Processing: Image Communication*.

Maia, H. A., Figueiredo, A. M. D. O., De Oliveira, F. L. M., Mota, V. F., and Vieira, M. B. (2015). A Video Tensor Self-Descriptor based on Variable Size Block Matching. *Journal of Mobile Multimedia*, 11(1&2):090–102.

Moreira, T., Menotti, D., and Pedrini, H. (2017). First-Person Action Recognition Through Visual Rhythm Texture Description. In *IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 2627–2631. IEEE.

Ng, J. Y.-H., Hausknecht, M., Vijayanarasimhan, S., Vinyals, O., Monga, R., and Toderici, G. (2015). Beyond Short Snippets: Deep Networks for Video Classification. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 4694–4702.

Peng, X., Wang, L., Wang, X., and Qiao, Y. (2016). Bag of Visual Words and Fusion Methods for Action Recognition: Comprehensive Study and Good Practice. *Computer Vision and Image Understanding*, 150:109–125.

Peng, X., Zou, C., Qiao, Y., and Peng, Q. (2014). Action Recognition with Stacked Fisher Vectors. In Fleet, D., Pajdla, T., Schiele, B., and Tuytelaars, T., editors, *European Conference on Computer Vision*, pages 581–595, Cham. Springer International Publishing.

Perez, E. A., Mota, V. F., Maciel, L. M., Sad, D., and Vieira, M. B. (2012). Combining Gradient Histograms using Orientation Tensors for Human Action Recognition. In *21st International Conference on Pattern Recognition*, pages 3460–3463. IEEE.

Phan, H.-H., Vu, N.-S., Nguyen, V.-L., and Quoy, M. (2016). Motion of Oriented Magnitudes Patterns for Human Action Recognition. In *International Symposium on Visual Computing*, pages 168–177. Springer.

Ravanbakhsh, M., Mousavi, H., Rastegari, M., Murino, V., and Davis, L. S. (2015). Action Recognition with Image based CNN Features. *arXiv preprint arXiv:1512.03980*.

Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., Huang, Z., Karpathy, A., Khosla, A., Bernstein, M., Berg, A. C., and Fei-Fei, L. (2015). ImageNet Large Scale Visual Recognition Challenge. *International Journal of Computer Vision*, 115(3):211–252.

Ryoo, M. S. and Matthies, L. (2016). First-Person Activity Recognition: Feature, Temporal Structure, and Prediction. *International Journal of Computer Vision*, 119(3):307–328.

Shi, F., Laganiere, R., and Petriu, E. (2015). Gradient Boundary Histograms for Action Recognition. In *IEEE Winter Conference on Applications of Computer Vision*, pages 1107–1114.

Simonyan, K. and Zisserman, A. (2014a). Two-Stream Convolutional Networks for Action Recognition in Videos. In *Advances in Neural Information Processing Systems*, pages 568–576.

Simonyan, K. and Zisserman, A. (2014b). Two-Stream Convolutional Networks for Action Recognition in Videos. In Ghahramani, Z., Welling, M., Cortes, C., Lawrence, N., and Weinberger, K., editors, *Advances in Neural Information Processing Systems 27*, pages 568–576. Curran Associates, Inc.

Soomro, K., Zamir, A. R., and Shah, M. (2012a). UCF101: A Dataset of 101 Human Actions Classes from Videos in the Wild. *arXiv preprint arXiv:1212.0402*.

Soomro, K., Zamir, A. R., and Shah, M. (2012b). UCF101: A Dataset of 101 Human Actions Classes from Videos in the Wild. *arXiv preprint arXiv:1212.0402*.

Szegedy, C., Vanhoucke, V., Ioffe, S., Shlens, J., and Wojna, Z. (2016). Rethinking the Inception Architecture for Computer Vision. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 2818–2826.

Torres, B. S. and Pedrini, H. (2016). Detection of Complex Video Events through Visual Rhythm. *The Visual Computer*, pages 1–21.

Tran, A. and Cheong, L. F. (2017). Two-Stream Flow-Guided Convolutional Attention Networks for Action Recognition. In *IEEE International Conference on Computer Vision Workshops*, pages 3110–3119.

Varol, G., Laptev, I., and Schmid, C. (2016). Long-Term Temporal Convolutions for Action Recognition. *arXiv preprint arXiv:1604.04494*.

Wang, H., Kläser, A., Schmid, C., and Liu, C.-L. (2011). Action Recognition by Dense Trajectories. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 3169–3176. IEEE.

Wang, H. and Schmid, C. (2013). Action Recognition with Improved Trajectories. In *International Conference on Computer Vision*, pages 3551–3558.

Wang, H., Yang, Y., Yang, E., and Deng, C. (2017a). Exploring Hybrid Spatio-Temporal Convolutional Networks for Human Action Recognition. *Multimedia Tools and Applications*, 76(13):15065–15081.

Wang, L., Ge, L., Li, R., and Fang, Y. (2017b). Three-stream CNNs for Action Recognition. *Pattern Recognition Letters*, 92:33–40.

Wang, L., Ge, L., Li, R., and Fang, Y. (2017c). Three-Stream CNNs for Action Recognition. *Pattern Recognition Letters*, 92(Supplement C):33–40.

Wang, L., Qiao, Y., and Tang, X. (2015a). Action Recognition with Trajectory-Pooled Deep-Convolutional Descriptors. In *Computer Vision and Pattern Recognition*, pages 4305–4314.

Wang, L., Xiong, Y., Wang, Z., and Qiao, Y. (2015b). Towards Good Practices for very Deep Two-Stream Convnets. *arXiv preprint arXiv:1507.02159*.

Wang, L., Xiong, Y., Wang, Z., Qiao, Y., Lin, D., Tang, X., and Van Gool, L. (2016a). Temporal Segment Networks: Towards Good Practices for Deep Action Recognition. In *European Conference on Computer Vision*, pages 20–36. Springer.

Wang, X., Farhadi, A., and Gupta, A. (2016b). Actions Transformations. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 2658–2667.

Wang, Z., Bovik, A. C., Sheikh, H. R., and Simoncelli, E. P. (2004). Image Quality Assessment: From Error Visibility to Structural Similarity. *IEEE Transactions on Image Processing*, 13(4):600–612.

Yeffet, L. and Wolf, L. (2009). Local Trinary Patterns for Human Action Recognition. In *IEEE 12th International Conference on Computer Vision*, pages 492–497. IEEE.

Zeiler, M. D. (2012). ADADELTA: An Adaptive Learning Rate Method. *arXiv preprint arXiv:1212.5701*.

Zhao, H., Gallo, O., Frosio, I., and Kautz, J. (2017a). Loss Functions for Image Restoration with Neural Networks. *IEEE Transactions on Computational Imaging*, 3(1):47–57.

Zhao, Y., Deng, B., Shen, C., Liu, Y., Lu, H., and Hua, X.-S. (2017b). Spatio-Temporal Autoencoder for Video Anomaly Detection. In *ACM on Multimedia Conference*, pages 1933–1941. ACM.

Zhu, Y., Lan, Z., Newsam, S., and Hauptmann, A. G. (2017). Hidden Two-Stream Convolutional Networks for Action Recognition. *arXiv preprint arXiv:1704.00389*.