Comparison of Deep Convolutional Networks for Action Recognition in Videos

Patrícia Kovaleski*, Leonardo Nunes†, Eduardo A. B. da Silva*
*Program of Electrical Engineering, Federal University of Rio de Janeiro
† Advanced Technology Labs, Microsoft
kovaleski@poli.ufrj.br, lnunes@microsoft.com, eduardo@smt.ufrj.br

***Abstract*—This work presents the implementation of deep convolutional networks for action recognition in videos based on the well-known two-stream architecture, that is composed of a temporal and a spatial stream. The development was done in order to replicate the one reported in the original paper using the Microsoft Cognitive Toolkit (CNTK). Different experiments were made in order to evaluate the performance of the two-stream in a public dataset when trained for different base network architectures and input data modality.**

I. Introduction

Action recognition consists in detecting if an action has occurred in a video given a predetermined set of options. Being one of the main challenges in Computer Vision, this task has been a highly researched topic in the academic community due to its wide applications in areas like automated surveillance [1], video indexing [2] and behavior analysis [3].

Unlike the case of image classification, the temporal component present in videos provides important additional information that considerably improve the recognition. For certain actions, data such as intensity and duration can be decisive for the correct classification, collaborating, for example, to distinguish a person walking from a person running. However, the spatial component is also fundamental, especially in more complex actions such as "peeling an apple" or "applying makeup", in which the object being interacted with plays a prominent role. Thus, most current recognition methods make use of spatio-temporal information when classifying an action.

Research in video recognition was significantly driven by advances in image recognition methods, often adapting them to deal with video data. Many approaches performed the extraction of spatio-temporal data features, which were encoded using representations like bag of visual words or their variants, in order to be used for classification [2], [4], [5].

With the return of neural networks to the public interest there were several attempts to develop a deep architecture for action classification. Many works used as input a stack of static frames, expecting the network to implicitly learn to identify spatio-temporal attributes [6]–[8]. Others explored the use of predefined spatial-temporal filters in its first layers [9].

In recent years, Convolutional Neural Networks have achieved a remarkable success in image classification and attempts to expand them for video recognition have emerged [10]–[12]. Some works [1] have proposed the use of 3D convolutions to extract simultaneously features from the spatial and the temporal dimensions. In [11], the use of various convolutional architectures for action recognition using a stack of consecutive static frames as input was compared. Almost no difference was found between the results obtained for a single or multiple input frames, indicating that the features learned did not capture the movement well. Based on these findings, [12] proposes a two-stream architecture in which spatial and temporal information are explored separately by two distinct convolutional networks. The results obtained for this method became the state of the art at the time and many recent works that have achieved superior performance [13]–[15] are based on this architecture.

The two-stream, just like most of the action classification approaches made until recently, considered trimmed videos datasets [16], [17], in which only one action would happen per frame. Untrimmed datasets [18], [19] are considerably more challenging, motivating many of the current efforts made in this area [20]. This paper will focus solely on trimmed actions as it is a building block for more complex action classification tasks.

In this paper, we present a comparison between several deep networks models when trained for action recognition in the two-stream architecture. The models were chosen based on their relevance in the current literature. In this way, we aim to offer an insight on how some popular network architectures perform considering their different computational complexity, parameters, and overall performance trade-offs. After trained, they can be widely used for fine-tuning in many of the recent works that are based on the two-stream, accelerating and improving their training. We also explore the adaptation for a real-time execution of this architecture [21], in which the temporal information is captured by the difference between sequential RGB frames.

II. The Two-Stream Architecture

Videos can be naturally decomposed into a spatial and a temporal component. The spatial part is present in the static images that carry information about objects and scenes. The temporal part describes the movement of the camera and the objects present in the video. Based on this central idea and having deep convolutional networks [22] established as the state of the art for image recognition, is proposed in [12] an architecture formed by two separate streams, spatial and temporal, that would have their results combined at the end.

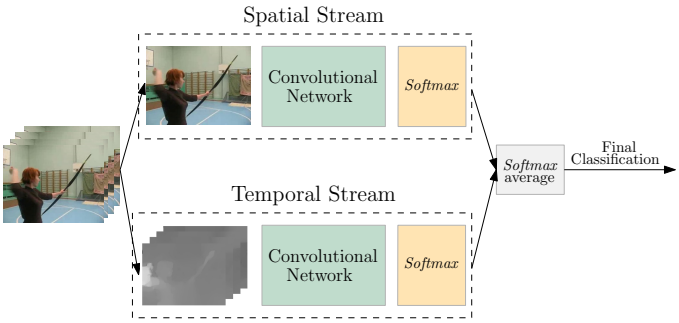

Fig. 1. Two-Stream architecture for video classification.

The two-stream architecture [12] is composed of two independent deep convolutional networks, as shown in Figure 1. The layers organization is the same for both; it corresponds to the CNN-M-2048 [23] architecture and is similar to the network presented in [24]. The final result for the two-stream architecture is obtained simply by averaging the individual results of each of its networks.

The spatial convolutional network receives individual video frames as input and performs action recognition based on still images, acting as an image classification network. The temporal convolutional network, on the other hand, receives as input multiple sequential frames with motion information. By selecting a sequential set of these frames, the motion along a small duration of the video is captured, thus making the identification of the action easier. The motion is described as the dense optical flow vectors [25] between two frames.

Recent works have explored other possible substitute inputs for the temporal stream since the computation of dense optical flow fields is impractical in real-time situations. In [21], the use of the difference of consecutive RGB frames as replacement for the optical flow is proposed. The RGB difference between consecutive frames can be considered a noisy estimate of the flow, indicating the regions where significant change is happening. A comparison between the different types of input data for the convolutional networks is shown in Figure 2.

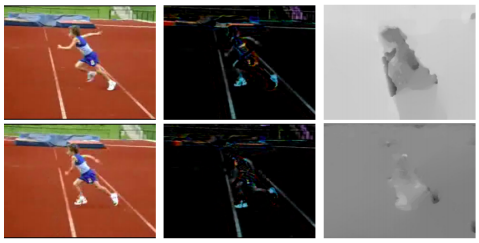

Fig. 2. Comparison between the three different types of input data: RGB (left), RGB difference (center) and optical flow (right). *Source:* [21]

III. Implementation

In this section we present the main topics related to this project implementation, the toolkit used, the base network architectures selected and the parameters and particularities of the training procedure.

A. The CNTK Toolkit

The Microsoft Cognitive Toolkit (CNTK), [26], is an open source toolkit developed by Microsoft for the distributed implementation of deep learning. It is widely applied in problems involving computer vision, speech recognition and textual analysis, many of the company's leading products use models trained by this tool. Examples include Skype's simultaneous translation system and the speech models used by Cortana's personal assistant.

The CNTK was used throughout the implementation of this project, from the definition of the models to their training and evaluation.

B. Network Base Architectures

Three main networks architectures were selected to be used as base in this work: VGG [27], Resnet [28] and Densenet [29]. These architectures are widely used in the current computer vision scenario, being the most usual choices in real applications. For each, we have considered its most common versions, totalizing eleven different networks, which are listed in Table I. The different versions of the same architecture vary in the number of layers, denoted by the number next to their name.

C. Training

The training process was similar for all networks, using minibatch stochastic gradient descent with momentum set to 0.9 and minibatch size of 256 [30]. In all cases, a data augmentation scheme composed of random crop of size 224×224 and horizontal flip was applied during the training. Particularities of each case are detailed bellow. The training was done in parallel using 4 to 8 Tesla P100 GPUs.

Spatial Network. Since the spatial network takes RGB images as input, pre-trained models on ImageNet [22] were used to initialize the network weights. The learning rate was set to 0.1 and decreased to 0.01 after 14000 mini-batches. The training stops after 3 epochs (approximately 20000 mini-batches).

Temporal Network with Optical Flow. Since the optical flow data is considerably different from that used in available pre-trained models, these networks were trained from scratch. To avoid over-fitting a dropout rate of 0.9 was applied. The learning rate was set to 0.1 and decreased to 0.01 and 0.001 after 50000 and 70000 mini-batches, respectively. The training stops after 24 epochs, that corresponds approximately to 80000 minibatches.

A 'pre-input' layer is added to the base network to deal with the sequence requirement of the multiple optical flow frames input. This layer is responsible for reading the frames in the correct order, applying the needed transformations and stacking then in an appropriate format to the base network. The mean frame value its subtracted, compensating for the global camera movement.

Temporal Network with RGB Difference. Pre-trained models on ImageNet were used to initialize the network weights, as sugested in [21]. The learning rate was set to 0.1

and decreased to 0.01 after 14000 mini-batches. The training stops after 3 epochs (approximately 20000 minibatches).

A pre-input layer is also added to the base network in this case. Its main goal is, besides ensuring the frame order, subtracting each two sequential frames to obtain the RGB difference between them.

IV. Evaluation

In this section we introduce the dataset used for training and evaluation of the models among with the metrics applied to measure their performance. We also show and discuss the results obtained.

A. Dataset and Metrics

The evaluation is performed in the UCF-101 [16] action dataset that consists of 101 different human actions distributed over 13320 clips. The clips are trimmed around the action and have on average 180 frames. This was one of the first datasets to have a large number of classes and to handle realistic videos, that usually contains camera movement, lightning change and partial occlusion. Because of this, it has been widely used for validation in action recognition research. There are currently larger and more challenging datasets than the UCF-101, such as those presented in [18], [19], [31]. However, it remains a very popular dataset and is widely used to benchmark the performance of different action recognition algorithms [14], [15], [21].

The evaluation is done using the first split and the performance is measured by the mean average accuracy over the classes. Following the testing scheme of [12], given a video, a set of 25 frames equally spaced in time is selected. For each frame 10 inputs are obtained by cropping a 224×224 area from its center and each of the four corners from the original and the flipped image. The final result for the video is then obtained by averaging the individual results of the sample frames and their crops.

B. Results

The results obtained for each stream individually and the combined result for the two-stream architecture are described in Table I. As expected the joint result of the spatial and temporal streams considerably increase the final performance. The use of RGB difference as temporal information was also validated, showing that this data carries complementary information relative to the spatial stream.

We can see that the performance of the spatial stream and the temporal stream with RGB difference increases with the depth of each network architecture. However, this increment is not seen in the temporal stream with optical flow, which had its results almost constant for all cases. This suggests an important characteristic of these optical flow networks, neither their architecture nor increasing the number of parameters influence their final performance.

Using different training methods or modifications in the two-stream architecture can lead to better results, as shown in many recent works [13], [15], [21], [32]. However, in the few works that published the performance of more than one of these architectures, the optical flow stream behavior remains quite similar. In Table II are the results obtained for the temporal stream in [32]. We can see that there is no direct relation between depth and better performance and that the results are also very similar for different architectures. Note that in [32] the training was done following the framework proposed in [21], which is an evolution of the two-stream architecture that significantly improves the results.

TABLE I
Results obtained for each network in the spatial and temporal streams and the two-stream architecture. The best result for each stream is in bold.

	Spatial	**Temporal**		**Two-Stream**	
Architecture (# parameters)	**RGB**	**OF**	**Diff**	**RGB + OF**	**RGB + Diff**
Resnet18 (12M)	76.7%	83.0%	79.4%	89.5%	85.1%
Resnet34 (22M)	78.6%	82.0%	80.2%	89.1%	86.1%
Resnet50 (26M)	80.6%	82.5%	83.5%	90.9%	87.5%
Resnet101 (44M)	82.0%	83.0%	84.8%	91.1%	88.8%
Resnet152 (60M)	82.3%	82.1%	85.7%	**91.3%**	89.4%
Densenet121 (8M)	78.7%	83.0%	84.2%	90.0%	87.6%
Densenet169 (14M)	80.5%	82.5%	84.4%	90.6%	88.3%
Densenet201 (20M)	82.3%	82.8%	85.8%	90.6%	88.9%
Densenet264 (33M)	**82.6%**	82.4%	**85.9%**	90.9%	**89.4%**
VGG16 (138M)	75.4%	**83.7%**	75.2%	89.4%	80.9%
VGG19 (144M)	74.5%	83.6%	76.3%	89.0%	81.5%

TABLE II
Results for the temporal stream using optical flow published for different network architectures. The results in [32] employ the Temporal Segment Networks (TSN) [21].

	Temporal stream (OF)	
Architecture	**[32] with TSN [21]**	**Ours**
Resnet50	86.8%	82.5%
Resnet101	88.7%	83.0%
Resnet152	87.3%	82.1%
VGG16	85.9%	83.7%

Now looking at the two-stream performance when using optical flow and RGB difference we note that the gap between them is around 3 percentage points. This is very significant considering that the RGB difference is a quite simple way to execute the two-stream architecture in real-time. Many applications of action recognition have real-time requirement, making the RGB difference an acceptable choice for these cases. In Table III we compare the execution time of some architectures using a minibatch size of 25. The speed is measure by frames per second (fps) in a Tesla K80 GPU.

Another important comparison to do, especially when dealing with real-world applications, is the number of learnable parameters in the network. Basically, the more parameters you have more expensive will be the training and the execution of the network. It also requires more attention during training to

avoid over-fitting. In Table I we have the approximate number of parameters of each network. Taking this into account we see that the VGG models tend to be the most unsuitable for real-world applications having an order of magnitude more parameters than the other models besides achieving the lower results.

TABLE III
EXECUTION TIME OF TWO-STREAM MODELS IN A K80 GPU.

Architecture	RGB + OF	RGB + Diff
Resnet18	13.4 fps	126.0 fps
Resnet50	11.8 fps	63.3 fps
Resnet152	10.2 fps	27.2 fps
Densenet121	12.8 fps	67.9 fps
Densenet264	10.7 fps	31.9 fps
VGG16	11.7 fps	41.7 fps
VGG19	11.2 fps	36.8 fps

V. CONCLUSION

In this paper we presented a comparison between several deep convolutional networks when used in the two-stream architecture. We explored the adaptation for the temporal stream proposed in [21] for a real-time implementation. The use of RGB difference as motion information leads to performances only 3% below, on average, to the one of the original optical flow data. Evaluating each stream individually, we have shown that there is a performance improvement with the depth of the networks for both the spatial stream and temporal stream with RGB difference. However, the network architecture does not have much effect on the temporal stream with optical flow, that presented an almost constant performance for all the networks. This shows a characteristic of the optical flow data, which indicates that leaning in deeper versions of the known networks does not lead to performance improvements.

In future works we intend to study some of the recent research that explores the use of different training procedures, architecture approaches and forms of representations of motion data. As we have shown, an approach with a different perspective of the action recognition task is needed to make the next big step towards its resolution.

ACKNOWLEDGMENT

This work is supported by CNPq (132979/2018-7) and Microsoft. This study was financed in part by the Coordenação de Aperfeiçoamento de Pessoal de Nível Superior - Brasil (CAPES) - Finance Code 001.

REFERENCES

[1] S. Ji, W. Xu, M. Yang, and K. Yu, "3d convolutional neural networks for human action recognition," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 35, no. 1, pp. 221–231, Jan 2013.

[2] I. Laptev, M. Marszalek, C. Schmid, and B. Rozenfeld, "Learning realistic human actions from movies," in *2008 IEEE Conference on Computer Vision and Pattern Recognition*, 2008, pp. 1–8.

[3] A. S. Jahagirdar and M. S. Nagmode, "Video based action recognition for behavior understanding - a study," *Digital Image Processing*, vol. 7, no. 10, pp. 293–300, 2015.

[4] H. Wang, A. Klser, C. Schmid, and C.-L. Liu, "Action recognition by dense trajectories," in *CVPR 2011*, 2011, pp. 3169–3176.

[5] H. Wang and C. Schmid, "Action recognition with improved trajectories," in *Proc. ICCV*, 2013, pp. 3551–3558.

[6] C. Desai, D. Ramanan, and C. Fowlkes, "Discriminative models for static human-object interactions," in *2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition - Workshops*, June 2010.

[7] V. Delaitre, I. Laptev, and J. Sivic, "Recognizing human actions in still images: a study of bag-of-features and part-based representations." in *British Machine Vision Conference 2010*, 2010, pp. 1–11.

[8] L. Li and L. Fei-Fei, "What, where and who? classifying events by scene and object recognition," in *2007 IEEE 11th International Conference on Computer Vision*, Oct 2007, pp. 1–8.

[9] H. Jhuang, T. Serre, L. Wolf, and T. A. Poggio, "A biologically inspired system for action recognition," in *2007 IEEE 11th International Conference on Computer Vision*, 2007, pp. 1–8.

[10] S. Ji, W. Xu, M. Yang, and K. Yu, "3d convolutional neural networks for human action recognition," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 35, no. 1, pp. 221–231, 2013.

[11] A. Karpathy, G. Toderici, S. Shetty, T. Leung, R. Sukthankar, and L. Fei-Fei, "Large-scale video classification with convolutional neural networks," in *Proc. CVPR*, 2014, pp. 1725–1732.

[12] K. Simonyan and A. Zisserman, "Two-Stream Convolutional Networks for Action Recognition in Videos," *ArXiv e-prints*, Jun. 2014.

[13] L. Wang, Y. Xiong, Z. Wang, and Y. Qiao, "Towards good practices for very deep two-stream convnets," *ArXiv e-prints*, 2015.

[14] B. Zhang, L. Wang, Z. Wang, Y. Qiao, and H. Wang, "Real-time Action Recognition with Enhanced Motion Vector CNNs," *ArXiv e-prints*, 2016.

[15] C. Feichtenhofer, A. Pinz, and A. Zisserman, "Convolutional two-stream network fusion for video action recognition," in *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

[16] K. Soomro, A. R. Zamir, and M. Shah, "UCF101: A dataset of 101 human actions classes from videos in the wild," *CoRR*, 2012.

[17] H. Kuehne, H. Jhuang, E. Garrote, T. Poggio, and T. Serre, "HMDB: a large video database for human motion recognition," in *Proceedings of the International Conference on Computer Vision (ICCV)*, 2011.

[18] W. Kay, J. Carreira, K. Simonyan, B. Zhang, C. Hillier, S. Vijayanarasimhan, F. Viola, T. Green, T. Back, P. Natsev, M. Suleyman, and A. Zisserman, "The kinetics human action video dataset," *CoRR*, vol. abs/1705.06950, 2017.

[19] C. Gu, C. Sun, S. Vijayanarasimhan, C. Pantofaru, D. A. Ross, G. Toderici, Y. Li, S. Ricco, R. Sukthankar, C. Schmid, and J. Malik, "AVA: A video dataset of spatio-temporally localized atomic visual actions," *CoRR*, vol. abs/1705.08421, 2017.

[20] S. Bandla and K. Grauman, "Active learning of an action detector from untrimmed videos," in *2013 IEEE International Conference on Computer Vision*, 2013, pp. 1833–1840.

[21] L. Wang, Y. Xiong, Z. Wang, Y. Qiao, D. Lin, X. Tang, and L. Val Gool, "Temporal segment networks: Towards good practices for deep action recognition," in *ECCV*, 2016.

[22] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Proceedings of Neural Information Processing Systems (NIPS)*, 2012, pp. 1106–1114.

[23] K. Chatfield, K. Simonyan, A. Vedaldi, and A. Zisserman, "Return of the Devil in the Details: Delving Deep into Convolutional Nets," *ArXiv e-prints*, May 2014.

[24] M. D. Zeiler and R. Fergus, "Visualizing and understanding convolutional networks," *CoRR*, vol. abs/1311.2901, 2013.

[25] T. Brox, A. Bruhn, N. Papenberg, and J. Weickert, "High accuracy optical flow estimation based on a theory for warping," in *European Conference on Computer Vision (ECCV)*, May 2004, pp. 25–36.

[26] Microsoft. The microsoft cognitive toolkit. [Online]. Available: https://docs.microsoft.com/en-us/cognitive-toolkit/

[27] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *CoRR*, vol. abs/1409.1556, 2014.

[28] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," *CoRR*, vol. abs/1512.03385, 2015.

[29] G. Huang, Z. Liu, and K. Q. Weinberger, "Densely connected convolutional networks," *CoRR*, vol. abs/1608.06993, 2016.

[30] N. Qian, "On the momentum term in gradient descent learning algorithms," *Neural Networks*, vol. 12, no. 1, pp. 145–151, 1999.

[31] M. Monfort, B. Zhou, S. A. Bargal, T. Yan, A. Andonian, K. Ramakrishnan, L. Brown, Q. Fan, D. Gutfruend, C. Vondrick *et al.*, "Moments in time dataset: one million videos for event understanding."

[32] Z. Liu, X. Zhang, L. Song, Z. Ding, and H. Duan, "More efficient and effective tricks for deep action recognition," 11 2017.