Using Scene Context to Improve Action Recognition

Juarez Monteiro[1], Roger Granada[1], Felipe Meneguzzi[2], and Rodrigo C. Barros[2]

School of Technology - Pontifícia Universidade Católica do Rio Grande do Sul
Av. Ipiranga, 6681, 90619-900, Porto Alegre, RS, Brazil
[1] Email: {juarez.monteiro, roger.granada}@acad.pucrs.br
[2] Email: {felipe.meneguzzi, rodrigo.barros}@pucrs.br

Abstract. Recently action recognition has been used for a variety of applications such as surveillance, smart homes, and in-home elder monitoring. Such applications usually focus on recognizing human actions without taking into account the different scenarios where the action occurs. In this paper, we propose a two-stream architecture that considers not only the movements to identify the action, but also the context scene where the action is performed. Experiments show that the scene context may improve the recognition of certain actions. Our proposed architecture is tested against baselines and the standard two-stream network.

Keywords: Action Recognition · Convolutional Neural Networks · Neural Networks

1 Introduction

Action recognition is one of the promising tasks in the computer vision area and has been employed in many tasks such as surveillance and assistance of the sick and disabled. Although recognizing actions is a trivial task for the human being, the automation of such task is particularly challenging in the real physical world, since it involves understanding the not only the movements that are being performed but also the context in which the action is happening. In this sense, the contextual information plays an important role, giving cues to disambiguate actions that are performed with the same movements. For example, observing a scene context, a human being can easily identify whether a swing movement is being performed in a tennis or a baseball match. To perform such task autonomously, we need an approach that is able to identify not only the moving parts of the image, but also the background of the image to identify the scene context.

In this paper, we address the problem of recognizing actions from videos by using a two-stream architecture, where two convolutional neural networks run in parallel, merging their features in a late fusion approach. Inspired by Silva *et al.* [12] that improve the object recognition by using the scene context, we build a two-stream neural network architecture where a stream performs the action recognition and another stream improves this recognition by identifying

the context where the action occurs. We perform experiments using our approach in two datasets for action recognition and compare our results with baselines and the state-of-the-art approaches.

This paper is organized as follows. Section 2 describes the related work and introduces the standard two-stream architecture and how it has been employed so far. Section 3 details our deep neural architecture based on two-stream for action recognition, whereas Section 4 presents all settings and data we use for assessing the performance of our proposed approach. Results are presented and discussed in Section 5 along with a comparison with baselines and the state-of-the-art results for each dataset. We finish this paper with our conclusions and future work directions in Section 6.

2 Related Work

Advances in hardware and greater availability of data have allowed deep learning algorithms such as Convolutional Neural Networks (CNNs) [5] to consistently improve on the state-of-the-art results when dealing with image-based tasks such as object recognition [7], detection, and semantic segmentation [3]. Extensions of CNN representations to the action recognition task in videos have been proposed in several recent works [8, 13, 15]. For example, Wang *et al.* [15] apply dynamic tracking attention model (DTAM), which is composed by a CNN and a Long-Short Term Memory (LSTM) to perform human action recognition in videos. Their architecture uses the CNN to extract features from images and the LSTM to deal with the sequential information of the actions. DTAM uses local dynamic tracking to identify moving objects, and global dynamic tracking to estimate the motion of the camera and correct the weights of the motion attention model.

Simonyan and Zisserman [13] propose the two-stream convolutional network architecture, which is composed by two streams running in parallel with a late fusion to merge both streams. The idea behind the two-stream is to mimic the visual cortex, which contains the ventral stream (responsible for object recognition) and the dorsal stream (responsible for recognizing motion) as two separate pathways. Thus, videos can be decomposed into spatial and temporal components: the spatial one that carries information about scene context, and the temporal one that conveys the motion across frames, indicating the movement of the observer and objects. Simonyan and Zisserman use the raw images in the spatial stream and pre-computed optical flow features in the temporal stream. Using a two-stream architecture with two different CNNs, Monteiro *et al.* [8] perform action recognition in a small egocentric dataset. They affirm that a two-stream architecture achieves better results than a single stream because each stream extracts different features from the same image. The extracted features are then merged by a late score fusion using a Support Vector Machine (SVM).

Scene recognition is a fundamental problem in computer vision and recently has been receiving an increasing attention [12, 16, 17]. As Wang *et al.* [16] affirm, a scene provides rich semantic information of the global structure providing a meaningful context. As scene context, we can understand as the place in which

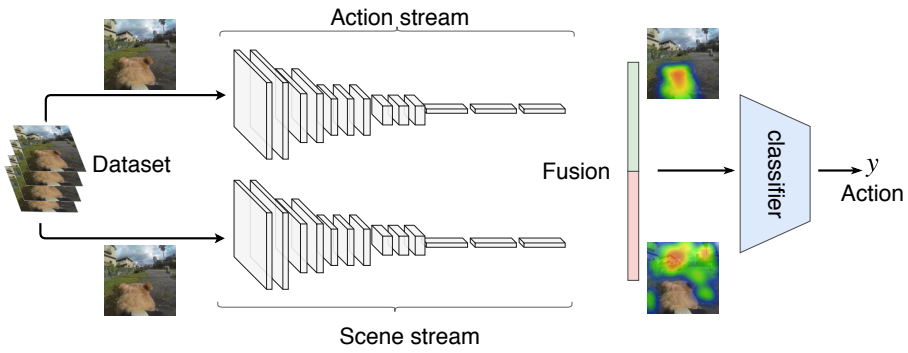

Fig. 1. Two-stream architecture composed by a stream to recognize actions and a stream to recognize the context scene with a late fusion.

the objects seat, *i.e.*, the background environment where actions occur. Unlike Monteiro *et al.* [8] and Simonyan and Zisserman [13], in this work, we associate the identification of the action with the scene context since we believe that the scene context may give interesting clues about the action that is being performed.

3 Recognizing Actions with Scene Context

To address the contextual awareness on action recognition, our approach aims to use the context of the scenes by fusing the information of the background with the information that identifies the action being performed in a two-stream architecture [13]. Our architecture is composed by a stream containing a CNN to identify the action happening in the current frame, and a CNN to identify where (scene context) the action is happening in the current frame, as illustrated in Figure 1. The idea of this architecture is that the CNN responsible for the action stream focuses on the movements that are being performed to identify an action, while the CNN responsible for the scene stream focuses on the background where the action happens. For example, consider two actions that contain similar movements, such as baseball swing and tennis swing. While the action stream may identify the swing performed in the action, the scene stream identifies the context where the swing is happening, increasing the chance to correctly classify the action. Features from both streams are connected in a late fusion approach and a classifier predicts the action performed on the input image.

Although our architecture allows different networks in each stream, we use two VGG-16 networks [14]. In order to extract features from the background environment (Scene stream), we use the weights of a CNN pre-trained using the Places365 [17] dataset, which contains only images with scenes. For the Action stream, we fine-tune a version of the VGG-16 with the weights pre-trained on the 1.2-million-image ILSVRC 2012 ImageNet dataset [10]. Finally, we train a multi-class Support Vector Machine (SVM) using the concatenation of both streams for the final classification.

Fig. 2. Examples of frames from each class of DogCentric (a) and UCF-11 (b) datasets.

4 Experiments

In this section, we describe the datasets and the main implementation details applied to our experiments.

4.1 Dataset

Our experiments are performed using two freely available datasets that contain a single action in each video. We select the datasets because they contain different characteristics: dataset containing an egocentric viewpoint of actions performed by dogs and a dataset containing a third-person viewpoint performed by humans. We detail each dataset as follows.

DogCentric Activity dataset[1] [4] consists of 209 videos containing 10 different action performed by 4 dogs as illustrated in Figure 2 (a). The dataset contains first-person videos taken from an egocentric animal viewpoint, *i.e.*, a wearable camera mounted on dogs' back records outdoor and indoor scenes, which are very challenging due to their strong camera motion. Following Monteiro *et al.* [9], we randomly select half of the videos of each action to the test set and the rest of the videos are separated into training and validation sets. Validation set contains 20% of the videos and the rest is separated to the training set.

UCF YouTube Action dataset[2] (hereafter called UCF-11) [6] consists of 1,600 videos extracted from YouTube containing 11 actions as illustrated in Figure 2 (b). Each video has 320×240 pixels and was converted to a frame rate of 29.97 fps and annotations were done accordingly, containing a single action associated with the entire video. As performed by Monteiro *et al.* [9], we divided the UCF-11 dataset into train, validation and test sets.

4.2 Network settings

In this work, we only train the Action stream, since we use the Scene stream as a feature extractor from a pre-trained version of the CNN. All deep models

[1] http://robotics.ait.kyushu-u.ac.jp/~hyumi/db/first_dog.html
[2] http://crcv.ucf.edu/data/UCF_YouTube_Action.php

developed in this work (including baselines) are implemented using *Keras*[3] and *TensorFlow*[4] frameworks. We pre-trained the Action stream CNN using the ImageNet dataset with weights being directly loaded from Keras core library. Training phase performs iterations using mini-batches of 128 images, applying mini-batch stochastic gradient with momentum (0.9), and using rectified linear unit (ReLU) as the activation of each convolution. We subtract all pixels from each image by the mean of each pixel from all training images. For all networks, we perform hyperparameter optimization using a grid search for *dropout* on the fully-connected layers and *learning rate* hyperparameters, since they are commonly changed when trying to learn a deep model. Due to space constraints, we show the results only for the setting that achieves the highest results in validation data. The best configuration for the Action stream contains 0.5 of *dropout* and 5e-4 of *learning rate* for the UCF-11 dataset and 0.95 of *dropout* and 5e-3 of *learning rate* for the DogCentric dataset. We limit the number of epochs to 30 with applying early stopping, where most of our experiments took no longer than 15 epochs to finish. The Scene stream contains a VGG-16 using weights of a CNN[5] pre-trained in the Places-365 [17] dataset. For the classification phase, we use the Crammer and Singer [1] implementation of the SVM from *scikit-learn*[6] with the default parameters.

4.3 Baselines

As deep learning approaches have become the state-of-the-art of different computer vision tasks [3, 7, 8, 15], we use as baselines the single stream networks, *i.e.*, only the Action stream and only the Scene stream, and the standard two-stream [13] configuration containing the Action stream and a Temporal stream. For the standard two-stream baseline, the Action stream has the same configuration of the Action stream in our approach. The Temporal stream in the two-stream baseline contains a dense optical flow representation [2] of adjacent frames, *i.e.*, vectors containing both horizontal and vertical displacements, regarding all points within frames. In order to generate the final image for each sequence of frames, we combine the 2-channel optical flow vectors and associate color to their magnitude and direction. Magnitudes are represented by colors and directions through hue values. The training phase of the Temporal stream follows the same settings as the Action stream, but the hyperparameters selected by grid search (*dropout* and *learning rate*). The best configuration for the Temporal stream contains 0.5 of *dropout* and 5e-3 of *learning rate* for both datasets. Due to space constraints, we do not insert in the paper the results achieved with the validation set, but these results, as well as our code, are freely available on our project's website[7].

[3] https://keras.io
[4] https://www.tensorflow.org
[5] https://github.com/GKalliatakis/Keras-VGG16-places365
[6] http://scikit-learn.org/
[7] https://github.com/jrzmnt/PlacesInAction

Table 1. Accuracy ($\mathcal{A}$), Precision ($\mathcal{P}$), Recall ($\mathcal{R}$) and F-measure ($\mathcal{F}$) achieved by the baselines, our approach and the state-of-the-art results.

		$\mathcal{A}$	$\mathcal{P}$	$\mathcal{R}$	$\mathcal{F}$
UCF-11	Action stream	0.71	0.70	0.71	0.70
	Scene stream	0.69	0.68	0.69	0.67
	Two-stream (as in [13])	0.71	0.73	0.71	0.71
	Our approach	**0.75**	**0.78**	**0.75**	**0.75**
	DTAM [15]	0.90	–	–	–
	Visual-DTAM [15]	0.91	–	–	–
DogCentric	Action stream	0.54	0.59	0.54	0.54
	Scene stream	0.48	0.54	0.48	0.48
	Two-stream (as in [13])	0.54	0.62	0.54	0.54
	Our approach	**0.53**	**0.58**	**0.53**	**0.53**
	PoT+ITF [11]	0.75	–	–	–
	2 CNNs-SVM-PP [8]	0.76	0.74	0.76	0.75

5 Results and Discussion

To evaluate our proposal and compare with baselines, we compared the output of each network using the test set. Table 1 shows the accuracy ($\mathcal{A}$), precision ($\mathcal{P}$), recall ($\mathcal{R}$) and F-measure ($\mathcal{F}$) scores for all experiments in each dataset.

In Table 1, we can see that the combination of Scene stream with Action stream performed by our approach increased the results achieved by the Action stream alone in the UCF-11 dataset. The addition of features from the background (Scene stream) increased the accuracy by 4 percentage points and the precision by 8 percentage points indicating that the context of the scene is helpful to identify the action that is being performed. The best results on this dataset are achieved in *Diving* and *Horse riding* actions, with 85% and 81% of accuracy respectively. On the other hand, the approach achieved the lowest accuracy for *Soccer juggling*, indicating that the action may not be dependent on the background. Checking images of this dataset, we can see that the *Soccer juggling* is performed in different places, thus confirming that the scene context is not relevant to identify this action.

Unlike the results achieved on UCF-11, when testing on the DogCentric dataset, our approach seems to be ineffective. The results achieved in the DogCentric dataset may be justified since usually the actions performed by a dog are not related to a fixed background as the actions performed by different sports. In fact, the actions that have some relation to the background, such as *Car*, where the dog is outside waiting for a car to pass by, the precision achieved 90%. The second highest precision score was achieved by the action *Drink*, where the background always contains a water bowl in which the dog drinks water. Actions that do not depend on the background, such as *Look left*, *Look right* or *Shake* achieve low precision scores 21%, 14% and 17% respectively.

Above the results achieved by our approach in Table 1, we can see that the baselines achieved lower scores for all measures when compared with our approach in the UCF-11 dataset, indicating that the scene context may improve the action recognition. A comparison with the standard two-stream baseline sug-

gests that the identification of the scene plays an important role when compared with the optical flow representation. Due to the camera movement on the back of the dog in DogCentric dataset, the scene identification is not very effective. The same problem occurs with the optical flow generation in a standard two-stream. Therefore, both approaches achieved approximately the same results when compared with the Action stream alone.

Below our approach in Table 1, we present the results achieved by the state-of-the-art for each dataset. Values containing a dash are not reported by the authors. As illustrated in Table 1, the results achieved by our approach are modest when compared with the results achieved by the state-of-the-art. However, it is important to note that our intention in this paper was not to achieve the state-of-the-art results, but instead, verify whether the scene context improves the action recognition. Wang *et al.* [15] achieved the state-of-the-art results for UCF-11 dataset using a combination of visual attention with dynamic tracking attention model (DTAM). Their approach uses a combination of CNN and LSTM as an attention mechanism, in order to capture the temporal aspect of an action. Monteiro *et al.* [8] uses the DogCentric dataset to apply a two-stream approach containing two different CNNs and a post-processing step which consists of smoothing the predicted classes by assigning to a target frame the majority voting of all frames within a window. This smoothing process intends to eliminate a few correctly predicted classes when they are in the middle of other classes.

6 Conclusions and Future Work

In this work, we developed an architecture for action recognition based on a two-stream CNN architecture. Unlike the standard two-stream architecture, our approach includes a stream focusing on recognizing actions and the other stream focusing on recognizing the context of the scene. Finally, we performed a late fusion using the concatenation of the features extracted from both streams. We performed experiments to validate our architecture and showed that our approach achieves better results when compared with the baselines composed by the streams separately, and a standard two-stream using optical flow in the temporal stream. A preliminary analysis demonstrates the importance of taking into account the context where the action occurs. Our intention in this paper was not to achieve the state-of-the-art results, but check whether the scene identification plays an important role in the action recognition. Thus, in a future work, we intend to expand our architecture by changing the action stream by the state-of-the-art algorithm to recognize actions taking into account the temporal aspect of the video.

Acknowledgement

The authors would like to thank CAPES/FAPERGS and Motorola Mobility for partially funding this research. We gratefully acknowledge the support of

NVIDIA Corporation with the donation of the Titan Xp GPU used for this research.

References

1. Crammer, K., Singer, Y.: On the algorithmic implementation of multiclass kernel-based vector machines. JMLR **2**(Dec), 265–292 (2001)
2. Farnebäck, G.: Two-frame motion estimation based on polynomial expansion. In: Proceedings of SCIA'03. pp. 363–370 (2003)
3. Ge, W., Yang, S., Yu, Y.: Multi-evidence filtering and fusion for multi-label classification, object detection and semantic segmentation based on weakly supervised learning. In: Proceedings of CVPR'18. pp. 1277–1286 (2018)
4. Iwashita, Y., Takamine, A., Kurazume, R., Ryoo, M.: First-person animal activity recognition from egocentric videos. In: Proceedings ICPR'14. pp. 4310–4315 (2014)
5. LeCun, Y., Bottou, L., Bengio, Y., Haffner, P.: Gradient-based learning applied to document recognition. Proceedings of the IEEE **86**(11), 2278–2324 (1998)
6. Liu, J., Luo, J., Shah, M.: Recognizing realistic actions from videos "in the wild". In: Proceedings of CVPR'09. pp. 1996–2003 (2009)
7. Lu, C., Su, H., Li, Y., Lu, Y., Yi, L., Tang, C.K., Guibas, L.J.: Beyond holistic object recognition: Enriching image understanding with part states. In: Proceedings of CVPR'18. pp. 6955–6963 (2018)
8. Monteiro, J., Aires, J.P., Granada, R., Barros, R.C., Meneguzzi, F.: Virtual guide dog: An application to support visually-impaired people through deep convolutional neural networks. In: Proceedings of IJCNN'17. pp. 2267–2274 (2017)
9. Monteiro, J., Granada, R., Aires, J.P., Barros, R.: Evaluating the feasibility of deep learning for action recognition in small datasets. In: Proceedings of IJCNN'18. pp. 1596–1603 (2018)
10. Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., Huang, Z., Karpathy, A., Khosla, A., Bernstein, M., Berg, A.C., Fei-Fei, L.: Imagenet large scale visual recognition challenge. International Journal of Computer Vision **115**(3), 211–252 (2015)
11. Ryoo, M.S., Rothrock, B., Matthies, L.: Pooled motion features for first-person videos. In: Proceedings of CVPR'15. pp. 896–904 (2015)
12. Silva, L.P., Granada, R., Monteiro, J., Ruiz, D.: Using scene context to improve object recognition. In: Proceedings of the KDMiLe 2017. pp. 105–112 (2017)
13. Simonyan, K., Zisserman, A.: Two-stream convolutional networks for action recognition in videos. In: Proceedings of NIPS'14. pp. 568–576 (2014)
14. Simonyan, K., Zisserman, A.: Very deep convolutional networks for large-scale image recognition. arXiv preprint arXiv:1409.1556 (2014)
15. Wang, C.Y., Chiang, C.C., Ding, J.J., Wang, J.C.: Dynamic tracking attention model for action recognition. In: Proceedings of ICASSP'17. pp. 1617–1621 (2017)
16. Wang, Z., Wang, L., Wang, Y., Zhang, B., Qiao, Y.: Weakly supervised patchnets: Describing and aggregating local patches for scene recognition. IEEE Transactions on Image Processing **26**(4), 2028–2041 (2017)
17. Zhou, B., Lapedriza, A., Khosla, A., Oliva, A., Torralba, A.: Places: A 10 million image database for scene recognition. IEEE Transactions on Pattern Analysis and Machine Intelligence **PP**(99), 1–14 (2017)