View metadata, citation and similar papers at core.ac.uk
brought to you by CORE
provided by Servicio de Difusión de la Creación Intelectual

CNN–LSTM Architecture for Action Recognition in Videos

Carlos Ismael Orozco[1], María Elena Buemi[2], and Julio Jacobo Berlles[2]

[1] Departamento de Informática, FCE. Universidad Nacional de Salta, Argentina
[2] Departamento de Computación, FCEyN. Universidad de Buenos Aires, Argentina
ciorozco.unsa@gmail.com, {mebuemi,jacobo}@dc.uba.ar

Abstract. Action recognition in videos is currently a topic of interest in the area of computer vision, due to potential applications such as: multimedia indexing, surveillance in public spaces, among others. In this paper we propose a CNN–LSTM architecture. First, a pre-trained VGG16 convolutional neuronal networks extracts the features of the input video. Then, a LSTM classifies the video in a particular class. To carry out the training and the test, we used the UCF-11 dataset. Evaluate the performance of our system using the evaluation metric in accuracy. We apply LOOCV with $k = 25$, we obtain $\sim 98\%$ and $\sim 91\%$ for training and test respectively.

Keywords: Action recognition · Convolutional neural network · Long short-term memory · UCF-11.

1 Introduction

The action recognition problem in videos is of great interest in the area of pattern recognition and computer vision due to its potential applications such as: multimedia indexation, information recovery, patient monitoring and control, automated surveillance in public spaces, among others. The objective of the action recognition systems is to classify each video into the class that represents the action that happens in the video. To this end, the interactions between the subjects and/or objects within it, must be accounted for. This problem has been investigated by other works:

Liu et al. [10] propose a framework for detecting and recognizing human actions. To achieve a robust estimation of the region of interest, they use a combination of optical flow together with a Harris $3D$ edge detector to obtain space-time information from video. Then, with the calculation of the local features SIFT and STIP, they train a universal model background (UBM) for the task in question.

Wang et al. [16] propose a dense trajectory approach. They take dense points in each frame of the video and track based on the optical flow displacement information.

Sharma et al. [13] They propose a model based on attention mechanisms for the task of recognizing actions in videos. They use an LSTM neural network

contemplating the spatial and temporal space of the video.

The objective of this work is to implement a video action recognition system. For this we propose the use of a CNN–LSTM architecture. A convolutional neural network extracts the features of the video while an LSTM neural network classifies the video into a certain category. The work is organized as follows: in the section 2, the general structure of the system is described; in section 3, the database used, the evaluation method, the experiments carried out and the results obtained are described. Finally, in section 4 the conclusions and future work are presented.

2 Our Proposal

In this paper we propose the use of a CNN–LSTM architecture. Figure 1 shows a general scheme of the system in its different stages. Input: the video is normalized for a total of 40 frames. CNN: A pre-trained VGG16 extracts the characteristics of the video by obtaining features of size 40×25088. LSTM: takes each vector from the previous stage and processes it in 320 LSTM units.Finally, the output stage consists of a dense layer with 11 nodes.

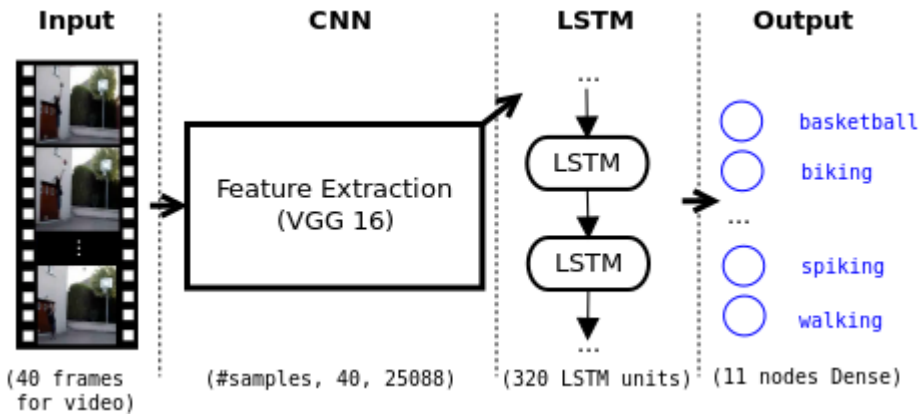

Fig. 1. System architecture. First a CNN VGG16 extracts the features. Then an LSTM classifies a certain class.

2.1 Convolutional Neural Network: VGG16

We use the convolutional architecture of VGG16 proposed by Zisserman et al. [14] because it obtained a very good result in the ImageNet Large-Scale Visual Recognition Challenge (ILSVRC- 2014) Classification and location tasks. Figure 2 shows the layers that make up the architecture.

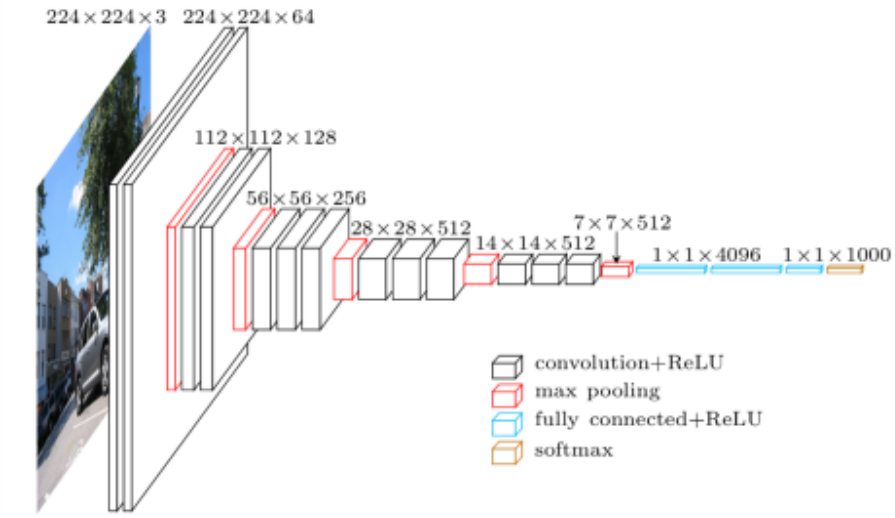

Fig. 2. VGG16 pre-trained implemented in the library Keras [5].

We realized a vectorization of the max pooling of layer 10 obtaining a vector of dimension $7 \times 7 \times 512 = 25088$ to feed the neural network LSTM.

2.2 Long Short-Term Memory

The neural networks LSTMs [7] are a special type of recurrent neural network (RNN) that are formulated in such a way that remembering information for long periods of time is their natural behavior.

We use the LSTM unit proposed by Donahue et al. [6]. Its general structure is shown in Figure 3. The main characteristic of an LSTM unit is a memory cell c which encodes, at each time step, the knowledge of the entries that have been observed up to that moment. The cell is modulated by three types of gates:

1. Input gate (i): Controls whether the current entry is considered (x_t).
2. Forget gate(f): Allows the LSTM to forget the previous memory (c_{t-1})
3. Output gate(o): Decides how much memory will be transferred to the hidden state (h_t)

They are calculated as:

$$i_t = \sigma(W_{xi}x_t + W_{hi}h_{t-1}) \tag{1}$$

$$f_t = \sigma(W_{xf}x_t + W_{hf}h_{t-1}) \tag{2}$$

$$o_t = \sigma(W_{xo}x_t + W_{ho}h_{t-1}) \tag{3}$$

$$c_t = f_t \odot c_{t-1} + i_t \odot \phi(W_{xc}x_t + W_{hc}h_{t-1}) \tag{4}$$

$$h_t = o_t \odot \phi(c_t) \tag{5}$$

Where: σ is the sigmoidal function, ϕ is the hyperbolic tangent, $\odot$ represents the product with the value of the gate and the weights of the matrix denoted by W_{ij}.

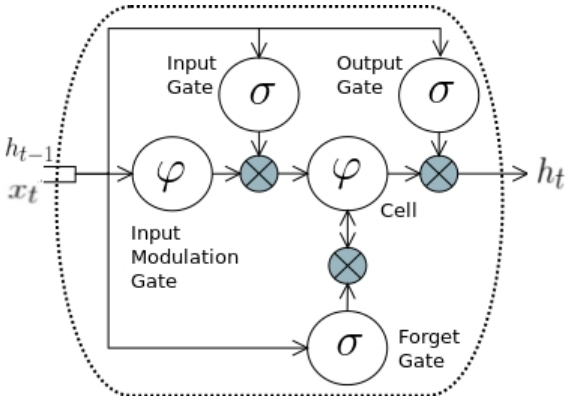

Fig. 3. LSTM unit

3 Experiments and Results

3.1 Database: UCF-11

The experiments carried out in this work use the UFC-11 database proposed by Liu et al. [11]. It is a collection of 1600 videos that belong to one of the following eleven classes: `basketball`, `biking`, `diving`, `golf_swing`, `horse_riding`, `soccer_juggling`, `swing`, `tennis_swing`, `trampoline_jumping`, `volleyball_spiking` and `walking`. Similar to the original setup [11] we use leave one out cross validation (LOOCV) for a per-defined set of 25 folds. Performance measure is calculated by average accuracy over all classes.

3.2 Experiments and Results

Our system was implemented in Python using the library Theano [2,3] on an Intel CORE i7-6700HQ computer with 16GB DDR3 memory and Ubuntu Operating System 16.04. The experiments were carried out on an NVIDIA Titan Xp GPU mounted on a server with a similar configuration.

Table 1 summarizes the results obtained by our system compared to other approaches cited in the bibliography. Our proposed architecture obtain better results.

System	Accuracy
Liu et al. [11]	71.2%
Liu et al. [10]	76.1%
Wang et al. [16]	84.2%
Sharma et al. [13]	85.0%
Cho et al. [4]	88.0%
CNN - LSTM (Our proposal)	**91.94 %**

Table 1. Results of the video classification using the database UCF-11 [11]. The Table summarizes the results obtained by our system comparing with others of the bibliography. Our results are better than those proposed by [11,10,16,13] in terms of accuracy.

4 Conclusions and future work

In this work we implement a video action recognition system, using a CNN–LSTM neural network. First, a VGG16 extracts the characteristics of the video. Then an LSTM neural network classifies the class to which it belongs. It was implemented in Python using the library Theano [2,3], trained and tested using the database [11].

We evaluated the performance of our proposal using the precision evaluation metrics. We obtained $\sim 98\%$ and $\sim 91\%$ for training and testing respectively.

As future work we consider the use of other databases, such as Hollywood2 [12], HMDB [8], UCF-101 [15] to make our system more robust. Another goal is to implement the attention mechanisms proposed by [1,9].

Acknowledgement

This work has been supported by Universidad Nacional de Salta (Proyecto CIUNSa A 2364)

References

1. Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. Neural machine translation by jointly learning to align and translate. *CoRR*, abs/1409.0473, 2014.
2. Frédéric Bastien, Pascal Lamblin, Razvan Pascanu, James Bergstra, Ian J. Goodfellow, Arnaud Bergeron, Nicolas Bouchard, and Yoshua Bengio. Theano: new features and speed improvements. Deep Learning and Unsupervised Feature Learning NIPS 2012 Workshop, 2012.

3. James Bergstra, Olivier Breuleux, Frédéric Bastien, Pascal Lamblin, Razvan Pascanu, Guillaume Desjardins, Joseph Turian, David Warde-Farley, and Yoshua Bengio. Theano: a CPU and GPU math expression compiler. In *Proceedings of the Python for Scientific Computing Conference (SciPy)*, June 2010. Oral Presentation.
4. Jungchan Cho, Minsik Lee, Hyung Jin Chang, and Songhwai Oh. Robust action recognition using local motion and group sparsity. *Pattern Recognition*, 47(5):1813 – 1825, 2014.
5. François Chollet et al. Keras. https://keras.io, 2015.
6. Jeff Donahue, Lisa Anne Hendricks, Sergio Guadarrama, Marcus Rohrbach, Subhashini Venugopalan, Kate Saenko, and Trevor Darrell. Long-term recurrent convolutional networks for visual recognition and description. *CoRR*, abs/1411.4389, 2014.
7. Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural Comput.*, 9(8):1735–1780, November 1997.
8. 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.
9. Natsuda Laokulrat, Sang Phan, Noriki Nishida, Raphael Shu, Yo Ehara, Naoaki Okazaki, Yusuke Miyao, and Hideki Nakayama. Generating video description using sequence-to-sequence model with temporal attention. In *COLING 2016, 26th International Conference on Computational Linguistics, Proceedings of the Conference: Technical Papers, December 11-16, 2016, Osaka, Japan*, pages 44–52, 2016.
10. D. Liu, M. Shyu, and G. Zhao. Spatial-temporal motion information integration for action detection and recognition in non-static background. In *2013 IEEE 14th International Conference on Information Reuse Integration (IRI)*, pages 626–633, Aug 2013.
11. Jingen Liu, Jiebo Luo, and Mubarak Shah. Recognizing realistic actions from videos in the wild. pages 1996 – 2003, 07 2009.
12. M. Marszalek, I. Laptev, and C. Schmid. Actions in context. In *2009 IEEE Conference on Computer Vision and Pattern Recognition*, pages 2929–2936, June 2009.
13. Shikhar Sharma, Ryan Kiros, and Ruslan Salakhutdinov. Action recognition using visual attention. *CoRR*, abs/1511.04119, 2015.
14. K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. *CoRR*, abs/1409.1556, 2014.
15. Khurram Soomro, Amir Roshan Zamir, and Mubarak Shah. UCF101: A dataset of 101 human actions classes from videos in the wild. *CoRR*, abs/1212.0402, 2012.
16. H. Wang, A. Klser, C. Schmid, and C. Liu. Action recognition by dense trajectories. In *CVPR 2011*, pages 3169–3176, June 2011.