GIBIS at MediaEval 2018: Predicting Media Memorability Task

Ricardo Manhães Savii[1,2], Samuel Felipe dos Santos[1], and Jurandy Almeida[1]

[1]GIBIS Lab, Instituto de Ciência e Tecnologia, Universidade Federal de São Paulo, Brazil
[2]Dafiti Group, Brazil
ricardo.savii@dafiti.com.br,{felipe.samuel,jurandy.almeida}@unifesp.br

ABSTRACT

This paper describes the GIBIS team experience in the *Predicting Media Memorability Task* at MediaEval 2018. In this task, we were required to develop an approach to predict a score reflecting whether videos are memorable or not, considering short-term memorability and long-term memorability. Our proposal relies on different learning strategies: for long-term memorability, we adopted k-NN regressors trained on hand-crafted motion features; and for short-term memorability, we trained deep learning models.

KEYWORDS

Media Memorability, k-NN Regressor, Deep Learning

1 INTRODUCTION

The Predicting Media Memorability task is part of the MediaEval 2018 Benchmarking Initiative for Multimedia Evaluation. The goal of this task is to automatically predict a memorability score for a video reflecting its probability to be remembered. For this, it is provided a dataset composed of 10,000 short, soundless videos split into 8,000 videos for the development set and 2,000 videos for the test set. Also, pre-computed visual features are provided to facilitate participation. For more details about this task, please, refer to [2].

In this paper, we explore two main approaches: (1) for long-term memorability, an ensemble of ten KNR (k-Nearest Neighbor Regressor) or SVR (Support Vector Regression) [4] trained on the provided HMP (Histogram of Motion Patterns) [1] feature; and (2) for short-term memorability, a deep learning model based on 3D convolutions and 3D pooling layers, known as C3D (Convolution3D) [5].

2 OUR APPROACH

The proposed approach exploits different strategies for long-term and short-term memorability. The former relies on hand-crafted motion features extracted with HMP whereas the latter uses data-driven features learned with C3D. One limitation of C3D is its capacity to capture subtle but long-term motion dynamics, as it requires to break a video into small clips. Unlike C3D, HMP captures motion dynamics of a video as a whole, and not just parts.

2.1 Long-Term Approach

Our proposed approach for the long-term memorability subtask consists of using the pre-computed HMP feature [1] in conjunction with two regression algorithms: SVR (Support Vector Regression) and KNR (k-Nearest Neighbor Regressor).

HMP encodes an entire video into a single histogram representing its overall motion dynamics. From this, we can consider the HMP vector as a hash identifying each video as a point in a high-dimensional space. This idea of space is the foundation for the use of the KNR and SVR algorithms [4].

For reproducibility, the KNR and SVR implementations used comes from the scikit-learn python package [4]. The steps for the experiment are: split the dev-set at random into ten folds, then train one regression model on each fold. In this way, we get ten different models. They are used as an ensemble to predict the memorability over the HMP features of the test-set. The average output is considered as the final score and we used the 95% confidence interval as the output confidence.

2.2 Short-Term Approach

For short-term memorability, we use a deep learning model based on the C3D architecture [5]. However, our C3D model has some differences from the original C3D model. Here, we include a multi-headed layer by adding two fully connected layers at the top of the C3D model. To provide confidence over prediction values, we implemented a multi-output model, a two-headed model. The heads are: (1) a regression output (i.e., sigmoid activation) used to predict the memorability score; and (2) a classification head predicting the discretized memorability bucket. The short-memorability score was discretized in 10 buckets and used as classes for prediction. In this way, the classification head using a softmax activation provides a confidence value over the responses of the regression head.

The implemented C3D model follows a 3D convolution and 3D max pooling architecture[1] and it outputs a fully connected layer with 2048 neurons. This is the first of three fully connected layers that feed a multi-head output for regression and classification. Figure 1 shows the network architecture of our first experiment.

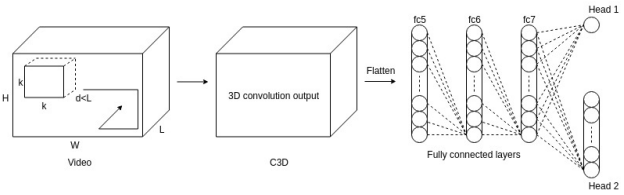

Figure 1: One-stream C3D architecture (adapted from [5]).

In our second experiment, we used the same C3D model, however, we consider a two-stream network. For this, a second C3D model receives as input the optical flow [3]. The outputs of each C3D model are concatenated to form the first fully connected layer.

[1]Our C3D implementation is available at: https://github.com/ricoms/deep_memorability/blob/master/deep_memorability/trainer2/video_c3d.py

MediaEval'18, 29-31 October 2018, Sophia Antipolis, France

The motivation for this two-stream network is to evaluate if our C3D model can improve its results with this extra information. For reproducibility, we used the dense version of optical flow provided with the OpenCV library[2]. Figure 2 shows the overall architecture for this experiment.

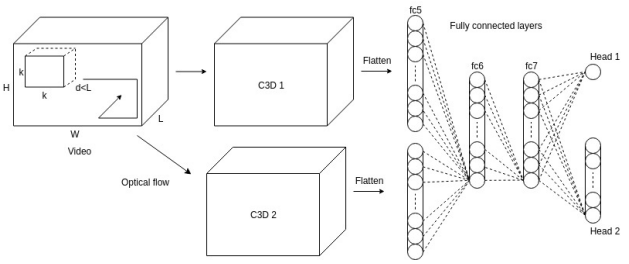

Figure 2: Two-stream C3D architecture (adapted from [5]).

For both networks, the input data are normalized to real values in the range $[-1, 1]$ and resized to 128×171 pixels. Also, the C3D model limits the input to a frame sequence with a predefined length (typically, 16 frames) and, for this reason, a sequence of 16 consecutive frames from each video was selected at random and used as input to the network. Optical flow generates a frame sequence with one less frame and, for easier the implementation, a last frame filled with zeros was appended at the end.

For training, a different loss function was used for each head: mean squared logarithmic error for the regression head and categorical crossentropy for the classification head. Then, a weighted sum of these individual losses with weights 1.0 and 0.7, respectively, was computed as the final loss to be minimized by a RMSProp optimizer with a learning rate of 0.0015.

3 RESULTS AND ANALYSIS

We submit three different runs configured as shown in Table 1. We calibrated the long-term memorability subtask through 10-fold cross-validation on the development data and use a holdout method with 10% of the development data for validation to calibrate the short-term memorability subtask. The evaluation metrics are: Spearman's rank correlation, Pearson correlation coefficient, and MSE (Mean Squared Error). The former is the official metric for the task.

Table 1: Configuration of the submitted runs.

Subtask	Run	Configuration
Long-term memorability	1	HMP + 20-NN regressor
Short-term memorability	1	1-C3D (video)
	2	2-C3D (video, optical flow)

Table 2 presents the results for the development and test sets considering the long-term memorability subtask. In the development set, we tested different regression models: SVR with RBF kernel and KNR. Also, the values experimented for the parameter k of KNR were 5, 20, and 30. Notice that KNR performs better than SVR for Spearman and Pearson metrics and the best results were achieved by KNR with $k = 20$. Therefore, we submit one run for the long-term memorability subtask considering our best result on the development set. From it, we achieved a Spearman value of 0.11845 for the test set.

[2]https://docs.opencv.org/3.4/d7/d8b/tutorial_py_lucas_kanade.html

Table 2: Long-term memorability results.

	Approach	Spearman	Pearson	MSE
Dev. Set	HMP + SVR kernel RBF	−0.026	−0.009	0.02
	HMP + 5-NN regressor	−0.004	−0.002	0.02
	HMP + 20-NN regressor	0.009	0.022	0.02
	HMP + 30-NN regressor	−0.003	0.014	0.02
Test Set	HMP + 20-NN regressor	0.11845	0.11966	0.02011

Table 3 presents the results for the development and test sets considering the short-term memorability subtask. The results for Spearman and Pearson metrics are not a number (NaN) and therefore they were not included in this table. The reason is because the C3D models assigned a same memorability score for all the videos. This turned impossible to calculate Spearman and Pearson correlation metrics due to lack of variance. Despite we tried several adjustments to hyper-parameters and pre-processing, it turned out during the experiments that the models did not improve results after reaching a value close to the average memorability score for the development set, possibly indicating a lack of fit.

Table 3: Short-term memorability results.

Approach	MSE	
	Dev. Set	*Test Set*
1-C3D (video)	0.0043	0.00702
2-C3D (video, optical flow)	0.0046	0.00699

4 DISCUSSION AND OUTLOOK

It is important to notice that HMP and C3D have an important difference: HMP captures motion dynamics of a video as whole whereas C3D is limited to a short window of fixed duration. An intention of future work is to analyze if features encoding long-term motion dynamics, like HMP or RNN (Recurrent Neural Network), are better for predicting video memorability than those capturing short-term motion dynamics, like C3D or ORB.

We can think about some reasons for the failure of our deep learning models. First, we constrain a full length video to a sequence of 16 consecutive frames. Smarter strategies to capture the temporal structure of a video, like RNN with LSTM (Long-Short Term Memory), could led to improvements. Second, we trained our deep neural networks from scratch. As the training set is rather small, data augmentation could be used to improve the results.

Another promising direction is to combine different features. For short memorability, we fused optical flow and video data. Would it improve results if we fuse video (visual data) and captions (textual data) provided for the task? Or other visual features, like HMP?

ACKNOWLEDGMENTS

We thank the São Paulo Research Foundation - FAPESP (grant 2016/06441-7), the Brazilian National Council for Scientific and Technological Development - CNPq (grants 423228/2016-1 and 313122/2017-2) and the Brazilian Federal Agency for Coordination for the Improvement of Higher Education Personnel - CAPES (grant 1703269) for funding. We gratefully acknowledge the support of NVIDIA Corporation with the donation of the Titan Xp GPU used for this research.

REFERENCES

[1] J. Almeida, N. J. Leite, and R. S. Torres. 2011. Comparison of Video Sequences with Histograms of Motion Patterns. In *IEEE International Conference on Image Processing (ICIP'11)*. Brussels, Belgium, 3673–3676.

[2] R. Cohendet, C-H. Demarty, N. Duong, M. Sjöberg, B. Ionescu, T-T. Do, and F. Rennes. 2018. MediaEval 2018: Predicting Media Memorability Task. In *Proc. of the MediaEval 2018 Workshop*. Sophia Antipolis, France.

[3] L. Fan, W-B. Huang, C. Gan, S. Ermon, B. Gong, and J. Huang. 2018. End-to-End Learning of Motion Representation for Video Understanding. In *IEEE International Conference on Computer Vision and Pattern Recognition (CVPR'18)*. Salt Lake City, UT, USA, 6016–6025.

[4] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. VanderPlas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. 2011. Scikit-learn: Machine Learning in Python. *Journal of Machine Learning Research* 12 (2011), 2825–2830.

[5] D. Tran, L. D. Bourdev, R. Fergus, L. Torresani, and M. Paluri. 2015. Learning Spatiotemporal Features with 3D Convolutional Networks. In *IEEE International Conference on Computer Vision (ICCV'15)*. Santiago, Chile, 4489–4497.