Video shot representation using a neural network

Nabil MADRANE
LIAD Laboratory
University Hassan II
Casablanca, Morocco

Said JAI ANDALOUSSI
LIAD Laboratory
University Hassan II
Casablanca, Morocco

Abderahim SEKKAKI
LIAD Laboratory
University Hassan II
Casablanca, Morocco

***Abstract*: This paper describes a work concerning the use of neural networks for learning both the static and dynamic structure of a video clip and thus generating signatures suitable for indexing and retrieval. We show how to design a neural network suitable for representing a video shot and we describe how the inner weight data of the neural network can be used for video shot retrieval and extraction of signatures. This paper also presents preliminary experimental results of our approach.**

Video indexing, neural networks, shot encoding, video signature

I. INTRODUCTION

It is remarkable to notice that 90% of the data in the world has been created in just the last two years. 80% of this data is unstructured data such as video or photos. This unstructured data demands a specific attention in the current Big Data realm. A number of challenging problems in the field of video indexing and retrieval in particular have been addressed.

In the field of video indexing and retrieval, one of the main challenges is to find a way to compare two video clips. Since these two videos may not have the same length and since video data contain a huge amount of data, the next problem is to find a way to represent a video with a short sequence of symbols, called a signature, by extracting both time and spatial features from the video sequence [1][2][3]. The generated signature, if designed appropriately, could be used to compute the similarity of 2 video sequences [4].

It however difficult to extract, in the general case, the discriminating features from a video clip. Various algorithms based on motion detection have been proposed [5][6]. The challenge here is to capture both meaningful spatial and temporal data, which are tightly coupled in video data [7][8].

In this paper we focus on how neural networks can be used to represent a video sequence. By forcing a neural network to learn how to reconstruct a sequence of frames, we force it to "encode" both spatial and time features and we make the assumption that this encoding is suitable for indexing and retrieval.

II. OVERVIEW

Our work is composed of two main parts. The first one concerns the representation of a video shot by using a neural network. This neural network is trained so that it can approximately produce image I_{t+1} from I_t, for any time t in the sequence. This part focuses on a single shot of the video, and we assume that a shot segmentation process has previously been applied to the video sequence. Inside a shot, it is thus assumed that the images are similar and that most of the difference is due to motion.

In the second part of our work, we address the problem of shot retrieval : given a single image, we want to extract the video shot to which this image pertains. For that purpose we use the neural network described in the first part of our work as a similarity calculator.

Finally, in the third part, we show how to effectively use the results of the first part in order to compute the similarity between 2 video shots, which is the first step towards an indexing and retrieval system. For that purpose, each shot is "encoded" by a specific neural network and we use the resulting neural weights as a signature.

III. SHOT ENCODING

Here "encoding" means representing the whole image sequence of a shot as a neural network so that it can be possible to approximately "regenerate" the video sequence from what the neural network has learned. More precisely, we are looking for a neural architecture so that

$I_{t+1} \sim \text{Forward}(I_t)$

where Forward() if the function that computes the output of the neural network from a given input image I_t.

This objective can be achieved by minimizing the reconstruction error :

$$E = \sum_{t=0}^{N-2} \| I_{t+1} - Forward(I_t) \|$$

978-1-4673-9795-7/15 $31.00 © 2015 IEEE
DOI 10.1109/CSCI.2015.163

A number of neural network learning algorithms have been designed, for specific neural architectures (multi-layer, self-organizing maps, etc.). However, our goal here is not to be able to achieve perfect reconstruction. We make the assumption that even with broad and imprecise learning, the resulting neural network (represented by the weights of its synapses) constitute a good representation of the video shot, suitable for indexing and retrieval.

So, given a specific shot of the video, composed of N frames I_t (t=0 to N-1), our goal is to train a neural network so that it encodes the dynamics of the shots, i.e. how the images are related to each other.

For this purpose, in the neural network that we designed (called the NN1 neural network), each neuron is connected to its 8 neighbors in current frame I_t, to the 9 neighbors neurons in frame I_{t-1} and to the 9 neighbors neurons in I_{t-2}. Thus each neuron is connected to 26 others neurons.

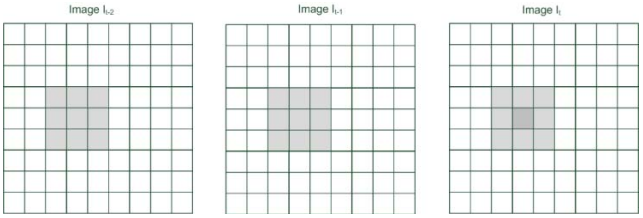

Figure 1. Neiborhod of a neuron

So, for each neuron, 8 connections capture the local spatial data and 18 connections capture the temporal data over the 2 previous frames.

This static representation of the neural network, where time is represented by a new layer, is not easy to implement since the number of layers depends on the number of frames. Event for a small video shot of 5 seconds at 25 fps, containing 125 frames of size 320x240, the neural network would contain too much neurons and synapses.

Fortunately, this static representation is actually equivalent to a dynamic neural structure with only 3 layers, if we use time-delayed (or recurrent) neurons. These neurons have a memory that remember their 3 previous states.

The state e_i^t of neuron *i* at time *t* is :

$$e_i^t = S\left(\alpha \sum_{j\in\pi_i} w_{ij} e_j^t + \beta \sum_{j\in\pi_i} w_{ij} e_j^{t-1} + \gamma \sum_{j\in\pi_i} w_{ij} e_j^{t-2}\right)$$

where S is the sigmoid function

$$S(x) = \frac{1}{1+e^{-x}}$$

w_{ij} is the weight of the connection between neurons *i* and *j*

π_i is the neighborhood of neuron *i*, constituted of its 8 surrounding neurons

α, β and γ are global weighting coefficients determined empirically. In our experiments we chose α=0.6, β=0.2 and γ=0.2 so that the intra-frame contribution (*i.e.* $\sum_{j\in\pi_i} w_{ij} e_j^t$) has more impact than the inter-frame contributions.

Each neuron is mapped to a single pixel, state e_i^t representing the gray level of the corresponding pixel. Color information is not taken into account.

Once the neural network is constructed, it is trained on the input video sequence (for an input sample I_t, the corresponding target output sample is I_{t+1}) by using a modified version of the Back-Propagation algorithm.

Convergence of the network is difficult to achieve under these conditions because there are too many neurons and synapses and not enough sample data to learn from. Our solution is to oversample the input data :

If (I_t,I_{t+1}) is a input/output pair of data that has to be learned by the neural network, then it is possible to artificially generate others pairs by modifying the luminance of both I_t and I_{t+1} with the same factor. By using different values of the luminance factor, we generate a number of artificial samples.

At the end of the learning process, we verify that our neural network can approximately reconstruct I_{t+1} from any I_t image. As previously mentioned the goal here is not to achieve close reconstruction : even if the convergence of the network is not achieved, the resulting neural weights are still capturing important spatial and temporal information, suitable for indexing and retrieval.

In conclusion, the neural network NN1 described in this section provides us a weight vector, resulting from the training of a specific shot by NN1. If the video sequence has K shots, each shot S_k (k=0 to K-1) is encoded with the neural network NN1 and is thus represented by a vector of weights W_k.

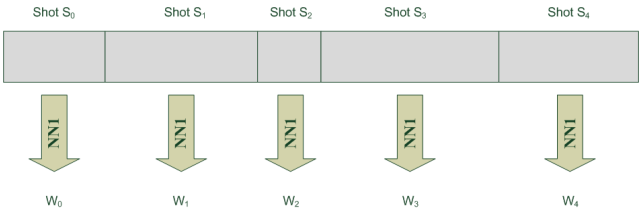

Figure 2. Encoding of each shot with the NN1 network

Our main idea is that a weight vector W_k constitutes a good representation of the corresponding video shot and can be used for shot retrieval (i.e. retrieve a shot from an image) and shot signature computation (in order to index video shots).

IV. SHOT RETRIEVAL

Since the NN1 neural network is able to approximately reconstruct I_{t+1} from any I_t image, it is interesting to compute the reconstruction error and use that error as the main criteria for determining if an image pertains or not to a shot.

This is called "shot retrieval" and the problem is the following : given a query image, how can we determine if that image should be considered as pertaining to a shot S_k or not ?

If I^*_{t+1} is the reconstructed image from I_t, then the reconstruction error is defined as $\varphi = \|I^*_{t+1} - \bar{I}\|$, where $\bar{I}$ is the mean image of the shot on which the test is made. This leads to the criteria :

$$\varphi < \sigma \rightarrow I \in S_k$$

i.e. if the reconstruction error is below a certain threshold σ then the image is declared as pertaining to the shot

In other words, if the image effectively pertains to the shot, the reconstructed image should be close to the mean image of the video shot. So we take the input image that we want to test, we perform Forward(I_t) in order to approximate I_{t+1} with the neural network and then we calculate the distance between this "reconstructed" image and the mean image of the shot. If the input image is very different from the ones present in the shot, the neural network will produce a noisy output image I_{t+1} and then the distance between this noisy image and the mean image of the shot will be large.

For practical applications, we suppose that the synapses weights of the neural network for each shot S_k are stored in a database. A query for an input image would consist on 1) calculating the reconstruction error between this image and each of the shots in the database and 2) choose the shot for which this reconstruction error is minimum and below a certain threshold.

V. SHOT SIGNATURE

We will now focus on how to use these weight vectors in order to generate a signature of each shot.

The size of a W_k vector is large, since each neuron is tied to a single pixel and has 26 connections. For an image size of 320x240 pixels, the size of a W_k vector is 320x240x26 = 1996800. In order to compute a compact signature from a W_k vector, we generate a binary pattern where each bit b_m is defined by

$$b_m = \begin{cases} 1 \; if \; |W_k^m| > 0.8 \\ 0 \; otherwise \end{cases}$$

This leads to a binary pattern containing 1996800 bits, i.e. 249600 bytes. This binary pattern constitute the signature of the whole shot S_k.

VI. SIMILARITY BETWEEN SHOTS

The similarity between two shots S_a and S_b is simply defined as the distance between their respective signatures. Since each signature is a binary pattern, we perform a logical AND between the two signatures and then count the number of bits with value 1.

VII. PRELIMINARY EXPERIMENTS

For these preliminary experiments, and in order to speed up computations and facilitate the convergence of the learning algorithm, we focused on a small set of 5 video clips. The size of the images have been resized to 160x120 pixels, so the number of neurons is 19200.

Video	Duration	Number of shots
Beverage	25	6
Perfume	30	9
City	35	7
Dance	25	8
News	25	6

For the indexing part, we first "encoded" each of the 36 video shots with the NN1 neural network. Then we transformed each resulting W_k weighting vector to a signature, as described in section V. The signature and the corresponding shot identifier were stored in a database.

For testing shot retrieval we used random images extracted from any of the 5 video clips and we performed a search on the database. We managed to achieve a 89% recognition rate, meaning that the neural network, even without achieving complete convergence, is still able to encode sufficient information in order to retrieve a shot from a single input image. This is encouraging and more work has to be done in this specific area.

VIII. DISCUSSION AND CONCLUSION

In this paper we have presented our work on how neural networks could be used to represent video shots. After splitting the video sequence into meaningful shots, we first capture the dynamics of each shot by forcing our neural network to encode the image sequence of that shot and, secondly, we use the results of the training to build signatures and extract shots from sample images.

Our future work will focus on using a learning algorithm similar to the one used for Self Organizing Maps (SOMs) instead of our modified version of the Back-Propagation algorithm. We will also make more experiments in order to determine the most appropriate time window (we currently use a time window of 3, since our neural network uses data from t, t-1 and t-2).

REFERENCES

[1] Chu-Hong Hoi, Wei Wang, and Michael R. Lyu, Anovel scheme for video similarity detection, International Conference of Image and Video Retrieval (CIVR2003), Urbana, IL, USA, 2003

[2] Krämer, P., Benois-Pineau, J. and Domenger, J.-P. (2006), Scene similarity measure for video content segmentation in the framework of a rough indexing paradigm. Int. J. Intell. Syst., 21: 765–783. doi: 10.1002/int.20159

[3] A Hybrid System of Signature Recognition Using Video and Similarity Measures, Hybrid Artificial Intelligence Systems, Lecture Notes in Computer Science Volume 8480, 2014, pp 211-220, Rafal Doroz, Krzysztof Wrobel, Mateusz Watroba.

[4] Kullback-Leibler similarity measures for effective content based video retrieval, R Priya; T N Shanmugam; R Bhaskaran, The Imaging Science Journal, Volume 61 Issue 7 (September 2013), pp. 541-555.

[5] Video Indexing: A Survey, Muhammad Nabeel Asghar, Fiaz Hussain, Rob Manton, International Journal of Computer and Information Technology (ISSN: 2279 – 0764) Volume 03 – Issue 01, January 2014.

[6] J.-w. Hsieh, S.-l. Yu, and Y.-s. Chen, "Motion-based video retrieval by trajectory matching," IEEE Transactions on Circuits and Systems for Video Technology, vol. 16, no. 3, pp. 396–409, Mar. 2006..

[7] C. Y. Y. Nakanishi and K. Tanaka, "Querying video data by spatiotemporal relationships of moving object traces," in Visual and Multimedia Information Management: IFIP TC 2/WG 2.6 Sixth Working Conference on Visual Database Systems, May 29-31, 2002, Brisbane, Australia. Kluwer Academic Pub, 2002, p. 357

[8] a. Mittal, "Addressing the problems of Bayesian network classification of video using high-dimensional features," *IEEE Transactions on Knowledge and Data Engineering*, vol. 16, no. 2, pp. 230–244, Feb. 2004.