ML Project 2
Motion-based Similarity Search in Videos of Confucian Rituals

Fadel Mamar Seydou, Alessandro Fornaroli, Razvan-Florin Mocan
Mentor: Yumeng Hou, Laboratory for Experimental Muselogy +
Machine Learning CS-433, EPFL, Switzerland

***Abstract*—This paper focuses on videos of re-enactment of ancient Confucian rituals, and develops a method to search videos based on the similarity in motion. The aim of this paper is to create a function that returns the k videos in the dataset which are the most similar to a given video. In order to properly analyse the videos, we treat each video as a time series, extracting motion-related features for each frame. Subsequently, we develop a model of similarity search based on Dinamic Time Warping and Radial Basis Function.**

I. Introduction

The Book of Li (or Book of Rites) is an ancient text of the classical Chinese period describing the cerimonial rituals, as well as social behaviour and manners that were deemed proper according to the teachings of Confucius.[3] This Machine Learning project is based on a selection of videos on these ancient Eastern rites. The aim of this project is to devise a method to retrieve the k videos most similar to a given video, based on the motions appearing in the video. Hence, we provide a method to compute the similarity between the motion that appears in different videos, and return the k videos that have the highest similarity score with respect to the input video.

II. Data and Challenges

A. The Datasets

The data that we used consisted of three datasets of videos, for a total of 213 videos of the ancient Confucian rituals:

- The first dataset, *posecuts_frontview*, consists of 40 videos showing the front view of a person performing determined actions in the framework of ancient Confucian rituals.
- The second one, *posecuts_sideview*, is relatively similar to the first dataset in the fact that it also consists of 40 videos. However, this time they show the side view of a person performing different actions.
- On the other hand, the third dataset, *bodyvocab*, comprises 133 videos of varied types of actions. The videos in this dataset are more heterogeneous and complex in their nature than in the other two, as the persons may perform different types of actions from different views, and other objects may appear.

Along with the videos, we have also been given some textual annotations for each video to be used for assessing the model.

B. Challenges of the Datasets

Each of the three datasets consists entirely of videos. Therefore, the first challenge that we had to address was to convert the videos into items that could be analysed more easily. As explained in the Feature Extraction part, we considered each video frame by frame, and obtained the data relative to bodyparts and their movement for each frame. Ultimately, each video was considered as a multi-dimensional time series, where each dimension represented a movement feature. We chose time series representation as it has relatively low cost in storage space and computational time. [6] Moreover, as we will see in Section IV, it allows to use certain similarity measures (such as dynamic time warping) which are useful in our model.

III. Data Preparation

A. Feature extraction

In order to extract the data regarding the angles and coordinates of the different body parts, we used PoseNet, a public model provided by Tensorflow [1] [5]. PoseNet does a pose estimation, finding the coordinates of certain salient bodyparts (eyes, nose, elbows, feet, and many others) known as keypoints.
PoseNet expects processed camera image as a result we have considered each video as a vector of frames. We have obtained the basic raw features from the resulting keypoints. Based on the confidence scores and their coordinates we have calculated the angles between joints and their lengths. In this way we have extracted the raw features from videos that we used to train our model. Figure 1 illustrates the raw features that have been extracted for a particular frame.

Figure 1. Features obtained from Pose Estimation of a frame

B. Feature expansion

To get a better representation of the intrinsic relations in the dataset we have applied the following feature expansion to the raw features:

- **Standardization**: this allows to better compare the different videos in the dataset as the dataset is center around a mean of 0. In order to carry out this step, the library *scikit-learn* [7] has been used.
- **Extraction of x and y components of the angles** : this is obtained through an application of cosine and sine.
- **Log-transform** of the square of the angles.
- **Angular velocities** along each axis : the velocities are computed across frames.
- **Angular accelerations** along each axis.
- **Polynomial expansion** along the x-axis : emphasizing the motion along the x-axis.

C. Dimensionality Reduction

After the feature expansion, in order to improve our dataset, we applied a dimensionality reduction in the form of principal component analysis (PCA). In order to do so, we used the library *scikit-learn* [7]. PCA allows to reduce the computational time taken for the similarity measure by reducing the dimensionality of data.

IV. Similarity search: main approach

A. General idea

We developed a function that returns the k videos in a dataset that are the most similar to a video input. In order to do so, we calculate the similarities between the input video and all other videos in the dataset using one of the similarity measures described in the following sections, and return the k videos with the highest similarity score.

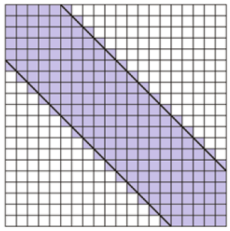
B. Dynamic time warping (DTW)

1) Optimization using Sakoe-Chiba local constraints: One common similarity measurement method for times series (we will consider the extracted keypoints to be time series) is dynamic time warping (DTW). It allows to look for similar sequences within time series (i.e. many-to-one comparison) instead of doing only one-to-one comparison. The idea is to compute a cumulative distance matrix where the last element computed represents the distance between the two time series. [2] The recurrence formulation of the algorithm is :

$$\gamma(i,j) = d(T_i, S_j) + \min\left(\gamma(i-1,j-1), \gamma(i-1,j), \gamma(i,j-1)\right),$$

where $d(T_i, S_j\ )$ represents the distance (e.g. Euclidean, Manhattan, Cosine) between the components i and j of the time series T and S respectively.

The algorithmic solution implemented using dynamic programming has a complexity of $O(m * n)$ where m and n represent the respective number of frames in each video. For very large videos, this may not be efficient. As a solution, the implementation using Sakoe-Chiba band (which is a local constraint on the warping path, illustrated in Figure2) reduces the complexity to $O(\delta(m+n))$ which is much better than the previous one, where δ is the window. [2] For the



Sakoe-Chiba band

Figure 2. Sakoe-Chiba band

implementation of this function, we used the library *tslearn*, which is specifically designed for time series analysis [10].

2) A novel approach to DTW using a soft-min: Another approach recently developed includes a **soft-min approach**. As the dynamic time warping is not differentiable, gradient-based method cannot be applied to speed up the process. Thus, a soft-min approach was developed by Cuturi and Blondel [4] that implements a backward method for computing the gradient of the differentiable loss function:

$$\text{soft-min}(a_1, \ldots, a_n) = -\gamma \log \sum_{i=1}^{n} e^{-\frac{a_i}{\gamma}}$$

In order to implement the method, the *tslearn* python library [10] has been used.

C. RBF-based approach

Finally, we have implemented a method based on the RBF (Radial basis function) kernel. [11] We feed in the mean of the computed distance matrix (i.e. combinations of frames between two videos).
This method can be expressed as:

$$\text{Score } (i,j) = e^{\frac{-\text{mean Distance}}{2*\sigma^2}}$$

For our implementation, the distance matrix is computed using the squared Euclidean distance, and σ is a hyper-parameter determined in a way to give a reasonable-sized region of similarity. Figure 3 depicts our motivation. Determining a good σ was detrimental as it influences the region of similarity[9]. When setting the value of σ, not being too strict can allow to boost the performance, especially when we mix the models. Thus, we set $\sigma = 1$.

D. Model averaging approach

In order to improve the results, we chose to **combine the three methods and perform a model averaging**

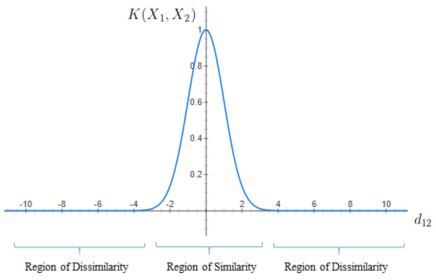


Figure 3. Motivation for the RBF-based method: a larger σ allows for a broader range of similarity

where only the best predictions would be given. Indeed, this gave a better result than just considering the methods by themselves although it comes with a drawback of a slight increase in computational time. We strongly believe that for a method not based on clustering, model averaging can produce descent results provided a good data preparation and cleaning.

V. Results and Evaluation

A. Model Evaluation and Similarity scores

To assess the performance of the model, we used a **F-1 metric** (i.e. the harmonic mean of precision and recall). [8] This is more reliable than an accuracy in our case where false positive and false negative should be heavily penalized. Regarding the similarity scores resulting from each method, a more practical approach would be to define a standardized score for all methods (DTW and RBF-based) and give weight to the most recurring "predictions", for the model averaging approach. In fact, in practice, small annotation of the data was provided to us. However, relying solely on a feedback function that uses annotations is not desirable, as it is not scalable.

B. Results

*1) Model performance **before** model averaging:* Figures 4 and 5 show the performance of the DTW and RBF-based methods on the set *posecuts_sideview*. We have noticed that the results with the soft-min DTW approach were pretty much similar to the standard DTW, thus we have not included another boxplot.

*2) Model performance **after** model averaging:* One finding that we made was that although the soft-DTW and the DTW provide similar results,"averaging" the three models give a higher performance than just considering two models out of three. Figure 6 shows the increase in performance after averaging. A similar performance was also obtained on the front view training dataset.

The Figure 7 below shows the **advantages of combining the model averaging approach with a preprocessing** of the dataset. It can be seen that the results are more robust (with respect to the F-1 score) with the model averaging as the medians are almost similar (for both cases) even if the interquartile range is bigger (without preprocessing).

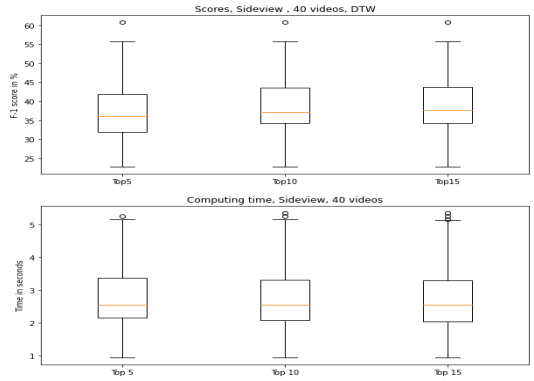


Figure 4. F1 score (top) and computational time (bottom) with the search model based on DTW with cosine similarity measurement on *posecuts_sideview*.

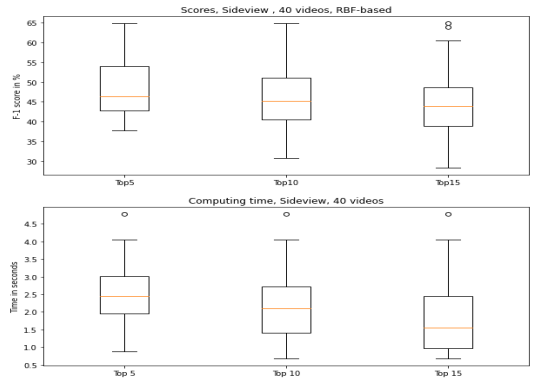


Figure 5. F1 score (top) and computational time (bottom) with the RBF-based search model on *posecuts_sideview*. The results are for $k = 5$, $k = 10$ and $k = 15$.

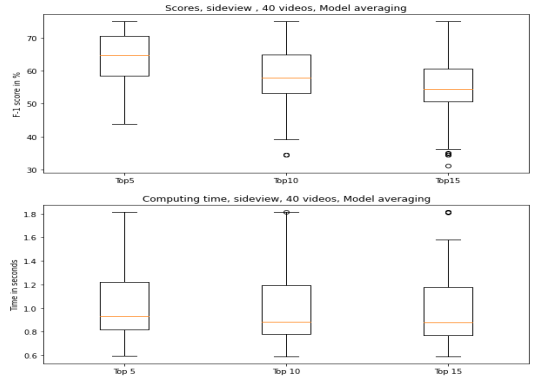


Figure 6. F1 score (top) and computational time (bottom) with Model averaging on *posecuts_sideview* **with preprocessed Dataset**.

One significant downside of not using a preprocessing is the increase in computing time as shown.

C. Results on bodyvocab

The global performance on the *bodyvocab* dataset is way below the one we had with the other sets. Many reasons can explain these results. The test dataset included many people instead of only one, although in many videos of the dataset the people were performing the same movements. Our model seems to be quite sensitive to this setup.Further improvement in dealing with multiple people can definitely

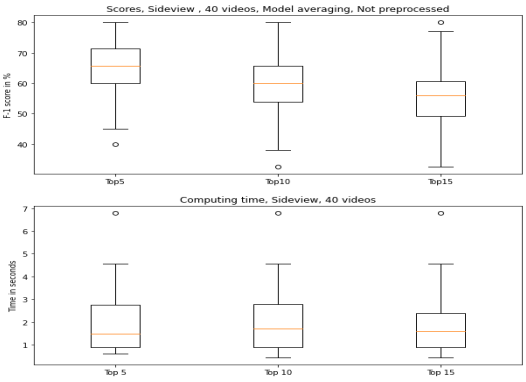


Figure 7. F1 score (top) and computational time (bottom) with Model averaging on *posecuts_sideview* **with unpreprocessed Dataset**.

improve the results.

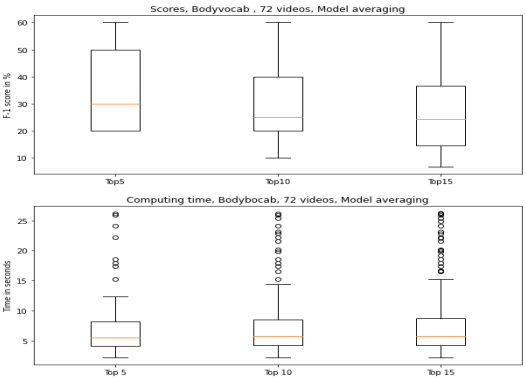


Figure 8. Test results for $k = 5$, $k = 10$, $k = 15$ for *bodyvocab*, showing F1 score (top) and computational time (bottom).

VI. Discussion

A. General remarks

Overall, the results that we obtain were rather satisfactory on the videos in *posecuts_frontview* and *posecuts_sideview*. We also managed to obtain satisfactory results with the videos in *bodyvocab*, even though these appeared to be less accurate than for the other two datasets, due to the greater complexity and additional noise.

A problem with our model is that it is scalable only to a certain extent to the entire database of videos on Confucian rites. Firstly, because of the large size of the database, it would take a very long time to compute the distance from the input to all other videos. Secondly, the other videos database present different features, objects, people and settings that were not accounted for in our simpler model. Further analysis and an improved feature extraction, using other models besides PoseNet, would be necessary to account for these new factors.

B. Ideas for improvement

We propose to investigate a **KD-tree based structure**for retrieval. This type of indexing extends the binary search tree to higher dimensions. It is possible to constitute the KD-tree based on a similarity calculation with **fundamental and redundant motions**. One challenge would be to keep the number of features (here the redundant and fundamental motions) to a reasonable number so that **maintenance and query can be done efficiently**. This idea has not been tested but we believe it can be interesting for further research. Due to the high dimensionality of the videos, constituting a base (i.e. the redundant and fundamental motions) for comparison can speed up the query, hence allowing a K-nearest neighbors approach to this problem.[12] With the Confucius rituals, many movements are similar, we can think of having an "orthogonal base of movements" from which all the other movements derive.

VII. Alternative Approach: Clustering of the Datasets

A. Methodology

After having devised a method that returns a list of similar videos, we chose to investigate the possibility of an different approach, to compare it to the main model and leave it for further research. This new method is based on clustering of the videos in the dataset.

In order to cluster each dataset, we used the library tslearn [10]. As in the previous part, we used the DTW algorithm to compute the distance between videos. In order to do the clustering, we used a K-means algorithm optimized for time series. We find the number of clusters to be

$$\text{number of clusters} = \frac{\text{size of the dataset}}{k}$$

. The size of each cluster will not be exactly k. However, k will be the average size of each cluster9, and if the clustering is balanced, then k will approximate to the cluster size.

Our method creates a clustering of all the dataset, and predicts and returns the cluster to which the input video belongs.

B. Results and Discussion

The results of the clustering-based method are illustrated by Figure 9.

In order for a clustering-based method to return exactly k videos, we should apply a form of soft clustering, where each video can belong to multiple clusters.

Moreover, the videos in a certain cluster are not necessarily *all the most similar* ones to the input video. Furthermore, as the number of cluster increases, an increasing number of cluster will consist of only one video, and would not therefore provide any other similar videos. Nonetheless, the clustering based model could potentially allow to see certain categorization of rituals that might not have been self-evident.

Alternative methods could improve the accuracy of the clustering. For example, Gaussian Mixture Models might achieve better results, as they account for variance within clusters. We leave such improvements for further research.

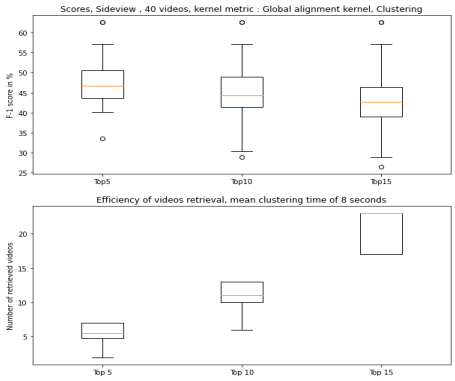


Figure 9. Test results for the clustering based method, showing for $k = 5$, $k = 10$ and $k = 15$ on *posecuts_sideview*, showing F1 score (top) and efficiency of the clustering (bottom).

VIII. Conclusion

Our project provided a method to retrieve videos with similar motion given a video from the Confucian Rites database. Further research would address the entire database, considering the previously mentioned scalability issues. Moreover, another question that will arise is to carry a clustering of motion in the entire database: this practice could potentially uncover previously unnoticed patterns regarding the ancient rites of Confucius.

References

[1] Martín Abadi, Ashish Agarwal, Paul Barham, Eugene Brevdo, Zhifeng Chen, Craig Citro, Greg S. Corrado, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Ian Goodfellow, Andrew Harp, Geoffrey Irving, Michael Isard, Yangqing Jia, Rafal Jozefowicz, Lukasz Kaiser, Manjunath Kudlur, Josh Levenberg, Dandelion Mané, Rajat Monga, Sherry Moore, Derek Murray, Chris Olah, Mike Schuster, Jonathon Shlens, Benoit Steiner, Ilya Sutskever, Kunal Talwar, Paul Tucker, Vincent Vanhoucke, Vijay Vasudevan, Fernanda Viégas, Oriol Vinyals, Pete Warden, Martin Wattenberg, Martin Wicke, Yuan Yu, and Xiaoqiang Zheng. TensorFlow: Large-scale machine learning on heterogeneous systems, 2015. Software available from tensorflow.org.

[2] Carmelo Cassisi, Placido Montalto, Marco Aliotta, Andrea Cannata, and Alfredo Pulvirenti. *Similarity Measures and Dimensionality Reduction Techniques for Time Series Data Mining*. 09 2012.

[3] Howard Curzer. Contemporary rituals and the confucian tradition: a critical discussion. *Journal of Chinese Philosophy*, 39, 06 2012.

[4] Marco Cuturi and Mathieu Blondel. Soft-dtw: a differentiable loss function for time-series. 03 2017.

[5] A. Kendall, M. Grimes, and R. Cipolla. Posenet: A convolutional network for real-time 6-dof camera relocalization. In *2015 IEEE International Conference on Computer Vision (ICCV)*, pages 2938–2946, 2015.

[6] V. Niennattrakul and C. A. Ratanamahatana. Clustering multimedia data using time series. In *2006 International Conference on Hybrid Information Technology*, volume 1, pages 372–379, 2006.

[7] 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. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.

[8] Yutaka Sasaki. The truth of the f-measure. *Teach Tutor Mater*, 01 2007.

[9] Sushanth Sreenivasa. Radial basis function (rbf) kernel: The go-to kernel.

[10] Romain Tavenard, Johann Faouzi, Gilles Vandewiele, Felix Divo, Guillaume Androz, Chester Holtz, Marie Payne, Roman Yurchak, Marc Rußwurm, Kushal Kolar, and Eli Woods. Tslearn, a machine learning toolkit for time series data. 01 2020.

[11] Lei Xu, Adam Krzyżak, and Alan Yuille. On radial basis function nets and kernel regression: Statistical consistency, convergence rates, and receptive field size. *Neural Networks*, 7(4):609 – 628, 1994.

[12] Yunyue Zhu. High performance data mining in time series: Techniques and case studies. 2004.