IAES International Journal of Artificial Intelligence (IJ-AI)
Vol. 9, No. 1, March 2020, pp. 40~45
ISSN: 2252-8938, DOI: 10.11591/ijai.v9.i1.pp40-45

Large-scale image-to-video face retrieval with convolutional neural network features

Imane Hachchane[1], Abdelmajid Badri[2], Aïcha Sahel[3], Yassine Ruichek[4]
[1,2,3]Laboratory of Electronics, Energy, Automation & Information Processing, Faculty of Sciences and Techniques Mohammedia, University Hassan II Casablanca, Mohammedia, Morocco
[4]IRTES-Laboratory SET, University of Technology Belfort Montbéliard, France

Article Info

Article history:

Received Aug 23, 2019
Revised Oct 10, 2019
Accepted Nov 3, 2019

Keywords:

BOVW
Classification
CNN, Faster R-CNN
Face retrieval
FV
Image processing
Image-to-video instance retrieval
Object recognition
Video retrieval

ABSTRACT

Convolutional neural network features are becoming the norm in instance retrieval. This work investigates the relevance of using an of the shelf object detection network, like Faster R-CNN, as a feature extractor for an image-to-video face retrieval pipeline instead of using hand-crafted features. We use the objects proposals learned by a Region Proposal Network (RPN) and their associated representations taken from a CNN for the filtering and the re-ranking steps. Moreover, we study the relevance of features from a finetuned network. In addition to that we explore the use of face detection, fisher vector and bag of visual words with those same CNN features. We also test the impact of different similarity metrics. The results obtained are very promising.

Corresponding Author:

Imane Hachchane,
Laboratory of Electronics, Energy, Automation & Information Processing,
Faculty of Sciences and Techniques Mohammedia,
University Hassan II Casablanca, Mohammedia, Morocco.
Email: hachchaneimane@gmail.com

1. INTRODUCTION

In this work we address the task of image to video face retrieval. With billions of images and videos created each day, it is essential to build tools for accessing and retrieving multimedia content efficiently. In the context of retrieval, image-to-video face retrieval is the task of identifying a specific frame or scene in a video or a collection of videos from a specific face instance in a static image.

On one hand, image-to-video retrieval is an asymmetric problem. Images only contain static information but videos have much richer visual information, like optical flow. Due to the lack of temporal information, standard techniques used for extracting video descriptors [1-4] cannot be directly used on static images. But, standard features for image retrieval [5-8] can be applied to video data by processing each frame as an independent image. Temporal information is usually compressed either by reducing the number of local features or by encoding multiple frames into a single global representation. On the other hand, face retrieval remains a challenging task because conventional image retrieval approaches, such as bag of visual words (BOVW), are difficult to adapt to the face domain [9].

Traditionally, image-to-video retrieval or face retrieval methods [10-12] are based on hand-crafted features (SIFT [13], BRIEF[14], etc.) and not much effort has been put so far into the adaptation of deep learning techniques, such as convolutional neural networks (CNN). CNNs trained with large amounts of data

can learn features generic enough to be used to solve tasks for which the network has not been trained [15]. For image retrieval, in particular, many works in the literature [7, 16] have adopted solutions based on standard features extracted from a pretrained CNN for image classification [17], achieving encouraging performances. Many CNN-based object detection pipelines have been proposed, but we are more interested in the latest ones. Faster R-CNN [18] uses a Region Proposal Network (RPN) that removes the dependence of object proposals from older CNN object detection systems. In Faster R-CNN, RPN shares features with the object-detection network in [19] to simultaneously learn prominent object propositions and their associated class probabilities. Although the Faster R-CNN is designed for generic object detection [20]. Demonstrated that it can achieve impressive face detection performance especially when retrained on a suitable face detection training set [21].

In this paper we try to fill this gap by exploring the relevance of on-the-shelf and fine-tuned features of an object detection CNN for image-to-video face retrieval. We exploit the features of a state-of-the-art pre-trained object detection CNN called Faster R-CNN. We use his end-to-end object detection architecture to extract global and local convolutional features in a single forward pass and test their relevance for image-to-video face retrieval. We also explore the use of face detection, Fisher Vector (FV) [4] and BOVW words with those same CNN features. The rest of this paper is organized as follows: Section 2 presents our research method, including our features extraction method and the raking and reranking strategies. Section 3 presents our results and discussions. Finally, we present our conclusions in Section 4.

2. METHODOLOGY

2.1. Datasets exploited

We evaluate our methodologies using the following datasets:

- YouTube Celebrities Face Tracking and Recognition Data (Y-Celeb) [22]: The dataset contains 1910 sequences of 47 subjects. All videos are encoded in MPEG4 at 25fps rate.
- YouTube Faces Database [23]: The data set contains 3,425 videos of 1,595 different people. All the videos were downloaded from YouTube. An average of 2.15 videos are available for each subject. The shortest clip duration is 48 frames, the longest clip is 6,070 frames, and the average length of a video clip is 181.3 frames.

The datasets used to finetune the network:

- FERET [24]: 3528 images, including 55 Query images. A framing box surrounding the target face is provided for query images.
- FACES94 [25]: 2809 images 2809 images, including 55 Query images. A framing box surrounding the target face is provided for query images.
- FaceScrub [26]: 55127 images

2.2. Video retrieval strategy:

This section describes the three major steps in our pipeline, we used:

1. Filtering step. We create image descriptors for query and database frames using CNN features. At testing time, the descriptor of the query is compared to all items in the database, which are then ranked according to a similarity measure. At this stage, the entire frame is considered as a query.
2. Spatial re-ranking. After the filtering step, the N upper elements are analyzed locally and re-ranked.
3. Query expansion (QE). We average the frame descriptors of the M higher elements of the first ranking with query descriptor to carry out a new search.

2.3. CNN-based representations

We explore the relevance of using CNN features for face image to video face retrieval. The query instance is defined by a bounding box above the query image. We use the features extracted from Faster R-CNN pre-trained models [18] as our global and local features. Faster R-CNN has a region proposal network that gives the locations in the image which have higher probabilities of having an object, and a classifier that labels each of those object proposals as one of the classes in the learning dataset [27]. We extract compact features from the activations of a convolutional layer in a CNN [27-28]. Faster R-CNN is faster on a global and local scale. We build a global frame descriptor by ignoring all the layers that work with object proposals and extract features from the last convolutional layer. Considering the extracted activations of a convolution layer for a frame, we group the activations of each filter to create a frame descriptor with the same dimension as the number of filters in the convolution layer, to do so both max and sum pool-ing strategies are considered and compared in section 3. We aggregate the activations of each window suggestion in the RoI Pooling layer to create regional descriptions [21].

We use the VGG16 architecture of Faster R-CNN to extract the global and local features. We choose that architecture because it performs better. It has been shown in previous works in the literature [21, 27] that the capabilities of deeper networks achieve better performance. The global descriptors are extracted from the last convolution layer "conv5_3" and are of dimension 512. The local features are grouped from the Faster R-CNN RoI clustering layer. All experiments were performed on a Nvidia GTX GPU.

2.4. Fine-tuning Faster R-CNN

Fine tuning the Faster R-CNN network allows as to obtain features specific to face retrieval and should help improve the performance of spatial analysis and re-ranking. To achieve this, we choose to fine-tune Faster R-CNN to detect the query faces. The resulting networks will be used to extract better local and global representations, and will be used to perform spatial reranking.

We chose to refine the model VGG16 Faster R-CNN, pre-trained with the objects of Pascal VOC, with two deferent datasets. The first network was refined using FERET and Faces94 datasets, we combine them to create one bigger dataset. We modify the output layer in the network to return 422 class probabilities (269 people in the FERET dataset plus 152 people in the Faces94 dataset, plus one additional class for the background) and their corresponding bounded bound box coordinates [21]. This new refined network will be called VGG(F-F), the training process took 2 hours 47 minutes. The second network was refined using FaceScrub dataset. We modify the output layer in the network to return 530 class probabilities (530 people, plus one additional class for the background) and their corresponding bounded bound box coordinates. Our second refined network will be called VGG(F-S)[21], the training took 2 hours 30 minutes.

We kept the Faster R-CNN's original parameters described in [19], but due to our smaller number of training samples we decreased the number of iterations from 80,000 to 20,000. We use the refined networks of the tuning strategy (VGG(F-S) & VGG(F-F)) on all datasets to extract image and region descriptors to perform a face retrieval.

2.5. Faster R-CNN features & Face detection

We evaluate the impact of using a face detection algorithm on our datasets and queries before using Faster R-CNN for feature extraction and the ranking and reranking strategies as described previously.

2.6. Faster R-CNN features & FVs

To explore the relevance of using FVs on CNN feature, for the image-to-video face retrieval task, we first extract the CNN features of each frame. We then apply Principal Component Analysis (PCA), Gaussian mixture model (GMM), L2 normalization on those features before using our FV function. Finally, as described before, we compute the similarity measure and use the ranking and reranking strategies.

2.7. Faster R-CNN features & BOVW

To explore the relevance of using BOVW with CNN feature, for the image-to-video face retrieval task, we first extract the CNN features of each frame. Then we apply the clustering, vector quantization and inverted indexing steps. Finally, as described before, we compute the similarity measure and use the reranking strategies.

3. RESULTS AND DISCUSSION

We evaluate the use of Faster R-CNN features for face image to video face retrieval. We experimented with six different similarity metrics. The results were similar and close but overall cosine performed better. Table 1 shows an example of our results when using features from an on the shelf network with VGG16 architecture trained on pascal dataset.

We carried out a comparative study of the sum and max-pooling strategies of the image-wise and region-wise descriptors. Table 2 summarizes most of our results. According to our experiments, the sum-pooling gives better performance than the max-pooling. It also shows the performance of Faster R- CNN with a VGG16 architectures trained on two different datasets (Pascal VOC and COCO), VGG16 trained on COCO performed better because the dataset is bigger and more diverse. Moreover, it presents the impact of spatial reranking and query expansion. Using the global features of Faster R-CNN on their own without any reranking strategy gives the best results. Spatial reranking & QE had no positive impact on the results. We should note that in average the offline feature extraction took 29.7 minutes while the online ranking steps took 3.7 seconds and the reranking strategy took 7 minutes for Y-Celeb dataset. For YouTube Faces Database, the offline feature extraction took 20 hours while the online ranking steps took only 85 seconds and the reranking strategy took 21 minutes.

Table 1. Mean Average Precision (mAP) Of Pretrained Faster R-CNN Models with Vgg16 Architectures on Pascal Dataset Using Different Similarity Measures Using Y-Celeb Dataset.

Similarity metric	Pooling	Ranking	Reranking	QE
Cosine	max	0.888	0.860	0.550
	sum	0.915	0.846	0.600
Manhattan	max	0.900	0.869	0.570
	sum	0.905	0.841	0.428
Euclidian	max	0.888	0.860	0.550
	sum	0.915	0.846	0.600
CityBlock	max	0.900	0.869	0.570
	sum	0.905	0.841	0578
L1	max	0.900	0.869	0.570
	sum	0.905	0.841	0.578
L2	max	0.888	0.860	0.550
	sum	0.915	0.846	0.603

Y-Celeb-Faces column present the results of using face detection on the Y-Celeb dataset. As we can see in Table 2 face detection did not improve the results. We should note that we were able to reduce the ranking time to 2.4 seconds on average. Table 2 show that the refined features slightly exceeded the raw features in the spatial reranking and the QE stages. But still, the global features of Faster R-CNN from VGG16 trained on COCO used without any reranking strategy give the best results.

Table 2. Mean Average Precision (mAP) of pre-trained Faster R-CNN models with VGG16 architectures. (P), (C), (F-S) AND (F-F) denote whether the network was trained with Pascal VOC, Microsoft COCO, FaceScrub or Feret & Faces94 images, respectively. With a comparison between sum and max pooling strategies. When indicated, QE is applied with M = 5

Network	Pooling	Y-Celeb			YouTube Faces Database			Y-Celeb-Faces		
		Ranking	Reranking	QE	Ranking	Reranking	QE	Ranking	Reranking	QE
VGG16	max	0.888	0.860	0.550	0.892	0.877	0.882	0.574	0.516	0.542
(P)	sum	0.915	0.846	0.600	0.897	0.886	0.891	0.618	0.486	0.511
VGG16	max	0.911	0.888	0.522	0.892	0.878	0.889	0.622	0.574	0.617
(C)	sum	0.926	0.807	0.512	0.903	0.882	0.896	0.705	0.538	0.551
VGG16	max	0.809	0.777	0.457	0.848	0.834	0.838	0.477	0.423	0.450
(F-S)	sum	0.917	0.843	0.578	0.882	0.873	0.874	0.635	0.509	0.519
VGG16	max	0.915	0.874	0.554	0.894	0.884	0.887	0.666	0.656	0.682
(F-F)	sum	0.924	0.899	0.621	0.896	0.892	0.893	0.715	0.612	0.646

When using FVs with Faster R-CNN features we can say that max pooling performed better, as shown in Table 3, but it is clear that using FVs is not a good idea. The mAP is very low (below 10%). We couldn't test on the YouTube Faces Database due to a Memory Error caused by the size of the dataset and the limitation of the hardware.

Table 3. Mean Average Precision (mAP) of pre-trained Faster R-CNN models with VGG16 architectures. (P) and, (C) denote whether the network was trained with Pascal VOC or Microsoft COCO. With a comparison between sum and max pooling strategies. When indicated, QE is applied with M = 5

Network	Pooling	Y-Celeb		
		Ranking	Reranking	QE
VGG16 (P)	max	0.097	0.102	0.097
	sum	0.097	0.100	0.102
VGG16 (C)	max	0.097	0.102	0.098
	sum	0.097	0.097	0.097

When using on BOVW with Faster R-CNN features we couldn't analyze the full results because we kept running into a Memory Error caused by the sizes of the datasets and the limitation of the hardware in addition to that the result obtained were not that encouraging. Table 4 present the results that we were able to get.

Table 4. Mean Average Precision (mAP) of pre-trained Faster R-CNN models with VGG16 architectures. (C) denote that the network was trained with Microsoft COCO. With a comparison between sum and max pooling strategies. When indicated, QE is applied with M = 5

Network	Pooling	Y-Celeb		
		Ranking	Reranking	QE
VGG16 (C)	max	0.032	0	0.097
	sum	0.032	-	-

Finally, we can clearly see in that the raw faster R-CNN features largely outperformed the other strategies with a mAP of 92.6%. Table 5 show comparison with State-of-the-art.

Table 5. Comparison with State-of-the-art. Results provided as mAP.

Method	Y-Celeb	YouTube Faces Database
NN [23]	-	0.145
O-SBoF[29]	-	0.471
RN-BOF[30]	-	0.465
Faster R-CNN features	0.926	0.903
Faster R-CNN features + FV	0.097	0.006
Faster R-CNN features +BOVW	0.032	0.001

4. CONCLUSION

This article explores the use of features from an object detection CNN for image-to-video face retrieval. It uses Faster R-CNN features as global and local descriptors. We have shown that the common similarity metric gives similar results. We also found that sum-pooling performs better than max-pooling in most cases, and contrary to our previous work [21] fine tuning does not improve the results. More importantly, we found that applying the similarity measure directly on the CNN feature of an off-the-shelf CNN trained on a large and diverse dataset gave the best results, and that using FVs or BOVW is memory consuming and is not suitable for CNN features in this case.

ACKNOWLEDGEMENTS

This work falls within the scope of Big Data and Connected Object (BDCO). We would like to thank the Hassan II University of Casablanca for financing this project.

REFERENCES

[1] A. Filgueiras De Araujo, "Large-Scale Video Retrieval Using Image Queries A Dissertation Submitted To The Department Of Electrical Engineering And The Committee On Graduate Studies Of Stanford University In Partial Fulfillment Of The Requirements For The Degree Of Doctor Of Philos," 2016.
[2] J. Y.-H. Ng, M. Hausknecht, S. Vijayanarasimhan, O. Vinyals, R. Monga, and G. Toderici, "Beyond Short Snippets: Deep Networks for Video Classification," Mar. 2015.
[3] K. Simonyan and A. Zisserman, "Two-Stream Convolutional Networks for Action Recognition in Videos," Jun. 2014.
[4] D. Tran, L. Bourdev, R. Fergus, L. Torresani, and M. Paluri, "Learning Spatiotemporal Features with 3D Convolutional Networks," Dec. 2014.
[5] A. Babenko, A. Slesarev, A. Chigorin, and V. Lempitsky, "Neural Codes for Image Retrieval," Apr. 2014.
[6] Y. Kalantidis, C. Mellina, and S. Osindero, "Cross-dimensional Weighting for Aggregated Deep Convolutional Features," Dec. 2015.
[7] A. S. Razavian, J. Sullivan, S. Carlsson, and A. Maki, "Visual Instance Retrieval with Deep Convolutional Networks," Dec. 2014.
[8] L. Wu, Y. Wang, Z. Ge, Q. Hu, and X. Li, "Structured deep hashing with convolutional neural networks for fast person re-identification," *Comput. Vis. Image Underst.*, vol. 167, pp. 63–73, Feb. 2018.
[9] C. Herrmann and J. Beyerer, "Fast face recognition by using an inverted index," 2015, vol. 9405, p. 940507.
[10] R. Arandjelovic and A. Zisserman, "Three things everyone should know to improve object retrieval," in *2012 IEEE Conference on Computer Vision and Pattern Recognition*, 2012, pp. 2911–2918.
[11] J. Pont-Tuset, P. Arbelaez, J. T. Barron, F. Marques, and J. Malik, "Multiscale Combinatorial Grouping for Image Segmentation and Object Proposal Generation," Mar. 2015.
[12] J. Long, E. Shelhamer, and T. Darrell, "Fully Convolutional Networks for Semantic Segmentation," Nov. 2014.
[13] D. G. Lowe, "Distinctive Image Features from Scale-Invariant Keypoints," *Int. J. Comput. Vis.*, vol. 60, no. 2, pp. 91–110, Nov. 2004.

[14] M. Calonder, V. Lepetit, C. Strecha, and P. Fua, "BRIEF: Binary Robust Independent Elementary Features," Springer, Berlin, Heidelberg, 2010, pp. 778–792.
[15] J. Y. H. Ng, M. Hausknecht, S. Vijayanarasimhan, O. Vinyals, R. Monga, and G. Toderici, "Beyond short snippets: Deep networks for video classification," *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, vol. 07-12-June, pp. 4694–4702, 2015.
[16] A. Araujo and B. Girod, "Large-Scale Video Retrieval Using Image Queries," *IEEE Trans. Circuits Syst. Video Technol.*, vol. XX, no. c, pp. 1–1, 2017.
[17] G. De Oliveira Barra, M. Lux, and X. Giro-I-Nieto, "Large scale content-based video retrieval with LIvRE," *Proc. - Int. Work. Content-Based Multimed. Index.*, vol. 2016-June, 2016.
[18] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 39, no. 6, pp. 1137–1149, 2017.
[19] R. Girshick, "Fast R-CNN," in *2015 IEEE International Conference on Computer Vision (ICCV)*, 2015, pp. 1440–1448.
[20] H. Jiang and E. Learned-Miller, "Face Detection with the Faster R-CNN," *Proc. - 12th IEEE Int. Conf. Autom. Face Gesture Recognition, FG 2017 - 1st Int. Work. Adapt. Shot Learn. Gesture Underst. Prod. ASL4GUP 2017, Biometrics Wild, Bwild 2017, Heteroge*, pp. 650–657, 2017.
[21] I. Hachchane, A. Badri, A. Sahel, and Y. Ruichek, "New Faster R-CNN Neuronal Approach for Face Retrieval," in *Lecture Notes in Networks and Systems*, vol. 66, 2019, pp. 113–120.
[22] Minyoung Kim, S. Kumar, V. Pavlovic, and H. Rowley, "Face tracking and recognition with visual constraints in real-world videos," in *2008 IEEE Conference on Computer Vision and Pattern Recognition*, 2008, pp. 1–8.
[23] L. Wolf, T. Hassner, and I. Maoz, "Face recognition in unconstrained videos with matched background similarity," in *CVPR 2011*, 2011, pp. 529–534.
[24] P. J. Phillips, H. Wechsler, J. Huang, and P. J. Rauss, "The FERET database and evaluation procedure for face-recognition algorithms," *Image Vis. Comput.*, vol. 16, no. 5, pp. 295–306, Apr. 1998.
[25] D. L. Spacek, "Faces94 a face recognition dataset," 2007.
[26] H.-W. Ng and S. Winkler, "A data-driven approach to cleaning large face datasets," in *2014 IEEE International Conference on Image Processing (ICIP)*, 2014, pp. 343–347.
[27] A. Salvador, X. Giro-I-Nieto, F. Marques, and S. Satoh, "Faster R-CNN Features for Instance Search," in *IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops*, 2016, pp. 394–401.
[28] G. Tolias, R. Sicre, and H. Jégou, "Particular object retrieval with integral max-pooling of CNN activations," Nov. 2015.
[29] N. Passalis and A. Tefas, "Spatial Bag of Features Learning for Large Scale Face Image Retrieval," 2017, pp. 8–17.
[30] N. Passalis and A. Tefas, "Learning Neural Bag-of-Features for Large-Scale Image Retrieval," *IEEE Trans. Syst. Man, Cybern. Syst.*, vol. 47, no. 10, pp. 2641–2652, 2017.