A Simple and Effective Technique for Face Clustering in TV Series

Vivek Sharma, M. Saquib Sarfraz, Rainer Stiefelhagen
CV:HCI, Karlsruhe Institute of Technology
{vivek.sharma, saquib.sarfraz, rainer.stiefelhagen}@kit.edu

Abstract

In this paper, we present a simple aggregation of frame-level CNN features in a face track to produce a track-level feature representation for face clustering in movies or videos. The approach is invariant of the image sequence and the number of frames the track has. We demonstrate the effectiveness of this strategy on three challenging benchmark video face clustering datasets: Big Bang Theory, Buffy the Vampire Slayer, and Notting Hill. Experiments using our straightforward strategy shows promising results on all the datasets. In addition, our strategy is useful in improving the baseline performance of generic face clustering methods without using any additional external constraints.

1. Introduction

Face clustering in videos has attracted quite a lot of attention, due to the potential applications in video summarization, content-based indexing & retrieval, video segmentation, and character interaction analysis. Even if considerable progress is made, face clustering is hugely affected due to issues like camera motion, continuously changing viewpoints, illumination, resolution and noise. To address these issues, several attempts have been made to cope with this by employing visual constraints [2, 12, 14, 15] based on face tracks, for example (1) the must-link, and (2) the must-not-link constraints, are often used to specify that if the two pairs of faces that appear in a track or frame should be linked together or not. However, these methods usually rely on hand-crafted features [2, 14], thus have been susceptible to cope with image appearance variation and other issues. This implies that the solution of the face clustering lies in the feature representation. A good feature representation of positive face pairs should have small intra-distance, and large inter-distance to that of the negative pairs in that feature space.

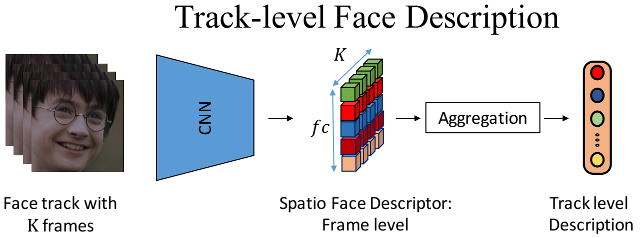

Figure 1: Track-level face description. Given a number of frames of a face track, we aggregate the track into a compact representation.

Recently, features derived from learning-based representation have been shown to outperform the hand-crafted descriptors, because they have the power of discovering and optimizing visual description for the specific task to be solved. In this context, Convolutional Neural Networks (CNN) are very effective feature learning methods [3, 6, 11]. The learned feature embeddings of VGG Deep Face (VDF) [9], and FaceNet [10] achieve state-of-the-art performance on the face recognition and verification tasks. As we show in this paper, the feature space learned by such a model is representative and already permits better face clustering under unconstrained appearance variations.

In this paper, we utilize this powerful feature representation, and extract features for each frame in the face track, and then aggregate the the frame-level features by averaging to form a track-level feature representation (see Fig. 1 for illustration). These feature are then fed to a clustering algorithm. We evaluate our method on three challenging video face clustering datasets, and show promising results in comparison to the state-of-the-art methods.

The rest of the paper is organized as follows. In Section 2, we discuss related work. Section 3 describes our proposed strategy. Experimental results are presented in Section 4. Finally, conclusions are drawn in Section 5.

2. Related Work

In this section, we review previous work on face clustering in videos. Cinbis et al. (ULDML) [2] utilize distance metric learning to automatically identify if two persons are same or not, by using pairs of faces within a track as positive

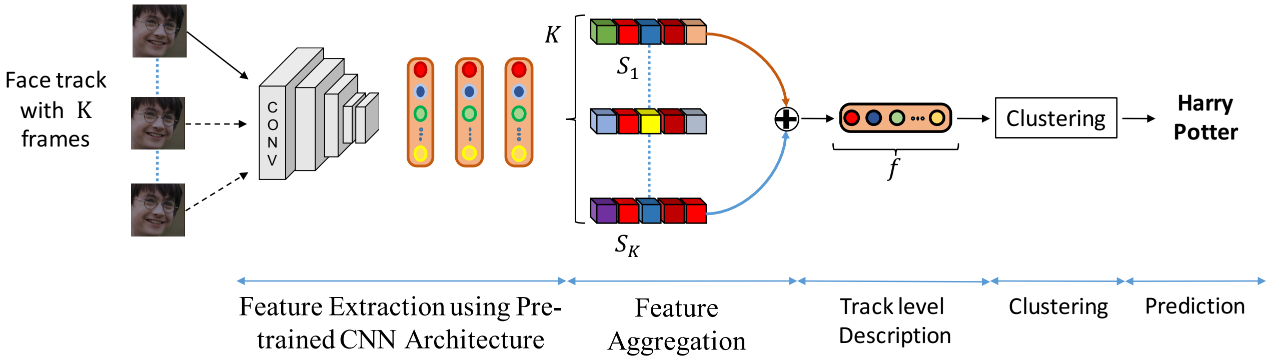

Figure 2: Overview of the full pipeline of our approach. Using a pre-trained network, which operates on RGB frames, the features maps are extracted for a given K frames in a track. The extracted features from each track are aggregated using simple element-wise aggregation operation. Finally, the aggregated track-level feature maps are fed to a clustering algorithm. The number of clusters is set to the actual number of main-casts in the video.

examples, while negative examples are obtained with must-not link constraints from pairs of face tracks of different people that appear together in a video frame. Wu et al. [14] explored Hidden Markov Random Fields (HMRF) to guide the clustering process by using label-level and constraint-level local smoothness priors as pairwise constraints. Xiao et al. [15] propose subspace clustering for face clustering in videos, by using prior knowledge while learning a low rank data representation via weighted block-sparse low rank representation (WBSLRR).

Parkhi et al [8] propose SIFT Fisher Vectors coding (VF^2) to aggregate all the frames in a video to form a video-level feature representation. In [12] using VF^2 features, Tapaswi et al. as a baseline merges tracks using agglomerative clustering assisted by cannot-link constraints. In [12], the authors further use video editing style towards merging tracks in the full episodes.

All the recent algorithms usually employ handcrafted features for face clustering thus the representation effectiveness is limited. Recently [16, 17] use CNNs to learn discriminative face feature representations for face clustering in videos. Zhang et al [16] use an improved triplet (Imp-Triplet) loss function in CNNs to learn positive and negative face pairs. Zhang et al [17] formulate the face clustering problem as a joint face representation adaptation and clustering (JFAC) approach. This is done by iterative discovery of weak pairwise identity constraints derived from potentially noisy face clustering result. In contrast to this, we use a deep face feature representation obtained from the VDF model [9]. We perform simple averaging of features of all the frames in the face tracks. As we will show, this is robust enough to embed in unique euclidean space of each identity without using any of the above mentioned constraints. We compare our straightforward approach with all the above mentioned methods.

3. Method

In a video, person tracking and identification has shown great success. Recently, several attempts have been made to come up with track-level representations [8] that encode all the frames together in a track. In our work, we exploit a deep CNN network's representative features to create a single feature map for the whole track. We expect that the feature space of each identity is unique, and should give near-perfect separation between other identities.

Consider there are N face tracks $(X_k^i, y_i)_{i=1}^N$, where each face track X^i is invariant of image sequence and can have varying number of frames k in the track, i.e. $X_k^i = \{x_1, \ldots, x_K\}$, $k \in [1, \ldots, K]$, and y_i is the corresponding label of the track, $y_i \in \{1, \ldots, T\}$, where T is the total number of assigned track-labels from the tracking algorithm. Using a pre-trained CNN network, the features maps of the last fully-connected layer (fc), for each image in the track, is extracted. The feature maps are vectors $\{S_1, \ldots, S_K\}$ of size $S \in \mathbb{R}^{D\times 1}$, where D denote the feature dimensions of the CNN fc feature maps. An aggregation function $\phi : S_1, S_2, \ldots, S_K \rightarrow f$, aggregates K frames to output a single aggregated feature map $f \in \mathbb{R}^{D\times 1}$. ϕ allows us to aggregate the whole track into a compact and robust single track-level feature representation. We investigate three different functions ϕ for feature aggregation, they are (i) element-wise average of track, (ii) element-wise maximum of track, and (iii) element-wise multiplication of track. Of all the aggregation functions, element-wise average of feature maps yielded best results and was therefore selected.

The resulting aggregated features are then ℓ_2 normalized $(f^{'} \leftarrow f/||f||_2)$ to be unit vectors. The feature vectors $f^{'}$ are then fed to a clustering algorithm to merge the tracks of each identity into the ideal number of clusters (C) i.e. the number of main-casts in the video. We compare two clustering algorithms in our work, they are (i) bottom-up hierarchical agglomerative clustering, and (ii) K-means clustering. The full pipeline is illustrated in Figure 2.

4. Evaluation

In this section, we first introduce the datasets, implementation details and evaluation metrics of the proposed approach. Then we demonstrate the applicability of our strategy, and finally, compare it with the state-of-the-art.

4.1. Datasets & Implementation Details

We conduct experiments on three challenging face clustering datasets, namely *Buffy the Vampire Slayer* (BF) [17], *Big Bang Theory* (BBT) [13, 16], and *Notting Hill* (Notting Hill) [14, 17]. Following the protocol of the current face track clustering studies [12, 16, 17], we evaluate on episodes 1-6 from season 5 for BF (BF05-01, . . . ,BF05-06), episode 1 from season 1 for BBT (BBT0101), and Notting Hill. For all the datasets, we use the same number of main-casts, while there is a difference in the number of tracks due to the difference in usage of different trackers. In particular, in our case the face tracks were obtained by a tracking-by-detection method using particle filter [4]. In Table. 1, we summarize the statistics of tracks for each dataset for our method, and compare the statistics of the tracks used for the same dataset by other papers.

We employ the VGG Deep Face (VDF) model [9] pre-trained on 2.6M face images. For feature extraction, the input RGB images are resized to a size of 227 × 227, and then mean-subtracted by value of 128. In particular, we extract the $fc7$ features of the network, resulting in 4096 dimensional descriptor vectors. The features are then ℓ_2 normalized before they are fed to the clustering algorithm. Following the protocol of the current face track clustering studies [16, 17], we set the number of clusters to the same number of main-casts.

4.2. Evaluation Metrics

We use three measures to evaluate the quality of clustering, (i) Accuracy: computed from a confusion matrix between the predicted cluster labels and the actual ground-truth classes., (ii) Weighted Clustering Purity (WCP) [12] is a measure to check the purity of each cluster, as we want to perform clustering with no errors. WCP is given as $W = \frac{1}{N}\sum_{c \in C} n_c \bullet p_c$, where N is the total number of tracks in the video, n_c is the number of tracks in the cluster $c \in C$, and its purity p_c is measured as the fraction of the largest number of tracks from the same label to n_c, and C is the total number of clusters., (iii) Operator Clicks Index (OCI-k) [5, 12] is a measure to report the clustering quality computed in terms of the number of clicks required to label all the face tracks for a given clustering. Given as $OCI - k = C + E$, where E is the total number of samples which are wrongly clustered in C clusters. All the measures of clustering evaluation are done at track-level. All the evaluation metrics are widely employed in video face clustering methods [12, 14, 17, 18].

Datasets	Main-casts	# Tracks (Ours)	# Tracks
BF0502	6	575	229 [17]
BBT0101	5	601	182 [13, 16]
Notting Hill	5	240	76 [14, 17]

Table 1: Comparison of statistics of the datasets, BF0502, BBT0101, Notting Hill used in our experiments.

4.3. Evaluation of aggregation functions

In this section, we investigate different aggregation functions (ϕ) to come up with a single compact and robust features representation for the whole track. The reported performance is the clustering accuracy in (%). Specifically, we explore three aggregation functions (i) element-wise multiplication (Mul), (ii) element-wise maximum (Max), and (iii) element-wise average (Avg). In Table 2, we report the performance. Of all the functions, averaging performs the best, and was therefore selected as a default aggregation function. Also, averaging should be preferred because it is less affected due to some erroneous frames of a wrong identity in a face track.

4.4. Comparison with the state-of-the-art

Finally, after exploring the aggregation function, we now compare our method with the current state-of-the-art approaches. Table 4, shows the comparison of our method to the published state-of-the-art in terms of clustering accuracy on BF0502, Notting Hill and BBT0101. A good feature representation should lie closer together in the embedding space to the features of its own identity without any addition constraints as we also observed in our evaluations. We can observe that our simple averaging of track-level VDF features (VDF ($fc7$,Avg)) outperforms all the methods on Notting Hill, and is second to the state-of-the-art on BF0502 and BBT0101. Interestingly, one can also observe that, our method of using a simple averaging of track-level representation is computationally efficient, and more effective and robust in comparison to all of the other methods. We expect that, any constraints modeled on top of this representation

Aggregation Function (ϕ)	BF0502	Notting Hill	BBT0101
Element-wise Multiplication (Mul)	58.53	88.81	76.13
Element-wise Maximum (Max)	82.31	99.08	78.91
Element-wise Average (Avg)	87.46	99.26	89.62

Table 2: Comparison of HAC clustering accuracy (%) using the aggregation functions for fusing the VDF $fc7$ features on BF0502, Notting Hill and BBT0101.

Episodes	**BF05-01**			**BF05-02**			**BF05-03**			**BF05-04**			**BF05-05**			**BF05-06**		
#tracks	630			779			974			668			646			843		
#ideal	11			15			13			15			18			18		
Measures	NC	WCP	OCI-k	NC	WCP	OCI-k	NC	WCP	OCI-k	NC	WCP	OCI-k	NC	WCP	OCI-k	NC	WCP	OCI-k
HAC-Neg [12]	534	1.000	534	688	1.000	688	852	1.000	852	566	1.000	566	575	0.999	576	751	1.000	751
VDF ($fc7$,Avg) +K-means (**Ours**)	534	0.952	571	688	0.965	718	852	0.975	877	566	0.945	611	575	0.964	601	751	0.970	780
TC [12]	466	1.000	466	598	1.000	598	730	1.000	730	494	1.000	494	507	0.998	508	643	1.000	643
VDF ($fc7$,Avg) +K-means (**Ours**)	466	0.939	513	598	0.950	641	730	0.963	767	494	0.925	556	507	0.946	547	643	0.942	700

Table 3: Comparison of our aggregated VDF features with state-of-the-art methods on the basis of WCP and OCI-k on season 5 of BF dataset (BF05) with 01-06 referring to episode number, where NC is the number of clusters.

is an add-on, and thus in turn can lead to more accurate clustering.

Furthermore, in Table 3, we compare the performance in terms of WCP and OCI-k on the episodes 01-06 of the BF sitcom series. In this case, our feature representation is compared with the SIFT based Fisher encoding (VF^2) [8] using (1) Hierarchical Agglomerative Clustering with Negative Constraints (HAC-Neg) [12], and (2) Scene-level clustering (TC) [12] on full episodes. For a fair comparison, we use the same baseline number of clusters (NC) as used in [12], and report the WCP and OCI-k measures. We can observe that the purity of the clusters using averaged feature representation is effective, and competitive enough to the other methods.

5. Discussion and Remarks

– **Impact of feature representation?** When CNN training is not performed, fusion of features could be beneficial or lead to adverse affects. We show that averaging of features of all the frames provide highly discriminative features for the whole track. Moreover, an averaging operation may prove robust to any noise such as an erroneous frame (of incorrect identity) in that track. Feature representation plays the most essential and crucial role to cluster face tracks of an identity in a video. If the feature representation is robust, we can expect that the face tracks of each identity shall be merged together in a unique cluster without modeling any additional constraints.

Method	BF0502	Notting Hill	BBT0101
ULDML [2]	41.62	73.18	57.00
HMRF [14]	50.30	84.39	60.00
PPC [7]	78.88	–	78.88
WBSLRR [15]	62.76	96.29	72.00
JFAC [17]	**92.13**	99.04	–
McAFC [18]	–	96.05	–
CMVFC [1]	–	93.42	–
Imp-Triplet [16]	–	–	**96.00**
VDF ($fc7$,Avg) +K-means (**Ours**)	87.46	98.31	89.20
+HAC (**Ours**)	87.46	**99.26**	89.62

Table 4: Comparison of clustering accuracy (%) of our aggregated VDF features with state-of-the-art methods on BF0502, Notting Hill, and BBT0101 dataset.

– **Does constraints help to merge tracks in face clustering?** Conventional techniques for face clustering use handcrafted features that are not very effective in the presence of illumination, and viewpoint variations. In this setting, *must-link* and *must-not-link* pairwise constraints are useful. However, as we show when the feature representation is more discriminative, one can obtain a similar or even better clustering performance without using these constraints. We thus conjecture that any modeling performed on a powerful representation is complimentary to using such constraints. An important consideration in clustering is to provide a method that can automatically infer the effective number of clusters. Interestingly the recent published state-of-the-art methods in face clustering [16, 17] seem not to focus on this and rather propose increasing the quality of face track representation using some external constraints. These methods then use a fixed number of clusters based on a priori information, such as a given number of characters in the video. As we have shown, under such a setting, the face representation can be readily used by simple off line features and learning an expensive model trying to model additional constraints is not that meaningful. One should actually focus more on an unsupervised clustering technique based on these features that can infer the optimal number of clusters itself. An interesting approach in this direction is described in [12] where using handcrafted track-level feature descriptors they incorporated video editing structure details as constraints to group similar descriptors together. The main idea there was to use the shot level, thread level and scene level track information to group tracks together. In this paper we show that given a more representative powerful description one probably does not need such constraints or external video editing information to group these together. Our next step is now to devise a method that can infer an optimal number of clusters. Our work shows that this could be done without relying on any external constraints since the feature representation is discriminative enough to learn the data grouping with relaxed thresholds.

References

[1] X. Cao, C. Zhang, C. Zhou, H. Fu, and H. Foroosh. Constrained multi-view video face clustering. *IEEE TIP*, 2015.

[2] R. G. Cinbis, J. Verbeek, and C. Schmid. Unsupervised metric learning for face identification in tv video. In *ICCV*, 2011.

[3] A. Diba, V. Sharma, and L. Van Gool. Deep temporal linear encoding networks. In *CVPR*, 2017.

[4] E. Ghaleb, M. Tapaswi, Z. Al-Halah, H. K. Ekenel, and R. Stiefelhagen. Accio: a data set for face track retrieval in movies across age. In *ICMR*, 2015.

[5] M. Guillaumin, J. Verbeek, and C. Schmid. Is that you? metric learning approaches for face identification. In *ICCV*, 2009.

[6] A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. In *NIPS*, 2012.

[7] Z. Lu and T. K. Leen. Penalized probabilistic clustering. *Neural Computation*, 2007.

[8] O. M. Parkhi, K. Simonyan, A. Vedaldi, and A. Zisserman. A compact and discriminative face track descriptor. In *CVPR*, 2014.

[9] O. M. Parkhi, A. Vedaldi, and A. Zisserman. Deep face recognition. In *BMVC*, 2015.

[10] F. Schroff, D. Kalenichenko, and J. Philbin. Facenet: A unified embedding for face recognition and clustering. In *CVPR*, 2015.

[11] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.

[12] M. Tapaswi, O. M. Parkhi, E. Rahtu, E. Sommerlade, R. Stiefelhagen, and A. Zisserman. Total cluster: A person agnostic clustering method for broadcast videos. In *ICVGIP*, 2014.

[13] B. Wu, S. Lyu, B.-G. Hu, and Q. Ji. Simultaneous clustering and tracklet linking for multi-face tracking in videos. In *ICCV*, 2013.

[14] B. Wu, Y. Zhang, B.-G. Hu, and Q. Ji. Constrained clustering and its application to face clustering in videos. In *CVPR*, 2013.

[15] S. Xiao, M. Tan, and D. Xu. Weighted block-sparse low rank representation for face clustering in videos. In *ECCV*, 2014.

[16] S. Zhang, Y. Gong, and J. Wang. Deep metric learning with improved triplet loss for face clustering in videos. In *Pacific Rim Conference on Multimedia*, 2016.

[17] Z. Zhang, P. Luo, C. C. Loy, and X. Tang. Joint face representation adaptation and clustering in videos. In *ECCV*, 2016.

[18] C. Zhou, C. Zhang, H. Fu, R. Wang, and X. Cao. Multi-cue augmented face clustering. In *ACM'MM*, 2015.