AN ONLINE ALGORITHM FOR CONSTRAINED FACE CLUSTERING IN VIDEOS

Prakhar Kulshreshtha, Tanaya Guha

Indian Institute of Technology, Kanpur

ABSTRACT

We address the problem of face clustering in long, real world videos. This is a challenging task because faces in such videos exhibit wide variability in scale, pose, illumination, expressions, and may also be partially occluded. The majority of the existing face clustering algorithms are offline, i.e., they assume the availability of the entire data at once. However, in many practical scenarios, complete data may not be available at the same time or may be too large to process or may exhibit significant variation in the data distribution over time. We propose an *online* clustering algorithm that processes data sequentially in short segments of variable length. The faces detected in each segment are either assigned to an existing cluster or are used to create a new one. Our algorithm uses several spatio-temporal constraints, and a convolutional neural network (CNN) to obtain a robust representation of the faces in order to achieve high clustering accuracy on two benchmark video databases (82.1% and 93.8%). Despite being an online method (usually known to have lower accuracy), our algorithm achieves comparable or better results than state-of-the-art offline and online methods.

***Index Terms*—** Online clustering, face clustering, deep features, constrained clustering, movie analysis

1. INTRODUCTION

We are interested in automatically clustering faces into disjoint groups appearing in long, real-world videos such that each group consists of faces from a single person. An effective solution to this problem has direct application to many areas, such as video summarization and indexing, content-based video retrieval, character discovery and analytics [1], and organizing and managing a large collection of faces [2]. This is a highly challenging task because faces in real world videos exhibit wide variability in scale, pose, illumination, expressions, appearance (owing to changes in hair style, make-up), and also may be partially occluded [2, 3].

The problem of face clustering is relatively less studied as compared to face recognition - its supervised counterpart. The dominant approach to face clustering (both in images and videos) is completely unsupervised, where the primary objective is to learn a suitable distance measure between the data samples [4, 5, 6, 7]. Several methods [8, 9] have proposed to use partial supervision to improve clustering performance. In the context of video-based clustering, significant improvement can be achieved by exploiting the temporal information about the occurrence of the faces [2]. For example, when two faces appear at the same time in a video, they can be safely assumed to belong to two different entities. Such constraints were used in a hidden Markov random field-based framework (HMRF) [2] to perform face clustering in videos from movies and TV series. Another constrained clustering approach, called the unsupervised logistic discriminative metric learning (ULDML) [10], learned a metric optimizing over similar temporal constraints. A constrained multiview video face clustering technique [11] used constrained sparse subspace representation of faces with constrained spectral clustering. Constrained or not, all clustering techniques ultimately computes a distance between faces in a feature space. An effective representation of faces hence can dramatically improve the performance of any clustering algorithm. Recent clustering approaches choose convolutional neural networks (CNN) to learn robust representation of faces [12, 5, 13]. Face clustering in videos has been shown to improve by using aggregated deep features [12], deep features with pairwise constraints [5], and by using deep features with triplet loss [13].

The approaches discussed above are all *offline* methods i.e., they assume the availability of the entire data at once. In this paper, we address a more challenging problem, that is of clustering faces in an *online* manner. In an online setting, a clustering algorithm does not have the luxury of 'seeing' the entire data simultaneously. Instead, data is available in small chunks, and prior information about the number of characters (cluster count) is also not available. Evidently, this is a more difficult scenario, and is known to trail offline methods in terms of performance. Nevertheless, online clustering can be highly useful in cases where the entire data is not available at once, or has to be processed and monitored for a very long duration (e.g., in surveillance systems). To the best of our knowledge, there is only one work on online face clustering in videos in the existing literature [14]. This work created small tracklets of faces from the video, and clustered them in an online fashion based on temporal coherence and the Chinese restaurant process (TCCRP) [14] . However, this online method tends to create multiple clusters for the same person thereby degrading the completeness (whether or not a cluster contains all data from the same class) of the clusters.

In this work, we propose an online method to cluster faces in real world videos as they appear in the video stream. Our algorithm processes a video in small segments or 'shots', where the shot boundaries are also detected automatically in an online manner. Faces are detected in each frame within a shot, and a spatial matching technique is used to create face tracks pertaining to individual characters. In order to obtain a robust representation of a face track, a CNN-based architecture, called the FaceNet [4], is employed. Using the deep features and several temporal constraints, we develop an online algorithm to cluster faces. Unlike most offline clustering methods, the proposed method does not require any prior knowledge of the number of clusters to be formed. The proposed method, despite being an online approach, achieves comparable or better clustering accuracy than several state-of-the-art offline methods on two real-world video databases.

2. PROPOSED APPROACH

In this section, we describe our online clustering approach in detail. It consists of two steps: (i) face track creation, and (ii) online clustering of the face tracks. Fig. 1 presents an overview of the proposed method.

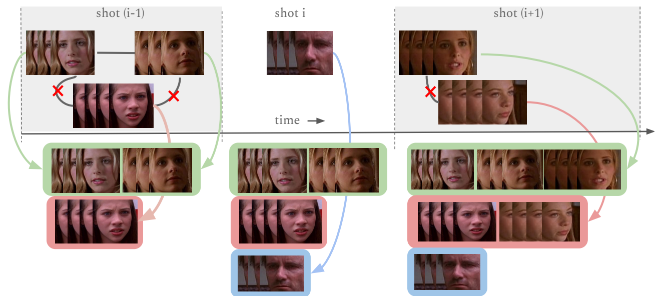


Fig. 1. Overview of the proposed online clustering algorithm.

2.1. Face track creation

In this step, we perform the following subtasks: shot boundary detection, face detection and face track creation.

Shot boundary detection: We process each movie at the shot level. A shot is defined as a contiguously recorded sequence of frames [15]. Given a movie, we first detect the shot boundaries in an online fashion. For our framework, the accuracy of shot boundary detection is not critical, hence we stick to the simple frame difference-based method [16] for detecting the shot boundaries. Let us consider a movie $\mathcal{M}$ that comprises T frames: $\mathcal{M} = \{I_t\}_{t=1}^{T}$. The i^{th} shot S_i is defined as a sequence of consecutive frames $\{I_{t_i}, I_{t_i+1} \ldots I_{t_{(i+1)}-1}\}$, where t_i is the i^{th} shot boundary, and t_1 is initialized as the first frame. The i^{th} shot boundary T_i is detected when the distance between consecutive frames exceeds a predefined threshold γ. This is given by $T_i \leftarrow t : \{\beta(I_{t-1}, I_t) \geq \gamma$, $|\beta(I_{t-2}, I_{t-1}) - \beta(I_{t-1}, I_t)| \geq \gamma\}$, where $\beta(\cdot)$ is average pixel difference between two frames.

Face detection and deep feature extraction: After detecting the shot boundary t_{i+1}, we obtain the i^{th} shot $\mathcal{S}_i$. We run a standard face detector [17, 18] on each frame in $\mathcal{S}_i$. Note that this frame level face detection can be done in parallel to searching for shot boundaries. The face detector returns the corner coordinates of the rectangular bounding box of each face detected in every frame. To build a robust representation of these faces, we employ a convolutional neural network (CNN) that can extract robust features from them. We use the *FaceNet* - a variant of CNN that is trained on a triplet loss involving matching and non-matching face patches [4]. The features learned using FaceNet have been shown to achieve state-of-the-art face verification and clustering results even using standard techniques, such as k-nearest neighbor (kNN) [4]. The learned features are also observed to be robust to pose variation, lighting conditions and image resolution [4]. Each face $\mathbf{f}^p$ is forward-passed through the pretrained FaceNet to obtain its d dimensional unit-norm feature-vector $\mathbf{v}^p$.

Face track creation: Instead of tracking individual faces, we use a simple yet effective strategy to combine the faces detected in consecutive frames to form a face track. Let us define the cropped-faces $\mathbf{f}^p$ and $\mathbf{f}^q$ pertaining to two faces detected in two consecutive frames. The overlap percentage $a(\cdot)$ between two faces is defined as:

$$a(p,q) = \frac{\text{area}(\mathbf{f}^p \cap \mathbf{f}^q)}{\max(\text{area}(\mathbf{f}^p), \text{area}(\mathbf{f}^q))} * 100 \quad (1)$$

where $\text{area}(\mathbf{f})$ is the area of the rectangular bounding box $\mathbf{f}$. The squared feature-distance between feature vectors $\mathbf{v}^p$ and $\mathbf{v}^q$ are defined as $\delta(p,q) = \|\mathbf{v}^p - \mathbf{v}^q\|^2$. If $a(p,q) > 0.85$ and $\delta(p,q) \leq 0.1$

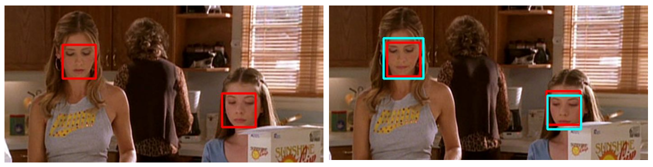


Fig. 2. Example of 85% spatial overlap between face pairs in two consecutive frames.

i.e. if the faces have more than 85% overlap and less than 0.1 feature distance in consecutive frames, they are considered to be of the same person (see Fig. 2). Detected faces that overlap in such way in consecutive frames are thus combined to form a sequence of cropped faces, called a *face track*, and the sequence of features corresponding to each of these faces is defined as a *featuretrack*.

2.2. Online face clustering

Our next task is to cluster the face tracks sequentially as they appear in each shot. We assume the availability of all face tracks in a single shot at a time. Our goal is to assign a face track belonging to a person who has appeared earlier to the correct existing cluster, and to form a new cluster for a face track belonging to a person appearing for the first time.

We cluster the face tracks across shots using their deep features. Let us consider the i^{th} shot $\mathcal{S}_i$ for processing. Upto this point, we have processed $(i-1)$ number of shots, and have obtained a set of L clusters corresponding to the L characters. The clusters are represented by their corresponding cluster centers $\mathcal{C} = \{\mathbf{c}_l\}_{l=1}^{L}$, where $\mathbf{c}_l \in \mathbb{R}^d$. A cluster center is simply defined as the mean feature vector obtained by averaging all features across all face tracks contained in that cluster. Let us also assume that there are K face tracks in $\mathcal{S}_i$, where each face track $\mathcal{F}_k$ is represented by its featuretrack set $\mathcal{V}_k$. Note that the number of clusters and the clusters themselves are dynamic since they evolve as each shot is processed. To this end, we define three matrices as follows:

- A *temporal constraint matrix* $\mathbf{Q} \in \mathbb{R}^{K \times K}$ is defined as

$$\mathbf{Q}(p,q) = \begin{cases} 0 & \text{if } \mathcal{F}_p \text{ and } \mathcal{F}_q \text{ overlap in time} \\ 1 & \text{otherwise} \end{cases} \quad (2)$$

  where, $p, q \in \{1, 2, \ldots, K\}$. The matrix $\mathbf{Q}$ enforces a temporal constraint on the face tracks such that if two face tracks have any overlap in time, they are considered to belong to two different characters, and hence, are assigned to different clusters.

- A *similarity matrix* $\mathbf{D} \in \mathbb{R}^{L \times K}$ which measures the similarity between a face track (represented by $\mathcal{V}_k$), and a cluster center c_l for a given shot.

$$\mathbf{D}(l,k) = d(\mathbf{c}_l, \mathcal{V}_k) = 4 - \frac{1}{N_k}\sum_{j=1}^{N_k} \|\mathbf{v}_k^j - \mathbf{c}_l\|_2^2 \quad (3)$$

  where $l = 1, 2, \ldots, L$, and $k = 1, 2, \ldots, K$. The second component is an average squared distance, the maximum value of of which is 4 (since each feature is a unit vector). By subtracting the distance from 4 we obtain a similarity value between $[0, 4]$.

- A *weight matrix* $\mathbf{W} \in \mathbb{R}^{L \times K}$ initialized with all ones.

Algorithm 1: Facetrack clustering for a given shot.

```
Input: Face track features in the current shot: ${\mathcal{V}_k}_{k=1}^K$,
       Initial clusters: $\mathcal{C}$
Output: Updated $\mathcal{C}$
Initialize: $\mathbf{ind} = [1, 2, \ldots, K]$, $\mathbf{W}$ = all-ones matrix.

Compute $\mathbf{Q}$, $\mathbf{D}$ using (2) and (3)
while length($\mathbf{ind}$) $> 0$ do
    if $\mathcal{C}$ not empty && $\max_{l,k}(\mathbf{D} \odot \mathbf{W}) \geq \tau$ then
        $(\hat{l}, \hat{k}) \leftarrow \text{argmax}_{l,k}(\mathbf{D} \odot \mathbf{W})$
        $k^* \leftarrow \mathbf{ind}[\hat{k}]$
        Update cluster center $\mathbf{c}_{\hat{j}}$ with $\mathcal{V}_{k^*}$
    else
        Add new cluster $(\hat{l}, \hat{k}) \leftarrow (L+1, 1)$
        $k^* \leftarrow \mathbf{ind}[\hat{k}]$
        $\mathbf{c}_{new} \leftarrow \text{mean}(\mathcal{V}_{k^*})$
        $\mathcal{C} \leftarrow \mathcal{C} \cup \mathbf{c}_{new}$
    end
    Recompute $\mathbf{D}$ for $\mathbf{c}_{\hat{l}}$
    $\mathbf{W}(\hat{l}, :) \leftarrow \mathbf{Q}(\hat{k}, :)$
    Delete $\mathbf{D}[:, \hat{k}], \mathbf{W}[:, \hat{k}], \mathbf{Q}[\hat{k}, :], \mathbf{Q}[:, \hat{k}], \mathbf{ind}[\hat{k}]$
end
```

Given $\mathcal{V}_1, \ldots, \mathcal{V}_K$ in $\mathcal{S}_i$ shot, our task is to assign them (one at a time) to their respective clusters or create a new cluster. This is done by computing the similarities between each face track and each cluster center, and obtaining the face track with the highest similarity to an existing cluster. For keeping track of the ids of face tracks we define a vector $\mathbf{ind} = [1, 2, \cdots K]$

$$(\hat{l}, \hat{k}) = \underset{l,k}{\text{argmax}}(\mathbf{D} \odot \mathbf{W}) \tag{4}$$

where $\odot$ denotes element wise product between two matrices. Let $k^* = \mathbf{ind}[\hat{k}]$. For $\max_{l,k}(\mathbf{D} \odot \mathbf{W}) \geq \tau$, where τ is a user-defined threshold, the faces in $\mathcal{V}_{k^*}$ are considered similar to those in the cluster $\mathbf{c}_{\hat{l}}$. If the face track $\mathcal{V}_{k^*}$ is assigned to the $\hat{l}^{th}$ cluster, we update the cluster center $\mathbf{c}_{\hat{l}}$ by averaging over the existing and the newly added faces. On the other hand, if $\max_{l,k}(\mathbf{D} \odot \mathbf{W}) < \tau$, then none of the clusters is considered close enough to $\mathcal{V}_{k^*}$, and hence, a new cluster $\mathbf{c}_{new}$ is created. Thus it belongs to a completely new person, we add a new cluster to the set of clusters $\mathcal{C} \leftarrow \mathcal{C} \cup \mathbf{c}_{new}$ with its center being the mean of all the vectors in $\mathcal{V}_{k^*}$. Note that since $\mathbf{W}$ is initialized as a matrix of all ones, it has no effect on the clustering of the first face track i.e., the face track with the highest similarity to any cluster. For the subsequent assignments of face tracks to clusters, $\mathbf{W}$ is updated to add temporal constraints.

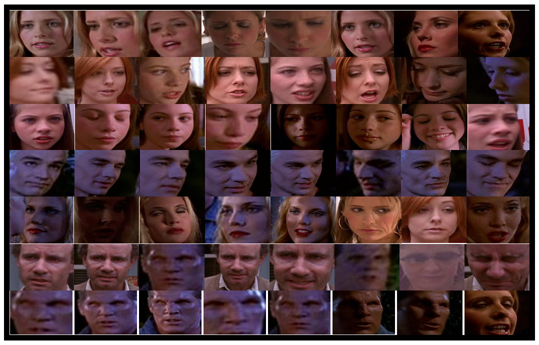
After $\mathcal{V}_{\hat{k}}$ is assigned to a cluster, we update the similarity matrix $\mathbf{D}$ and the weight matrix $\mathbf{W}$ as follows.

- Case I: $\mathcal{V}_{\hat{k}}$ is assigned to an existing cluster $\hat{l}$

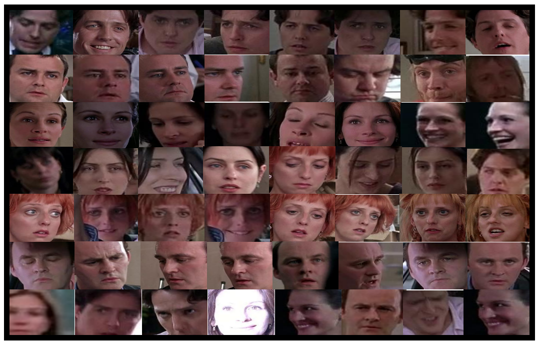
$$\mathbf{W}(\hat{l}, :) \leftarrow \mathbf{Q}(\hat{k}, :) \tag{5}$$

This updated $\mathbf{W}$ will make $\mathbf{D} \odot \mathbf{W}$ zero for all the face tracks having any temporal overlap with $\mathcal{V}_{k^*}$ in the $\hat{l}^{th}$ row.

$$\mathbf{D}(\hat{l}, k) = d(\mathbf{c}_{\hat{l}}, \mathcal{V}_k) \quad \text{for } k \in [1, |\mathbf{ind}|] \tag{6}$$



(a) 6 character clusters, 1 noisy cluster (7th row) in BF2006.



(b) 6 character clusters, 1 noisy cluster (7th row) in NH2016.

Fig. 3. Sample faces from all the clusters created by the proposed online algorithm on the two video databases.

- Case II: $\mathcal{V}_{\hat{k}}$ is assigned to a new cluster

$$\hat{l} = |\mathcal{C}| + 1 \tag{7}$$
$$\mathcal{C} \leftarrow \mathcal{C} \cup \mathbf{c}_{new} \tag{8}$$
$$\mathbf{W}(\hat{l}, :) \leftarrow \mathbf{Q}(\hat{k}, :) \tag{9}$$
$$\mathbf{D}(\hat{l}, k) = d(\mathbf{c}_{new}, \mathcal{V}_k) \text{ for } k \in [1, |\mathbf{ind}|] \tag{10}$$

As track $\mathcal{V}_{k^*}$ is processed and sent into a cluster, its id is removed i.e. $\hat{k}^{th}$ element of $\mathbf{ind}$, $\hat{k}^{th}$ column of $\mathbf{D}$ and $\mathbf{W}$, and $\hat{k}^{th}$ row and column of $\mathbf{Q}$ are removed. This process goes on till all the tracks in the shot are processed, and then we move to next shot. Algorithm 1 summarizes this online clustering process for a given shot [1].

3. PERFORMANCE EVALUATION

In this section, we present detailed evaluation of the proposed online face clustering algorithm, and compare its performance with several existing algorithms on two video databases.

3.1. Databases

We have used two real-world video databases which are commonly used to benchmark face clustering algorithms: (i) The *Buffy* database (BF2006) [3, 2] containing 229 face tracks of 6 characters (17,337 faces altogether) extracted from the episode 2, season 5 of the TV series *Buffy - the Vampire Slayer*. The database includes the frame number, bounding box coordinates, track ids, and the character names for each face. (ii) The *Notting Hill* database (NH2016) [19]

[1]Code and experiments available at https://github.com/ankuPRK/COFC

Table 1. Comparison with the baselines (all with FaceNet features)

	BF2006		NH2016	
	V measure	*F* score	*V* measure	*F* score
GMM	**0.80**	**0.82**	0.79	0.72
Kmeans	0.78	0.81	0.78	0.72
Proposed	0.68	0.73	**0.89**	**0.94**

Table 2. Comparison with the online face clustering method

	BF2006		NH2016	
	TCCRP [14]	Proposed	TCCRP [14]	Proposed
Homogeneity	**0.93**	0.68	**0.92**	0.88
Completeness	0.44	**0.69**	0.44	**0.89**
V measure	0.60	**0.68**	0.58	**0.89**
clusters	57	7	61	7

that contains 277 facetracks of 7 characters (19,278 faces altogether) from the movie *Notting Hill*. It contains the frame numbers, bounding boxes, track ids, features and character names for each face in the database.

3.2. Results and discussion

For the videos in each database, we obtain shot boundaries, create face tracks, extract deep features and cluster the faces using our proposed algorithm (see Algorithm 1). The value of the threshold parameter τ is set to 2.80 and 2.85 for the BF2006 and the NH2016 database. Sample faces from all the clusters created by our algorithm are presented in Fig. 3. For BF2006, we get a cluster for each of the 6 characters, and for NH2016, we get a cluster for 6 / 7 characters.

Comparison with baselines: To evaluate the performance of the proposed method, we first create two baselines: Gaussian mixture model (GMM) with FaceNet features, and Kmeans with FaceNet features. We compare the proposed online method with these baseline methods in terms of *V measure* [20] and *F score* [21] in Table 1. The *V measure* (bounded between 0 to 1) is an entropy-based measure of cluster homogeneity and completeness. The *F score* (bounded between 0 to 1) is the geometric mean of precision and recall. For both the metrics higher value indicates better performance. Results in Table 1 show that the performance of the baselines is slightly better than the proposed method on the BF2006 database, while the proposed algorithm outperforms the baselines on NH2016 database. Note that the baselines are offline methods, while our proposed method is online.

Comparison with online face clustering: As mentioned earlier, to the best of our knowledge only one work on online face clustering [14] in the literature. We used the original TCCRP code provided by the authors [14]. Although not originally used with FaceNet features, we combine TCCRP with FaceNet features for better performance, and used a tracklet length of 10. For both TCCRP and our method, no detected face is rejected, and no cluster created by the algorithms is discarded. We compare the proposed method with TCCRP in terms of three entropy based-measures [20]: *homogeneity* score, *completeness* score and their harmonic-mean i.e., the *V measure* (see Table 2). F score has not been used as the number of clusters formed by TCCRP and our method are quite different, and hence a fair comparison is not possible. Table 2 shows that TCCRP has higher cluster homogeneity, but this is achieved at the cost of overclustering (note the large number of clusters created by TCCRP) and thereby degrading completeness. Our method achieves

Table 3. Performance comparison with the state-of-the-art (offline) clustering methods in terms of clustering accuracy (%)

Method	**BF2006**	**NH2016**
ULDML [10]	49.29	43.82
cHMRF [19]	61.87	47.94
FaceNet + aCNN [12]	**89.90**	90.17
FaceNet + GMM	84.21	73.46
FaceNet + Kmeans	82.92	71.66
Proposed	82.12	**93.84**
Proposed + GMM	93.79	94.17

significantly higher completeness and V measure while discovering a more accurate number of clusters.

Comparison with the state-of-the-art: We also compare the performance of our method with several state-of-the-art face clustering methods: (i) ULDML [10], (ii) a recently proposed constrained clustering method - the coupled HMRF (cHMRF) [19], and (iii) an aggregated CNN feature-based clustering (aCNN) [12] Performances of all the method are compared in Table 3.2 in terms of clustering accuracy (expressed in %) which compares the predicted cluster labels with the ground truth labels. Note that all the methods in Table 3.2 are offline methods, where the entire data, information about the face tracks, and the cluster counts are provided as an input to the algorithms. For the proposed online method, however, no information about the face tracks or cluster counts are made available.

Note that the baseline clustering techniques, such as GMM and Kmeans when combined with deep (FaceNet) features outperform sophisticated clustering methods such as cHMRF. The performance of our algorithm on the BF2006 database is superior to that of cHMRF and ULDML, and is comparable to Kmeans. On the NH2016 database, our algorithm outperforms all its offline counterparts, achieving a clustering accuracy of 93.8%. Furthermore, we show that if GMM (similar trend is noticed with Kmeans) is initialized with the cluster centers discovered by the proposed method (see the last row of Table 3.2), even better clustering performance is achieved. This indicates the possibility of developing hybrid methods where the advantages of both online and offline methods can be exploited.

4. CONCLUSION

We addressed the problem of face clustering in real-world videos. The majority of the existing face clustering algorithms are offline that requires the entire data to be available simultaneously. The main contribution of this work is to propose a new online clustering algorithm that performs as good or better than existing online or offline methods. Our online algorithm uses CNN-based features for robust representation of faces, and enforces several spatio-temporal constraints to cluster the faces sequentially as they appear. The proposed algorithm achieves high clustering accuracy on two real-world video databases, outperforming all its offline counterparts on one database and showing comparable performance on the other. A natural extension of our work is to use multivariate Gaussians to represent a cluster instead of the centroids. Also, our algorithm at present only creates new clusters (if required), but does not further split or fuse existing clusters. Functionalities such as online splitting and fusing of clusters can further improve the clustering performance.

5. REFERENCES

[1] Tanaya Guha, Che-Wei Huang, Naveen Kumar, Yan Zhu, and Shrikanth S Narayanan, "Gender representation in cinematic content: A multimodal approach," in *Proceedings of the 2015 ACM on International Conference on Multimodal Interaction*. ACM, 2015, pp. 31–34.

[2] Baoyuan Wu, Yifan Zhang, Bao-Gang Hu, and Qiang Ji, "Constrained clustering and its application to face clustering in videos," in *Computer Vision and Pattern Recognition (CVPR), 2013 IEEE Conference on*. IEEE, 2013, pp. 3507–3514.

[3] Mark Everingham, Josef Sivic, and Andrew Zisserman, "Hello! my name is... buffy–automatic naming of characters in tv video," .

[4] Florian Schroff, Dmitry Kalenichenko, and James Philbin, "Facenet: A unified embedding for face recognition and clustering," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015, pp. 815–823.

[5] Zhanpeng Zhang, Ping Luo, Chen Change Loy, and Xiaoou Tang, "Joint face representation adaptation and clustering in videos," in *European conference on computer vision*. Springer, 2016, pp. 236–251.

[6] Yi Sun, Yuheng Chen, Xiaogang Wang, and Xiaoou Tang, "Deep learning face representation by joint identification-verification," in *Advances in neural information processing systems*, 2014, pp. 1988–1996.

[7] Quoc V Le, "Building high-level features using large scale unsupervised learning," in *Acoustics, Speech and Signal Processing (ICASSP), 2013 IEEE International Conference on*. IEEE, 2013, pp. 8595–8598.

[8] Ming Du and Rama Chellappa, "Face association across unconstrained video frames using conditional random fields," in *European Conference on Computer Vision*. Springer, 2012, pp. 167–180.

[9] Lior Wolf, Tal Hassner, and Itay Maoz, "Face recognition in unconstrained videos with matched background similarity," in *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*. IEEE, 2011, pp. 529–534.

[10] Ramazan Gokberk Cinbis, Jakob Verbeek, and Cordelia Schmid, "Unsupervised metric learning for face identification in tv video," in *Computer Vision (ICCV), 2011 IEEE International Conference on*. IEEE, 2011, pp. 1559–1566.

[11] Xiaochun Cao, Changqing Zhang, Chengju Zhou, Huazhu Fu, and Hassan Foroosh, "Constrained multi-view video face clustering," *IEEE Transactions on Image Processing*, vol. 24, no. 11, pp. 4381–4393, 2015.

[12] M. Saquib Sarfraz Sharma, Vivek and Rainer Stiefelhagen., "A simple and effective technique for face clustering in tv series," .

[13] Shun Zhang, Yihong Gong, and Jinjun Wang, "Deep metric learning with improved triplet loss for face clustering in videos," in *Pacific Rim Conference on Multimedia*. Springer, 2016, pp. 497–508.

[14] Adway Mitra, Soma Biswas, and Chiranjib Bhattacharyya, "Bayesian modeling of temporal coherence in videos for entity discovery and summarization," *IEEE transactions on pattern analysis and machine intelligence*, vol. 39, no. 3, pp. 430–443, 2017.

[15] Jeroen Vendrig and Marcel Worring, "Systematic evaluation of logical story unit segmentation," *IEEE Transactions on Multimedia*, vol. 4, no. 4, pp. 492–499, 2002.

[16] Johan Mathe, "Shotdetect: A free software which detects shots and scenes from a video," 2012.

[17] Davis E. King, "Dlib-ml: A machine learning toolkit," *Journal of Machine Learning Research*, vol. 10, pp. 1755–1758, 2009.

[18] Navneet Dalal and Bill Triggs, "Histograms of oriented gradients for human detection," vol. 1, pp. 886–893, 2005.

[19] Baoyuan Wu, Bao-Gang Hu, and Qiang Ji, "A coupled hidden markov random field model for simultaneous face clustering and tracking in videos," *Pattern Recognition*, vol. 64, pp. 361–373, 2017.

[20] Andrew Rosenberg and Julia Hirschberg, "V-measure: A conditional entropy-based external cluster evaluation measure," in *Proceedings of the 2007 joint conference on empirical methods in natural language processing and computational natural language learning (EMNLP-CoNLL)*, 2007.

[21] Edward B Fowlkes and Colin L Mallows, "A method for comparing two hierarchical clusterings," *Journal of the American statistical association*, vol. 78, no. 383, pp. 553–569, 1983.