VideoClusterNet: Self-Supervised and Adaptive Face Clustering for Videos

Devesh Walawalkar and Pablo Garrido

Flawless AI
{devesh.walawalkar,pablo.garrido}@flawlessai.com
https://www.flawlessai.com/

Abstract. With the rise of digital media content production, the need for analyzing movies and TV series episodes to locate the main cast of characters precisely is gaining importance. Specifically, Video Face Clustering aims to group together detected video face tracks with common facial identities. This problem is very challenging due to the large range of pose, expression, appearance, and lighting variations of a given face across video frames. Generic pre-trained Face Identification (ID) models fail to adapt well to the video production domain, given its high dynamic range content and also unique cinematic style. Furthermore, traditional clustering algorithms depend on hyperparameters requiring individual tuning across datasets. In this paper, we present a novel video face clustering approach that learns to adapt a generic face ID model to new video face tracks in a fully self-supervised fashion. We also propose a parameter-free clustering algorithm that is capable of automatically adapting to the finetuned model's embedding space for any input video. Due to the lack of comprehensive movie face clustering benchmarks, we also present a first-of-kind movie dataset: MovieFaceCluster. Our dataset is handpicked by film industry professionals and contains extremely challenging face ID scenarios. Experiments show our method's effectiveness in handling difficult mainstream movie scenes on our benchmark dataset and state-of-the-art performance on traditional TV series datasets.

1 Introduction

Video Face Clustering can be defined as the task of grouping together human faces in a video among common identities. It contributes significantly to several other research domains, such as video scene captioning [39], video question answering [49], and video understanding [52]. Having an understanding of the spatial location, face size, and identity of the characters that appear in specific scenes is essential for all the aforementioned tasks. Clustering faces in a video is a challenging unsupervised problem that has garnered a lot of interest over the past few decades [38, 40, 55, 63]. Given the rise in the creation of video production content and the subsequent need for its analysis, face clustering in the movie/TV series domain has garnered significant interest in the last couple of years [41, 47]. It serves as an effective editing tool for movie post-production

personnel, helping them select scenes with a specific group of characters, among other benefits. We thus primarily focus on the video production content domain for evaluating our proposed method, given its closeness to real-world scenarios and use cases.

The video production content domain often provides an unique set of challenges for face clustering, in terms of large variations in facial pose, lighting, expression and appearance (Fig. 1). In specific domains with high-quality standards, such as movies that possess a unique cinematic style[1], performance of face identification (ID) models trained on generic large-scale datasets is often limited for such domains (Tab. 6). Furthermore, hand labeling a large cast of characters, often present in movies/TV series, can be very time-consuming and error-prone. As a result, the inherent challenges in video face clustering and difficulties in hand labeling often limit video-specific model training. In this paper, we propose an algorithm that successfully tackles these limitations. Specifically, our proposed method adapts a generic face ID model to a specific set of faces and their observed variations in a given video in a fully self-supervised fashion. Also, traditional deep face clustering algorithms [66] present certain limitations, which can be categorized into two main types. The first group [14, 19] adopts a bottom-up approach to clustering and incorporates a pre-defined distance function to compare face embeddings, thus requiring a user-defined threshold to specify a positive match. The second group [10, 32] follows a top-down approach and requires the number of known clusters or minimum cluster sample count as input for its initialization. Thus, both these groups have shortcomings in terms of requiring non-intuitive user-defined parameters. We present a novel agglomerative clustering algorithm that improves on these limitations. It is fully automated without the need for any user input and uses a distance metric that is optimized for the given model-learned embedding space.



Fig. 1: Select hard case clusters predicted using our algorithm from within **MovieFaceCluster** dataset. Trivial face represents an easy ID sample for each cluster. The term "varying parameter" depicts the dominant image attributes that are particularly challenging for a given face crop. It is not part of the dataset annotations but is depicted for enhanced reader understanding.

Overall, our proposed method VideoClusterNet can be divided into two main stages. The first stage involves a fully self-supervised learning (SSL) based finetuning of a generic face ID model on a given set of video faces. The finetuning task is formulated as an iterative optimization task facilitated through alternating stages of model finetuning and coarse face track matching. The SSL finetuning bootstraps itself by soft grouping together high-confidence matching tracks

[1] Particular movie features include high resolution, high dynamic range, and large facial attribute variations.

at regular training intervals. The second stage involves a track clustering algorithm that adopts the loss function used for SSL finetuning as a distance metric. Our clustering algorithm computes a custom matching threshold for each track and combines tracks with high-confidence matches in an iterative bottom-up style.

Given the unique in-the-wild challenges in video production domain, the academic research community lacks a standardized video dataset benchmark for real-world performance evaluation. Thus, we also present a novel video face clustering dataset, which incorporates challenging movies hand-selected by experienced film post-production specialists. We conduct extensive experiments of our proposed method on this dataset to validate its effectiveness for character clustering in mainstream movies. In addition, we provide results on selects benchmark datasets, showing that our method attains state-of-the-art performance.

In summary, **we propose the following contributions:** 1) A fully self-supervised video face clustering algorithm, which progressively learns robust identity embeddings for all faces within a given video face dataset, facilitated via iterative soft matching of faces across pose, illumination, and expression variations observed in the dataset. 2) A self-supervised model finetuning approach that, unlike prior works, relies only on positive match pairs, removing any dependence on manual ground truth cluster labels or use of temporal track constraints to obtain negative match pairs. 3) A deep learning-based similarity metric for face clustering, which automatically adapts to a given model's learned embedding space. 4) A novel video face clustering algorithm that does not depend on any user-input parameters.

Additionally, we present a new comprehensive movie face clustering dataset to better evaluate video face clustering algorithms on real-world challenges.

2 Related Work

We review prior work in video-based face clustering and list out some deep learning metric and self-supervised learning based methods since they form an important component of our approach.

Auxiliary labels assisted Video Face Clustering: Single frame-based face clustering has drawn a lot of attention in the past few decades. For a detailed survey, please refer to [66]. For the video domain, early work focused on using additional information available from TV series episodes/movies. Specifically, methods such as [5, 11, 20, 21, 36, 38, 40, 46] utilize aligned captions, transcripts, dialogues, or a combination of the above with detected faces to perform identity clustering.

Contextual Information based Video Face Clustering: Following work using supplementary labels, methods such as [18, 59] leverage contextual information, e.g., clothing and surrounding scene contents, while [37] use aligned audio to help localize faces. Alternatively, [63] incorporate gender information

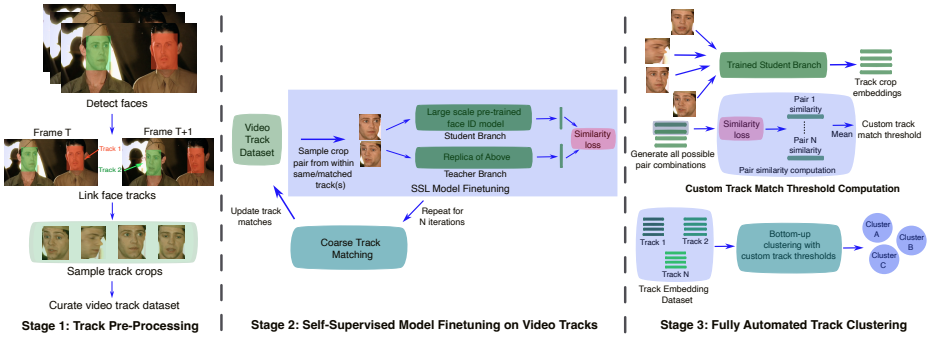


Fig. 2: Overview of VideoClusterNet: Stage 1: Given the temporal continuity in the video domain, faces detected in consecutive frames are first locally grouped into tracks using a motion tracking algorithm. Stage 2: A large-scale pre-trained face ID model is finetuned on these tracks using temporal self-supervision (w/ only positive pairing), via learning through natural and augmented face variations within each track. The finetuning is bootstrapped by soft-matching tracks across common identities. Performing these two steps alternatively helps the model better understand the given set of faces. Stage 3: An agglomerative clustering algorithm based on a model-learned similarity metric groups common identity tracks.

along with temporal constraints through face motion tracking. Unlike previous approaches, our method requires no explicit contextual information.

Video Face Clustering using temporal feature aggregation, 3D convolutions: Another line of work, such as in [22, 31], incorporates mechanisms to aggregate deep learning-based features of a given face track to provide a single track level embedding, which is in turn used to perform non-temporal face clustering. Recent approaches, such as [25], adopt 3D convolutions inside the feature extractor to model temporal identity information better. Our method utilizes temporal information in a more flexible way, thus allowing the use of any feature encoder architecture.

Temporal Track Constraints based Video Face Clustering: A large body of methods focuses on generating identity labels through the creation of positive image pairs. They track a given face across consecutive frames and negative pairs through co-occurring tracks. Approaches such as [8, 27, 57] apply such temporal constraints in semi/fully supervised settings, whereas methods like [1, 12, 13, 26, 42, 47, 48, 53, 54, 55, 56] use temporal constraints in an unsupervised manner, with the majority of them adopting some contrastive pair loss formulation. We significantly improve on this major trend by skipping negative pairs selection and, thus, any complex mining strategy for obtaining them.

Deep Metric Learning: Deep face clustering inherently relies on having embeddings of the same identity closer to each other and of different identities farther away in the representation space. Approaches such as [9, 29, 34, 45, 58, 60] focus on optimizing such a space and require defining a face similarity metric to improve video face clustering performance. Most approaches incorporate a contrastive-based similarity metric, such as triplet loss, to help obtain an embedding space optimized for a given set of faces in a video. We adopt the metric defined in Eq. (2) that does not rely on any negative pairing, thereby avoiding

any sub-optimality induced through incorrect negative pair selection. In this paper, we also utilize this metric for final video track clustering, which provides enhanced performance since the embeddings are optimized w.r.t. the metric itself.

Joint Representation and Clustering: [41] proposes two methods, TSiam and SSiam, to generate positive and negative pairs for model finetuning on a set of given video faces. TSiam adopts track-level constraints, i.e., sampling faces within the same and co-occurring tracks for positive and negative pairing, respectively. SSiam mines hard contrastive pairs using a pseudo-relevance feedback (pseudo-RF) inspired mechanism [57]. Both methods employ complex modules that depend on finetuned parameters to mine negative pairs. In contrast, our proposed method does not depend on negative pairs at all, making it much simpler and more generalized.

[61] incorporates a Markov Random Field (MRF) model to assign coarse track cluster labels, used as weak supervision for iteratively training a feature encoder. Negative pairs are mined through specific temporal constraints to boot start optimization of MRF. Unlike related prior works, we present a model adaptation stage that is highly effective due to the self-distillation procedure and its sole dependence on positive match pairs. Our final clustering algorithm directly benefits from the model finetuning stage by incorporating the self-supervised training objective as a distance metric. Such a learned metric helps boost clustering performance as it evaluates inter-track distances in an embedding space, which is explicitly optimized for reduction in observed same identity distances.

3 Method

3.1 Overview

A high-level overview of our proposed method is shown in Fig. 2. The following subsections describe in detail the prominent components of our approach.

3.2 Face Track Pre-Processing

In a standard frame rate video, frame content within the same scene gradually varies w.r.t. its temporal neighboring frames. To exploit this temporal stability for face clustering, we first locally cluster detected faces in a video by motion tracking, akin to all major prior works [9, 41, 55]. This pre-processing stage consists of four components.

First, scene cuts are detected in the given video, which divides it into contiguous separate sections, here coined as shots. Each scene cut represents a major change in scene composition, either involving a camera angle or scene setting change. We employ a threshold-based scene cut detection algorithm implemented in PySceneDetect library [3]. Second, we utilize a face detection algorithm to find all visible faces in each frame of the processed shots. We adopt RetinaFace [16] as our face detector due to its current state-of-the-art benchmark performance.

Third, the detected face crops are evaluated for their face ID quality by thresholding facial attributes based on blurriness and crop size. Crops failing the quality test are directly labeled as *Unknown*. Fourth, detected faces within a given shot are locally linked into a face track using a motion tracking algorithm. We adopt the state-of-the-art method, BoT-SORT [2], to generate tracks. For each face track t, a face crop I is sampled every 12-th frame, i.e., $t = [I_{t_1}, I_{t_2}, ..., I_{t_n}]$, where $t_n = 12 * n + f_1$ and f_1 denotes the original frame index for the first frame in the track's sampled set t. This particular frame interval assumes a video frame rate of 24 fps and ensures, in most cases, that there is a significant change in either facial pose and/or expressions through the track duration.

3.3 Task Objective Formulation

Following [22, 54], we consider a set of all detected tracks within a given video, which can be denoted by $T = \{t_j | j = 1, 2, ..., N\}$ for a set of N tracks. The face clustering objective can be defined as assigning an unique cluster id d for each track t_j, where all tracks with the same id belong to an unique facial identity in the set T. Note that the ground truth number of clusters is undefined. More formally,

$$t_j^d = \{-1, 1, 2, ...\},\ \forall j = \{1, 2, ..., N\}, \tag{1}$$

where $d = -1$ indicates the *Unknown* face cluster. This cluster represents tracks with the majority of their faces flagged as failures in any of the previously mentioned face attributes tests or if the face ID model was uncertain about it. The latter case is detailed in Sec. 3.6.

3.4 Self Supervised Model Finetuning

To adapt a large-scale pre-trained face ID model to a specific set of faces, we incorporate the notion of finetuning the model for that face track set. Traditional supervised finetuning would require human supervision, i.e. ground truth labels, which can be tedious depending on the number of tracks involved. To alleviate this problem, several approaches in the domain of self-supervised feature learning (SSL) have recently been proposed [7, 24, 35, 51]. Especially interesting are methods that only use positive pairs for contrastive-based learning [23, 65]. Inspired by [64], we adopt a self-distillation-based SSL method that uses a teacher-student mechanism and positive pairs.

First, we modify the technique to perform finetuning rather than training from scratch. As shown in Fig. 3, given the pre-trained face-ID model, which has no specific architecture limitations, we attach a randomly initialized multilayer perceptron (MLP) as a model head. For a Transformer model architecture [17], separate heads are attached for the class and patch token embeddings, respectively. The base model with the attached head(s) is duplicated to create a teacher branch, with the original one designated as the student branch.

Second, image pairs are curated from each track's sampled crop set. Images in a given pair are first passed through a set of augmentations, then an embedding pair is obtained as the output of the two constructed branches. We adopt a similarity loss to compare these embeddings, presented as follows:

$$L_{ssl} = -1 * softmax\left(\frac{embed_t - c}{temp}\right) * \log(softmax(embed_s)), \tag{2}$$

where $embed_t$ and $embed_s$ are the embeddings from teacher and student branches, respectively. Here, c denotes a rolling average teacher embedding computed across training batches, and $temp$ is a fixed softening temperature. Respective loss gradients are back-propagated through the student branch weights only, while the teacher branch weights are updated via a moving average of the student weights at regular training intervals. Since the model is finetuned on all face tracks that need to be clustered in a self-supervised fashion, the training and validation sets are identical[2].

As the branch heads are randomly initialized, each of the branch's base model is frozen for an initial training phase. First, the heads are updated separately, akin to the description above. Then, both the base model and the heads are updated. Such a structured training regime encourages the model branch heads to produce robust and consistent embeddings for a given facial identity across the observed range of poses, expressions, lighting and appearance changes, thus improving overall clustering performance for that specific video. Model finetuning hyperparameters such as training epochs, batch size, and learning rate[2] are kept constant and are agnostic to a given dataset's attributes.

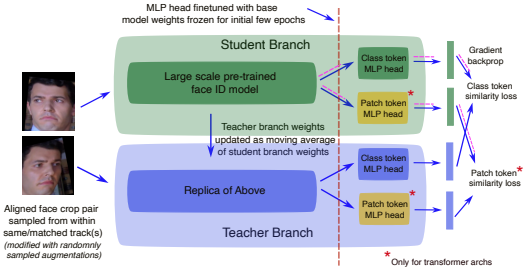


Fig. 3: Self-Supervised Model Finetuning: Face crop pair sampled from within same/matched track(s) is passed through a student and teacher branch, respectively. Gradients w.r.t. similarity loss are backpropagated only through the student branch, while the teacher weights are updated as moving average of student weights. Random augmentation set includes horizontal flipping, rotation, and color temperature variations.

Specific image augmentations[2] are applied during the finetuning process. This ensures that the model observes enhanced facial variations during its finetuning and helps reduce its primary dependence on natural variations present in the video. Tab. 2 provides empirical evidence in support of this fact wherein the track count ranges between 119 and 917 across our dataset movies. Our method manages to achieve significant performance improvements over benchmark methods across all movies despite these notable track count variations, evident in Tab. 3.

[2] Refer supplementary material for further details

3.5 Coarse Track Matching

Since a given face track is limited to being within a shot, there is no significant variation in lighting and/or face appearance across the track, which theoretically puts an upper bound on the model learning capacity.

However, for real-world scenarios such as those likely to be observed in movies and TV series, such parameters can vary greatly throughout the video. To account for such variations and facilitate further model learning, we perform fully automated coarse matching of tracks across the entire dataset. Image pairs generated from such coarse-matched tracks enable the model to better adapt to specific lighting and appearance variations encountered in a given face across the entire dataset. This notion is supported by our experimental findings in Sec. 4.

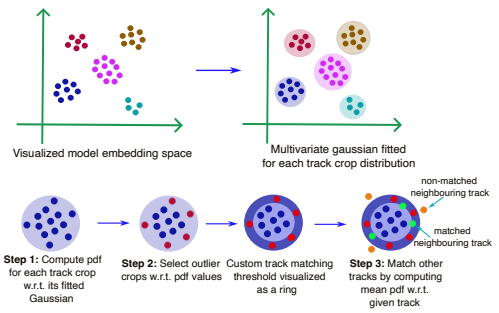


Fig. 4: Coarse Face Track Matching: A Multivariate Gaussian is fitted to every track crop distribution. Then, a custom track matching threshold is computed using outlier crops pdf values. Neighboring tracks having a mean pdf value higher than a custom threshold are soft-matched with the given track.

For coarse track matching, we leverage multiple sampled crops of the same identity in a given face track. To model this track crop distribution, we found empirically that fitting a multivariate normal distribution on all track crop embeddings works the best. Mathematically, we adopt

$$N_{t_j}(\mu_{t_j}, \Sigma_{t_j}) = (2\pi)^{-d/2} * \det(\Sigma_{t_j})^{-1/2} * \exp\left(\frac{-1}{2}(x-\mu_{t_j})^\top \Sigma_{t_j}^{-1}(x-\mu_{t_j})\right), \quad (3)$$

where $\mu_{t_j} \in \mathbb{R}^d$ and $\Sigma_{t_j} \in \mathbb{R}^{d\times d}$ are the mean and covariance matrix for the j^{th} face track, computed using all its sampled face crop embeddings. Here, d denotes the number of dimensions of the fitted distribution, which equals the dimension of track crop embeddings.

To automatically set a custom threshold value for matching a given face track to other neighboring tracks in the model learned embedding space, we resort to the probability density function (pdf) values of a given track's crop embeddings. Specifically, the pdf values of all crop embeddings are computed w.r.t. their parent track's fitted distribution using Eq. (3). We then consider the lowest 25% of these values and compute their mean. Experimental values ranging from 5% to 40% were considered, with value of 25% empirically providing optimal true positive matches and avoiding any false positive matches. This provides a customized matching threshold, which is illustrated as a ring around the track's distribution in Fig. 4. To get coarse matches for a given track, we compute a mean embedding for every other dataset track and its corresponding pdf value w.r.t.

the given track's fitted distribution. If a neighboring track's pdf value is equal to or higher than that given track's custom match threshold, then the track pair's distributions have significant overlap, hinting at a strong face identity match.

Further, given coarse matches for every dataset face track, we curate face crop pairs across these track matches for the next iteration of model finetuning. In particular, for each face crop in a given track, we randomly sample a track from a set of its coarsely matched tracks. The image pair is created by randomly sampling a crop from within the sampled track. We empirically found that this image pairing mechanism works better than other more complex strategies, such as thresholding inter track euclidean/cosine distances. If a track has no coarse matches, then we create pairs from within the same track.

Algorithm 1: Face Track Clustering

Input:
Filtered Face Tracks $T = \{t_j | j = 1, 2, ..., N\}$
$\ni\ t_j = \{I_{t_1}, I_{t_2}, ...I_{t_n} | t_n = 12 * n + f_1\}$,
finetuned model θ_{ft}, **Similarity metric** S
Stage 3.1 (Compute track crop embedding set):
for t_j *in* T **do**
  for I_{t_n} *in* t_j **do**
    $E_{t_n} \longleftarrow \theta_{ft}(I_{t_n})$
  end
  $t_{jE} = \{E_{t_1}, E_{t_2}, ..., E_{t_n}\}$
end
$T_E = \{t_{1E}, t_{2E}, ...t_{NE}\}$
Stage 3.2 (Compute custom track threshold):
for t_{jE} *in* T_E **do**
  $Sim_{t_{jE}} = \{S(E_{t_l}, E_{t_m}) \ \forall (l, m) \in {}^nC_2\}$
  $Thres_{t_j} \longleftarrow mean(Sim_{t_{jE}})$
end
$T_{thres} = \{Thres_{t_1}, ..., Thres_{t_N}\}$
Stage 3.3 (Perform track clustering):
Initialize $i = 0, C = \{C_j = \{t_{jE}\} \forall j = 1, 2, ..., N\}$
repeat
  for C_j *in* C **do**
    for C_k *in* C *if* $k \neq j$ **do**
      $Sim_{jk} \longleftarrow mean(\{S(t_{jE_a}, t_{kE_b}) \ \forall (a, b) \in {}^nC_2\})$
      if $Sim_{jk} < Thres_{t_j}/Thres_{t_k}$ **then**
        $C_j \longleftarrow merge(C_j, C_k)$
      end
    end
  end
  $C = link_merges(C)$
  $N_{C_i} = cluster_count(C)$
  Repeat Stage 2 for new merged cluster set C
  $i = i + 1$
until $N_{C_i} - N_{C_{i-1}} = 0$;
Output: Clustered track IDs C

3.6 Track Face Quality Estimation

In complex face identification scenarios, excluding bad quality crops/tracks becomes essential for coarse track matching and final clustering. Bad face quality of a given track often relates to model uncertainty, which can result in sub-optimal performance for coarse matching and final clustering phases. To automatically estimate the face quality of track crops, we adopt SER-FIQ [50], which utilizes a dropout layer to determine consistency in embeddings across multiple model iterations. For bad quality crops, the learned model that is uncertain about them would predict embeddings with high variance, thus resulting in a low-quality score.

To compute the face quality score for a given track (tqs), we adapt SER-FIQ to work on a track level by obtaining scores for each track crop and averaging them. To detect bad quality tracks, we adopt the median absolute deviation (MAD) [30] to detect outlier tracks based on dataset track score distribution and compute a threshold value[2]. Low quality score tracks are filtered out from coarse track matching and final clustering modules and their final track cluster IDs are assigned as *Unknown*.

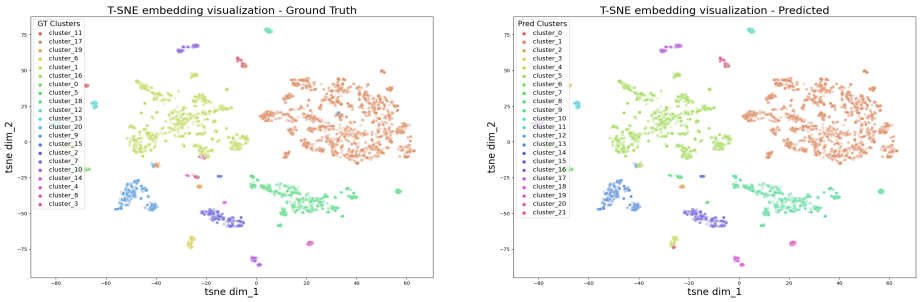


Fig. 5: Comparative t-SNE embedding visualizations [33] on **MovieFaceCluster:The Hidden Soldier** dataset. *Left:* Ground Truth (GT), *Right:* Our method. Each dot in the diagram above represents the finetuned model's extracted embedding for a given face crop I_{t_n} in a given track's sampled crop set t. Face embeddings assigned to a given color constitute a single cluster. Our method predicts almost perfectly the cluster designations (22 clusters) w.r.t. ground truth (21 clusters). Also note that our method correctly assigns cluster IDs to certain outlier tracks in GT. Such tracks pose a significant face clustering challenge owing to their distance from respective GT cluster centers.

3.7 Track Clustering Algorithm

To cluster tracks across common identities, we utilize the SSL loss function in Eq. (2) as an embedding similarity metric. Prior works incorporate Euclidean or other pre-defined distance metrics to compare model embeddings [41, 47]. Our proposed metric has a significant benefit. A finetuned model's embedding space is directly optimized w.r.t. this metric, thus making it optimal for evaluating embedding similarity. As such, there are no implicit assumptions made about the space through generic distance functions. Unlike other methods that define a global matching threshold [14], our approach provides enhanced performance through the adaptive custom threshold computed for each track. Such a thresholding mechanism helps the algorithm automatically adjust to how effectively the model can match a given track's identity across a true positive match pair.

Our proposed clustering methodology is detailed in Algorithm 1. We begin by creating all possible pair combinations among sampled face crops within a given track. Here, we exclude tracks filtered out by the track face quality estimation module described in Sec. 3.6. A pair's similarity value is computed via the loss metric by passing the respective face crops through each of the model branches. As the loss metric is not commutative, a mean value is computed by alternatively sending both images through each of the branches. A custom track matching threshold is set as the average of all curated pair similarity values in the given track. This threshold represents quantitatively how well the model matches face crops belonging to a given common facial identity since the crops are part of the same track. We repeat this step for all tracks in the dataset.

The next stage comprises merging tracks in a bottom-up approach, akin to Hierarchical Agglomerative Clustering (HAC) [14]. Initially, each track is assigned an individual cluster, and tracks are iteratively merged if they satisfy a matching criterion. It involves creating all pairwise combinations of face crops across both tracks for a given candidate track pair. Given the similarity (loss)

value for every crop pair combination, taking a mean across them provides a matching potential value for that candidate track pair. If this match potential is lower than either of the track's custom match thresholds, then the track pair is considered a positive match. This process is repeated for all possible track combinations in the dataset. All positively matched pairs are searched for common tracks so that they can be combined together into a bigger cluster. For example, if track pairs (1,2) and (2,3) are matched, then tracks 1, 2, and 3 are combined together. This entire matching process is run in an iterative fashion. For later iterations, where clusters could have more than one track, mean track embeddings are considered instead of a combined set of face crops for cluster pair matching, to avoid exponentially growing match computations. The algorithm terminates when no new clusters are merged in the new iteration.

4 Results

In this section, we present our experimental analysis on popular benchmark datasets and our curated movie dataset.

4.1 Benchmark Datasets

Following prior work [4, 41, 47], we evaluate our proposed method on TV series episodes of Big Bang Theory (BBT) and Buffy The Vampire Slayer (BVS), specifically the first six episodes of BBT season 1 and BVS season 5, respectively. BBT is a TV series with a primarily indoor setting, a cast of 5∼8 different characters, and 625 average face tracks per episode. Here, all shots include wide-angle scenes, and faces are relatively small. The most common face ID challenges are pose and lighting variations. BVS poses different challenges. The main cast comprises 12∼18 characters, and there are 919 average face tracks per episode. Shots are mainly captured outdoors, and scenes are dark. It also has more close-up shots and, thus, larger face sizes. Detailed statistics on these datasets can be found in Tab. 1 in [4]. Tab. 1 compares our method with state-of-the-art methods on BBT and BVS, respectively.

Method	BBT S01 Episode						
	S1E1	S1E2	S1E3	S1E4	S1E5	S1E6	Combined
SCTL [54]	66.48	-	-	-	-	-	-
TSiam [41]	96.4	-	-	-	-	-	-
SSiam [41]	96.2	-	-	-	-	-	-
MLR [4]	95.18	94.16	77.81	79.35	79.93	75.85	83.71
BCL [47]	98.63	98.54	90.61	86.95	89.12	81.07	89.63
CCL [42]	98.2	-	-	-	-	-	-
VCTRSF [53]	99.39	**99.84**	97.58	96.41	98.47	93.33	94.20
Ours⋆†	**99.70**	99.67	**98.60**	**98.80**	**99.10**	**97.10**	**98.70**

Method	BVS S05 Episode						
	S5E1	S5E2	S5E3	S5E4	S5E5	S5E6	Combined
HMRF [55]	-	50.3	-	-	-	-	-
WBSLRR [56]	-	62.7	-	-	-	-	-
TSiam [41]	-	92.46	-	-	-	-	-
SSiam [41]	-	90.87	-	-	-	-	-
CP-SSC [44]	-	65.2	-	-	-	-	-
MvCorr [43]	-	97.7	-	-	-	-	-
MLR [4]	71.99	61.27	66.60	67.07	69.59	61.72	66.37
BCL [47]	92.08	79.76	84.00	84.97	89.05	80.58	83.62
CCL [42]	-	92.1	-	-	-	-	-
Ours⋆†	**96.30**	**99.10**	**98.70**	**97.43**	**99.00**	**96.78**	**96.10**

Table 1: WCP/Clustering Accuracy on BBT-S01 and BVS-S05. ⋆We use ArcFace-R100 [15] as our pre-trained base model. Combined results indicate clustering performance on set of face tracks from all six episodes combined together. † For fair literature comparison, we use the same face detection, tracking, and clustering labels as provided in [41, 47], thereby not utilizing our proposed advanced pre-processing modules in order to effectively compare pure track clustering performance against literature methods.

Dataset	TV series Episode/Movie	Attribute			
		Unique characters	**Track Count**	**Unique ethnicity** (other than White/Caucasian) (cast percentage)	**Avg. Track Face Quality Score** †
The Big Bang Theory (BBT)	S01E01	6	647	A (Minor)	0.7193
	S01E02	5	613	A (Minor)	0.7108
	S01E03	7	562	A (Minor)	0.7094
	S01E04	8	568	A (Minor)	0.7140
	S01E05	6	463	A (Minor)	0.7177
	S01E06	6	651	A (Minor)	0.7111
		Average: 6.33	Total: 3504	Unique Count: 1	Average: 0.714
Buffy The Vampire Slayer (BVS)	S05E01	12	786	None	0.7090
	S05E02	13	866	None	0.7117
	S05E03	14	1185	None	0.7150
	S05E04	15	852	None	0.7125
	S05E05	15	733	None	0.7081
	S05E06	18	1055	None	0.7142
		Average: 14.5	Total: **5477**	Unique Count: 0	Average: 0.712
MovieFaceCluster	An Elephant's Journey	18	562	None	0.7112
	Armed Response	14	119	AA (Major), ME (Major)	0.7085
	Angel Of The Skies	29	319	None	0.7150
	Death Do Us Part (2019)	7	395	AA (Major)	0.7177
	American Fright Fest	37	457	AA (Minor)	0.7098
	The Fortress	64	917	A (Major)	0.6918
	Under The Shadow	9	143	ME (Major)	0.7134
	The Hidden Soldier	21	594	A (Major)	0.7056
	S.M.A.R.T. Chase	10	113	A (Major)	0.7110
		Average: **23.2**	Total: 3619 ††	Unique Count: **3**	Average: **0.706**

Table 2: Specific dataset attribute comparisons across BBT, BVS and our MovieFaceCluster dataset. A: Asian, AA: African American, ME: Middle Eastern characters. † Score is computed as the average of all track quality scores as part of a given TV series episode/movie. A given track quality score is computed as the average of face quality scores for each of its sampled crops, using SER-FIQ [50] and ArcFace-R100 [15] as the pre-trained model for extracting embeddings. Our lower average face quality score indicates that MovieFaceCluster contains on average more challenging cases for face clustering compared to literature datasets. †† Only tracks containing decent quality face crops were added as part of each movie dataset, resulting in a slightly lower track count. Please refer to the supplementary material for further dataset analysis and comparisons.

4.2 Metrics

We define two primary metrics at face track level for evaluating video face clustering performance, namely Weighted Cluster Purity/Accuracy (WCP) and Predicted Cluster Ratio (PCR). WCP is defined as the fraction of common identity tracks in a predicted cluster, weighted by the cluster track count. PCR is the ratio between the predicted cluster and ground truth cluster count. Note that a ratio closer to 1 is deemed better.

4.3 Movie Dataset

Mainstream movies present challenges for face clustering due to extreme pose, illumination, and appearance variations. Considering the lack of significant benchmark datasets in the academic research community, we present a new movie benchmark dataset named **MovieFaceCluster** [3], containing a collection of movies, hand-selected by film post-production specialists, with unique face clustering challenges [2]. Given the set of movies, we run the preprocessing mentioned in Sec. 3.2 to obtain a specific track dataset for each movie. We hand-label each track with an ID using the main character cast from that track's parent

[3] The MovieFaceCluster dataset can be downloaded from here

movie. Also, false detections and extreme unidentifiable tracks are pre-filtered out using the track face quality estimation module mentioned in Sec. 3.6. Tab. 2 provides detailed statistics and comparisons of this curated dataset versus literature datasets. Overall, our presented dataset provides more unique characters, comparable track count, higher ethnic diversity and more challenging face tracks compared to publicly available datasets.

Tab. 3 also compares our method to other state-of-the-art (SoTA) approaches on this benchmark to provide empirical evidence of our algorithm's effectiveness. We implemented all five literature methods due to the lack of their public code implementations [2]. Fig. 5 shows a t-SNE visualization of our learned embeddings and clustering performance on one dataset movie, and Fig. 1 illustrates some hard case clustered tracks using our proposed method on MovieFaceCluster.

Method	**Movie**								
	An Elephant's Journey (2019)	Armed Response	Angel Of The Skies	Death Do Us Part (2019)	American Fright Fest	The Fortress	Under The Shadow	The Hidden Soldier	S.M.A.R.T. Chase
	Weighted Cluster Accuracy (%) & Pred Cluster Ratio (Pred / GT)								
TSiam [41]	90.7 & 1.44	84.9 & 1.36	77.1 & 0.62	92.9 & 1.57	89.3 & 0.83	68.6 & 0.69	71.8 & 2.11	90.7 & 1.33	79.6 & 1.70
SSiam [41]	88.1 & 1.61	86.6 & 1.21	75.5 & 0.59	94.4 & 1.28	86.2 & 0.78	71.1 & 0.73	68.3 & 2.33	88.7 & 1.24	82.3 & 1.80
JFRAC [61]	91.4 & 1.33	85.2 & 1.50	73.4 & 0.62	90.8 & 0.71	91.5 & 0.86	65.3 & 0.77	73.1 & 2.00	92.6 & 1.19	85.8 & 1.70
CCL [42]	89.5 & N.A.†	89.7 & N.A.†	75.0 & N.A.†	95.4 & N.A.†	87.2 & N.A.†	62.7 & N.A.†	77.4 & N.A.†	84.0 & N.A.†	89.9 & N.A.†
VCTRSF [53]	96.3 & N.A.†	92.2 & N.A.†	77.7 & N.A.†	96.5 & N.A.†	91.3 & N.A.†	78.8 & N.A.†	78.7 & N.A.†	94.4 & N.A.†	88.4 & N.A.†
Ours	**97.2 & 1.11**	**94.1 & 0.93**	**85.9 & 0.72**	**98.0 & 1.14**	**97.6 & 0.92**	**89.3 & 1.02**	**82.5 & 1.88**	**98.5 & 1.04**	**93.8 & 1.50**

Table 3: Quantitative comparisons on each MovieFaceCluster dataset movie. For a fair comparison, we incorporate ArcFace-R100 [15] as the pre-trained feature extractor for all reported methods, including ours. We outperform SoTA methods w.r.t. cluster accuracy and predicted cluster ratio. Details on our implementation of all comparative methods can be found in the supplementary material. †Number of ground truth clusters is required as input for these methods, so PCR isn't a valid performance metric while also being a major limitation for these methods.

5 Ablative Analysis

We ablate the central components of our method and analyze limitations and future directions.

Model Finetuning We ablate on the effectiveness of generic face ID model finetuning to a given set of face tracks as part of our proposed method. Tab. 4 provides a comparison of clustering performance w/ and w/o using the model finetuning module. Note that our proposed clustering algorithm depends on the similarity metric learned during the finetuning stage. As such, to compare both methods in a fair way, we adopt a baseline clustering algorithm, i.e., HAC with average linkage and cosine distance metric. Performing model finetuning results in roughly 6% increase in cluster accuracy, which underlines its usefulness.

Final Clustering Algorithm We further ablate on the performance of our final clustering approach vs. baseline algorithm, i.e., HAC in Tab. 5. Here, we keep the model finetuning stage constant in all methods to compare fairly. As for HAC, we take the mean of a given track's sampled crop embeddings to obtain a representative track embedding. We further ablate on the loss function as a similarity metric. Specifically, we compare our final clustering algorithm with one that uses Euclidean and Cosine distances as similarity metrics. Our approach involving the loss metric outperforms all other methods.

Generic face ID Model Architectures In Tab. 6, we ablate on our approach's generalization capabilities to incorporate any generic face ID model, fairly agnostic to its architecture class. Specifically, we compare some prominent face ID models from both CNN and Transformer architecture classes incorporated as part of our method, against using them (w/o finetuning) along with the baseline clustering method (HAC). Regardless of the incorporated face ID model, our finetuning method provides roughly a $5 \sim 12\%$ performance boost, which underlines our method's capability to adapt to and improve any generic face ID model.

Limitations and Future Work The use of a generic face ID model means that any pre-existing model biases may also be propagated through our method. For example, if the generic model has learned an incorrect similarity between two distinct facial identities, then our algorithm might adapt to it and provide a false positive cluster for that given pair. A future direction could be to automatically detect such biases, such that a given pair's embeddings are specifically pulled apart. This could be done by incorporating an outlier detection technique based on pair similarity values for a cluster's tracks. Also, given that we finetune on a set of face tracks, it might not be optimal for real-time applications depending on the track set size.

Method	Cluster Accuracy (%)
Baseline-(Non-Finetuned)	86.10
Ours-(Finetuned)	**91.52**

Table 4: Ablation for model finetuning module, using ArcFace-R100 as the base model. Experiments are performed on **MovieFaceCluster: The Hidden Soldier** dataset and HAC as the final clustering algorithm.

Clustering Algorithm	Similarity Metric	Cluster Accuracy (%)	Cluster Ratio (Pred/GT)
Baseline (HAC)	Cosine	91.52	1.43 (30/21)
Ours	Cosine	93.70	2.0 (42/21)
Ours	Euclidean	96.50	3.5 (74/21)
Ours	**Loss Func.**	**98.50**	**1.04 (22/21)**

Table 5: Ablation for final clustering algorithm, compared with baseline HAC, and using predefined metrics within our algorithm. Experiments are performed on **MovieFaceCluster: The Hidden Soldier** dataset and using ArcFace-R100 as the base model.

Face ID Model	Cluster Acc. (%)	
	Baseline	Ours
FaRL-P16 [62]	78.7	**90.2**
VGGFace2-R50 [6]	84.2	**95.7**
ArcFace-R100 [15]	86.1	**98.5**
AdaFace-R100 [28]	86.9	**98.4**

Table 6: Ablation for various base face ID models incorporated in our method. We perform comparisons using our proposed method and pre-trained model + baseline clustering (HAC). Experiments are performed on **MovieFaceCluster: The Hidden Soldier** dataset.

6 Conclusion

We present a novel video face clustering algorithm that specifically adapts to a given set of face tracks through a fully self-supervised mechanism. This helps the model better understand and adapt to all observed variations for a given facial identity across the entire video without any human-in-the-loop label guidance. Our fully automated approach to video face clustering specifically helps avoid any sub-optimal solutions that may be induced from non-intuitive user-defined parameters. In addition, using a model-learned similarity metric over generic distance functions helps provide SoTA video face clustering performance over other competing methods. Extensive experiments and ablation studies on our presented comprehensive movie dataset and traditional benchmarks underline our method's effectiveness under extremely challenging real-world scenarios.

Bibliography

[1] Aggarwal, A., Pandya, Y., Ravindranathan, L.A., Ahire, L.S., Sethu, M., Nandy, K.: Robust actor recognition in entertainment multimedia at scale. ACM International Conference on Multimedia (ACM MM) pp. 2079–2087 (2022) 4

[2] Aharon, N., Orfaig, R., Bobrovsky, B.: Bot-sort: Robust associations multi-pedestrian tracking. arXiv preprint arXiv:2206.14651 (2022) 6

[3] API, P.: (2022), `https://www.scenedetect.com/api/`, version 0.6.0.3 5

[4] Bäuml, M., Tapaswi, M., Stiefelhagen, R.: Semi-supervised learning with constraints for person identification in multimedia data. In: Conference on Computer Vision and Pattern Recognition (CVPR). pp. 3602–3609. IEEE (2013) 11

[5] Berg, T.L., Berg, A.C., Edwards, J., Maire, M., White, R., Teh, Y.W., Learned-Miller, E.G., Forsyth, D.A.: Names and faces in the news. In: Conference on Computer Vision and Pattern Recognition (CVPR). pp. 848–854. IEEE (2004) 3

[6] Cao, Q., Shen, L., Xie, W., Parkhi, O.M., Zisserman, A.: Vggface2: A dataset for recognising faces across pose and age. In: International Conference on Automatic Face & Gesture Recognition. pp. 67–74. IEEE (2018) 14

[7] Chen, T., Kornblith, S., Norouzi, M., Hinton, G.E.: A simple framework for contrastive learning of visual representations. In: International Conference on Machine Learning (ICML). pp. 1597–1607. PMLR (2020) 6

[8] Cherniavsky, N., Laptev, I., Sivic, J., Zisserman, A.: Semi-supervised learning of facial attributes in video. In: Trends and Topics in Computer Vision: ECCV 2010 Workshops. pp. 43–56. Springer (2010) 4

[9] Cinbis, R.G., Verbeek, J., Schmid, C.: Unsupervised metric learning for face identification in tv video. In: International Conference on Computer Vision (ICCV). pp. 1559–1566. IEEE (2011) 4, 5

[10] Comaniciu, D., Meer, P.: Mean shift: A robust approach toward feature space analysis. IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI) **24**(5), 603–619 (2002) 2

[11] Cour, T., Sapp, B., Nagle, A., Taskar, B.: Talking pictures: Temporal grouping and dialog-supervised person recognition. In: Conference on Computer Vision and Pattern Recognition (CVPR). pp. 1014–1021. IEEE (2010) 3

[12] Dahake, R., et al.: Face recognition from video using threshold based clustering. Turkish Journal of Computer and Mathematics Education (TURCOMAT) **12**(1S), 272–285 (2021) 4

[13] Datta, S., Sharma, G., Jawahar, C.V.: Unsupervised learning of face representations. In: International Conference on Automatic Face & Gesture Recognition. pp. 135–142. IEEE (2018) 4

[14] Defays, D.: An efficient algorithm for a complete link method. The Computer Journal **20**(4), 364–366 (1977) 2, 10

[15] Deng, J., Guo, J., Xue, N., Zafeiriou, S.: Arcface: Additive angular margin loss for deep face recognition. In: Conference on Computer Vision and Pattern Recognition (CVPR). pp. 4690–4699. IEEE (2019) 11, 12, 13, 14

[16] Deng, J., Guo, J., Zhou, Y., Yu, J., Kotsia, I., Zafeiriou, S.: Retinaface: Single-shot multi-level face localisation in the wild. In: Conference on Computer Vision and Pattern Recognition (CVPR). pp. 5203–5212. IEEE (2020) 5

[17] Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., Uszkoreit, J., Houlsby, N.: An image is worth 16x16 words: Transformers for image recognition at scale. In: International Conference on Learning Representations (ICLR). Springer (2021) 6

[18] El Khoury, E., Sénac, C., Joly, P.: Face-and-clothing based people clustering in video content. In: Proceedings of the International Conference on Multimedia Information Retrieval. pp. 295–304. ACM (2010) 3

[19] Ester, M., Kriegel, H., Sander, J., Xu, X.: A density-based algorithm for discovering clusters in large spatial databases with noise. In: Knowledge Discovery and Data Mining. pp. 226–231. AAAI Press (1996) 2

[20] Everingham, M., Sivic, J., Zisserman, A.: Hello! my name is... buffy" – automatic naming of characters in TV video. In: British Machine Vision Conference (BMVC). pp. 899–908 (2006) 3

[21] Everingham, M., Sivic, J., Zisserman, A.: Taking the bite out of automated naming of characters in tv video. Image and Vision Computing **27**(5), 545–559 (2009) 3

[22] Gong, S., Shi, Y., Kalka, N.D., Jain, A.K.: Video face recognition: Component-wise feature aggregation network (c-fan). In: International Conference on Biometrics. pp. 1–8. IEEE (2019) 4, 6

[23] Grill, J., Strub, F., Altché, F., Tallec, C., Richemond, P.H., Buchatskaya, E., Doersch, C., Pires, B.Á., Guo, Z., Azar, M.G., Piot, B., Kavukcuoglu, K., Munos, R., Valko, M.: Bootstrap your own latent - A new approach to self-supervised learning. In: Advances in Neural Information Processing Systems (NeurIPS) (2020) 6

[24] He, K., Fan, H., Wu, Y., Xie, S., Girshick, R.B.: Momentum contrast for unsupervised visual representation learning. In: Conference on Computer Vision and Pattern Recognition (CVPR). pp. 9726–9735. IEEE (2020) 6

[25] Huo, J., van Zyl, T.L.: Unique faces recognition in videos. In: International Conference on Information Fusion (FUSION). pp. 1–7. IEEE (2020) 4

[26] Kalogeiton, V., Zisserman, A.: Constrained video face clustering using 1nn relations. In: British Machine Vision Conference (BMVC) (2020) 4

[27] Kapoor, A., Hua, G., Akbarzadeh, A., Baker, S.: Which faces to tag: Adding prior constraints into active learning. In: International Conference on Computer Vision (ICCV). pp. 1058–1065. IEEE (2009) 4

[28] Kim, M., Jain, A.K., Liu, X.: Adaface: Quality adaptive margin for face recognition. In: Conference on Computer Vision and Pattern Recognition (CVPR). pp. 18750–18759. IEEE (2022) 14

[29] Law, M.T., Urtasun, R., Zemel, R.S.: Deep spectral clustering learning. In: International Conference on Machine Learning (ICML). vol. 70, pp. 1985–1994. PMLR (2017) 4
[30] Leys, C., Ley, C., Klein, O., Bernard, P., Licata, L.: Detecting outliers: Do not use standard deviation around the mean, use absolute deviation around the median. Elsevier Journal of Experimental Social Psychology **49**(4), 764–766 (2013) 9
[31] Liu, Z., Hu, H., Bai, J., Li, S., Lian, S.: Feature aggregation network for video face recognition. In: International Conference on Computer Vision (ICCV). pp. 990–998. IEEE (2019) 4
[32] Lloyd, S.P.: Least squares quantization in pcm. In IEEE Transactions on Information Theory **28**(2), 129–136 (1982) 2
[33] der Maaten, L.V., Hinton, G.: Visualizing data using t-sne. Journal of Machine Learning Research **9**(11) (2008) 10
[34] Mensink, T., Verbeek, J., Perronnin, F., Csurka, G.: Metric learning for large scale image classification: Generalizing to new classes at near-zero cost. In: European Conference on Computer Vision (ECCV). vol. 7573, pp. 488–501. Springer (2012) 4
[35] van den Oord, A., Li, Y., Vinyals, O.: Representation learning with contrastive predictive coding. arXiv preprint arXiv:1807.03748 (2018) 6
[36] Ozkan, D., Duygulu, P.: A graph based approach for naming faces in news photos. In: Conference on Computer Vision and Pattern Recognition (CVPR). pp. 1477–1482. IEEE (2006) 3
[37] Paul, G., Elie, K., Sylvain, M., Jean-Marc, O., Paul, D.: A conditional random field approach for audio-visual people diarization. In: International Conference on Acoustics,Speech and Signal Processing (ICASSP). pp. 116–120. IEEE (2014) 3
[38] Pham, P.T., Moens, M.F., Tuytelaars, T.: Cross-media alignment of names and faces. IEEE Transactions on Multimedia (TMM) **12**(1), 13–27 (2009) 1, 3
[39] Rohrbach, A., Torabi, A., Rohrbach, M., Tandon, N., Pal, C.J., Larochelle, H., Courville, A.C., Schiele, B.: Movie description. International Journal on Computer Vision (IJCV) **123**(1), 94–120 (2017) 1
[40] Satoh, S., Nakamura, Y., Kanade, T.: Name-it: Naming and detecting faces in news videos. IEEE Multimedia **6**(1), 22–35 (1999) 1, 3
[41] Sharma, V., Tapaswi, M., Sarfraz, M.S., Stiefelhagen, R.: Self-supervised learning of face representations for video face clustering. In: International Conference on Automatic Face & Gesture Recognition. pp. 1–8. IEEE (2019) 1, 5, 10, 11, 13
[42] Sharma, V., Tapaswi, M., Sarfraz, M.S., Stiefelhagen, R.: Clustering based contrastive learning for improving face representations. In: IEEE International Conference on Automatic Face and Gesture Recognition. pp. 109–116. IEEE (2020) 4, 11, 13
[43] Somandepalli, K., Hebbar, R., Narayanan, S.: Robust character labeling in movie videos: Data resources and self-supervised feature adaptation. IEEE Transactions on Multimedia **24**, 3355–3368 (2021) 11

[44] Somandepalli, K., Narayanan, S.S.: Reinforcing self-expressive representation with constraint propagation for face clustering in movies. In: International Conference on Acoustics,Speech and Signal Processing (ICASSP). pp. 4065–4069. IEEE (2019) 11
[45] Song, H.O., Jegelka, S., Rathod, V., Murphy, K.: Deep metric learning via facility location. In: Conference on Computer Vision and Pattern Recognition (CVPR). pp. 5382–5390. IEEE (2017) 4
[46] Tang, Z., Zhang, Y., Li, Z., Lu, H.: Face clustering in videos with proportion prior. In: International Joint Conference on Artificial Intelligence (IJCAI). pp. 2191–2197. Annual AAAI Conference on Artifical Intelligence (AAAI) (2015) 3
[47] Tapaswi, M., Law, M.T., Fidler, S.: Video face clustering with unknown number of clusters. In: International Conference on Computer Vision (ICCV). pp. 5027–5036. IEEE (2019) 1, 4, 10, 11
[48] Tapaswi, M., Parkhi, O.M., Rahtu, E., Sommerlade, E., Stiefelhagen, R., Zisserman, A.: Total cluster: A person agnostic clustering method for broadcast videos. In: Indian Conference on Computer Vision Graphics and Image Processing. pp. 7:1–7:8. ACM (2014) 4
[49] Tapaswi, M., Zhu, Y., Stiefelhagen, R., Torralba, A., Urtasun, R., Fidler, S.: Movieqa: Understanding stories in movies through question-answering. In: Conference on Computer Vision and Pattern Recognition (CVPR). pp. 4631–4640. IEEE (2016) 1
[50] Terhörst, P., Kolf, J.N., Damer, N., Kirchbuchner, F., Kuijper, A.: SER-FIQ: unsupervised estimation of face image quality based on stochastic embedding robustness. In: Conference on Computer Vision and Pattern Recognition (CVPR). pp. 5650–5659. IEEE (2020) 9, 12
[51] Tian, Y., Krishnan, D., Isola, P.: Contrastive multiview coding. In: European Conference on Computer Vision (ECCV). pp. 776–794. Springer (2020) 6
[52] Vicol, P., Tapaswi, M., Castrejón, L., Fidler, S.: Moviegraphs: Towards understanding human-centric situations from videos. In: Conference on Computer Vision and Pattern Recognition (CVPR). pp. 8581–8590. IEEE (2018) 1
[53] Wang, Y., Dong, M., Shen, J., Luo, Y., Lin, Y., Ma, P., Petridis, S., Pantic, M.: Self-supervised video-centralised transformer for video face clustering. IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI) **45**(11), 12944–12959 (2023) 4, 11, 13
[54] Wu, B., Lyu, S., Hu, B., Ji, Q.: Simultaneous clustering and tracklet linking for multi-face tracking in videos. In: International Conference on Computer Vision (ICCV). pp. 2856–2863. IEEE (2013) 4, 6, 11
[55] Wu, B., Zhang, Y., Hu, B., Ji, Q.: Constrained clustering and its application to face clustering in videos. In: Conference on Computer Vision and Pattern Recognition (CVPR). pp. 3507–3514. IEEE (2013) 1, 4, 5, 11
[56] Xiao, S., Tan, M., Xu, D.: Weighted block-sparse low rank representation for face clustering in videos. In: European Conference on Computer Vision (ECCV). pp. 123–138. Springer (2014) 4, 11

[57] Yan, R., Zhang, J., Yang, J., Hauptmann, A.G.: A discriminative learning framework with pairwise constraints for video object classification. IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI) **28**(4), 578–593 (2006) 4, 5

[58] Yang, J., Parikh, D., Batra, D.: Joint unsupervised learning of deep representations and image clusters. In: Conference on Computer Vision and Pattern Recognition (CVPR). pp. 5147–5156. IEEE (2016) 4

[59] Zhang, L., Kalashnikov, D.V., Mehrotra, S.: A unified framework for context assisted face clustering. In: Proceedings of the 3rd ACM conference on International conference on multimedia retrieval. pp. 9–16. ACM (2013) 3

[60] Zhang, S., Gong, Y., Wang, J.: Deep metric learning with improved triplet loss for face clustering in videos. In: Advances in Multimedia Information Processing. vol. 9916, pp. 497–508. Springer (2016) 4

[61] Zhang, Z., Luo, P., Loy, C.C., Tang, X.: Joint face representation adaptation and clustering in videos. In: European Conference on Computer Vision (ECCV). pp. 236–251. Springer (2016) 5, 13

[62] Zheng, Y., Yang, H., Zhang, T., Bao, J., Chen, D., Huang, Y., Yuan, L., Chen, D., Zeng, M., Wen, F.: General facial representation learning in a visual-linguistic manner. In: Conference on Computer Vision and Pattern Recognition (CVPR). pp. 18697–18709. IEEE (2022) 14

[63] Zhou, C., Zhang, C., Fu, H., Wang, R., Cao, X.: Multi-cue augmented face clustering. In: ACM International Conference on Multimedia (ACM MM). pp. 1095–1098. ACM (2015) 1, 3

[64] Zhou, J., Wei, C., Wang, H., Shen, W., Xie, C., Yuille, A.L., Kong, T.: Image bert pre-training with online tokenizer. In: International Conference on Learning Representations (ICLR). Springer (2022) 6

[65] Zhou, P., Zhou, Y., Si, C., Yu, W., Ng, T.K., Yan, S.: Mugs: A multi-granular self-supervised learning framework. arXiv preprint arXiv:2203.14415 (2022) 6

[66] Zhou, S., Xu, H., Zheng, Z., Chen, J., Li, Z., Bu, J., Wu, J., Wang, X., Zhu, W., Ester, M.: A comprehensive survey on deep clustering: Taxonomy, challenges, and future directions. arXiv preprint arXiv:2206.07579 (2022) 2, 3