VideoClusterNet: Self-Supervised and Adaptive Face Clustering for Videos

Supplementary Material

Devesh Walawalkar and Pablo Garrido

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

1 Implementation Details

1.1 Model Architecture

As part of our algorithm, any CNN or Transformer architecture can be incorporated as the base Face ID model. For the attached MLP head, we incorporate a 3-layer network with GELU activation [4]. The length of the head's output feature embedding is kept the same as the length of the base model's output embedding. Unnormalized model output embeddings are used for both training and the final clustering stage. Also, teacher branch head weights are initialized separately from their student counterpart. When using a transformer class base model, the head is shared between the class and path token embeddings in both student and teacher branches, respectively (see Fig. 3 in the main paper).

1.2 Data Pre-Processing

Generation of face tracks as part of given video dataset curation for finetuning and/or clustering purposes involves processing the video through the four-stage pipeline, as mentioned in the main paper Sec. 3.2. To further augment variations in a sampled face crop pair, 'global' and 'local' views are created from the original crops. Specifically, crops are randomly cropped and resized with a scale between 0.7 and 1.0 to create multiple global crops, while a scale between 0.4 and 0.6 provides local crop variants. Given a total of 6 views for each non-cropped original pair, we set the global and local crop count to 2 and 4, respectively. The loss for a given original pair is computed as the sum of contrastive losses for each possible pair of constituent global and local crops. We further apply horizontal flipping and color temperature variations as additional augmentations.

Face alignment is performed for each face crop prior to the model finetuning and final clustering stages to further enhance model learning of facial features and, in turn, for better face clustering performance. More specifically, a given face crop is first resized to the pre-trained model's expected input image size. Then, face landmarks (five points) are predicted for that crop (we incorporate

the landmarks provided as an auxiliary output by the face detection model RetinaFace [2]). Given the facial landmarks, a similarity transform is computed w.r.t. a mean landmark template. An affine warp is then computed to align the face within the face crop.

1.3 Model Finetuning

For the first finetuning iteration, we train the model for 30 epochs. Note that within this iteration, the first 10 epochs are dedicated to isolated finetuning of branch heads since they contain randomly initialized weights. All subsequent iterations include 10 training epochs without the isolated head training stage since the heads are not randomly reinitialized for every new iteration. The teacher branch weights are updated with the exponential moving average of the student branch weights after every epoch. We adopt AdamW optimizer [5] and an initial learning rate (lr) of 1×10^{-4}. We use cosine decay scheduling and reduce the lr to a final value of 1×10^{-5}. The initial lr is linearly warmed up for the first 5 epochs for each iteration from a starting value of 5×10^{-6}. Experiments were conducted using a Nvidia A10 GPU with 24 GB VRAM, running CUDA 12.0, with code implemented in PyTorch 1.13 [6]. For finer details on the loss function and training hyperparameters, please refer to [15].

1.4 Baseline Methods Implementation Details

All baseline methods were implemented in Pytorch 1.13 [6]. For a fair comparison, we utilize ArcFace-R100 [1] as the base feature extraction model for all baseline methods, including ours. We would be making the baseline method codebase publicly available for the community's sake and result reproducibility.

Implementation for TSiam and SSiam [9] Since finer details on training hyperparameters are unavailable, we assume standard values for batch size (32 for TSiam) and training epochs (100). Besides, no image augmentations are added during training. We remark that we generate results on the MovieFaceCluster dataset tracks, excluding any bad face quality tracks that the original method struggles to cluster. A global threshold, which was empirically deemed to be optimal for the entire MovieFaceCluster dataset, was set as the cut-off threshold for the hierarchical agglomerative clustering module. Please refer to [9] for the rest of the implementation details.

Implementation for JFRAC [14] We implement a Markov Random Field (MRF) approach in Python from scratch for face clustering. For the rest of the implementation details, please refer to [14].

Implementation for CCL [10] As part of the proposed system, the FINCH algorithm [8] was incorporated from Scikit-learn library [7]. The respective paper

recommends using the clusters generated from the 2nd partition as weak supervisory labels. However, we noticed that the 1st partition itself provided good clusters, while the second partition somewhat degraded the clustering performance. Hence, we incorporated labels from the 1st partition. All training parameters, including training epochs, batch size, and learning rate, as well as the positive and negative pair sampling mechanism, were rigorously followed as detailed in the paper. For the rest of the implementation details, please refer to [10].

Implementation for VCTRSF [13] Due to the lack of a detailed description of the modifications made to the standard ViT architecture [3], we assumed standard parameters for implementing the video transformer model. Specifically, the number of heads in each layer was kept at 16, 4 layers were incorporated within the ViT model, and the input patch size was kept at 1x1 (since the model's input are pre-extracted embeddings in our case rather than standard images). The hidden and output embedding dimensions were kept at 512 and 256, respectively. The set value for learning rate (lr) ratio multiplier p wasn't mentioned in the paper. We set the value to 1.0 in our reported experiments, resulting in the lr for transformer model architecture being the same as the one for updating video centers. For additional implementation details (which weren't modified in our implementation), please refer to [13].

2 Algorithm pseudo-code

We present the pseudo-code of the different steps in our proposed method in Algorithm 1. The mentioned stage numbers correspond to the main paper Fig. 2.

Algorithm 1: VideoClusterNet

Input:
Face Tracks $T = \{t_j |\ j = 1, 2, ..., N\}$
$\ni t_j = \{I_{t_1}, I_{t_2}, ...I_{t_n} |\ f_{n+1} - f_n = 12\}$ (obtained from stage 1)
pretrained model θ_m, **cluster iterations** *total iters*
Stage 2: Self Supervised Model Finetuning
$\theta_s, \theta_t \longleftarrow\ replicate(\theta_m)$
$\theta_s \longleftarrow\ \theta_s + attach\ head(\theta_h)$
$\theta_t \longleftarrow\ \theta_t + attach\ head(\theta_h)$
$T_{filtered} = T,\ T_{cm} = None$
for *i in total iters* **do**
  $\theta_{s_i}, \theta_{t_i} \longleftarrow\ finetune\ model(\theta_{s_{i-1}}, \theta_{t_{i-1}}, T_{filtered}, T_{cm})$
  $T_{fq} \longleftarrow\ face\ quality\ estimation(T, \theta_{s_i})$
  $T_{filtered} \longleftarrow\ filter\ outliers(T, T_{fq})$
  $T_{cm} \longleftarrow\ track\ coarse\ matching(\theta_{s_i}, T_{filtered})$
  $\theta_{ft} \longleftarrow\ \theta_{s_i}$
end
Stage 3: Fully Automated Face Track Clustering
$T_{fq} \longleftarrow\ face\ quality\ estimation(T, \theta_{ft})$
$T_{filtered} \longleftarrow\ filter\ outliers(T, T_{fq})$
$C \longleftarrow\ cluster\ tracks(T_{filtered}, \theta_{ft})$
Output: Clustered track IDs C

3 Additional details for track quality estimation

Threshold value for filtering out bad face quality tracks For a given set of N tracks, a quality score threshold is computed as follows:

$$thres(tqs(N)) = \overline{tqs(N)} - (2.7 \times MAD(tqs(N))) \quad (1)$$

where $tqs(t_n)$ is the face quality score computed for the n^{th} track using technique detailed in main paper Sec. 3.6, $tqs(N) = \{tqs(t_1), tqs(t_2), ..., tqs(t_j)\} \; \forall \; j = \{1, 2.., N\}$. Tracks having a score lower than $thres(tqs(N))$ are filtered out from both coarse track matching and final clustering modules. The value of 2.7 was first loosely set by fitting a Gaussian distribution onto the given set of track face quality scores. To select/filter the lower 1.5% outliers, which are often less than the threshold of (mean - 2.8×std) in a Gaussian distribution, we sampled values in the range of 2.6 to 3.0 and empirically found that 2.7 worked optimally for our test set consisting of a wide range of movies.

4 MovieFaceCluster Dataset Curation

Movie / TV Series	Specific Face ID challenges
An Elephant's Journey (2019)	Bright outdoor scenes, American cast
Armed Response	Low light scenes, facial occlusions with military helmets, sunglasses, etc., African American and Middle Eastern Cast
Angel Of The Skies	Unique heavy occlusions with oxygen masks in bright settings
Death Do Us Part (2019)	Low light scenes, Extreme facial expressions like screaming, Extreme poses, Rapid movements, African American Cast
American Fright Fest	Facial occlusions like see-through masks, sunglasses, extreme poses
The Fortress	Facial occlusions like headgear, Large main cast with primarily Asian characters
Under The Shadow	Low light scenes, Middle Eastern Cast
The Hidden Soldier	Low Light scenes, Asian Cast
S.M.A.R.T. Chase	Extreme lighting in some scenes, Asian Cast
Big Bang Theory (S1E01-06)	Mainly Indoor scenes in constant well-lit environments, American cast
Buffy The Vampire Slayer (S5E01-06)	Overall darker scenes, American Cast

Table 1: Specific Face ID challenges presented by each movie, as part of MovieFaceCluster dataset and literature benchmark datasets

The MovieFaceCluster dataset provides challenging face ID tracks within a set of hand-selected mainstream movies. These challenges involve large variations in pose, appearance, illumination, and occlusions that are unavailable in any generic movie face ID datasets. To the best of our knowledge, this is the most comprehensive Video Face Clustering dataset for movies to be open-sourced. It consists of a total of 3619 face tracks across 209 different identities spanning nine movies. Each constituent movie has a unique set of characteristics in terms of number of characters, average track length, character age, ethnicity, and background environments, among other factors. Please refer to Tab. 1 for further specific details on it.

The dataset comprises tracks and face box spatial locations corresponding to a global movie frame index, for a given frame rate. These tracks are generated using the preprocessing module explained in detail in the main paper Sec. 3.2. As part of this processing stage, any bad quality tracks are discarded from the dataset. Also, each dataset movie consists of a mix of main and secondary characters. We

remark that an unique face identity, which has at least two good quality tracks, is included as part of the movie dataset.

Responsible Dataset Curation Our proposed dataset comprises face crop data for all lead characters in a movie and their respective proxy identity labels. We would like to report that all nine movie videos were procured from a US based movie distributor company, facilitated through an internal usage license. Personal human subject information in the form of a face crop is part of our dataset. The character identity is, however, anonymized by assigning a random number identifier rather than the character's listed name to each face crop. Following the dataset release format adopted by BBT and BVS datasets, our dataset consists of global frame index and face bounding box location for each respective face crop, in place of face crop image data. This help us maintain compliance with our internal usage license and requires the dataset user to obtain the respective movie video through alternate means.

5 Video Face Clustering Dataset Comparison

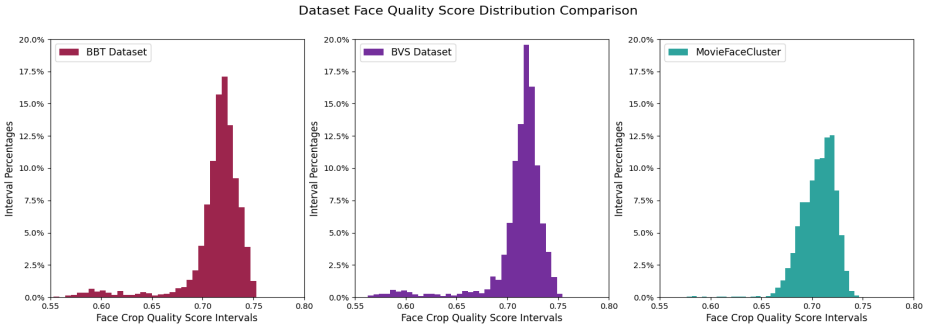

Fig. 1: Comparison of track crop quality score distributions across The Big Bang Theory (BBT), Buffy, The Vampire Slayer (BVS), and our MovieFaceCluster dataset. For a given dataset, the face crop quality score is computed for each of its constituent track. It is estimated as the average of scores calculated for a given track's sampled crop set (using SER-FIQ [12] and ArcFace-R100 [1] as the pre-trained model for extracting embeddings). The distribution mean is relatively lower for our MovieFaceCluster dataset compared to other benchmark datasets, along with more bias towards lower quality score interval - 0.66 to 0.7. This provides empirical evidence for our dataset containing more challenging cases for face clustering due to lower face quality scores.

In order to provide insights into the uniqueness of our proposed MovieFaceCluster dataset compared to existing literature, in Fig. 1, we present a dataset percent histogram comparison across face quality scores computed per dataset track. Similarly, in Fig. 2, we present a dataset percent histogram comparison for face crop parameters that are highly relevant for face clustering, namely scene lighting and face blur level. Scene lighting values are estimated as the average of lightness (L) parameter values in a given face crop image converted to HLS space. Face crop blur is estimated using a Singular Value Decomposition (SVD) based

method [11]. Significant differences were not observed in face pose attributes across all datasets.

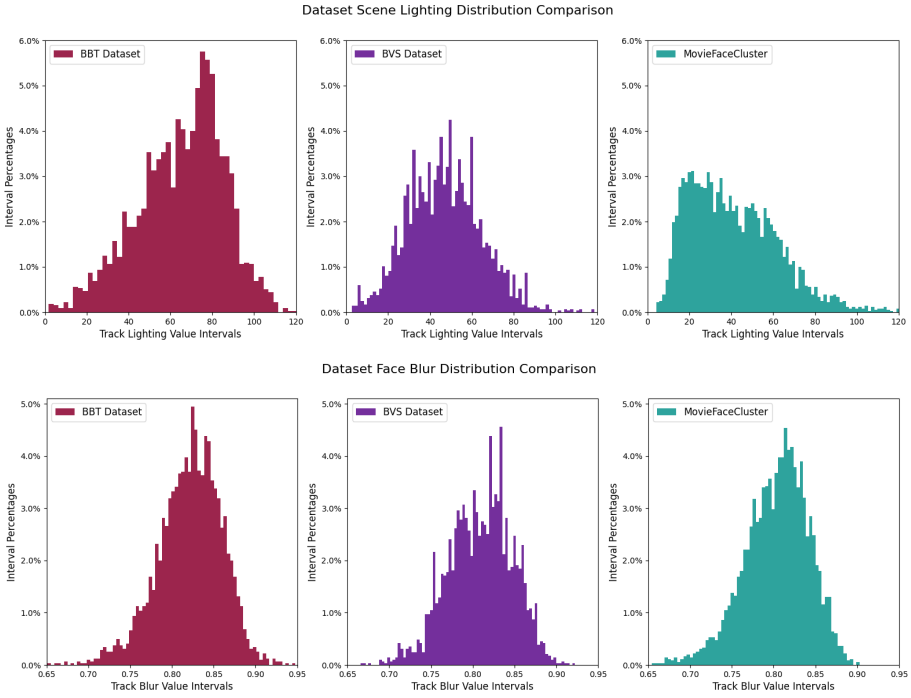

Fig. 2: Dataset attribute histogram comparisons for scene lighting and face blur across The Big Bang Theory (BBT), Buffy, The Vampire Slayer (BVS) and our MovieFaceCluster dataset. Each respective dataset attribute is computed per track, taken as the average of each of its sampled crop attributes. For scene lighting, MovieFaceCluster has higher distribution variance compared to literature datasets, with more bias towards darker lighting (values have a positive correlation to the amount of track scene lighting). For face blur, MovieFaceCluster has a higher sample count in 0.65 to 0.75 value range compared to literature datasets, providing evidence of higher blur levels in track face crops (blur values have negative correlation to amount of blur present in face crops).

With reference to (Main paper Tab. 2), our proposed MovieFaceCluster dataset has, on average, a lot more unique character faces and significantly better cast racial diversity. Also, it manages to obtain a much lower average face quality score, alluding to the fact that our dataset contains, on average, more challenging data samples w.r.t. face clustering/identification task.

With reference to Fig. 1, our proposed MovieFaceCluster dataset contains a higher percentage of dataset tracks having lower face quality scores compared to literature datasets. In addition, Fig. 2 signifies, firstly, that MovieFaceCluster has higher variance in scene lighting across its face samples with a bias towards a lot darker scenes, which helps provide harder scenarios for face clustering. Secondly, for facial blur, our dataset has a larger track count with high facial

blur compared to literature datasets, which again helps provide more challenging scenarios for face clustering.

6 Dataset Visual Comparisons

Fig. 3 presents a visual comparison of select face crops from BBT, BVS and our proposed MovieFaceCluster dataset.

Fig. 3: Visual comparison of literature versus our proposed MovieFaceCluster dataset. As evident, our dataset contains, on average, more challenges for face clustering task. Each depicted cluster is predicted using our proposed method. The term "varying parameter" depicts the dominant image attributes that are particularly challenging for a given face crop. It is not part of the available dataset annotations but is simply mentioned for enhanced reader understanding.

7 Additional t-SNE Visualizations

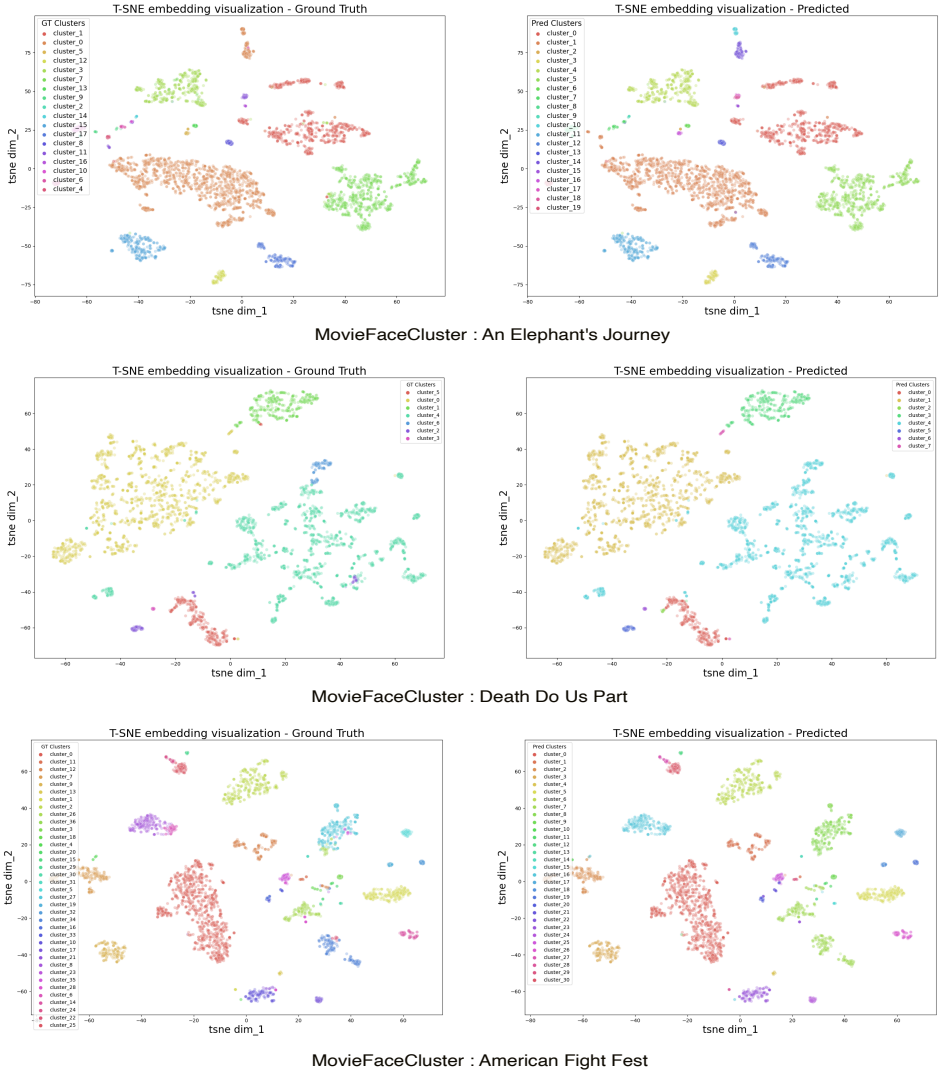

Fig. 4: Comparative t-SNE embedding visualizations on select **MovieFaceCluster** movie datasets. *Left:* Ground truth, *Right:* Our method. Each dot in the diagram above represents the finetuned model's extracted embedding for a face crop I_{t_n} in a given track's sampled crop set t. Face embeddings assigned to a given color constitute a single cluster.

8 Finetuning Iteration Ablation

This section provides ablation results for optimal Self-Supervised Learning (SSL) finetuning iteration count, performed on the *MovieFaceCluster: The Hidden Soldier* and *An Elephant's Journey* dataset. Our overall experiments showed that a minimum of 5 iterations are required to obtain optimal results. Each iteration here denotes a single model finetuning stage followed by the coarse matching stage. Additional few iterations (2 to 4) are necessary for harder datasets. The above assumes that the first iteration is run for about 30 epochs, with each succeeding one comprising 10 epochs. For iteration 10 and above, our method might over-cluster the dataset, i.e., it finds sub-clusters within the optimal clusters, optimizing for variants of a given character. Note that the first SSL finetuning iteration involves image pair generation from within the same track only. For all succeeding iterations, image pairs are generated across coarse-matched tracks, facilitated through the coarse track matching process detailed in main paper Sec. 3.5.

SSL Finetuning Iteration	Accuracy / WCP (%)	Pred Cluster Ratio/ PCR (Pred / GT)
1	52.56	0.476 (10/21)
2	72.12	0.571 (12/21)
3	91.38	1.114 (24/21)
4	96.93	1.000 (21/21)
5	**98.50**	**1.048 (22/21)**
6	98.41	1.095 (23/21)
7	98.47	1.095 (23/21)
8	98.50	1.190 (25/21)
9	98.50	1.238 (26/21)
10	98.41	1.238 (26/21)

SSL Finetuning Iteration	Accuracy / WCP (%)	Pred Cluster Ratio/ PCR (Pred / GT)
1	43.21	0.388 (7/18)
2	82.94	0.555 (10/18)
3	92.41	0.667 (12/18)
4	95.18	0.889 (16/18)
5	**97.20**	**1.111 (20/18)**
6	96.92	1.167 (21/18)
7	96.63	1.167 (21/18)
8	97.05	1.277 (23/18)
9	96.92	1.238 (24/18)
10	97.05	1.333 (24/18)

Table 2: Ablation for SSL finetuning iterations count, presented for **MovieFaceCluster:The Hidden Soldier** (left) and **MovieFaceCluster:An Elephant's Journey** (right) dataset. Iterative finetuning and coarse matching for 5 iterations provide optimal results in terms of both metrics.

9 Training And Evaluation Timings

Our proposed algorithm comprises two main stages from a computation standpoint: 1) Model SSL finetuning and 2) Final clustering. Tab. 3 presents run times of each of these stages for each movie of MovieFaceCluster dataset.

Statistics	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	Average (Per Track)
Track Count	562	119	319	395	457	917	143	594	113	-
Finetuning Iterations	5	6	5	8	10	9	6	5	5	-
Model Finetuning (mins)	193.76	38.45	110.16	397.24	298.41	645.06	57.21	307.12	48.19	0.579
Final Clustering (secs)	66.36	6.74	28.42	40.82	61.94	167.32	13.93	81.32	9.13	0.132

Table 3: Training iteration count and runtimes for SSL finetuning and final clustering computed on MovieFaceCluster dataset. Here, the training iteration count represents the SSL finetuning iteration at which the process was terminated, and final clustering was performed.

Bibliography

[1] 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) 2, 5

[2] 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) 2

[3] 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) 3

[4] Hendrycks, D., Gimpel, K.: Gaussian error linear units (gelus). arXiv preprint arXiv:1606.08415 (2016) 1

[5] Loshchilov, I., Hutter, F.: Decoupled weight decay regularization. In: International Conference on Learning Representations (ICLR). Springer (2019) 2

[6] Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., Killeen, T., Lin, Z., Gimelshein, N., Antiga, L., Desmaison, A., Köpf, A., Yang, E.Z., DeVito, Z., Raison, M., Tejani, A., Chilamkurthy, S., Steiner, B., Fang, L., Bai, J., Chintala, S.: Pytorch: An imperative style, high-performance deep learning library. In: Advances in Neural Information Processing Systems (NeurIPS). pp. 8024–8035 (2019) 2

[7] Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., et al.: Scikit-learn: Machine learning in python. Journal of machine learning research **12**(Oct), 2825–2830 (2011) 2

[8] Sarfraz, S., Sharma, V., Stiefelhagen, R.: Efficient parameter-free clustering using first neighbor relations. In: Conference on Computer Vision and Pattern Recognition (CVPR). pp. 8934–8943. IEEE (2019) 2

[9] 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) 2

[10] 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) 2, 3

[11] Su, B., Lu, S., Tan, C.L.: Blurred image region detection and classification. In: ACM International Conference on Multimedia (ACM MM). pp. 1397–1400. ACM (2011) 6

[12] 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) 5

[13] 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) 3

[14] 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) 2

[15] 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) 2