Pattern Analysis and Applications (2025) 28:88
https://doi.org/10.1007/s10044-025-01464-3

ORIGINAL ARTICLE

TVFace: towards large-scale unsupervised face recognition in video streams

Atif Khurshid[1] · Bostan Khan[2] · Muhammad Shahzad[3] · Muhammad Moazam Fraz[1]

Received: 3 October 2024 / Accepted: 30 March 2025
© The Author(s), under exclusive licence to Springer-Verlag London Ltd., part of Springer Nature 2025

Abstract
Recent advances in deep learning have led to significant improvements in face recognition systems, but face clustering, particularly in video streams, remains a challenging problem. Current video face clustering approaches are primarily tailored for short-form content, such as movies and television shows, that features a limited number of face images and individuals. The few existing large-scale face datasets are derived from web images and do not effectively capture the complexities of the video domain. In view of these limitations, we present TVFace, the first large-scale dataset of face images extracted from long-form video content. TVFace has been sourced from public live streams of international news channels and contains a total of 2.6 million face images of 33 thousand individuals. To address the challenge of identity annotation in unstructured video streams, we design a semi-automatic annotation framework that combines unsupervised face clustering with human validation, ensuring scalable and high-quality labeling. TVFace is well suited to evaluate and advance face representation and identity classification components of face recognition systems across both image and video domains. We also demonstrate the effectiveness of TVFace in evaluating real-time person retrieval systems using a novel tree-search-based Hierarchical Retrieval Index tailored for online face clustering. In conclusion, our work centers around the preparation of TVFace, a dataset poised to reshape the landscape of face recognition in the video domain, making it a crucial resource for the research community. The dataset and code are available at https://github.com/Vision-At-SEECS/streamface.

Keywords Face dataset · Video face clustering · Visual analysis · Live television · Hierarchical retrieval index

1 Introduction

Face clustering is a fundamental task in computer vision that allows for the grouping of similar faces without any prior knowledge of their identities [1]. It enables efficient organization and retrieval of faces, facilitating large-scale face recognition in various domains including human-robot interaction [2], surveillance [3, 4], and forensic investigation [5]. Facial analysis in video streams is a particularly interesting application of face clustering, where it can be used to extract valuable insights by automated aggregation of appearances and screen time [6]. For example, the clustering of faces appearing on news television can allow researchers and media monitors to conduct large-scale studies to answer socially relevant questions, such as biases in media coverage [6]. The identification and association of faces with demographic attributes can also allow broadcasters to analyze audience engagement and preferences, in order to optimize content production.

✉ Muhammad Moazam Fraz
moazam.fraz@seecs.edu.pk

Atif Khurshid
akhurshid.mscs20seecs@seecs.edu.pk

Bostan Khan
bostan.khan@mdu.se

Muhammad Shahzad
m.shahzad2@reading.ac.uk

[1] School of Electrical Engineering and Computer Science, National University of Sciences and Technology (NUST), Islamabad, Pakistan

[2] Division of Intelligent Future Technologies, Mälardalens University, Västerås, Sweden

[3] Department of Computer Science, University of Reading, Whiteknights House, RG6 6UR Reading, United Kingdom

Published online: 15 April 2025

Video stream analytics systems are limited to unsupervised clustering techniques due to the unscalability of traditional approaches to face recognition, such as verification and identification, which require one-to-one comparison between faces. Consequently, their computational complexity degrades significantly with increasing number of images and classes [7]. In addition, the unknown nature of individuals makes it impractical to train classification models. Although there exist large datasets of celebrity faces [8–10], enabling efficient identification of well-known personalities within video streams, the challenge lies in identifying unknown individuals, and unsupervised clustering provides the only viable solution for this task.

Video face clustering has been studied extensively in recent years [11–14] but these studies are generally conducted on short-form content, such as movies and television shows, with datasets [15–18] limited to only a few hundred thousand images of a few thousand individuals. This limitation arises because short-form content typically has a restricted runtime of only a few hours, inherently capping the number of unique frames and face appearances available for clustering. In contrast, long-form content, such as live television broadcasts, spans several hundred hours of continuous footage, significantly increasing the volume of facial data. This extended runtime can enable the creation of more diverse and comprehensive datasets containing millions of images. Although several million-scale face datasets [8–10, 19, 20] are available, these are based on web images of celebrities and do not represent the challenges of the video domain, such as frequent variations in photometric properties and non-discriminatory facial attributes, like pose and expression.

We present TVFace, the first large-scale dataset of face images sourced from video streams. It consists of 2.6 million images of 33 thousand individuals, extracted from live broadcasts of 22 international news channels. The dataset is organized into 22 subsets, each corresponding to a specific television channel, and features a diverse demographic distribution, with the channels originating from 15 countries across five continents. The dataset is also annotated for six facial attributes (mask, age, gender, ethnicity, expression, and pose) using state-of-the-art models for face analysis.

Identity annotation is the main challenge in extracting face datasets from video streams due to the complete lack of even weak supervision for labelling. In web images, associated metadata can be used to identify faces while movie-based datasets can rely on lists of cast members to facilitate annotation. However, faces extracted from long-form video content, such as live television, are completely unstructured. Therefore, we design StreamFace, a semi-automatic annotation framework based on unsupervised clustering, in order to facilitate manual annotation of large collection of face images. Face clustering is employed to generate an initial labeling that groups the most similar faces, followed by manual annotation that merges overlapping clusters and removes noisy ones. Moreover, faces from each television channel are annotated independently to enhance labelling accuracy since small-scale face recognition systems are generally more accurate than large-scale systems.

We also design a Hierarchical Retrieval Index for fast face matching in online person identification to demonstrate the effectiveness of the proposed dataset in evaluating real-time person retrieval systems. The index consists of a tree-based data structure where leaves contain feature vectors representing face images, while the internal nodes store their mean feature vectors. This results in a hierarchical ordering of feature vectors similar to a dendrogram. Given a query feature vector of a new face image, its approximate nearest neighbors can be determined by traversing down the tree like tree search. The index can be employed for several tasks in face recognition including identification and online clustering.

The following is a summary of our contributions:

- We develop frameworks for automated extraction of face images from live streams and semi-automatic identity annotation of unstructured collections of face images.
- We prepare the first large-scale dataset of face images extracted from long-form video content using the aforementioned frameworks. The dataset is well-suited for the evaluation of face representation and identity classification components in both image and video domains, as well as for multiple tasks including identification and clustering.
- We design a Hierarchical Retrieval Index based on tree search that can be employed for fast face matching in several face recognition tasks. The index is used to demonstrate the effectiveness of the proposed dataset in evaluating real-time person retrieval.

The remainder of this paper is organized as follows. Section 2 contains an overview of existing research relevant to our problem. Section 3 details the dataset preparation methodology while Sect. 4 describes the proposed Hierarchical Retrieval Index, followed by experiments and results in Sect. 5. The results, implications and limitations of this study are discussed in Sect. 6.

2 Related work

This section contains a discussion on the limitations of current face datasets in order to identify the research gaps that need to be addressed, along with an overview of current face clustering and identification techniques.

2.1 Face datasets

Face datasets are crucial for the development of face recognition systems as these provide a standardized benchmark for evaluating and comparing different approaches. These datasets also help researchers to identify the biases and limitations of their models, guiding future development. A good face dataset should consist of a large and diverse collection of face images, accurate identity labels, and facial attribute annotations, in order to ensure that the recognition system is robust to variability in facial appearance.

Face recognition benchmarks [16, 21–24] are generally quite small due to the infeasibility of manual annotation at large scales. In recent years, some large-scale datasets [8–10, 20] have also been made publicly available but these contain high quality web images of celebrities and are ill-suited for the evaluation of recognition in unconstrained settings. Moreover, image-based datasets rarely feature the range of variability in photometric properties and non-discriminatory facial attributes characteristic of the video domain, and even non-celebrity datasets [19] are subject to these limitations. Video face datasets are curated from movies and television shows that either feature a small number of face images [17, 27] or are limited in terms of the number of classes [25, 26].

Face datasets are often biased towards certain demographic (age, gender, and ethnicity) and non-demographic (pose, expression, hairstyle, and accessories) facial attributes [28]. In fact, a lot of research in recent years has focused on the correction of these biases and the development of invariant recognition systems [29–36]. There has been a particular emphasis on the refinement of existing datasets alongside the development of new ones [20, 37]. The LFW [21] dataset, for example, has several variants focusing on different biases [38, 39].

2.2 Face clustering

Face clustering has been the subject of significant research over the years, especially for unsupervised identity classification. The initial approaches were generally designed for small-scale applications, like photo album tagging, that often featured less than a hundred individuals and a few thousand images [40]. More recent studies have focused on the development of novel clustering techniques for large-scale applications [1, 41–44].

The availability of large-scale datasets also led to interest in supervised clustering techniques [45–48]. These are generally based on graph convolutional networks (GCNs) [49], deep neural networks that extend convolutions to graph-structured data, that learn complex patterns in affinity graphs based on nearest neighbors calculated using feature vectors in deep feature spaces. Although GCN-based approaches have been shown to be more accurate than unsupervised clustering techniques on some datasets, they are fairly sensitive to hard or noisy samples while also carrying a large computational footprint as they require long training times on large datasets [50].

Clustering can also facilitate the annotation of completely unstructured collections of face images by generating an initial rough labelling. However, face clustering techniques are generally not very accurate, especially at scale, and require post-processing for the removal of impure clusters. Nech et al. [19], for example, use intra-cluster pairwise distance distributions to detect impure clusters in MegaFace2. Although this is a completely automated technique, it relies on the accuracy of face representation models that is generally quite poor in videos due to frequent variations in pose, expressions, and photometric properties. Another approach is to perform partial clustering that groups only the most similar faces, resulting in multiple clusters for the same individual which can then be combined by human annotators [51, 52]. This method allows for manual annotation of large-scale datasets since labelling images by groups significantly reduces the annotation workload.

2.3 Face identification

Face identification typically requires pairwise comparisons between feature vectors, known as face matching [53]. The vectors of known faces are saved in a database, called the gallery, and probe vectors of unknown faces are compared with all vectors in the gallery based on some distance metric, like cosine distance. A probe is assigned an class if its distance to the class's exemplar in the gallery is less than some threshold.

As the computational efficiency of such methods degrades with scale [7], there has been considerable interest in large-scale face identification, especially in recent years as face datasets have become larger and more challenging. These studies [7, 54–56] aim to reduce the search space, either by using approximate search to find top candidates or by filtering out imposters. However, such methods often require the transformation of deep features into simpler representations that facilitate fast searching but inevitably lead to a loss of crucial discriminative information. Hence, they

achieve computational efficiency at the cost of classification accuracy. The proposed hierarchical retrieval index, on the other hand, aims to facilitate fast identity classification by reducing the number of comparisons without compromising the discriminatory power of deep features.

3 TVFace dataset

This section describes the dataset preparation process, from extraction of face images and labeling of person identity using a semi-automatic annotation framework to the final dataset and its statistics.

3.1 Data sources

We used YouTube live streams of international news channels as the source of the dataset since these are publicly available and easily accessible. These live streams not only feature a large number of individuals, most of whom are non-celebrities, but the distribution of screen time per person also varies significantly. Some people, such as anchors and politicians, appear very frequently while others may only be on screen for a few seconds, resulting in a long-tailed dataset. We selected 22 channels from 15 countries around the world to ensure a diverse demographic distribution in the dataset. Video frames were sparsely sampled for face image extraction since a dataset of video clips would have an unmanageable memory footprint at this scale. Although this results in the loss of continuous temporal information, a more discrete version is still preserved by maintaining metadata about the temporal order of faces in the form of timestamps.

A dataset sourced from videos, instead of web images, can more accurately represent the challenges of the video domain such as variations in photometric properties and non-discriminatory facial attributes like pose and expression. The extraction of face images while maintaining their temporal ordering makes the dataset suitable for the evaluation of face recognition models in both image and video domains. It also facilitates the task of online clustering, which is the most realistic scenario for a real-world video analytics system. Moreover, the smaller memory footprint of face images allows for a large number of individuals to be included in the dataset.

3.2 Face extraction

The live streams of selected channels were accessed over a period of two months and downloaded at the highest resolution available. We employed video compression-based keyframe extraction and content-based frame analysis to reduce the number of stored frames. Key frames allow for sparse sampling of high-quality images with minimal noise and motion blur while adapting to the variability of scene changes. These were extracted using the PyAV package. Content-based analysis was used to exclude empty, blurry, and duplicate frames. Empty frames were determined by analyzing the edge count produced by the Canny edge detector, blurriness detected using Variance of Laplacian, while duplicate frames were identified using template matching with the previous frame. Once selected, the frames were stored to disk with their timestamps as filenames.

RetinaFace [57] was used to detect and extract faces from stored frames. We implemented minimum thresholds for confidence and size of detected faces. The predicted bounding boxes were enlarged to include the region around the faces, which were then cropped out and resized to 256×256 pixels. The extracted face images were stored to disk with their filenames derived from the filenames of their respective frames, thus preserving the temporal ordering of faces using the timestamps of their on-screen appearances.

3.3 Identity annotation

We designed a semi-automatic framework for labelling unstructured collections of face images. It consists of a clustering step, whereby the collection is partitioned into an unknown number of clusters with high purity, followed by manual annotation to remove noisy clusters and merge clusters of the same individual. Annotation was performed for each television channel independently at first to optimize accuracy and efficiency since small-scale face recognition systems are generally more accurate than large-scale systems and clustering algorithms are also limited by memory constraints. This was followed by a global annotation step, in which the manually cleaned clusters from all television channels were merged using the same manual cleaning methodology. We note that the number of individuals appearing on multiple channels is fairly small since every channel has a unique cast of presenters and contributors. International news stories can be featured on multiple channels but these generally involve politicians and celebrities who can be identified quite accurately using existing face recognition systems.

3.3.1 Clustering

We deployed an ensemble of two feature representation models, to generate the feature vectors. The first consists of a ResNet34 backbone trained on the CASIA-WebFace [8] dataset using ArcFace loss [58] that outputs 512-dimensional feature vectors. The other comprises an Inception-ResnetV2 backbone and is trained on the VGGFace2 [20]

dataset using softmax loss. It also outputs 512-dimensional feature vectors. Both models were imported pretrained from the DeepFace library [59]. The outputs of individual models were L2 normalized and combined using summation, followed again by normalization. Face alignment was performed by taking a close-crop of the face and rotating the image until eyes were horizontally level.

Agglomerative clustering with complete linkage [60] was selected as the clustering algorithm since it is parameterized by the pairwise distances between feature vectors, which are directly optimized by feature representation models. The complete linkage criteria is also designed to yield closely-knit, "spherical" clusters, resulting in groupings of high purity. Cosine distance was used as the distance metric. Intrinsic evaluation, based on the Silhouette Coefficient [61], was employed, alongside manual verification, to determine the optimal distance threshold for clustering. This value of cosine distance was found to lie between 0.2 and 0.3 for different channels.

As hierarchical agglomerative clustering with complete linkage has a memory complexity quadratic in the number of samples, we partitioned the collection of feature vectors into batches and performed clustering on each batch independently. Each cluster was then represented by the mean of its members and a global clustering operation was performed on the mean feature vectors from all batches to yield the final grouping.

Although this approximation can introduce inaccuracies, the temporal ordering of faces can be exploited to minimize its impact. Due to the structure of television programming, on-screen appearances of people are concentrated within certain time periods. Therefore, we arranged the collection of feature vectors according to the timestamps of their respective faces and created temporally ordered batches where all faces extracted from contiguous scenes were represented in the same batch. This does mean, however, that the appearances of the same person at two distant time periods cannot be clustered in the first phase and the global clustering operation is aimed as the solution to this problem.

The main drawback of batched clustering is that it necessitates stricter thresholds, leading to an increase in the number of clusters for each individual and the annotation workload. However, in our experience, the impact on cluster purity and annotation accuracy is rather small (see Sect. 5.5.1). Additionally, we observed that batch size had little impact on clustering performance, provided it was sufficiently large. In our experiments, we set the batch size to 20,000 for channels with fewer than 100,000 images and 50,000 for those with more images.

3.3.2 Manual cleaning

Clustering results in two kinds of erroneous groupings that need fixing: multiple clusters for the same individual and multiple individuals in one cluster. The former were corrected by merging all clusters for the same individual while the latter were excluded from the dataset.

Manual cleaning was performed on cluster pairs so that the annotator could examine them side by side to determine whether they contained images of the same person, while also checking both clusters for noise. For all cluster pairs, at most K face images were sampled from both clusters and shown to the human annotator. If all faces belonged to the same person, the clusters were marked for merging. If either of the clusters contained multiple individuals or non-face images, it was marked as noisy and excluded from the dataset. The value of K was empirically to 25 for all channels. Since the number of negative cluster pairs is far greater than positive pairs, the search space was limited to a list of most similar clusters. The mean feature vector from each cluster was used to compute pairwise cosine similarity between all clusters. A lenient threshold was then applied to exclude all pairs that were clearly dissimilar and the remaining pairs, sorted by similarity score, were selected for manual examination.

The manual cleaning process was repeated a few times to the satisfaction of the annotator. As the initial clusters do not completely represent a class, being merely groups of very similar faces, the list of most similar clusters may not include some positive pairs if they fail to meet the similarity threshold. However, as clusters are merged, more and more faces of the same individuals are grouped together, and the resulting clusters shape up to better represent their respective classes, i.e. the clustering becomes a more semantically significant partition. Consequently, on a successive computation of the list of most similar clusters, the similarity between a class representative and a missed cluster of the same class may exceed the threshold, thus covering the missed cluster.

After manual cleaning, the remaining clusters were considered as classes representing specific individuals. We assigned a number as person ID to each class in descending order of the number of images contained in the class. Hence, the most populous class was assigned an ID of 0 while the least populous class was assigned $n-1$, where n is the total number of classes. The ordering allowed us to easily identify classes containing less than 10 face images, which were then excluded from the dataset.

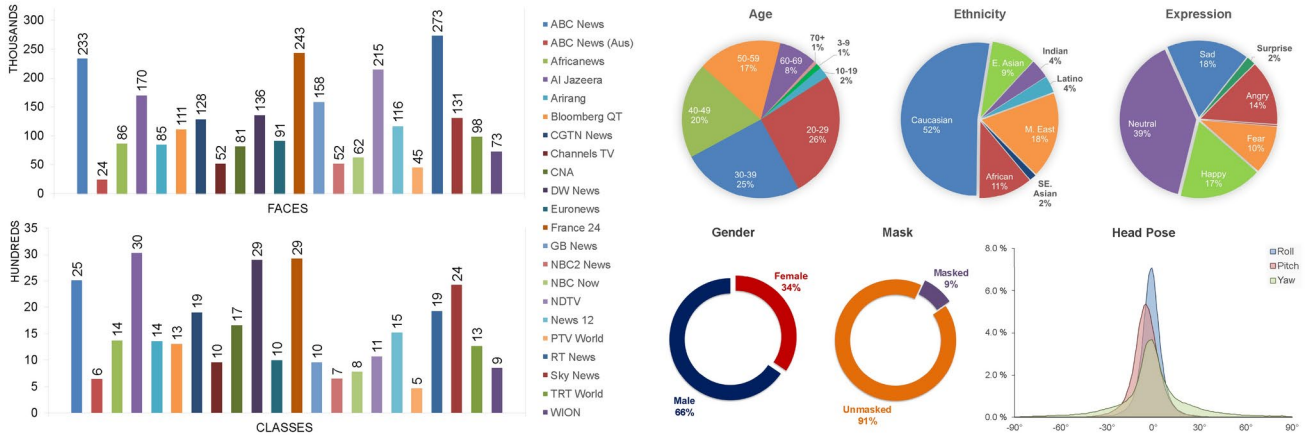


Fig. 1 Dataset Overview: TVFace consists of 2.6 million face images and 33 thousand classes divided into 22 subsets, one for each television channel. Figure on the left shows the channel-wise distribution of images and classes while figures on the right contain aggregate distributions of facial features estimated using state-of-the-art face analysis models

Table 1 Benchmark datasets for face recognition

Dataset	Classes	Images	Videos	Sources
LFW [21]	5749	13 K	–	
MS1M [9]	100 K	10 M	–	Web
MegaFace2 [19]	672 K	4.7 M	–	Images
WebFace [10]	2 M	42 M	–	
IJB-A [22]	500	5712	2085	
IJB-B [23]	1845	21 K	7011	Both
IJB-C [24]	3531	31 K	11 K	
Buffy [15]	20	–	12 K	
BBT [17]	10	–	3759	
SAIL-MCB [25]	80	–	9773	Videos
YTF [16]	1595	–	3425	
iQIYI-VID [26]	5000	–	600 K	
TVFace (Ours)	33 K	2.6 M	–	Live TV

3.4 Attribute annotation

We also used pretrained models to automatically annotate several facial attributes like age, gender, ethnicity, pose, expression, and mask. Pre-trained models are commonly used to accelerate annotation tasks and it is a widely accepted methodology in the field. Several large-scale face datasets [10, 19] have been automatically annotated using pre-trained models. We have adopted the same approach here, and have also included the confidence scores of the predicted labels in the annotations provided with the dataset to indicate their reliability. Age, gender, and ethnicity were classified using Fairface [37] while yaw, pitch, and roll angles were predicted using the WHEnet [62] model for head pose estimation. The Emotion model from DeepFace [59] was used to predict face expressions. Face masks were detected using a MobileNetV2 model [63] trained on a collection of masked face datasets.

3.5 Dataset statistics

A total of 3,525,183 faces were detected and extracted from 5,916,600 frames during the data collection phase. Automated clustering partitioned the dataset into 251,889 clusters. After manual cleaning, which involved merging and exclusion of clusters, the remaining 33,462 classes containing 2,663,373 face images constitute the proposed dataset.

The dataset was extracted from television channels around the world and features a diverse demographic distribution (Fig. 1). However, due to the limitations of publicly accessible live streams, only 15 countries from 5 continents are represented in the dataset, with South America missing. There is also a bias towards news channels from USA and Western Europe. Although international news channels highlight people from around the world, the aggregate demographic distribution of the dataset does indicate an over-representation of the Caucasian ethnicity.

The nature of news television also means that a significant number of subjects in face images maintain a frontal pose looking towards the camera, reducing the variations in head pose. Face expressions seem to be fairly diverse, with a significant number of happy, sad, and angry faces, while the neutral expression is most common. There is a slight gender imbalance with more male faces, though there is a considerable female minority. A substantial number of masked faces are also featured in the dataset.

TVFace is also a long-tailed dataset with significant class imbalance due to variations in screen time of different individuals (Fig. 3). Most face images belong to classes of frequently appearing individuals such as anchors, reporters, analysts, politicians and celebrities while most classes contain only a small number of face images corresponding to non-celebrities featured in news stories or non-recurring programs such as documentaries.

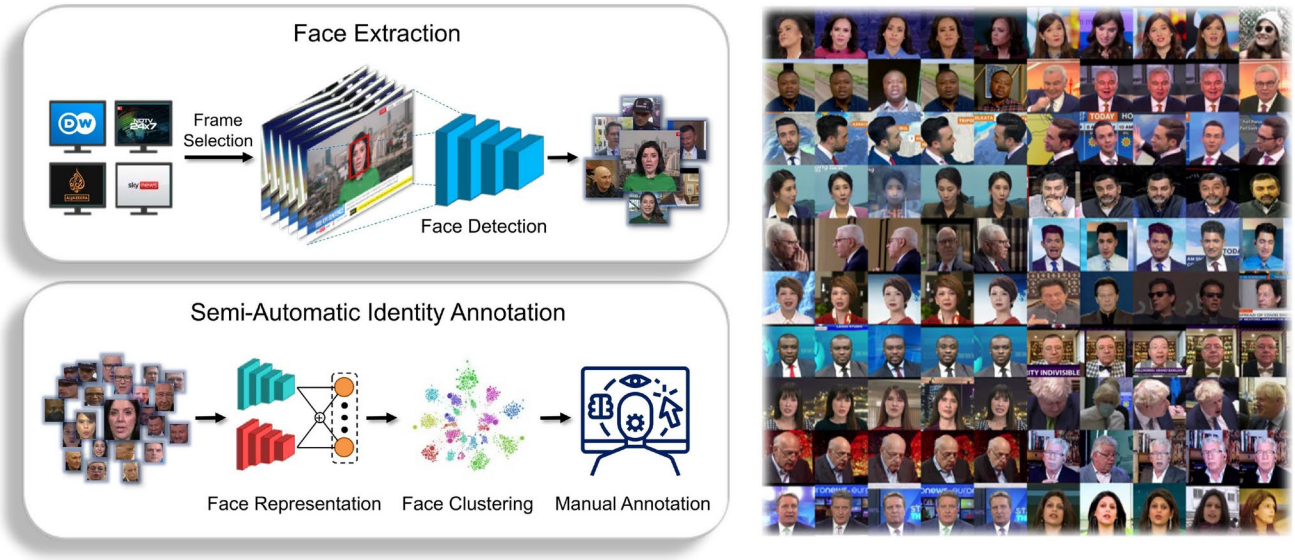


Fig. 2 Dataset preparation pipeline. Faces are extracted from live streams of television channels and labeled using a clustering-based semi-automatic annotation framework

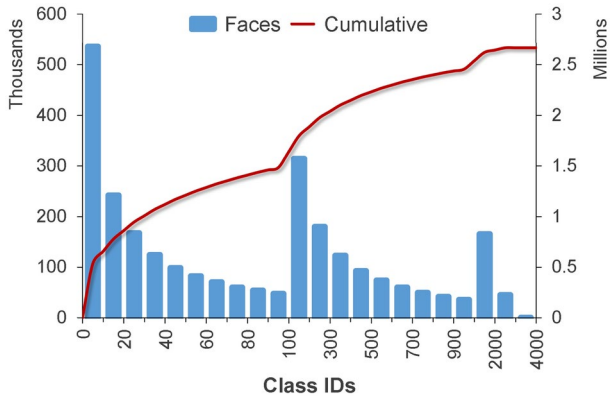


Fig. 3 Class membership distribution aggregated across all channels

4 Hierarchical retrieval index

Person retrieval in videos requires a stream of face images to be identified in real time. This is essentially an online clustering problem where samples are labeled one-at-a-time, as soon as they appear, without a priori knowledge of all faces. A naive approach for real-time person retrieval is to compare each new face with all faces already present in the gallery and determine its identity based on nearest neighbors. However, this is not scalable, especially as the number of faces in the gallery grows over time. Therefore, a more efficient and scalable approach is required for practical use.

4.1 Centroid matching

Centroid matching based on vector quantization is a basic approximation that can be introduced to enhance computational efficiency in face identification. Vector quantization is the representation of a vector by its closest centroid in a codebook. It can be deployed for fast face identification by partitioning the gallery into classes and quantizing each feature vector to its class centroid. This effectively reduces the number of comparisons required to determine the identity of a query vector to the number of classes. In the online scenario, the codebook is built up over time alongside the gallery. Given a query vector, if its distance to the closest centroid is within a threshold, it is assigned to the same class. Otherwise, it is given a new label and its respective centroid added to the codebook.

Centroid matching decreases the computational complexity of face identification from $\mathcal{O}(n)$ to $\mathcal{O}(c)$, where n and c are the number of faces and classes, respectively. However, the number of classes can not only be quite large, especially in long-form video content, but will continue to increase over time. Consequently, there is need for a matching technique with a more favourable computational complexity in terms of both the number of faces and classes involved.

4.2 Hierarchical retrieval index

We propose a Hierarchical Retrieval Index to facilitate fast identity classification by reducing the number of comparisons required for face matching without compromising the discriminatory power of deep features. It consists of a tree structure where leaf nodes contain feature vectors

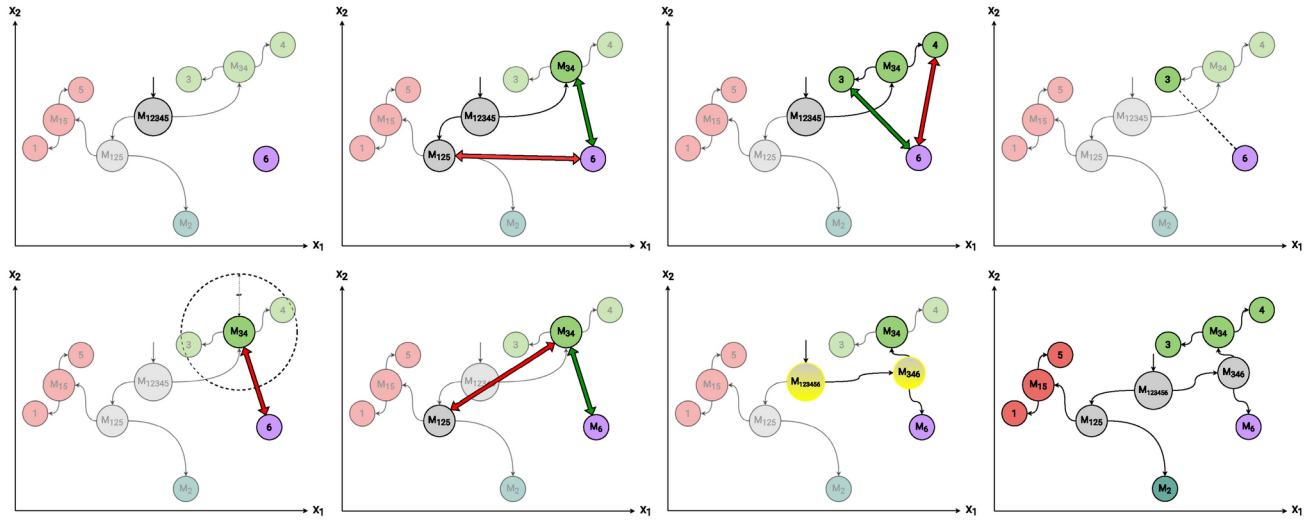


Fig. 4 A visual demonstration of search (top) and insert (bottom) operations on a hierarchical index in two-dimensional feature space. There are five leaf nodes representing faces, numbered 1 to 5 whereas the nodes labelled M_i are internal nodes. Node 6 represents a probe face that is first used in the search operation and then inserted into the index. The locations of the nodes correspond to the values of their respective feature vectors in the 2-D space and the four colors (red, blue, green, and purple) represent classes

representing face images while internal nodes store the mean feature vectors of their descendant leaf nodes. Leaf nodes representing the same class are grouped together under a single internal node, referred to as an endnode, which serves as the centroid for that class. The internal nodes are ordered according to pairwise distances like a dendrogram, enabling face matching to be performed as in a search tree, significantly reducing the number of comparisons required for face matching. The index is built iteratively and supports real-time search, insert and delete operations.

4.2.1 Search

The search operation determines which leaf node in the index is most similar to a probe feature vector. Beginning at the root node, we traverse down the tree following, at each node, the path of the child closest to the probe. Figure 4 (top) demonstrates the search operation performed on a hierarchical index containing feature vectors of 5 faces $(f_1, f_2, f_3, f_4, f_5)$ in order to find the vector closest to the probe f_6. Starting at the root node M_{12345}, the distances between its children (M_{125}, M_{34}) and the probe feature vector are calculated. As f_6 is closer to $f_{M_{34}}$ than $f_{M_{125}}$, the former node is selected for further exploration. Next, the distances between the probe and M_{34}'s children, 3 and 4, are calculated and 3 is chosen as the closest leaf node, bringing the search operation to its conclusion.

4.2.2 Insert

The insert operation adds a new feature vector to the index while maintaining the hierarchical ordering of all feature vectors in terms of pairwise similarity. As such, it is preceded by a search operation to determine the endnode closest to the new vector. If the distance between the vector and the endnode lies within a set threshold, it is added to the endnode's class as a new leaf node. Otherwise, it is assigned to a new class consisting of a single leaf node attached to its representative endnode. This subtree is then connected to the index as a sibling of the nearest endnode. Additionally, if the two endnodes are closer to each other than any of their siblings, they are grouped under a new internal node, as shown in Fig. 5, where the two endnodes and their new parent are denoted with N, S and P, respectively.

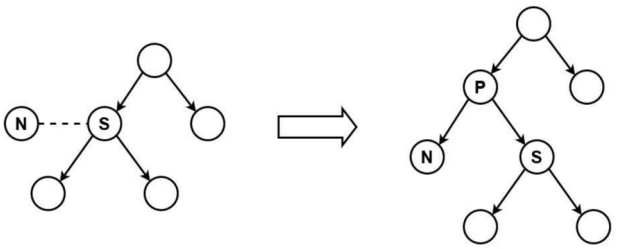


Fig. 5 Illustration of insertion process where two nodes are grouped under a new internal node

The attributes of all ancestors of the new node are also updated to make the index consistent with new data. The formulas for calculation of these values are shown below, where f_{c_i} is the feature vector, l_{c_i} is the class label, and d_{c_i} is the number of descendant leaf nodes for the i-th child. The values are calculated locally at each internal node, using only the attributes of its children.

$$\text{Value} = \sum_{i=1}^{C} \frac{f_{c_i} d_{c_i}}{d_{c_i}}$$

$$\text{Label} = \begin{cases} l_{c_i} & \text{if} l_{c_i} = l_{c_j}, \forall i, j \\ None & \text{else} \end{cases}$$

$$\text{Descendants} = \sum_{i=1}^{C} d_{c_i}$$

Figure 4 (bottom) shows the insertion of a new feature vector f_6 into the index after its closest endnode has been identified by the search operation depicted in Fig. 4 (top). As the distance between the new node 6 and its closest endnode M_{34} is greater than the threshold, it is assigned to a new class represented by endnode M_6. In this case, the distance between endnodes M_6 and M_{34} is smaller than the distance of siblings M_{34} and M_{125}. Therefore, node M_6 replaces node M_{125} as the new sibling of node M_{34} and a new internal node M_{346} is added as parent to the new siblings. Finally, the attributes of internal nodes M_{346} and M_{123456} are updated sequentially.

4.2.3 Delete

The delete operation removes leaf nodes from the index using sample IDs of their corresponding feature vectors. If the leaf node is the only member of its class, its parent endnode is also deleted. After the removal of these nodes, the attributes of all internal nodes on their path to the root are updated using the same procedure as followed during insertion.

4.2.4 Application

The proposed hierarchical index has been designed for the task of ordering a collection of face images according to their pairwise similarity by ensuring that similar feature vectors are inserted close to each other in the tree data structure. This property enables its application for several tasks in face recognition, and information retrieval in general.

In face identification, all images in the gallery can be registered in the index to facilitate fast face matching against probe images. The gallery can also be expanded in real-time with minimal impact on query computation time and feature vector integrity. Online face clustering is the primary use-case for the index and can be achieved simply by assigning a label to all feature vectors during the insert operation. The index can also be used for the calculation of approximate nearest neighbors by modifying the search operation which currently returns only the first nearest neighbours. The Hierarchical Retrieval Index is also well-suited for very-large scale scenarios involving databases and distributed deployment where a hierarchical ordering of the gallery can significantly reduce query times. For example, the highest layers of the index may be cached in primary memory to reduce search space for database queries, or the index may also be partitioned across a distributed computing cluster.

4.3 Limitations

The most prominent limitation of the proposed index is the reliance on mean feature vectors as representatives of their descendants. This essentially means that any query from the index can only return an approximate match rather than performing an exhaustive search. For example, during the search operation, a query vector is compared with the means of several partitions of feature vectors and only one partition is selected for further exploration according to the principles of tree search. However, the query vector may lie on the boundaries of some partitions and be closer to a feature vector in one of the rejected partitions than any vector in the accepted partition.

Another issue that limits the current implementation of the index is the unbalanced nature of the tree structure. The height and width of the tree are dependent on the feature vectors of faces registered in the index, and thus subject to variation. As the time complexity of all operations on the index is dependant on these factors, an unbalanced tree can significantly diminish the gains in computational efficiency achieved by indexing.

5 Experiments and results

5.1 Evaluation metrics

We used several metrics for clustering performance evaluation in this study including Purity, Adjusted Rand Index (ARI), Normalized Mutual Information (NMI), Pairwise F-score (F_P) and BCubed F-score (F_B) [64].

Cluster purity is defined as the weighted average of maximal precision values across all clusters and it measures the extent to which clusters contain data points from a single ground-truth class. A higher purity score indicates better clustering quality, but it does not penalize excessive splitting of clusters.

$$\text{Purity} = \frac{1}{N} \sum_{k} \max_{j} |C_k \cap L_j| \tag{1}$$

where N is the total number of data points and $|C_k \cap L_j|$ denotes the number of common points in cluster k and class j.

Rand Index corresponds to pairwise accuracy, the proportion of pairs that are clustered together or apart in both predicted and true partitions. It is adjusted between -1 and 1 to ensure a value close to 0.0 for random labelling.

$$\mathrm{ARI}=\frac{\sum_{ij}\binom{n_{ij}}{2}-\left[\sum_i\binom{a_i}{2}\sum_j\binom{b_j}{2}\right]/\binom{N}{2}}{\frac{1}{2}\left[\sum_i\binom{a_i}{2}+\sum_j\binom{b_j}{2}\right]-\left[\sum_i\binom{a_i}{2}\sum_j\binom{b_j}{2}\right]/\binom{N}{2}} \tag{2}$$

where n_{ij} is the number of elements in both cluster i and ground-truth class j, a_i is the total number of elements in cluster i, b_j is the total number of elements in ground-truth class j, and N is the total number of elements.

NMI measures the agreement of predicted groupings with ground truth labels based on mutual information, normalized against chance. Its values range from 0 to 1, with higher values indicating better clustering.

$$\mathrm{NMI} = \frac{2I(C,L)}{H(C)+H(L)} \tag{3}$$

where

$$I(C,L) = \sum_i \sum_j P(i,j)\log\frac{P(i,j)}{P(i)P(j)} \tag{4}$$

$$H(C) = -\sum_i P(i)\log P(i) \tag{5}$$

$$H(L) = -\sum_j P(j)\log P(j) \tag{6}$$

$P(i)$, $P(j)$, and $P(i,j)$ represent probability distributions of cluster assignments and ground-truth labels.

Pairwise F-score is calculated using pairwise precision and recall scores obtained by counting pairs, true and false, positive and negative.

$$P_{\mathrm{P}} = \frac{\mathrm{TP}}{\mathrm{TP}+\mathrm{FP}}, \quad R_{\mathrm{P}} = \frac{\mathrm{TP}}{\mathrm{TP}+\mathrm{FN}} \tag{7}$$

On the other hand, precision and recall for BCubed F-score are calculated per sample based only on samples in its cluster and class.

$$P_{\mathrm{B}} = \frac{1}{N}\sum_i \frac{|\mathrm{cluster}(i)\cap\mathrm{class}(i)|}{|\mathrm{cluster}(i)|} \tag{8}$$

$$R_{\mathrm{B}} = \frac{1}{N}\sum_i \frac{|\mathrm{cluster}(i)\cap\mathrm{class}(i)|}{|\mathrm{class}(i)|} \tag{9}$$

The generalizability of face representation models was evaluated using ROC curves of True and False Positive Rates for face verification, where TPR is the fraction of genuine pairs with similarity score above the threshold while FPR is the fraction of imposter pairs with similarity score above the threshold. Face classification performance was judged based on F-scores.

5.2 Face clustering

We evaluated several unsupervised and supervised face clustering algorithms on the TVFace dataset. Faces in each subset were clustered independently and graded using ground truth labels. In unsupervised algorithms, all face images from each subsets were used for clustering. For supervised methods, the classes in each subset were split into two equal-sized train and test sets, in order to ensure that training and testing are performed on disjoint sets of classes. Hyperparameters were selected based on best pairwise and BCubed F-scores.

For unsupervised clustering algorithms, we considered a traditional clustering method DBSCAN [65], along with two recently proposed unsupervised clustering algorithms, Approximate Rank-Order (ARO) [41], and FINCH[43]. ARO is similar to agglomerative clustering but uses a distance measure based on shared nearest neighbors of face images while FINCH performs iterative clustering based on connected components in an adjacency matrix derived only from first neighbor relations. For supervised clustering, we experimented with three Graph Convolutional Network (GCN)-based methods: L-GCN [46], GCN-V [47], and STAR-FC [48]. L-GCN employs graph learning to infer the likelihood of linkage between image pairs in sub-graphs that depict local context, while GCN-V predicts vertex confidence and edge connectivity using GCNs to perform clustering. STAR-FC integrates a structure-preserved subgraph sampling strategy to enable training on very large-scale datasets.

Figure 6 summarizes the aggregate performance of all clustering algorithms on the TVFace dataset. A more detailed analysis is provided in Table 2. ARO and DBSCAN were in tight competition for the best unsupervised algorithm, with ARO gaining a slight edge. GCN-V showed the best performance among supervised algorithms, with L-GCN finishing last. Overall, unsupervised algorithms fared a lot better than supervised algorithms, perhaps due to the relatively small number of training samples, numbering from tens of thousands to over a hundred thousand across different subsets. Another reason could be the unsuitability of TVFace as a training dataset, as evident from experiments in Sect. 5.4.

Table 3 compares the performance of different face clustering algorithms on three large-scale datasets, MS1M [9],

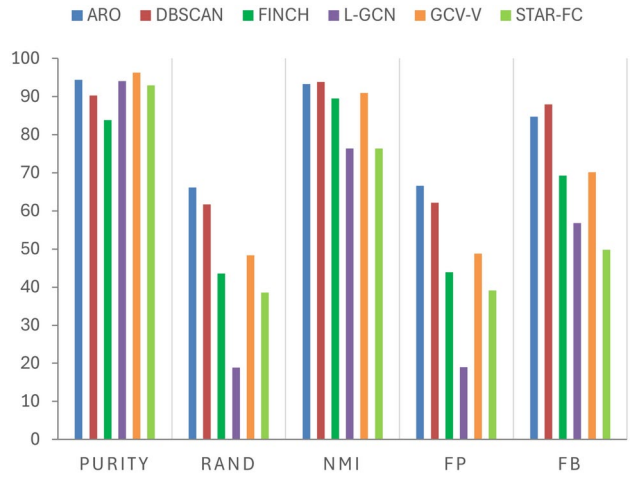


Fig. 6 Comparison of unsupervised (ARO, DBSCAN, FINCH) and supervised (L-GCN, GCN-V, STAR-FC) face clustering algorithms on TVFace dataset

ARO and HAC achieve the best performance on WebFace42 with TVFace being a close second, while the positions are reversed in the case of DBSCAN. MS1M is the most difficult to cluster dataset for all three unsupervised algorithms. However, supervised algorithms L-GCN and GCN-V achieve the highest scores on MS1M while STAR-FC [48] benefits greatly from its large-scale training optimization with excellent scores on WebFace42, the largest of the three datasets. TVFace appears to be the most difficult dataset for supervised clustering techniques, with scores lower than even unsupervised algorithms.

5.3 Online face clustering

We evaluated centroid matching and Hierarchical Retrieval Index for the task of online face clustering on the MS1M [9] dataset, a benchmark for face clustering, and each of the WebFace42 [10], and TVFace. The scores for TVFace are aggregated across all of its subsets except for HAC for which the scores represent the average of only three subsets, ABC News (Aus), Channels TV, and PTV World, due to memory constraints.

Table 4 Evaluation of centroid matching and Hierarchical Retrieval Index for online clustering and their comparison with traditional offline clustering techniques

Method	MS1M			TVFace		
	F_P	F_B	Time	F_P	F_B	Time
ARO [41]	52.8	52.9	26.6 m	66.6	84.7	4.2 h
HAC [60]	54.4	69.5	12.7 h	68.4	81.9	8.4 h
DBSCAN [65]	63.4	66.5	1.7 h	62.3	87.9	4.5 h
Centroid	83.4	82.6	5.6 h	84.9	88.2	3 h
HRI	76.9	75.5	18.8 h	77.5	80.6	5.7 h

subsets of TVFace. Table 4 reports the results of online clustering experiments, alongside a comparison with traditional offline clustering techniques.

Online clustering models outperformed traditional clustering algorithms on both datasets. Centroid-based

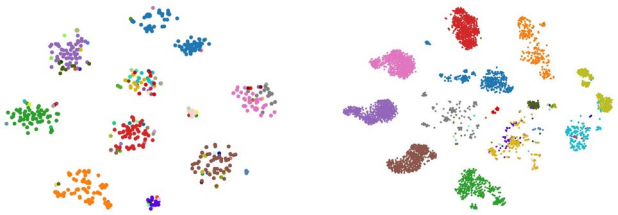
Table 2 Comparison of unsupervised and supervised face clustering algorithms on TVFace dataset

Method	Clusters	Purity (%)	Rand (%)	NMI (%)	F_P	F_B
ARO [41]	130,210	94.4	66.2	93.3	66.6	84.7
DBSCAN [65]	75,039	90.3	61.7	93.9	62.2	88.0
FINCH [43]	24,895	83.9	43.6	89.5	43.9	69.3
L-GCN [46]	530,196	94.1	18.9	76.4	19.0	56.8
GCN-V [47]	27,810	96.3	48.4	91.0	48.8	70.2
STAR-FC [48]	552,521	93.0	38.6	76.4	39.1	49.8

Table 3 Comparison of face clustering algorithms on MS1M, WebFace42, and TVFace datasets

Method	MS1M		WebFace42		TVFace	
	F_P	F_B	F_P	F_B	F_P	F_B
ARO [41]	52.8	52.9	76.9	88.8	66.6	84.7
HAC [60]	54.4	69.5	74.3	85.5	68.4	81.9
DBSCAN [65]	63.4	66.5	60.2	77.9	62.3	87.9
L-GCN [46]	75.8	81.6	–	–	19.1	56.8
GCN-V [47]	83.5	82.6	–	–	48.8	70.2
STAR-FC [48]	88.3	86.3	95.4	94.9	39.1	49.8

The scores on MS1M and WebFace42 were obtained from [48]



(a) MS1M (b) ABC News

Fig. 7 t-SNE visualizations of clustering using Hierarchical Retrieval Index on MS1M and ABC News datasets. Each point is a feature vector representing a face image and is colored according to its predicted cluster label

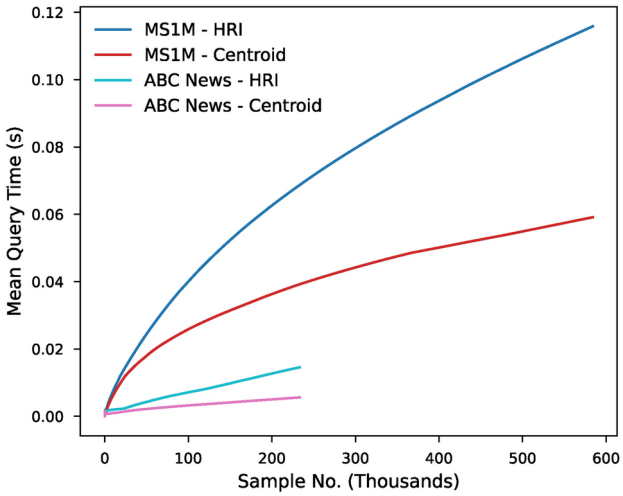


Fig. 8 Query time comparison between centroid and hierarchical matching on MS1M and ABC News datasets

identification was the most accurate technique, almost matching the performance of state-of-the-art supervised clustering models on *part1_test* subset [45] of the MS1M dataset. The Hierarchical Retrieval Index was comparatively less accurate as it is essentially an approximation of centroid matching. The scores on TVFace are higher across the board as clustering was applied on its subsets individually, with each subset containing significantly fewer samples than the MS1M subset. Moreover, the large number of classes and significant intra-class variation in MS1M increase the likelihood of impure clusters. This difference is also apparent from Fig. 7 that shows t-SNE visualizations of ten classes from MS1M and the ABC News subset of TVFace, colored according to predicted cluster labels.

The Hierarchical Retrieval Index is also considerably slower than centroid matching, with twice its mean query time (Fig. 8). Here, query time refers to the time taken to label a new sample and add it to the gallery. In centroid matching, it is only dependent on the search operation since insertion has a constant time complexity. However, the insertion operation in hierarchical index introduces a significant computational overhead to ensure its consistency with new data. Consequently, the index needs further optimization to minimize this overhead if it is to be a viable alternative to centroid matching. That said, HRI does have a better computational complexity than centroid matching, with its average complexity being closer to logarithmic in number of classes. Regardless, at the scale of our experiments, where the number of classes was only a few thousand, this difference in complexity did not have a significant impact.

5.4 Face representation

Although TVFace is primarily a benchmark dataset, we also investigated its potential for training face representation models. This is not a straightforward task, however, as such models require large datasets containing high quality images of a diverse set of faces with varying facial features and minimal noise, whereas TVFace consists of faces captured in unconstrained environments with significant photometric distortions. Moreover, it is split across multiple subsets, one for each television channel, which cannot be simply merged without accounting for class overlap as some individuals may appear on multiple channels. Therefore, we begin with fine-tuning a pretrained representation model on each subset independently to analyze its impact on feature generalizability.

A ResNet34-based ArcFace [58] model pretrained on the CASIA-WebFace [8] dataset was selected as the baseline for these experiments. All subsets were partitioned into training and testing sets in an $80:20$ ratio. The training sets were used to fine-tune the baseline model for 2 epochs at a learning rate of 0.0001, preceded by a warm-up epoch at 0.1 with the backbone frozen. The fine-tuned models were then evaluated for face verification on the LFW [21] and YTF [16] benchmarks.

The ROC curves of 5 fine-tuned models are shown in Fig. 9 alongside the baseline. On LFW, verification accuracy is diminished for very low false positive rates and improved for false positive rates above 0.001, whereas the opposite trend is observed on YTF. Models fine-tuned on the Arirang and Africanews subsets of TVFace perform worse on LFW at all false positive rates due to the differences in the demographic distributions of training and test sets. The contrasting results obtained on image-based LFW and video-based YTF highlight the significance of the domain gap between images and videos.

We also tested the fine-tuned models for face classification on all subsets of the TVFace dataset. A single-layer neural network, with feature vectors as its input, was trained for each subset and used to classify the identities in its test set. Figure 10 shows the F-scores of each subset's classifier when using different face representation models.

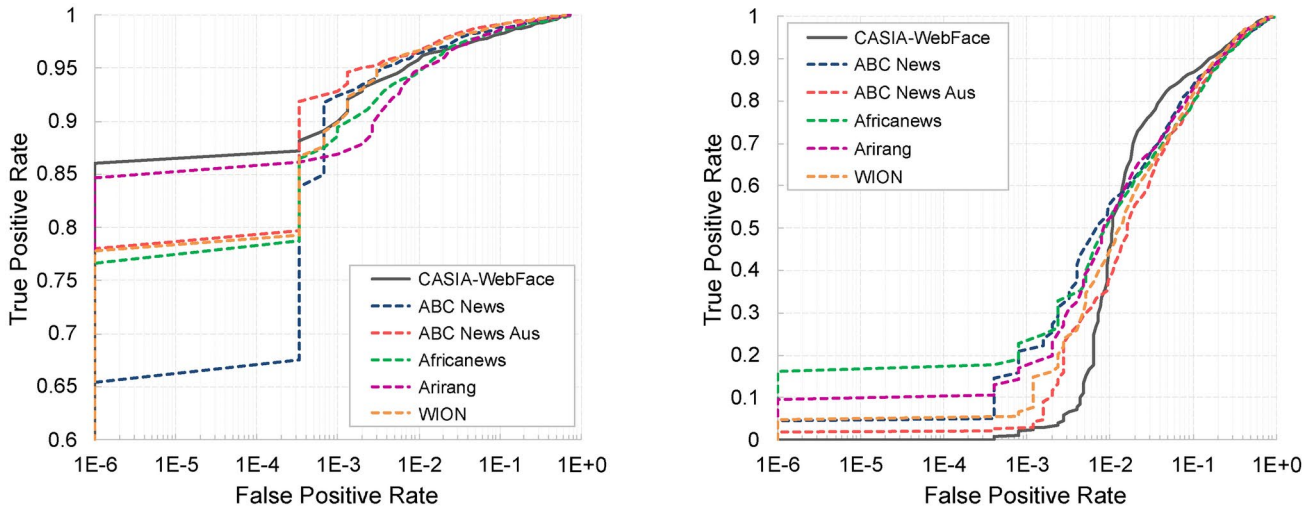


Fig. 9 ROC curves of models fine-tuned on different TVFace subsets, for face verification on LFW and YTF benchmarks

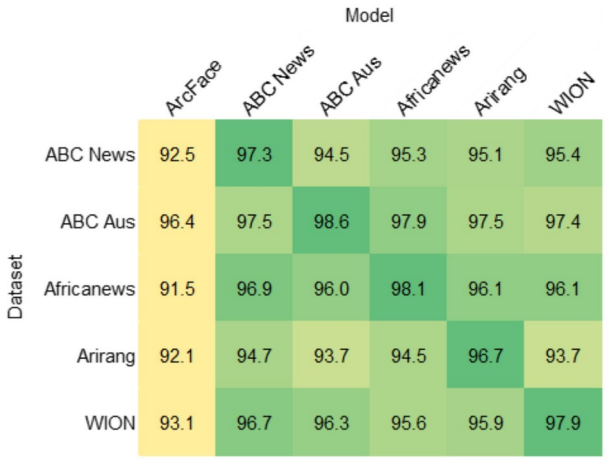


Fig. 10 F-Scores for face classification using representation models trained on different TVFace subsets

Unsurprisingly, the best performing representation model for each subset was the one fine-tuned on it. However, the models appear better suited for feature representation in the video domain even when trained on different subsets.

Additionally, we explored several dataset preprocessing techniques to enhance model training. As TVFace is a long-tailed dataset, random sampling with replacement was employed to acquire 50 images from each class in order to overcome class imbalance. The impact of image quality was evaluated by selecting only large faces that occupied more than 5% of a video frame. Furthermore, we experimented with the exclusion of certain classes that could be deemed unsuitable for face representation learning, e.g. classes with few images or high intra-class similarity. The former was achieved simply by selecting classes with at least 50 images while the latter required computation of intra-class distances based on feature vectors extracted during the annotation process, followed by the removal of classes with mean intra-class cosine distance of 0.1 or less.

The impact of these preprocessing techniques on face verification performance can be observed in Fig. 11 that shows ROC curves for models fine-tuned on ABC News. Balancing classes leads to substantial improvement in true positive rates on the LFW dataset. This performance is further enhanced when only large images or diverse classes are included in the training data, and combining the three techniques results in the best trained model. Excluding infrequent classes roughly approximates this combination since these classes feature background characters represented only by a small number of noisy images. Performance on the YTF dataset, however, is fairly steady for all techniques.

Figure 12 summarizes face classification results of the aforementioned models. The base model trained without any preprocessing achieved the highest F-scores on multiple subsets, though excluding classes with limited intra-class diversity also resulted in comparable performance. The preprocessing techniques do seem to enhance feature generalizability as their scores on subsets other than ABC News are fairly good.

Finally, we combined all subsets of TVFace into a single training dataset containing 2.4 million images and 29,501 classes by merging overlapping classes using hierarchical clustering. The mean feature vectors of all classes were clustered with single linkage at a cosine distance threshold of 0.2, and only the most populous class from each cluster was retained. We then applied all three preprocessing techniques discussed above to generate TVFace+, a dataset

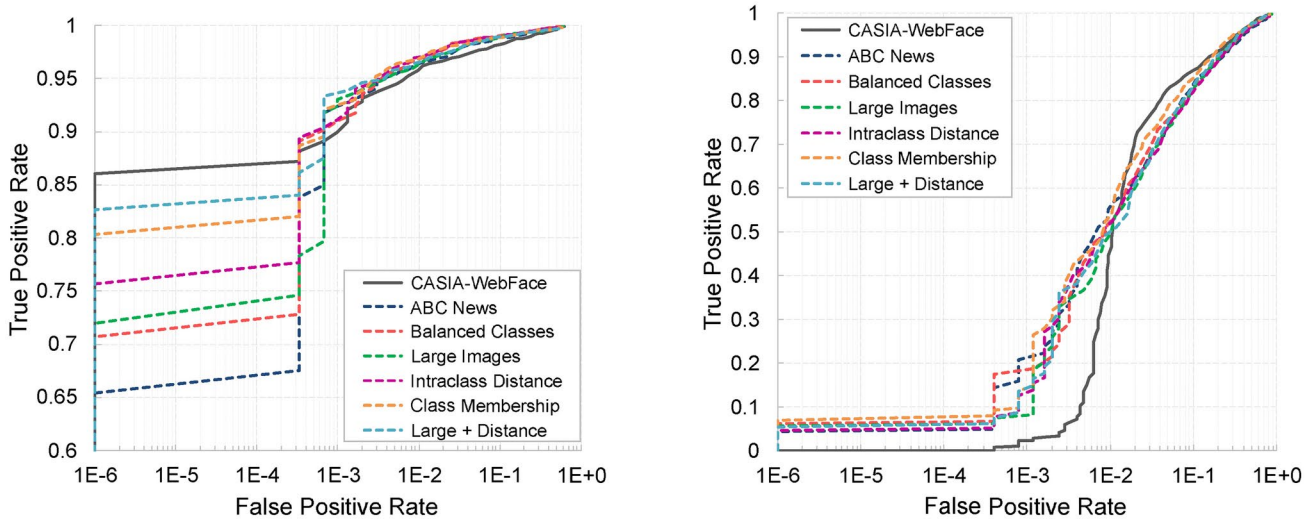


Fig. 11 ROC curves of models fine-tuned on the ABC News subset using different preprocessing techniques, for face verification on LFW and YTF benchmarks

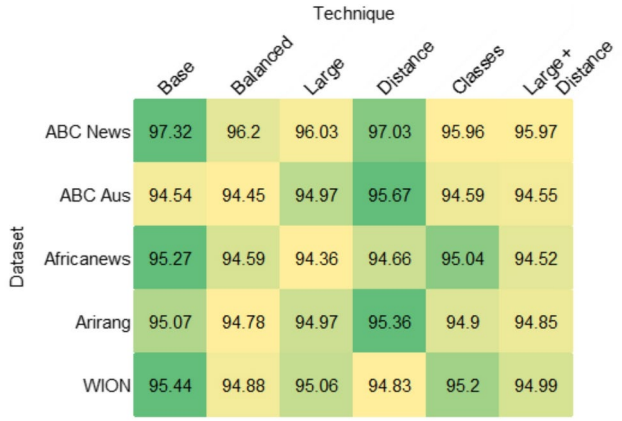


Fig. 12 F-Scores for face classification using representation models trained on ABC News with different preprocessing techniques

comprising $578,750$ images and $11,575$ classes. These datasets were used not only to fine-tune the aforementioned baseline but also to prepare new models from scratch by training a ResNet34 backbone for 5 epochs using ArcFace [58] loss at a learning rate of 0.001.

Figure 13 shows the ROC curves of these models on both face verification benchmarks. Fine-tuning on the complete dataset significantly improves verification accuracy at all false positive rates, with video-based YTF benefiting the most. It is interesting to note that while preprocessing improves fine-tuned performance on LFW, it has a negative impact on YTF. This can be attributed to characteristic differences between the two datasets, with LFW being an image-based dataset with relatively high-quality images, and YTF a video-based dataset with more low-quality images and significant variations in pose, illumination, and expression. The preprocessing techniques appear to remove a large number of challenging images that aid the model in adapting to the video domain, leading to poor results on YTF.

On the other hand, models trained from scratch performed quite poorly in comparison with the baseline. This is possibly because TVFace contains face images captured in unconstrained environments, including low-quality or even occluded and masked faces, whereas representation models are generally trained on high quality web images with clear facial features. Another potential reason is the lack of significant intra-class variations in most classes due to the short data collection period during which people appeared on screen infrequently and their appearances remained largely unchanged. In fact, the main sources of intra class diversity in the dataset are photometric variations, such as illumination and video artifacts, and volatile facial attributes such as expression, pose, hairstyle, and accessories. Consequently, only anchors and prominent international celebrities comprise diverse classes.

Figure 14 presents a comparison of intra-class distance distribution between TVFace and MS1M [9], a commonly used dataset for training face representation models. Feature vectors generated during the annotation process were used to compute intra-class distances in TVFace. For MS1M, we used the feature vectors of *part1_test* subset provided by [45]. Although these feature vectors are not directly comparable, having been generated by different representation models, it is apparent that intra-class diversity in TVFace is fairly limited.

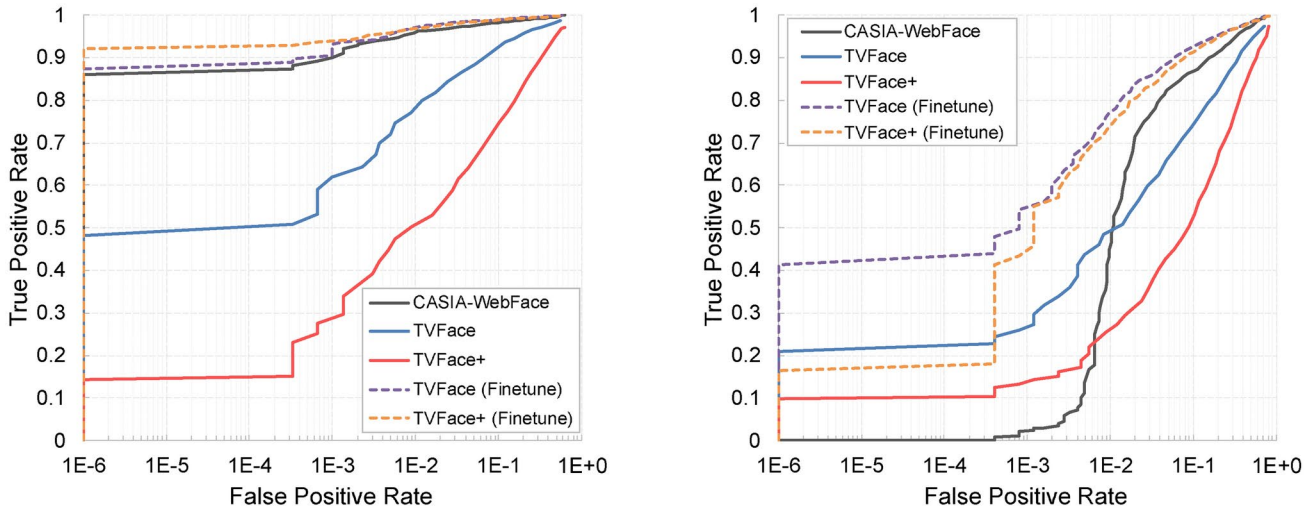


Fig. 13 ROC curves of models trained and fine-tuned on TVFace and TVFace+ datasets, for face verification on LFW and YTF benchmarks

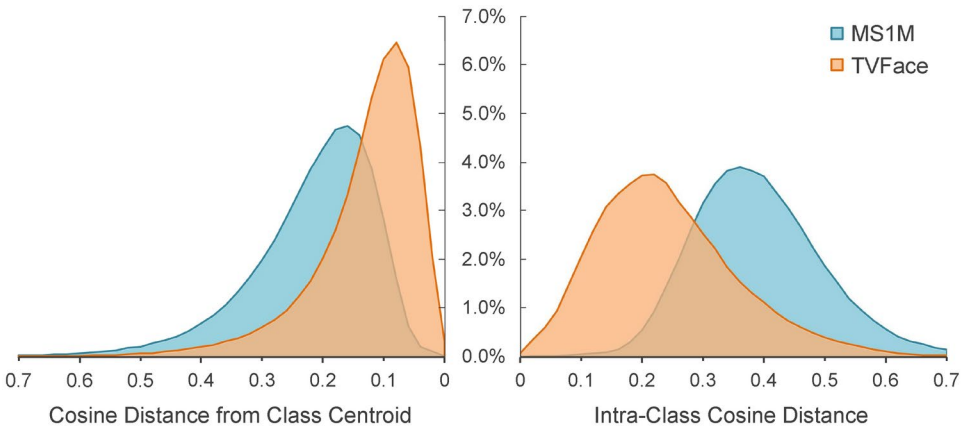


Fig. 14 Comparison of intra-class distance distribution between TVFace and MS1M [9] datasets

Table 5 Intrinsic evaluation of batched hierarchical agglomerative clustering at different merging thresholds τ

Channel	τ	Clusters	Silhouette
News12	0.20	9147	0.24
	0.25	6312	0.36
	0.30	4861	0.45
	0.35	3126	0.41
	0.40	2432	0.28
Euronews	0.20	8943	0.24
	0.25	7116	0.36
	0.30	4952	0.40
	0.35	3480	0.31
	0.40	2145	0.19

5.5 Ablation studies

5.5.1 Identity annotation

The proposed semi-automatic annotation framework includes an unsupervised clustering step that groups together obviously similar images in order to reduce the workload of human annotators. We use hierarchical agglomerative clustering to perform this task of partitioning the collection into an unknown number of clusters. However, the original algorithm is modified to incorporate a batching protocol for scalability. It also requires a threshold hyperparameter that needs to be fine-tuned according to the dataset. We employed intrinsic evaluation, based on Silhouette Coefficient [66], alongside manual verification, to determine the optimal distance threshold. Table 5 shows a comparison of Silhouette scores for two channels at different thresholds.

Table 6 Impact of batching protocol on the performance of hierarchical agglomerative clustering

Channel	Method	Clusters	Purity
ABC News (Aus)	HAC	921	99.7
	HAC-B	729	99.3
Channels TV	HAC	1,260	93.2
	HAC-B	985	87.9
PTV World	HAC	755	99.4
	HAC-B	458	94.7

Table 7 Impact of the merging threshold parameter, τ, on the performance of batched hierarchical agglomerative clustering

Channel	τ	Clusters	Purity	F_P	F_B
Africanews	0.2	2559	98.9	68.4	86.1
	0.3	1308	91.9	86.4	90.1
	0.4	805	76.0	75.0	81.1
Arirang	0.2	5901	99.8	50.9	69.2
	0.3	2104	96.0	77.8	86.2
	0.4	864	79.0	81.5	81.7
Bloomberg	0.2	4601	100	72.6	82.0
	0.3	1982	99.3	87.5	92.8
	0.4	1307	96.3	93.4	94.5

We also evaluated the batching and threshold optimization techniques using ground truth labels from the annotated dataset. The original and batched variants of agglomerative clustering were used to cluster several subsets of the TVFace dataset at different values of the threshold hyperparameter. The best threshold was selected for each method based on pairwise and BCubed F-scores. A comparison of the best partitions generated by both algorithms is shown in Table 6. Across the three subsets, HAC predicts more clusters than HAC-B, leading to higher cluster purity. The most probable cause of this behaviour is the global clustering operation in the batching protocol where clusters from different batches are combined based on their centroids.

The choice of distance threshold was evaluated by applying the batched hierarchical agglomerative clustering algorithm at different thresholds on several subsets of the TVFace dataset. Table 7 shows the results of this experiment for three subsets. It is apparent that cluster purity is highest in the $0.2-0.3$ range, although such strict thresholds also increase the number of clusters, causing pairwise and BCubed F-scores to suffer.

5.5.2 Hierarchical retrieval index

The impact of certain design choices on clustering performance of the Hierarchical Retrieval Index also merits closer inspection. In our proposed methodology, the value of an internal node is calculated as the mean of its descendant leaf nodes, since these nodes represent actual faces. An alternative is to calculate this value as the unweighted mean of the node's children. The choice of distance metric for computing feature similarity may also be an important factor to consider for the model.

Table 8 Evaluation of design choices in Hierarchical Retrieval Index for online clustering on ABC News (Aus) dataset

Mean	Distance	F_P	F_B	Time
Children	Cosine	80.6	85.3	42 s
	Euclidean	85.4	84.6	35 s
Leaf Nodes	Cosine	92.3	91.6	75 s
	Euclidean	92.5	91.8	62 s

The results of these ablation studies are reported in Table 8. Using the feature vectors of all descendant leaf nodes for calculating internal node values produces better results than using only the feature vectors of immediate children, as the internal nodes are more representative of their descendants in the former case. However, it also makes the index more susceptible to unbalanced distribution of vectors in the feature space, leading to slightly reduced efficiency. Clustering performance is also unaffected by the choice of distance metric in the case of leaf node-based mean calculations, although euclidean distance leads to slightly faster computation time.

Figure 15 visualizes the depths of all leaf nodes in the complete index under different averaging strategies and distance metrics. The color gradient of sample labels suggests that the children-based index readily generates new clusters for new samples. This is because the mean of a node's children does not adequately represent its descendants, leading to inaccurate search operations. As the nearest neighbors of a new feature vector are not found, it has little chance of being added to an existing cluster and instead forms a new one. In contrast, search operations in leaf nodes-based index are fairly accurate, allowing samples to find the correct clusters regardless of insertion order.

6 Discussion

6.1 Limitations of content-based identity annotation

Identity annotation is the main challenge in the preparation of face datasets. Web image-based datasets are prevalent because metadata from search engines and databases can be used to derive fairly accurate identity labels. In contrast, videos offer too little information for completely automatic annotation. Weak supervision is available in the form of temporal constraints but these can only be utilized to define similar and dissimilar groups of face images. Movies and serials can benefit from scripts and lists of cast members but the grouping of faces must still be performed using existing face recognition systems. In long-form video content, automatic annotation is limited to content-based

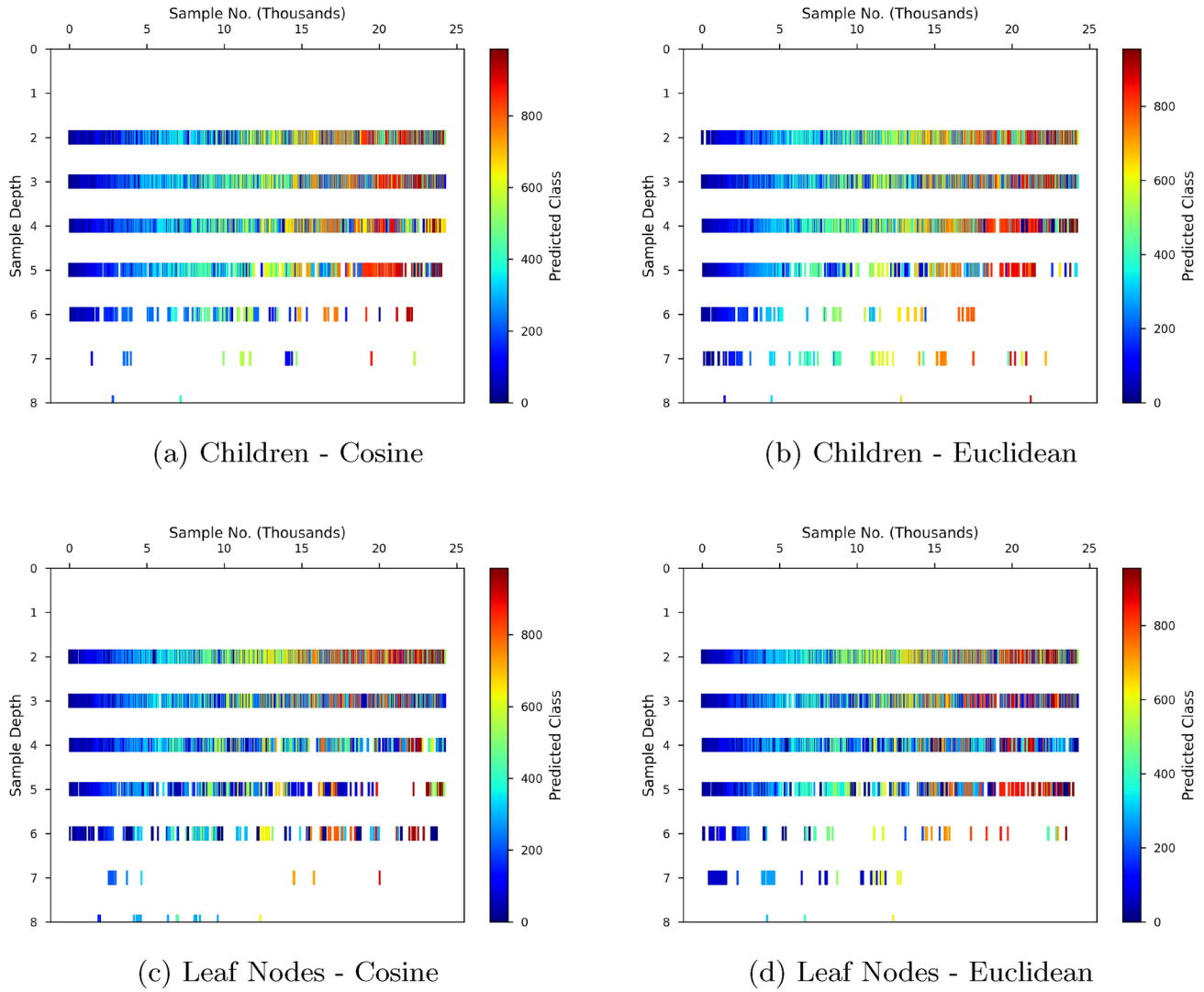
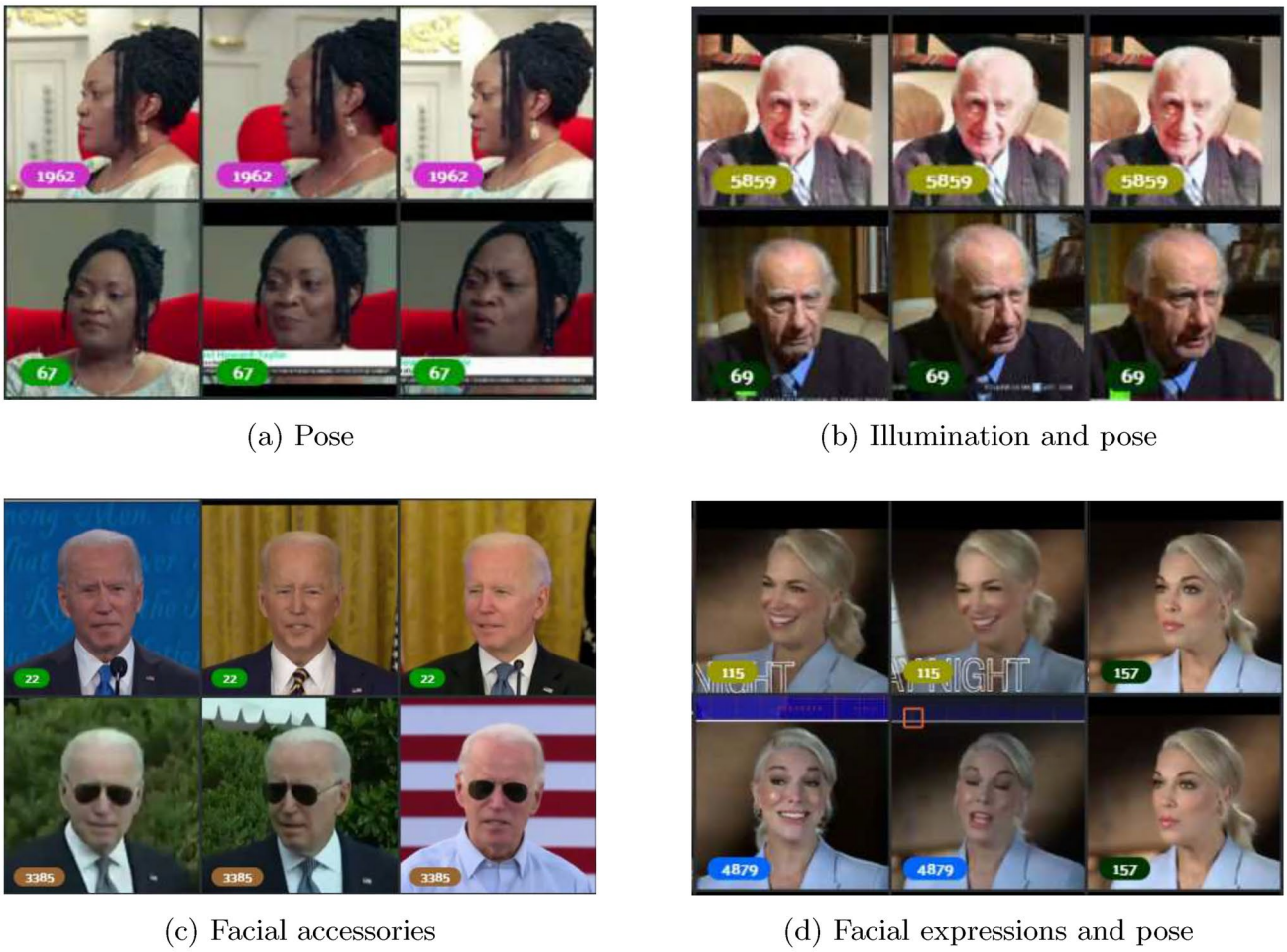


Fig. 15 Depths distribution of leaf nodes in the complete index using different averaging strategies and distance metrics

retrieval techniques, such as the clustering of feature vectors extracted from face images, as there is very little information about the identities of people appearing in the video streams. This can lead to biases towards retrieval techniques used for annotation.

For example, our identity annotation framework uses batched hierarchical agglomerative clustering (HAC-B) to partition the dataset into small, pure clusters in order to facilitate manual annotation. Even though this step is followed up by a cleaning process where clusters of the same individual are merged and impure clusters are separated, HAC-B achieves the highest test scores for face clustering on TVFace. This is because impure clusters contain samples that HAC-B cannot group accurately and removing these samples from the dataset makes it easier for the algorithm to cluster the remaining data points. It also allows for lenient thresholding where multiple clusters belonging to the same individual can be merged without the interference of difficult samples, further increasing clustering accuracy. In fact, all algorithms used in the automatic annotation process, from feature representation and clustering models to hyperparameter optimization protocol, introduce biases in the final labeling that can only be minimized by manual cleaning. Therefore, it is important to be cognizant of these biases when using a dataset for the development of face recognition systems.

It is also pertinent to note that the proposed manual cluster merging process is not exhaustive, since the human annotator is only shown cluster pairs that are similar enough so as to merit closer inspection. This similarity is calculated based on feature vectors and is, thus, subject to the same inaccuracies as any other face recognition system. We attempt to rectify this situation by performing the manual annotation process iteratively, but this can only yield a better



(a) Pose (b) Illumination and pose

(c) Facial accessories (d) Facial expressions and pose

Fig. 16 Qualitative evaluation of inaccurate face clustering due to variations in non-discriminatory facial features. Numbers at bottom left denote cluster IDs

approximation. Consequently, it is possible that the dataset contains multiple classes corresponding the same individual if their mean feature vectors are sufficiently distant.

6.2 Balancing diversity in a composite dataset

TVFace is a composite dataset consisting of 22 subsets that have each been annotated independently with the assumption that the number of individuals appearing in multiple subsets is fairly small and can generally be detected using existing face recognition systems. Each subset corresponds to a television channel and is thus unique in terms of its photometric properties and demographic distributions. This allows the developers of face recognition systems to modify the photometric properties and demographic distributions of the dataset by combining different subsets. For example, the performance of a recognition system on ABC News, Africanews, and CNA subsets can be compared to identify biases towards particular ethnicities.

6.3 Insights into deep feature representations

In applying the face recognition pipeline for identity labeling of the TVFace dataset and the following manual annotation phase, we observe several patterns that provide valuable insights into the workings and limitations of face representations models and may prove useful for development of novel methods to overcome these challenges.

Deep feature representations are trained to jointly model all facial features, including non-discriminatory features, such as facial expression and pose. Consequently, variations in non-discriminatory features can lead to significant changes in the feature vector, resulting in false predictions if the vector crosses class boundaries. Figure 16 shows some examples of faces grouped into different clusters due to changes in illumination, pose, expressions and facial accessories. These false negatives can be reduced by loosening the threshold for class boundaries but that would also increase false positives and face representation systems need to be improved to ease this trade-off.

Another interesting observation is that feature vectors of noisy samples, images containing blurry or occluded faces, tend to be concentrated in a few regions of the feature space. In fact, when the threshold for face clustering is slackened, the output often includes a few mega-clusters containing most of the noisy samples. This is perhaps because these images do not have well-defined facial features, resulting in the domination of non-discriminative features and causing them to be grouped together during face clustering.

The feature space also appears to be partitioned based on ethnic attributes, like skin color, as feature vectors belonging to people from the same ethnicities tend to cluster together. However, this partition is quite biased towards the Caucasian ethnicity, with the average inter and intra-cluster distances in American and European channels being greater than those in African or East Asian channels. This allows for the detection of subtle differences in facial features of Caucasian faces whereas ethnic features tend to dominate in faces of other ethnicities. We also had to use tighter distance thresholds for automatic clustering during annotation for non-Western channels. Even face attribute models appear to be biased towards Caucasian faces as their prediction confidences were considerably lower for faces from non-Western channels.

7 Conclusion

The objective of this study was to renew interest in and promote the development of face representation and identification models capable of large-scale, unsupervised face recognition in videos. As such, we proposed a large-scale dataset of face images extracted from news channels in order to more accurately represent the challenges of the video domain. The TVFace dataset is suitable for evaluating feature representation and matching techniques on the tasks of face identification and clustering. We used a semi-automatic annotation framework based on unsupervised clustering to facilitate manual identity labeling of face images in the dataset. We also designed a Hierarchical Retrieval Index to demonstrate the effectiveness of the proposed dataset in evaluating online face clustering, which is the most realistic scenario for a large-scale, unsupervised person retrieval system. We hope that the proposed dataset can prove a valuable resource for future research in this area.

A natural extension of this study is the development of a video face dataset for training face recognition models. This dataset may contain only high-quality images of frequently appearing individuals, and the list of channels can be extended to compensate for the reduction in number of classes. Another avenue for future work is to generate manual annotations for facial attributes such as masks, expressions, and pose, which can provide a more comprehensive understanding of facial features and allow for the development of robust recognition and analysis systems. A masked face dataset, as well as a face expressions dataset, could be particularly useful in this regard as there are quite a few masked faces in the current dataset and videos often feature a wide distribution of facial expressions.

Funding We acknowledge the support from the Islamic World Educational, Scientific, and Cultural Organization (ICESCO) by establishing the ICESCO Chair of Data Science and Analytics for Business at the National University of Sciences and Technology (NUST), Islamabad, Pakistan.

Data availability The data supporting the findings of this study are available from the corresponding author upon reasonable request.

Declarations

Conflict of interest The authors have no relevant financial or non-financial interests to disclose.

Ethical approval We have considered potential ethical concerns in the selection of television channels as data sources and taken measures to ensure privacy and responsible data use. Notably, the dataset does not contain any personally identifiable information about the individuals included. Each person is labeled only with an anonymous identifier within the dataset, without any association with their name or other personal details. Additionally, since the dataset is constructed from publicly available broadcasts, it does not involve private or sensitive information that would require an ethical review or approval. We have also implemented strict measures to prevent direct downloads of the dataset. Access is only granted through a request process, requiring researchers to provide information about their intended use. Each request undergoes a review to ensure that the dataset is used strictly for academic and research purposes.

References

1. Shi Y, Otto C, Jain AK (2018) Face clustering: representation and pairwise constraints. IEEE Trans Inf Forensics Secur 13:1626–1640
2. Wang Y, Shen J, Petridis S, Pantic M (2019) A real-time and unsupervised face re-identification system for human-robot interaction. Pattern Recogn Lett 128:559–68
3. Zhang L, Kalashnikov DV, Mehrotra S (2013) A Unified Framework for Context Assisted Face Clustering. In: ACM International Conference on Multimedia Retrieval, pp. 9–16
4. Malach T, Pomenkova J (2020) Optimal face templates: the next step in surveillance face recognition. Pattern Anal Appl 23(2):1021–1032
5. Otto C, Klare B, Jain AK (2015) An efficient approach for clustering face images. In: International Conference on Biometrics, pp. 1–10
6. Hong J, Crichton W, Zhang H, Fu DY, Ritchie J, Barenholtz J, Hannel B, Yao X, Murray M, Moriba G, Agrawala M, Fatahalian K (2021) Analysis of faces in a decade of US cable TV news. In: ACM SIGKDD Conference on Knowledge Discovery and Data Mining, pp. 3011–3021

7. Wang D, Otto C, Jain AK (2017) Face Search at Scale. IEEE Trans Pattern Anal Mach Intell 39:1122–1136
8. Yi D, Lei Z, Liao S, Li SZ (2014) Learning face representation from scratch. arXiv preprint arXiv:1411.7923, 1–9
9. Guo Y, Zhang L, Hu Y, He X, Gao J (2016) MS-Celeb-1M: A dataset and benchmark for large-scale face recognition. In: European Conference on Computer Vision, pp. 87–102
10. Zhu Z, Huang G, Deng J, Ye Y, Huang J, Chen X, Zhu J, Yang T, Lu J, Du D, Zhou J (2021) WebFace260M: a benchmark unveiling the power of million-scale deep face recognition. In: Computer Vision and Pattern Recognition, pp. 10492–10502
11. Zhang Z, Luo P, Loy CC, Tang X (2016) Joint face representation adaptation and clustering in videos. In: European Conference on Computer Vision, pp. 1–10
12. Jin S, Su H, Stauffer C, Learned-Miller E (2017) End-to-end face detection and cast grouping in movies using erdös-rényi clustering. In: International Conference on Computer Vision, pp. 1–10
13. Sharma V, Tapaswi M, Sarfraz MS, Stiefelhagen R (2019) Self-supervised learning of face representations for video face clustering. In: International Conference on Automatic Face and Gesture Recognition, pp. 1–8
14. Tapaswi M, Law M, Fidler S (2019) Video Face clustering with unknown number of clusters. In: International Conference on Computer Vision, pp. 5026–5035
15. Sivic J, Everingham M, Zisserman A (2009) "Who are you?" - Learning person specific classifiers from video. In: Computer Vision and Pattern Recognition, pp. 1145–1152
16. Wolf L, Hassner T, Maoz I (2011) Face recognition in unconstrained videos with matched background similarity. In: Computer Vision and Pattern Recognition, pp. 529–534
17. Bauml M, Tapaswi M, Stiefelhagen R (2013) Semi-supervised Learning with Constraints for Person Identification in Multimedia Data. In: Computer Vision and Pattern Recognition, pp. 1–8
18. Ghaleb E, Tapaswi M, Al-Halah Z, Ekenel HK, Stiefelhagen R (2015) Accio: A data set for face track retrieval in movies across age. In: ACM International Conference on Multimedia Retrieval, pp. 1–10
19. Nech A, Kemelmacher-Shlizerman I (2017) Level playing field for million scale face recognition. In: Computer Vision and Pattern Recognition, pp. 3406–3415
20. Cao Q, Shen L, Xie W, Parkhi OM, Zisserman A (2018) VGGFace2: A dataset for recognising faces across pose and age. In: International Conference on Automatic Face and Gesture Recognition, pp. 67–74
21. Huang GB, Mattar M, Berg T, Learned-Miller E (2008) Labeled faces in the wild: a database for studying face recognition in unconstrained environments. In: Workshop on Faces in 'Real-Life' Images: Detection, Alignment, and Recognition, Marseille, France, pp. 1–14
22. Klare BF, Klein B, Taborsky E, Blanton A, Cheney J, Allen K, Grother P, Mah A, Burge M, Jain AK (2015) Pushing the frontiers of unconstrained face detection and recognition: IARPA Janus Benchmark A. In: Computer Vision and Pattern Recognition, pp. 1931–1939
23. Whitelam C, Taborsky E, Blanton A, Maze B, Adams J, Miller T, Kalka N, Jain AK, Duncan JA, Allen K, Cheney J, Grother P (2017) IARPA Janus Benchmark-B face dataset. In: Computer Vision and Pattern Recognition Workshops, pp. 592–600
24. Maze B, Adams J, Duncan JA, Kalka N, Miller T, Otto C, Jain AK, Niggel WT, Anderson J, Cheney J, Grother P (2018) IARPA Janus Benchmark - C: face dataset and protocol. In: International Conference on Biometrics, pp. 158–165
25. Somandepalli K, Hebbar R, Narayanan S (2022) Robust character labeling in movie videos: data resources and self-supervised feature adaptation. IEEE Trans Multimed 24:3355–3368
26. Liu Y, Peng B, Shi P, Yan H, Zhou Y, Han B, Zheng Y, Lin C, Jiang J, Fan Y, Gao T, Wang G, Liu J, Lu X, Xie D (2018) iQIYI-VID: A large dataset for multi-modal person identification. arXiv preprint arXiv:1811.07548, 1–11
27. Guillaumin M, Verbeek J, Schmid C (2009) Is that you? Metric learning approaches for face identification. In: International Conference on Computer Vision, pp. 498–505
28. Terhorst P, Kolf N, Huber M, Kirchbuchner F, Damer N, Morales A, Fierrez J, Kuijper A (2021) A comprehensive study on face recognition biases beyond demographics. IEEE Transactions on Technology and Society 3, 16–30
29. Zhao J, Cheng Y, Xu Y, Xiong L, Li J, Zhao F, Jayashree K, Pranata S, Shen S, Xing J, Yan S, Feng J (2018) Towards pose invariant face recognition in the wild. In: Computer Vision and Pattern Recognition, pp. 2207–2216
30. He M, Zhang J, Shan S, Kan M, Chen X (2020) Deformable face net for pose invariant face recognition. Pattern Recogn 100:107–113
31. Huang Z, Zhang J, Shan H (2021) When age-invariant face recognition meets face age synthesis: a multi-task learning framework. In: Computer Vision and Pattern Recognition, pp. 7278–7287
32. Robinson JP, Livitz G, Henon Y, Qin C, Fu Y, Timoner S (2020) Face recognition: too bias, or not too bias? In: Computer Vision and Pattern Recognition, pp. 1–10
33. Zhao J, Yan S, Feng J (2022) Towards Age-Invariant Face Recognition. IEEE Trans Pattern Anal Mach Intell 44:474–487
34. Nawshad MA, Saadat A, Fraz MM (2023) Boosting facial recognition capability for faces wearing masks using attention augmented residual model with quadruplet loss. Mach Vis Appl 34(6):108
35. Nawshad MA, Fraz MM (2023) Improving masked face recognition using dense residual unit aided with quadruplet loss. In: Image and Vision Computing, pp. 345–360
36. Liu Y, Chen J, Li Y, Wu T, Wen H (2024) Joint face normalization and representation learning for face recognition. Pattern Anal Appl 27(2):1–15
37. Karkkainen K, Joo J (2021) FairFace: Face attribute dataset for balanced race, gender, and age for bias measurement and mitigation. In: Winter Conference on Applications of Computer Vision, pp. 1547–1557
38. Zheng T, Deng W, Hu J (2017) Cross-Age LFW: a database for studying cross-age face recognition in unconstrained environments. arXiv preprint arXiv:1708.08197, 1–10
39. Zheng T, Deng W (2018) Cross-pose LFW: A database for studying cross-pose face recognition in unconstrained environments. Technical report, Beijing University of Posts and Telecommunications
40. Zhao M, Teo YW, Liu S, Chua T-S, Jain R (2006) Automatic person annotation of family photo album. In: International Conference on Image and Video Retrieval, pp. 163–172
41. Otto C, Wang D, Jain AK (2018) Clustering Millions of Faces by Identity. IEEE Trans Pattern Anal Mach Intell 40:289–303
42. Lin W-A, Chen -C, Castillo CD, Chellappa R (2018) Deep density clustering of unconstrained faces. In: Computer Vision and Pattern Recognition, pp. 8128–8137
43. Sarfraz S, Sharma V, Stiefelhagen R (2019) Efficient parameter-free clustering using first neighbor relations. In: Computer Vision and Pattern Recognition, pp. 8926–8935
44. Tan SK, Wang X (2024) A novel two-stage omni-supervised face clustering algorithm. Pattern Anal Appl 27(3):83
45. Yang L, Zhan X, Chen D, Yan J, Loy CC, Lin D (2019) Learning to cluster faces on an affinity graph. In: Computer Vision and Pattern Recognition, pp. 2293–2301
46. Wang Z, Zheng L, Li Y, Wang S (2019) Linkage based face clustering via graph convolution network. In: Computer Vision and Pattern Recognition, pp. 1117–1125

47. Yang L, Chen D, Zhan X, Zhao R, Loy CC, Lin D (2020) Learning to cluster faces via confidence and connectivity estimation. In: Computer Vision and Pattern Recognition, pp. 13366–13375
48. Shen S, Li W, Zhu Z, Huang G, Du D, Lu J, Zhou J (2021) Structure-aware face clustering on a large-scale graph with 10^7 nodes. In: Computer Vision and Pattern Recognition, pp. 9085–9094
49. Kipf TN, Welling M (2017) Semi-supervised classification with graph convolutional networks. In: International Conference on Learning Representations, pp. 1–10
50. Nguyen X-B, Bui DT, Duong CN, Bui TD, Luu K (2021) Clusformer: A transformer based clustering approach to unsupervised large-scale face and visual landmark recognition. In: Computer Vision and Pattern Recognition, pp. 10847–10856
51. Cui J, Wen F, Xiao R, Tian Y, Tang X (2007) EasyAlbum: An interactive photo annotation system based on face clustering and re-ranking. In: SIGCHI Conference on Human Factors in Computing Systems, pp. 367–376
52. Tian Y, Liu W, Xiao R, Wen F, Tang X (2007) A face annotation framework with partial clustering and interactive labeling. In: Computer Vision and Pattern Recognition, pp. 1–8
53. Wang M, Deng W (2021) Deep face recognition: a survey. Neurocomputing 429:215–244
54. Wu Z, Ke Q, Sun J, Shum HY (2011) Scalable face image retrieval with identity-based quantization and multireference reranking. IEEE Trans Pattern Anal Mach Intell 33(10):1991–2001
55. Wang D, Hoi SCH, He Y, Zhu J, Mei T, Luo J (2014) Retrieval-based face annotation by weak label regularized local coordinate coding. IEEE Trans Pattern Anal Mach Intell 36(3):550–563
56. Dong X, Kim S, Jin Z, Hwang JY, Cho S, Teoh ABJ (2020) Open-set face identification with index-of-max hashing by learning. Pattern Recogn 103:107277
57. Deng J, Guo J, Ververas E, Kotsia I, Zafeiriou S (2020) Retinaface: single-shot multi-level face localisation in the wild. In: Computer Vision and Pattern Recognition, pp. 5202–5211
58. Deng J, Guo J, Xue N, Zafeiriou S (2019) ArcFace: additive angular margin loss for deep face recognition. In: Computer Vision and Pattern Recognition, pp. 4685–4694
59. Serengil SI, Ozpinar A (2020) LightFace: A hybrid deep face recognition framework. In: Innovations in Intelligent Systems and Applications Conference, pp. 1–5
60. Defays D (1977) An efficient algorithm for a complete link method. Comput J 20(4):364–366
61. Rousseeuw PJ (1987) Silhouettes: A graphical aid to the interpretation and validation of cluster analysis. J Comput Appl Math 20:53–65
62. Zhou Y, Gregson J (2020) Whenet: Real-time fine-grained estimation for wide range head pose. In: British Machine Vision Conference, pp. 1–17
63. Deb C (2022) Face mask detection. GitHub. https://github.com/chandrikadeb7/Face-Mask-Detection
64. Amigó E, Gonzalo J, Artiles J, Verdejo F (2009) A comparison of extrinsic clustering evaluation metrics based on formal constraints. Inf Retr 12(4):461–486
65. Ester M, Kriegel H-P, Sander J, Xu X (1996) A density-based algorithm for discovering clusters in large spatial databases with noise. In: International Conference on Knowledge Discovery and Data Mining, pp. 226–231
66. Rousseeuw PJ (1987) Silhouettes: a graphical aid to the interpretation and validation of cluster analysis. J Comput Appl Math 20:53–65

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Springer Nature or its licensor (e.g. a society or other partner) holds exclusive rights to this article under a publishing agreement with the author(s) or other rightsholder(s); author self-archiving of the accepted manuscript version of this article is solely governed by the terms of such publishing agreement and applicable law.