Unsupervised learning of co-occurrences for face images retrieval

Thomas Petit, Pierre Letessier, Stefan Duffner, Christophe Garcia

Unsupervised learning of co-occurrences for face images retrieval

Thomas Petit, Pierre Letessier, Stefan Duffner, Christophe Garcia

Unsupervised learning of co-occurrences for face images retrieval

Thomas PETIT
tpetit@ina.fr
Institut National de l'Audiovisuel
Univ Lyon, INSA Lyon, LIRIS (UMR 5202 CNRS)
Bry-sur-Marne, France

Pierre LETESSIER
pletessier@ina.fr
Institut National de l'Audiovisuel
Bry-sur-Marne, France

Stefan DUFFNER
stefan.duffner@liris.cnrs.fr
Univ Lyon, INSA Lyon, LIRIS (UMR 5202 CNRS)
Villeurbanne, France

Christophe GARCIA
christophe.garcia@liris.cnrs.fr
Univ Lyon, INSA Lyon, LIRIS (UMR 5202 CNRS)
Villeurbanne, France

ABSTRACT

Despite a huge leap in performance of face recognition systems in recent years, some cases remain challenging for them while being trivial for humans. This is because a human brain is exploiting much more information than the face appearance to identify a person. In this work, we aim at capturing the social context of unlabeled observed faces in order to improve face retrieval. In particular, we propose a framework that substantially improves face retrieval by exploiting the faces occurring simultaneously in a query's context to infer a multi-dimensional social context descriptor. Combining this compact structural descriptor with the individual visual face features in a common feature vector considerably increases the correct face retrieval rate and allows to disambiguate a large proportion of query results of different persons that are barely distinguishable visually.

To evaluate our framework, we also introduce a new large dataset of faces of French TV personalities organised in TV shows in order to capture the co-occurrence relations between people. On this dataset, our framework is able to improve the mean Average Precision over a set of internal queries from 67.93% (using only facial features extracted with a state-of-the-art pre-trained model) to 78.16% (using both facial features and faces co-occurrences), and from 67.88% to 77.36% over a set of external queries.

CCS CONCEPTS

• **Information systems** → **Image search**; *Document structure.*

KEYWORDS

person recognition, facial recognition, retrieval, co-occurrences

ACM Reference Format:
Thomas PETIT, Pierre LETESSIER, Stefan DUFFNER, and Christophe GARCIA. 2021. Unsupervised learning of co-occurrences for face images retrieval. In *ACM Multimedia Asia (MMAsia '20), March 7–9, 2021, Virtual Event, Singapore.* ACM, New York, NY, USA, 7 pages. https://doi.org/10.1145/3444685.3446265

1 INTRODUCTION

Following the recent advances in deep learning, facial recognition techniques have improved significantly during the last decade [6, 14, 17, 19]. The most advanced ones can nowadays compete with humans on tasks such as person verification or identification. On the well-known face verification benchmark LFW [9], a score of 99.6% has been reached [17]. However, these state-of-the-art methods are still prone to many mistakes that no human would make; this is because a human brain does not only focus on facial features to recognize a human face, but is using also contextual information. For example, humans have been able to obtain an accuracy of 94.27% on the LFW protocol, with all faces masked in the original images [12]. A very powerful contextual information is the co-occurrences of people appearing together: on TV, for example, it will always be easier to identify a member of a music band if you see them alongside the other ones, than alone. The same applies to many local politicians, football players, actors…

Several approaches have been proposed in order to improve facial recognition tasks, such as classification, verification, or retrieval, using additional contextual information. The nature of this contextual information is diverse: it can be a visual context, including the whole body and clothes of the person of interest, a social context from social media metadata, or categorical data. However, to our knowledge, no attempt has been made at inferring the social relationship of persons occurring in a large database in an unsupervised manner, and using it to improve the retrieval of persons in that database.

Our goal is as follows: for one face, observed in a TV show alongside other people, we want to retrieve all other instances of the same person in a large dataset of TV show.

We introduce an unsupervised method that provides a contextual embedding for groups of faces appearing together, based on their estimated co-occurrence relationships with other faces in the dataset. It can be divided in three main steps:

1) We use a soft clustering on face descriptors to approximate the ground truth entities with clusters of faces.
2) We build a probabilistic co-occurrence matrix to map the clusters to a contextual feature space capturing their co-occurrence relationships.
3) We fuse the contextual embeddings of all faces observed together into a common embedding that best represents them all.
We experimentally show that this method can be used to substantially improve face retrieval by merging facial feature vectors with their corresponding contextual embedding. We evaluate our approach on a dataset of 548,686 faces, organized under 138,381 TV shows containing the faces of people appearing together, in order to naturally embed their social relationships. We show that our method can increase the mean Average Precision obtained when retrieving a set of internal queries in our dataset from 67.93%, when using only facial features extracted with a pre-trained model, to 78.16%. We also show that our method reaches similar performance on a set of external queries.

2 RELATED WORK

2.1 Social context for person recognition

Using contextual information in order to improve person recognition is something that has been widely studied in the past. Some are focusing on additional visual information, such as clothes or specific body parts [13, 21].
The approach proposed by Stone et al. [18] uses the social relationships between users of a social media to recognize them. To do this, they define potential functions based on the connections of the users in the social media network, and the co-occurrences of people in already annotated photos. All of this information is directly available from the social media. Unfortunately, this cannot be extended to different situation where social relationships between subjects is completely unknown.
Coelho de Castro and Nowrozin [5] introduced a theoretical Bayesian model for person identification, in which the "context" is defined as a discrete latent variable. This latent variable is drawn from a probability that follows a Dirichlet prior, to allow for new context to appear. Since all persons observed together are sharing the same "context", this latent variable is supposed to embed the co-occurrence relations between people.
In the work from Huang et al. [10], both a visual context and a social context are used for person recognition, in photo albums or movies dataset. The visual context consists in different regions of the persons, like their upper body or whole body, while the social context is divided between the person-to-person relationships (i.e. person co-occurrences), and the event-to-person relationships, which aims at identifying events in the photo albums or movies where each person is expected to appear. However, this method is applied to classifying persons from a query set, based on a gallery set from which those relationships are learned. This assumes that we have at our disposal an already annotated subset from which we can learn a social context. This is not the case for our retrieval task in a non-annotated dataset.

2.2 Existing datasets of face images

The improvement of deep learning models for facial recognition has been made possible by an increase in both the size and the quality of the available datasets [1, 4, 7, 15]. Amongst the most recent ones, the VGGFace2 dataset [4] contains 3.3M images, divided in 9,000 unique identities. Other datasets have been built specifically in order to address the issue of faces observed under unconstrained settings, and to take advantage of some contextual information. The *People in Photo Albums* (PIPA) dataset [21] contains 37K images in which appear 63K instances of 2,356 unique identities. This is to date one of the most important dataset of people co-occuring together in different contexts; however the number of instances and unique people is quite low in comparison to what can be seen on TV over a few years. We can also mention the *Celebrity Together* dataset [22] which contains 194K images where 2,622 labeled celebrities appear together. However, there is no evidence that the co-occurrences of the celebrities in this dataset are representative of their social context.

3 DATASET

3.1 Motivation

Our motivation is to be able to retrieve faces in a large dataset of TV shows by exploiting both the visual features of the faces observed in them and various contextual information. However, evaluating a model on this task requires to build a ground truth, which would be a really tedious task on a large dataset of videos. We instead decided to build a new dataset of face images, organized to reflect the co-occurrences of faces observed under real conditions. This is the dataset we use to evaluate our method.

3.2 Dataset structure

We introduce a new dataset[1] of face images scraped with a search engine and selected to be representative of the people that can be observed on French TV. Our dataset is consisting of 548,686 instances of 42,655 unique persons. All of those instances are distributed into 138,381 shows. These shows are organised in order to contain images of people who did appear in the same TV show once, to capture the co-occurrence relationship between them.
The dataset was build as follows: Using a list of TV shows and of people occurring in them, we scraped a search engine in order to build a dataset of images of these people, that we annotated semi-automatically: for each queried person, we applied a clustering on the faces returned by the search engine to remove undesired results. We removed duplicates and annotated manually ambiguous pairs of classes for which the distance between facial descriptors obtained with a pre-trained model was low. For all remaining images, the corresponding ground truth is the name used as a query when scraping with a search engine.
To each TV show in our list, we assign, when available, an instance of all people that appear in it. No instance can be assigned to more than one show. If all instances of one person have been assigned, that person is no longer considered in the upcoming shows. Only shows with at least two instances assigned to them are kept. See Fig. 1 for an example.

[1] https://github.com/ina-foss/co-occurring_faces_in_tv

Finally, we obtain 548,686 instances of 42,655 unique identities distributed between 138,381 shows. Between 2 and 55 instances are assigned to each show, with an average of 3.96 instances per show. Every unique identity has on average 12.86 instances.
Because the TV shows used to build this dataset are real TV shows broadcasted between January 1990 and January 2020 on French TV, it contains co-occurrences of persons representative of the French television. To our knowledge, this is the largest public dataset of faces to contain such a large amount of information on social relationships between each subject.

3.3 Baseline

For our baseline, we use a previously trained model to compute facial features. It was trained as follows:

3.3.1 Training set and images pre-processing. We used the VGGFace2 training split [4] as our training set that chose for its large size and its large number of distinct identities. We first pre-processed the images using the OpenCV face detection network [2] to detect the face positions in the images. Then, we use the dlib library [11] for facial landmark detection and alignment. And finally, we extract the aligned faces in images of 256×256 pixels that are given to the network.

Figure 1: Example of shows containing face images. Donald J. Trump, Recep Tayyip Erdogan and 4 french journalists appeared together on the TV news on channel *France 2* the 16/10/2019. We sample one image for each one of them and assigned them to a common show (six first faces above). Another show is built with images of Edwige Antier, Marina Carrère d'Encausse and Michel Cymès who all appeared on a show on channel *France 5*, the 25/07/2001 (three last faces above).

3.3.2 Neural network architecture. This model is based on a ResNet18 architecture [8] trained from scratch. It outputs feature vectors of 128 dimensions.

3.3.3 Loss function. The loss function used for training is the triplet loss, introduced in [17] for similarity metric learning. This model thus learns a projection into a 128-dimensional vector space where embeddings from similar face images are supposed to be close and dissimilar ones further apart.

This model achieves a score of 98.98% on the LFW verification protocol. On the evaluation protocol defined in 5.1, using the facial features extracted with this model, we obtained a mAP over the query set of 67.93%.
For all of methods described in this paper, only this model is used. We will show how we can increase the mAP on our query set from 67.93%, with this trained model only, to more than 78% by exploiting the co-occurrence relationships of people.

4 METHODOLOGY

4.1 Main approach

In order to exploit social relationships between entities to better recognize each face, we first need to identify those entities. In other works [10, 21], the entities are already known, as the ultimate goal is not to retrieve other faces, but to classify a query in a set of known identities, for which examples of social relationships are known. In our case, however, we do not dispose of this information.

An overview of our approach is represented in Fig. 2. We call X the set of instances in our dataset, and S the list of shows, which are subsets of X. For practical reasons and for readability, in the rest of this document, X refers interchangeably to both the set of faces observed in all shows and to the set of facial features extracted from these faces with our model (see subsection 3.3).

As the real identities of all our instances is unknown, we estimate these identities *via* a clustering of all of the face images from our dataset, based on their visual features. Although there may be some outliers or wrongly grouped faces, the list of identified clusters will then be used as a proxy for the list of real identities, and we will infer the social relationships among the clusters. We will compare different clustering methods for approximating the real identities in section 4.2.

Once identities have been approximated with clusters, we map each cluster to a multidimensional feature space so that clusters containing many co-occurring instances are mapped to close vectors in the new feature space, while clusters with no co-occurring instances are mapped to distant feature vectors. This part is detailed in section 4.3.

To each show $s \in S$ we can now assign a contextual feature $\mathbf{y}_s$ that combines the features of all the clusters appearing in that show. The new combined contextual feature $\mathbf{y}_s$ should be representative of all instances of s while being robust to outliers. We compare two methods to combine the contextual features of the clusters appearing in a show in subsection 4.4. The first one is using

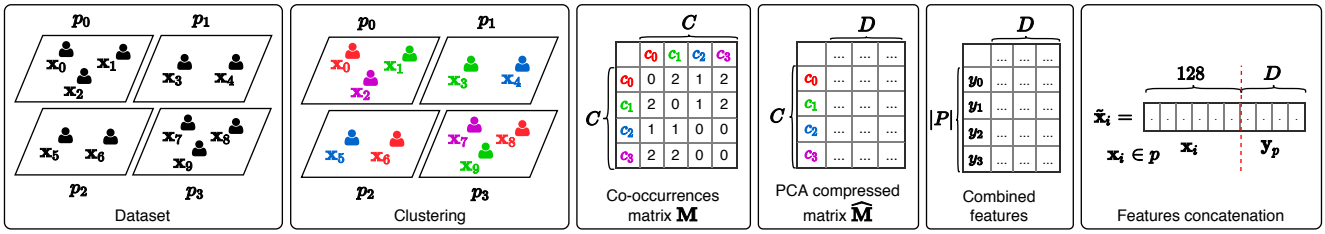

Figure 2: Our proposed framework. Similar faces appearing in the different shows s_i are first grouped together into C clusters using our approximate soft-clustering approach, operating on visual feature vectors $\mathbf{x}_i$. Cluster co-occurrences are registered in a specific co-occurrence matrix $\mathbf{M}$ which is used to learn a contextual descriptor matrix $\widehat{\mathbf{M}}$. For each show $s \in S$, the contextual descriptors of all faces $\mathbf{x}_i \in s$ are then merged into a common descriptor $\mathbf{y}_s$ that best represent them. Finally, the retrieval of faces is performed on the concatenation $\tilde{\mathbf{x}}_i$ of the facial descriptors $\mathbf{x}_i$ and the contextual descriptors $\mathbf{y}_s$.

their average value, while the second is using the geometric median.

Finally, we define $\tilde{\mathbf{x}}$ for $\mathbf{x} \in s$ as the concatenation of the facial feature vector of the instance $\mathbf{x}$ and the contextual descriptor $\mathbf{y}_s$ of show s, after normalization. Retrieval is now applied on the concatenated feature vectors $\tilde{\mathbf{x}}$.

4.2 Clustering methods

4.2.1 HDBSCAN. We compare three different clustering approaches. The first one is the HDBSCAN clustering method [16], which is based on DBSCAN [3]. Its main advantage is that it does not require to specify the number of expected clusters, as it is the case in other approaches such as k-means. The only input it requires is the minimum size of the clusters, which we set to 3 in order to identify even the smaller ones. With HDBSCAN, every point can be assigned to a cluster, or, if it is too far from any identified cluster, be considered as an outlier.
The vectors $\mathbf{c}(\mathbf{x})$ can now be defined as one-hot encoders ($\mathbf{c}(\mathbf{x})_i = 1$ if $\mathbf{x}$ has been assigned to cluster i, 0 otherwise). If an instance $\mathbf{x}$ is considered as an outlier, then $\mathbf{c}(\mathbf{x}) = \mathbf{0}$. This means outliers are not accounted for when building the co-occurrence matrix $\mathbf{M}$ (see section 4.3).

4.2.2 Hard-clustering without outliers. In the second approach, we assign all points, including those identified as outliers, to the nearest clusters identified by HDBSCAN. The vectors $\mathbf{c}(\mathbf{x})$ can once again be defined as one-hot encoders. This means we have more available data to build the co-occurrence matrix $\mathbf{M}$ (see section 4.3) and to combine contextual feature vectors for each show (see section 4.4); however, these additional data are more likely to be noisy.

4.2.3 Approximated soft-clustering. The last approach is a trade-off between excluding the outliers like in the first approach and using all available information like in the second approach, by using soft clustering.
For practical reasons, the probability distribution of all instances are not computed over all identified clusters. Instead, the clusters are once again identified with HDBSCAN. The probability distribution is computed over the k-nearest clusters for each instance and a probability of all other clusters is arbitrarily assigned to zero. Also, all cluster distributions are considered isotropic Gaussians with a common covariance matrix $\Sigma = \sigma\mathbf{I}$, and all clusters are considered to have the same prior probability $P(c_i)$. With these assumptions, the posterior probabilities are equal to the likelihood: $P(c_i|\mathbf{x}) \propto P(\mathbf{x}|c_i)$. For an instance $\mathbf{x}$ and a cluster c_i with mean value μ_i, the estimated probability $P(c_i|\mathbf{x})$ is computed as follows:

$$P(c_i|\mathbf{x}) = \mathbf{c}(\mathbf{x})_i \propto \begin{cases} \exp\left(-\frac{||\mathbf{x}-\mu_i||}{2\sigma}\right) & \text{if } c_i \in k\text{-nearest clusters of x} \\ 0 & \text{otherwise} \end{cases} \quad (1)$$

The case $k = 1$ is equivalent to the hard clustering described in 4.2.2.

4.3 Co-occurrence matrix

After clustering X (detailed in subsection 4.2), C clusters are identified. For each instance $\mathbf{x} \in X$, we define a vector $\mathbf{c}(\mathbf{x}) \in \mathbb{R}^C$ that denotes the posterior probability of each identified cluster given $\mathbf{x}$:

$$\mathbf{c}(\mathbf{x})_i = P(c_i|\mathbf{x}) \text{ for } \mathbf{x} \in X \text{ and for } i \in \{1...C\} \quad (2)$$

that is, $\mathbf{c}(\mathbf{x})$ represents the probability mass function over all clusters for a given instance.

We then define a probabilistic co-occurrence matrix $\mathbf{M} \in \mathbb{R}^{C\times C}$ as follows:

$$\mathbf{M}_{i,j} = \begin{cases} \sum_{s\in S} \sum_{x_1,x_2\in p, x_2\neq x_1} \mathbf{c}(\mathbf{x_1})_i \mathbf{c}(\mathbf{x_2})_j & \text{if } i \neq j \\ 0 & \text{otherwise} \end{cases} \quad (3)$$

The value $\mathbf{M}_{i,j}$, for $i \neq j$, is thus equal to

$$\sum_{s\in S} \sum_{x_1,x_2\in s, x_2\neq x_1} P(c_i|\mathbf{x_1})P(c_j|\mathbf{x_2})$$

This element denotes the expected number of shows in which an instance assigned to cluster i co-occurs with an instance assigned to cluster j.

As C can be quite large, the matrix $\mathbf{M}$ is rather sparse containing a great number of zeros as well as some redundancy. Further, each line/column of $\mathbf{M}$ represents a contextual description that is very long compared to the visual feature vectors. Thus the fusion of both creates an imbalance that degrades the final retrieval performance. Therefore, a Principal Component Analysis (PCA) is performed on $\mathbf{M}$ and the D most important components are kept ($D < C$), transforming it into a new matrix $\widehat{\mathbf{M}} \in \mathbb{R}^{C\times D}$.

Each cluster c_i, for $i \in \{1...C\}$ is now associated with a (compressed) contextual feature vector $\widehat{\mathbf{M}}_i \in \mathbb{R}^D$.

4.4 Combining contextual features for each show

For a show s, the list of clusters that occur in that show is the list of clusters to which are assigned instances of that show: $\{\mathbf{c}(\mathbf{x})\}_{\mathbf{x} \in s}$. These clusters themselves have a contextual feature vector that embeds their co-occurrence relationships. In order to map a show in the same contextual feature space, different approaches are possible.

4.4.1 Average value. The first, straightforward approach consists in simply computing the feature vector $\mathbf{y}_s$ of show s as the average feature vector of the clusters appearing in it:

$$\mathbf{y}_s = \frac{\sum_{\mathbf{x} \text{ in } s} \sum_{i=1}^{C} \mathbf{c}(\mathbf{x})_i \widehat{\mathbf{M}}_\mathbf{i}}{\sum_{\mathbf{x} \text{ in } s} \sum_{i=1}^{C} \mathbf{c}(\mathbf{x})_i} \tag{4}$$

4.4.2 Geometric median. The second approach consists in computing the feature vector $\mathbf{y}_s$ not as the average value, but as the geometric median of the feature vectors of the clusters occurring in s. This allow for $\mathbf{y}_s$ to be more robust to single feature vectors that differ considerably from the other ones, which might be the case when an instance has been assigned to the wrong cluster.

$$\mathbf{y}_s = \arg\min_{\mathbf{y} \in \mathbb{R}^D} \sum_{\mathbf{x} \text{ in } s} \sum_{i=1}^{C} \mathbf{c}(\mathbf{x})_i ||\widehat{\mathbf{M}}_\mathbf{i} - \mathbf{y}||_2 \tag{5}$$

The algorithm used for inferring the geometric median is the one detailed in [20].

Note that for both approaches, the feature vector of each cluster i is weighted by the probability $\mathbf{c}(\mathbf{x})_i = P(c_i|\mathbf{x})$ for $\mathbf{x} \in s$. This means that in the case of soft clustering, only the k-nearest clusters identified in 4.2.3 are accounted for.
With the HDBSCAN clustering, it is possible that no instance of a show s have been assigned to a cluster: $\sum_{\mathbf{x} \text{ in } s} \sum_{i=1}^{C} \mathbf{c}(\mathbf{x})_i = 0$. In this case, $\mathbf{y}_s$ is set to the average value of all clusters contextual feature vectors.

5 EXPERIMENTS

5.1 Experiments setup

Our dataset of faces distributed in TV shows is partitioned into two disjoint sets: a training set of 137,381 TV shows, containing 544,863 faces, and a test set of 1,000 TV shows of 3,823 faces, 3,770 of which belonging to people also appearing in the training set. We apply our pre-processing on the training set (i.e. clustering of its instances, computation and reduction of the co-occurrence matrix). Our method to improve the retrieval of faces is evaluated on both internal and external queries. The internal queries are 9,820 instances from the training set, belonging to people appearing at least one more time in the training set. The external queries are the 3,770 instances of the test set belonging to people who also appear in the training set.
The mean Average Precision (mAP) obtained on each set of queries is used as the performance metric. We compare the performance of our approach for the different clustering methods (explained in 4.2) applied on the training set, and the different feature vector combinations explained in 4.4.
As a baseline, we use the model for facial features extraction described in section 3.3. With the face descriptors extracted with this model, we obtain a mAP of 67.93% on the internal query set and a mAP of 67.88% on the external query set.

5.2 Internal queries results and analysis

The results obtained with internal queries in the different configurations for clustering and for feature vector combination are detailed in Table 1. The mAP is computed on the retrieval of the vectors $\tilde{\mathbf{x}}$. The dimensionality of the contextual features $\mathbf{y}_s$ for $s \in s$ is set to $D = 300$. We observe an increase of the mAP on the query set from 67.93% using facial features only, to 78.16% by exploiting the co-occurrences between the identified clusters, in the best configuration.
Figure 3 shows the evolution of the mAP in the different configurations as a function of the number of additional components from the contextual features concatenated to the 128-dimensional face descriptors. We can notice the score keeps increasing slightly for a contextual features dimension D higher than 300.

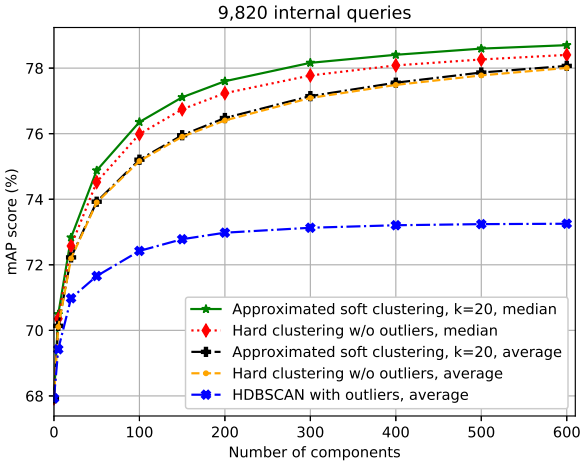

Figure 3: Mean Average Precision over the internal queries as a function of the number of additional components from the contextual features, for different clustering methods and configurations.

The results obtained using the HDBSCAN algorithm, where some points are considered as outliers, are far below those obtained when clustering all points or when using soft clustering. If some of the points might be wrongly clustered, the gain we get from this additional data is higher than the loss due to these errors.
Also, in the case of hard clustering without outliers, the clustering errors can be mitigated by the geometric median (configuration (4) in Table 1) when computing a show feature vector. The geometric median will be more robust to outliers in the contextual feature space, hence a small gain compared to the average value

Table 1: mAP over 9,820 internal queries under different configurations, for 300 additional components:

Clustering algorithm	HDBSCAN		Hard clustering w/o outliers		Soft clustering $k = 20$		Face descriptors only
Feature vector combination	Average (1)	Median (2)	Average (3)	Median (4)	Average (5)	Median (6)	
mAP over 20K queries	73.13%	72.97%	77.09%	77.78%	77.15%	**78.16%**	**67.93%**

Table 2: Mean Average Precision over 3,770 external queries under different configurations, for 300 additional components:

Clustering algorithm	Hard clustering w/o outliers		Soft clustering $k = 20$		Face descriptors only
Feature vector combination	Average (3)	Median (4)	Average (5)	Median (6)	
mAP over 20K queries	76.15%	76.93%	76.22%	**77.36%**	**67.88%**

(configuration (3)). The best results are obtained when combining the geometric median with the soft clustering (configuration (6)). For a soft clustering approximated over the $k = 20$ nearest clusters of each instance, the mAP obtained while retrieving faces from the query set reaches 78.16%.

With this best model, the average precision increased for 8,325 of the 9,820 internal queries. It decreased for 782 other queries and remained unchanged for the 713 remaining ones.

5.2.1 Impact of the number of instances co-occurring with the query. We observe that the variation of the Average Precision for a given query does not depend on the amount of people occurring in the show in which it appears : the improvement is visible for shows with only 2 persons and remains stable for shows with more people.

5.2.2 Impact of the number of instances of the queried identity in the dataset. Our model's performance is slightly lower to the baseline when retrieving a face with only 1 or 2 other instances in our dataset.

This can be due to the fact that identities with fewer instances are more likely to be assigned to neighbor clusters with co-occurrence relationships that do not match their owns'. In this case, the contextual information used to improve our results is noisy. The gain becomes clearly noticeable for identities occurring at least 4 times, which are more likely to form their own cluster and to take advantage of our approach.

5.2.3 Impact of the parameter k for the approximated soft-clustering on the results. In our soft clustering approach, the probability $P(c_i|\mathbf{x})$ of an instance x to belong to a cluster c_i is approximated as described in 4.2.3 and Eq. 1. We only assign a non-zero probability to the k-nearest clusters. Setting k to a small value means we only focus on very similar clusters (according to our model detailed in 3.3), while choosing a higher value of k means the instance could also belong to clusters that are much more dissimilar, if their contextual embeddings match.

We show that even if the mAP slightly varies with the value of k, the choice of this parameter is not significant : for a given configuration where the contextual feature vectors have 300 components, and the geometric median is used to combine each show contextual features, the mAP increases from 78.12% for $k = 2$ to 78.16% for $k = 20$ and then reaches a plateau for higher values of k.

5.3 External queries results and analysis

External queries are first assigned or soft-assigned to the nearest clusters identified in the gallery set. The contextual features of the external shows are computed based on these clusters. The results obtained with external queries in the different configurations for clustering and for feature vector combination are detailed in Table 2. The mAP is computed on the retrieval of the combined vectors $\tilde{\mathbf{x}}$. Similarly to what has been done with internal queries, the dimensionality of the contextual features $\mathbf{y}_s$ for $s \in S$ is set to $D = 300$. The evolution of the mAP as a function of the dimensionality D is displayed in Fig. 4. While the scores obtained on these external queries is slightly lower to what was observed with internal queries, they remain substantially higher to our baseline.

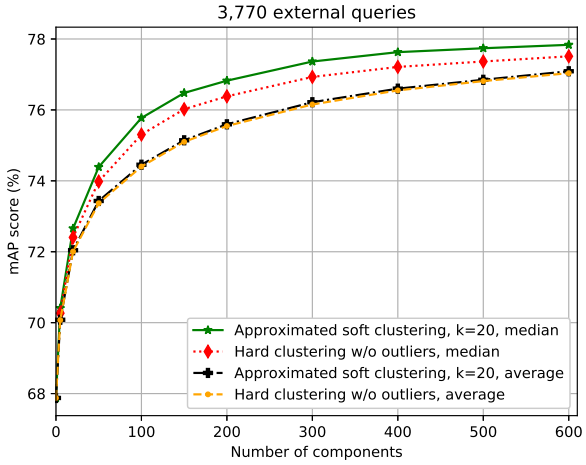

Figure 4: Mean Average Precision over the external queries as a function of the number of additional components from the contextual features, for different clustering methods and configurations.

6 CONCLUSION

In this paper, we describe a new unsupervised method to learn a contextual descriptor mapping for a set of unlabeled faces occurring together. Using a previously trained model to extract facial feature vectors, we compute for each one of these faces a probability distribution over a set of possible contextual embeddings. The probability distributions of these co-occurring faces are then combined to determine a contextual embedding that best represent the whole context. We show that the fusion of the facial feature vectors and the combined contextual descriptor yields considerably better results in the retrieval task than the facial feature vectors alone.

We evaluated our method on a new dataset, built in order to capture the diversity of people on TV and their co-occurrence relationships in the different shows. We show that using no additional data, the results of our method are clearly superior to those obtained using only facial features on a retrieval task.

REFERENCES

[1] Ankan Bansal, Anirudh Nanduri, Carlos D Castillo, Rajeev Ranjan, and Rama Chellappa. 2017. Umdfaces: An annotated face dataset for training deep networks. In *2017 IEEE International Joint Conference on Biometrics (IJCB)*. IEEE, 464–473.

[2] G. Bradski. 2000. The OpenCV Library. *Dr. Dobb's Journal of Software Tools* (2000).

[3] Ricardo JGB Campello, Davoud Moulavi, and Jörg Sander. 2013. Density-based clustering based on hierarchical density estimates. In *Pacific-Asia conference on knowledge discovery and data mining*. Springer, 160–172.

[4] Q. Cao, L. Shen, W. Xie, O. M. Parkhi, and A. Zisserman. 2018. VGGFace2: A dataset for recognising faces across pose and age. In *International Conference on Automatic Face and Gesture Recognition*.

[5] Daniel Coelho de Castro and Sebastian Nowozin. 2018. From face recognition to models of identity: A Bayesian approach to learning about unknown identities from unsupervised data. In *Proceedings of the European Conference on Computer Vision (ECCV)*. 745–761.

[6] Jiankang Deng, Jia Guo, Niannan Xue, and Stefanos Zafeiriou. 2018. Arcface: Additive angular margin loss for deep face recognition. *arXiv preprint arXiv:1801.07698* (2018).

[7] Yandong Guo, Lei Zhang, Yuxiao Hu, Xiaodong He, and Jianfeng Gao. 2016. Ms-celeb-1m: A dataset and benchmark for large-scale face recognition. In *European Conference on Computer Vision*. Springer, 87–102.

[8] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2016. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 770–778.

[9] Gary B Huang, Marwan Mattar, Tamara Berg, and Eric Learned-Miller. 2008. Labeled faces in the wild: A database forstudying face recognition in unconstrained environments. In *Workshop on faces in'Real-Life'Images: detection, alignment, and recognition*.

[10] Qingqiu Huang, Yu Xiong, and Dahua Lin. 2018. Unifying identification and context learning for person recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2217–2225.

[11] Davis E. King. 2009. Dlib-ml: A Machine Learning Toolkit. *Journal of Machine Learning Research* 10 (2009), 1755–1758.

[12] Neeraj Kumar, Alexander Berg, Peter N Belhumeur, and Shree Nayar. 2011. Describable visual attributes for face verification and image search. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 33, 10 (2011), 1962–1977.

[13] Haoxiang Li, Jonathan Brandt, Zhe Lin, Xiaohui Shen, and Gang Hua. 2016. A multi-level contextual model for person recognition in photo albums. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 1297–1305.

[14] Weiyang Liu, Yandong Wen, Zhiding Yu, Ming Li, Bhiksha Raj, and Le Song. 2017. Sphereface: Deep hypersphere embedding for face recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 212–220.

[15] Ziwei Liu, Ping Luo, Xiaogang Wang, and Xiaoou Tang. 2015. Deep Learning Face Attributes in the Wild. In *Proceedings of International Conference on Computer Vision (ICCV)*.

[16] Leland McInnes and John Healy. 2017. Accelerated hierarchical density based clustering. In *2017 IEEE International Conference on Data Mining Workshops (ICDMW)*. IEEE, 33–42.

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

[18] Zak Stone, Todd Zickler, and Trevor Darrell. 2008. Autotagging facebook: Social network context improves photo annotation. In *2008 IEEE computer society conference on computer vision and pattern recognition workshops*. IEEE, 1–8.

[19] Yaniv Taigman, Ming Yang, Marc'Aurelio Ranzato, and Lior Wolf. 2014. Deepface: Closing the gap to human-level performance in face verification. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 1701–1708.

[20] Yehuda Vardi and Cun-Hui Zhang. 2000. The multivariate L1-median and associated data depth. *Proceedings of the National Academy of Sciences* 97, 4 (2000), 1423–1426.

[21] Ning Zhang, Manohar Paluri, Yaniv Taigman, Rob Fergus, and Lubomir Bourdev. 2015. Beyond frontal faces: Improving person recognition using multiple cues. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 4804–4813.

[22] Yujie Zhong, Relja Arandjelovic, and Andrew Zisserman. 2018. Compact deep aggregation for set retrieval. In *Proceedings of the European Conference on Computer Vision (ECCV)*. 0–0.