Context-aware person recognition in TV programs

Thomas Petit

N°d'ordre NNT : 2022LYSEI049

THESE de DOCTORAT DE L'UNIVERSITE DE LYON
opérée au sein de
INSA Lyon

Ecole Doctorale N° 512
InfoMaths

Spécialité/ discipline de doctorat :
Informatique

Soutenue publiquement le 10/06/2022, par :
Thomas PETIT

Context-aware person recognition in TV programs

Devant le jury composé de :

BENOIS-PINEAU, Jenny Professeure des Universités, UNIVERSITE BORDEAUX	Présidente
HUDELOT, Céline Professeure des Universités, CENTRALE-SUPELEC	Rapporteure
TAMINE-LECHANI, Lynda Professeure des Universités, PAUL-SABATIER	Rapporteure
GRAVIER, Guillaume Directeur de Recherche, IRISA	Examinateur
SOULIER, Laure Maître de Conférences, SORBONNE UNIVERSITE	Examinatrice
LEFEBVRE, Grégoire Docteur, Chercheur, ORANGE LABS	Examinateur
GARCIA, Christophe Professeur des Universités, INSA-LYON	Directeur de thèse
DUFFNER, Stefan Maître de Conférences (HDR), INSA-LYON	Co-directeur de thèse
LETESSIER, Pierre Ingénieur de Recherche, INA	Invité

Département FEDORA – INSA Lyon - Ecoles Doctorales

SIGLE	ECOLE DOCTORALE	NOM ET COORDONNEES DU RESPONSABLE
CHIMIE	**CHIMIE DE LYON** **https://www.edchimie-lyon.fr** Sec. : Renée EL MELHEM Bât. Blaise PASCAL, 3e étage secretariat@edchimie-lyon.fr	**M. Stéphane DANIELE** C2P2-CPE LYON-UMR 5265 Bâtiment F308, BP 2077 43 Boulevard du 11 novembre 1918 69616 Villeurbanne **directeur@edchimie-lyon.fr**
E.E.A.	**ÉLECTRONIQUE, ÉLECTROTECHNIQUE, AUTOMATIQUE** **https://edeea.universite-lyon.fr** Sec. : Stéphanie CAUVIN Bâtiment Direction INSA Lyon Tél : 04.72.43.71.70 secretariat.edeea@insa-lyon.fr	**M. Philippe DELACHARTRE** INSA LYON Laboratoire CREATIS Bâtiment Blaise Pascal, 7 avenue Jean Capelle 69621 Villeurbanne CEDEX Tél : 04.72.43.88.63 **philippe.delachartre@insa-lyon.fr**
E2M2	**ÉVOLUTION, ÉCOSYSTÈME, MICROBIOLOGIE, MODÉLISATION** **http://e2m2.universite-lyon.fr** Sec. : Sylvie ROBERJOT Bât. Atrium, UCB Lyon 1 Tél : 04.72.44.83.62 secretariat.e2m2@univ-lyon1.fr	**M. Philippe NORMAND** Université Claude Bernard Lyon 1 UMR 5557 Lab. d'Ecologie Microbienne Bâtiment Mendel 43, boulevard du 11 Novembre 1918 69 622 Villeurbanne CEDEX **philippe.normand@univ-lyon1.fr**
EDISS	**INTERDISCIPLINAIRE SCIENCES-SANTÉ** **http://ediss.universite-lyon.fr** Sec. : Sylvie ROBERJOT Bât. Atrium, UCB Lyon 1 Tél : 04.72.44.83.62 secretariat.ediss@univ-lyon1.fr	**Mme Sylvie RICARD-BLUM** Institut de Chimie et Biochimie Moléculaires et Supramoléculaires (ICBMS) - UMR 5246 CNRS - Université Lyon 1 Bâtiment Raulin - 2ème étage Nord 43 Boulevard du 11 novembre 1918 69622 Villeurbanne Cedex Tél : +33(0)4 72 44 82 32 **sylvie.ricard-blum@univ-lyon1.fr**
INFOMATHS	**INFORMATIQUE ET MATHÉMATIQUES** **http://edinfomaths.universite-lyon.fr** Sec. : Renée EL MELHEM Bât. Blaise PASCAL, 3e étage Tél : 04.72.43.80.46 infomaths@univ-lyon1.fr	**M. Hamamache KHEDDOUCI** Université Claude Bernard Lyon 1 Bât. Nautibus 43, Boulevard du 11 novembre 1918 69 622 Villeurbanne Cedex France Tél : 04.72.44.83.69 **hamamache.kheddouci@univ-lyon1.fr**
Matériaux	**MATÉRIAUX DE LYON** **http://ed34.universite-lyon.fr** Sec. : Yann DE ORDENANA Tél : 04.72.18.62.44 yann.de-ordenana@ec-lyon.fr	**M. Stéphane BENAYOUN** Ecole Centrale de Lyon Laboratoire LTDS 36 avenue Guy de Collongue 69134 Ecully CEDEX Tél : 04.72.18.64.37 **stephane.benayoun@ec-lyon.fr**
MEGA	**MÉCANIQUE, ÉNERGÉTIQUE, GÉNIE CIVIL, ACOUSTIQUE** **http://edmega.universite-lyon.fr** Sec. : Stéphanie CAUVIN Tél : 04.72.43.71.70 Bâtiment Direction INSA Lyon mega@insa-lyon.fr	**M. Jocelyn BONJOUR** INSA Lyon Laboratoire CETHIL Bâtiment Sadi-Carnot 9, rue de la Physique 69621 Villeurbanne CEDEX **jocelyn.bonjour@insa-lyon.fr**
ScSo	**ScSo*** **https://edsciencessociales.universite-lyon.fr** Sec. : Mélina FAVETON INSA : J.Y. TOUSSAINT Tél : 04.78.69.77.79 melina.faveton@univ-lyon2.fr	**M. Christian MONTES** Université Lumière Lyon 2 86 Rue Pasteur 69365 Lyon CEDEX 07 **christian.montes@univ-lyon2.fr**

*ScSo : Histoire, Géographie, Aménagement, Urbanisme, Archéologie, Science politique, Sociologie, Anthropologie

Abstract

The automatic recognition and retrieval of faces can be a useful tool for exploiting and promoting large datasets, such as the archival collection of TV shows stored by INA. Although face recognition solutions have improved dramatically in the last decade, they unfortunately remain prone to mistakes, more especially with a large number of faces and a large number of different identities.
The various TV shows are however quite standardised, meaning that it is most of the time easy for anyone to tell what a TV show is about in a glimpse, be it a sport show, an entertainment show or a newscast. Though implicit, this standardisation of the TV shows applies in numerous ways, from the visual appearance of the show to the broadcast time. Moreover, we also know that the contextual information plays a major role in helping the human brain recognizing people, and that, in fact, we seldom recognize people based on their facial appearance only. This also applies to TV shows, where the various contextual information can help us identify who is likely or not to appear in a given show.
The goal of this thesis is to identify and to exploit the contextual modalities available and potentially useful for the identification of the people appearing in TV shows. For each one of these modalities, we extract the information as a feature descriptor which can be combined to the facial feature descriptor to either retrieve other instances of the same person or to identify them.
More especially, we focus on how the social relationships of the people appearing in the shows make them more likely to appear with some people than with others. We introduce an unsupervised method for identifying simultaneously the participants of a TV show, by estimating their probably to appear together based on previous unannotated observations.
We also study the visual context of the shows and we highlight how the background and other visual cues can help to successfully identify difficult faces.
Finally, we explore how useful can be the contextual modalities such as the time of broadcast or the thematic tags assigned to each show, by evaluating the improvement they bring on the face recognition task and how redundant they can be with the other modalities.

Résumé

L'identification automatique et la recherche par similarité des visages peut s'avérer être un outil utile pour la fouille de grandes bases de données telles que les archives télévisuelles de l'INA. Bien que les outils de reconnaissance faciale aient grandement progressé récemment, ils ne sont pas pour autant exempts d'erreurs, notamment lorsque la quantité de visages et le nombre de personnalités à reconnaître deviennent trop grands.

En revanche, les programmes télévisés sont généralement très codifiés, de telle manière qu'il est aisé pour chacun de dire en quelques secondes d'une émission s'il s'agit d'une émission sportive, de divertissement ou d'actualité. Cette codification des programmes, bien qu'implicite, peut s'étendre de l'apparence visuelle du plateau au choix du créneau horaire. Par ailleurs, nous savons aussi aujourd'hui que le contexte, au sens large, joue un rôle important pour le cerveau afin de reconnaître des individus, et que l'on ne reconnaît en réalité que très rarement des visages de par leur apparence seule. Ceci s'applique aussi bien évidemment aux programmes télévisés, où ces informations nous permettent donc de prédire qui est susceptible ou non de participer à une émission donnée.

L'objectif de cette thèse est ainsi d'exploiter l'ensemble des informations contextuelles disponibles et potentiellement utiles pour l'identification des personnalités apparaissant dans les programmes télévisés. Pour chacune de ces modalités, nous en extrayons l'information, qui combinée aux descripteurs faciaux des sujets à reconnaître, permettra d'améliorer la recherche de nouvelles instances ou la classification des visages.

Nous nous intéressons notamment aux relations sociales entre les différents participants faisant que certains sont plus susceptibles d'apparaître ensemble à la télévision que d'autres. Nous proposons ainsi une méthode non-supervisée pour identifier simultanément l'ensemble des participants à un programme télévisé, en estimant leur probabilité d'apparaître conjointement.

Dans une seconde partie, nous nous intéressons aux informations contenues dans le contexte visuel des programmes télévisé et montrons que les arrière-plans visibles à l'écran peuvent aider à d'identifier avec succès les visages ambigus.

Nous explorons aussi les modalités contextuelles telles que les heures de diffusion ou les catégorisations thématiques des programmes, pour lesquelles nous évaluons l'apport d'informations utiles à la reconnaissance des participants ainsi que leur redondance avec les autres modalités étudiées.

Acknowledgments

Je tiens tout d'abord à remercier Christophe et Stefan, pour leur encadrement ainsi que leurs conseils tout au long de ces trois années.

Je souhaite aussi remercier Pierre, bien évidemment, pour son aide et son implication sans lesquelles ces travaux n'existeraient tout simplement pas !

Merci beaucoup, aussi, à l'ensemble de l'équipe de la recherche de l'INA, que ce soit pour nos échanges constructifs comme pour leur accueil et tout le reste. Je pense en particulier à Nicolas, Agnès, Zeynep, Louis, Laurent, Steffen, Abdelkrim, David, Rémi, et tous les autres.
Un grand merci aussi à Valérie et à Elisabeth, pour leur aide lorsque l'on a besoin d'elles, ainsi que pour leur bonne humeur !

Merci à mes parents pour m'avoir permis de suivre les études que je souhaitais.
Merci Yousra, pour m'avoir accompagné et soutenu ces trois années et pour tout le reste.

Merci, enfin, à tous ceux qui prendront la peine de lire cette thèse, et notamment aux rapporteurs !

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Motivation

INA is a french public company whose goal is amongst others to archive french television broadcasts. To this day, the archival collection of INA contains over 20 million hours of TV shows. Because INA captures in real time about 170 TV and radio channels, this collection increases by almost 1.5 million hours every year.
For this reason, being able to fully or partially automate the archival collection annotation, and more specifically to automate the person recognition task, is a major industrial matter for INA. A face retrieval tool based on facial similarity, or a face verification model, will find many uses for documentation ends, for example by helping archivists with name suggestions for labelling unknown faces, but also for promoting the archival collection by allowing users to browse content that has not yet been labeled manually, and would have remained inaccessible otherwise.

While automatic facial recognition has seen considerable improvements in the last decade, thanks especially to the advances in machine learning, the performance of these algorithms is not perfect yet when applied to large scale datasets like INA archival collection. Also, the fact that the state-of-the-art performance for facial recognition models have been obtained using enormous private datasets of large digital media companies such as Facebook or Google suggests that we are unlikely to outperform them without a similar amount of annotated data [SKP15].
However, we assume that the specificity of the INA archival collection could be leveraged to help identify the participants of TV shows and hence improve these generic face recognition models. Indeed, television programs are often standardised in many ways that implicitly describe what kind of show they are and what people are likely to appear in them. These codes describing the TV shows can be observed in various aspects such as the television schedule, the visual appearance of the TV sets, the people appearing in them, and many more.
Moreover, we also know that the contextual information, such as the visual back-

ground or the location at which they are usually seen, plays a major role in the recognition of people for human agents; we implicitly learn contextual information alongside pure face appearances to help us recognize people, to the the extent that we may fail at recognizing known faces, were they to appear in unexpected settings. In practice, however, this occurs quite rarely and such information thus proves to be very useful.

For these reasons, we aim at exploring and exploiting the contextual information of the TV shows in this thesis in order to better identify the participants appearing in them.

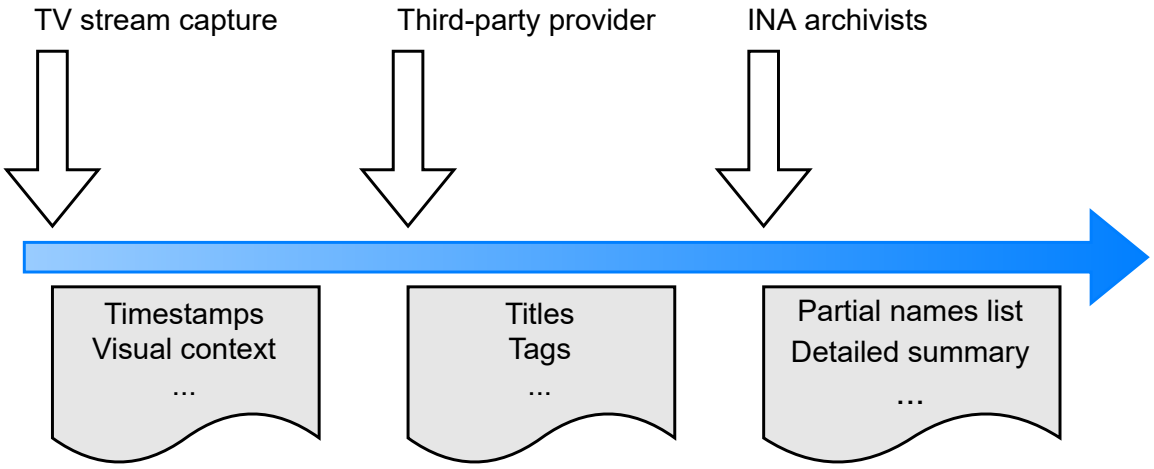

Figure 1.1: Available contextual meta-data provided at various steps for the INA archival collection.

The contextual meta-data available in the archival collection of INA are gathered at different steps and by different actors. First, some information is directly available from the capture of the TV stream: it is the case for the date and time of diffusion, the visual information contained in the video, and so on. Additional information is later provided by third-party companies which detail roughly the TV shows with their titles and a few descriptive tags. Finally, the INA archivists will enrich these information with more details like a non-exhaustive list of the participants occurring in the shows, a detailed summary, and more (see Fig 1.1). Due to the cost these annotations, the more detailed the provided information is, the least frequent it is across the collection; hence, even though the timestamps and visual information are available for every show, the descriptive tags are not always provided, and the list of participants and detailed summaries are only available for a few shows among the most interesting ones (mainly TV news in the main channels).

1.2 Problem statement and goals

Our first goal in this thesis is to identify the contextual modalities of interest in the scope of person identification or retrieval. Our second objective is to find an optimal fusion scheme for both the facial appearances and the contextual information that maximize the score on these tasks. Those two goals will often be addressed simultaneously. To do this, we will consider the contextual information that is available to us and try to learn a continuous embedding from it. The advantage of continuous vector representations is that they can easily be merged with pre-existing facial feature embeddings. By providing complementary information, we aim at reaching more accurate and more confident person recognition results.

It is to be noted that while our ultimate goal is indeed to identify the faces appearing in TV shows, the facial recognition problem itself does not lie in the scope of this thesis: instead, we want to devise a solution that takes advantage of the various contextual modalities but that is independent for the choice of facial feature embeddings and could be used with different face recognition models.

1.3 Contributions

Our contributions are the following:

Chapter 5 details our works on leveraging the statistical information of the relationships between the participants within the shows. We introduce an unsupervised method for taking advantage of the fact that many people tend to occur together on different TV shows. By identifying the different faces appearing in a query TV show simultaneously, we have been able to improve the performance on the task of retrieving other faces of the same individuals in a large gallery set of TV shows. This is done by identifying the frequent identities appearing in the gallery set and by computing a social embedding for each one of these identities that describe with whom they are the most likely to appear. When merged with the facial feature descriptors, these social embeddings describe more precisely the identity of the participants of the TV shows and hence improve the retrieval of other instances.

In Chapter 6, we investigated how categorical descriptive tags could be used to this same end, and showed that they could indeed lead to an improvement in retrieving other instances of people. However, we have also been able to observe that the information conveyed in these categorical tags tends to be redundant with, though less precise than, the information derived from the simultaneous co-occurrences of the participants in several TV shows, leading us to revise our appreciation of this source.

Chapter 7 focuses on the visual information surrounding the faces in the image. We studied how a TV show can be described by its visual context and its visual background and how they can contain information about the people appearing in this same show. By building a large dataset of TV frames, we have been able to learn a model for extracting visual embeddings specific to TV shows. We show that this visual context embeddings can be beneficial for distinguishing different faces in ambiguous cases. Several approaches have been experimented in order to identify these ambiguous or difficult cases.

Finally, Chapter 8 describes how we also tried to take advantage of the temporal information present in the broadcast schedule. After determining that the temporal information in the broadcast time of the TV shows can be broken down into linear and periodical information, we used a recently proposed frequency transform based on deep learning to identify the relevant periods at which people are likely to appear again on TV. This allowed us to learn a vector representation of scalar timestamps to enrich the facial feature descriptors.

All of these experiments have been made possible by the development of several datasets, build for the purpose of investigating the contribution of the various contextual modalities within the scope of person identification or retrieval. These datasets, which are made public, are to our knowledge the largest ones built specifically with a focus on the contextual information of TV shows. They are described in detail in Chapter 4.

1.4 Publications

Parts of the works described in this thesis have been introduced in the following publications:

- Thomas Petit, Pierre Letessier, Stefan Duffner, and Christophe Garcia. Unsupervised learning of co-occurrences for face images retrieval. In *Proceedings of the 2nd ACM International Conference on Multimedia in Asia*, pages 1–7, 2021
- Thomas Petit, Pierre Letessier, Stefan Duffner, and Christophe Garcia. Exploiting Visual Context to Identify People in TV Programs. In *International Conference on Computer Analysis of Images and Patterns*, pages 220–230. Springer, 2021

These publications correspond in particular to the chapters 5 and 7 of this thesis, which address the issue of the social context and the relationships between the participants of TV shows, and the background and visual context.

Part I

Related works

Chapter 2

Research background on face analysis

The facial recognition task, being closely related to the human-machine interaction problem, arose quickly as a major milestone in the artificial intelligence domain. Thus, despite being a complex issue, the automatic recognition of faces has been studied very early compared to other computer vision problems. Because the devise of an optimal facial feature descriptor is not the ultimate purpose of this thesis, we will not go into much details into the large range of existing methods for face recognition. We will, however, describe shortly the early methods of automatic face recognition and explain the current state-of-the-art approaches for identifying faces, upon which we will develop the contextual information analysis.
An overview of the current state-of-the-art methods for contextual information analysis for person recognition is available in Chapter 3.

2.1 Face recognition methods

2.1.1 Historical methods

The very first attempts at describing mathematically human faces where quite naive: they consisted in identifying facial landmarks such as the edges of the eyes or the tip of the nose, using a simple edge detector [Kan74]. The faces were finally described as a sequence of handcrafted features like the ratio of distances between some of these landmarks, or the angles between them. Such features are depicted in Fig. 2.1 Of course, this method was very inefficient and inapplicable as soon as the lighting was insufficient or the subject was not facing properly the camera.

Later emerged a novel approach, following the increased computational power available at the time: the *Eigenfaces* [TP91, BHK97], and shortly later the *Fisherfaces* [CLY+92, BHK97]. Both methods consist in decomposing the face images to analyze into a combination of principal components.
The *Eigenfaces* are obtained after performing a Principal Component Analysis (PCA)

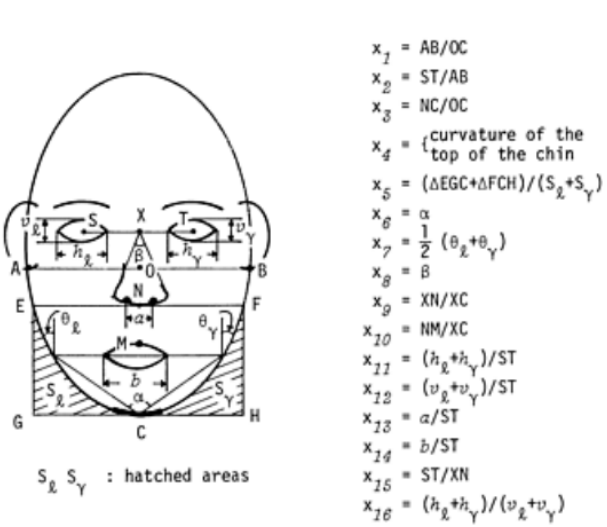

Figure 2.1: Kanade descriptors (figure from [Kan74]).

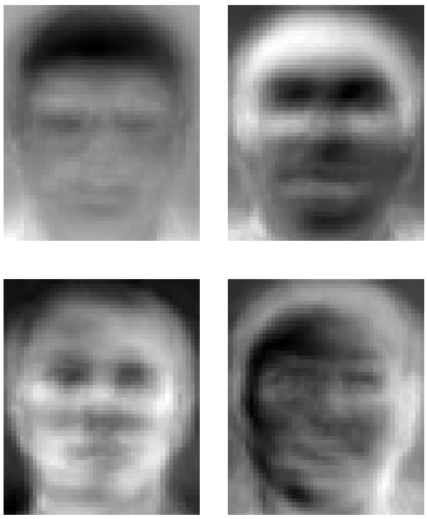

Figure 2.2: Eigenfaces (figure from [TP91]. ©AT&T Laboratories Cambridge).

on a training set of face images. Each face image to analyze is then decomposed as the sum of the average face of the training set and a linear combination of these principal components, called *eigenfaces*. Examples of such principal components are illustrated in Fig. 2.2. The more components are retained, the more precise the reconstruction is supposed to be. The *Fisherfaces* method is quite similar, except that the principal components are obtained through a Linear Discriminant Analysis (LDA), allowing them to better represent the extra-class variance while being more robust to intra-class variance (i.e. being less sensitive to the variations of appearance of a single person).
These methods, despite being much more effective than the previous ones, still suffered from the same flaws: the necessity for the face images to be sufficiently lighted, perfectly frontal and aligned. However, they set the path to automatically learned features by showing that they could be more efficient than handcrafted ones.

Another popular image feature representation used to describe images is based on the Scale Invariant Feature Transform (SIFT) [Low04], which have been widely used after their introduction for various computer vision tasks, like object detection or recognition with promising results. Many attempts have then followed to apply them on face recognition problems [BLGT06, GJ09, LMT+07]. SIFT descriptors are 128-dimensional vectors describing the local gradients of the image around some specific keypoints. Their main advantage compared to the previously used image descriptors is that they are supposed to be invariant in regard to scale, rotation or translation of the image, which allows for more robustness w.r.t. to common face image variations and different acquisition conditions. However, they rely on

the detection of prominent facial landmarks, which may be difficult under more challenging acquisition conditions. Moreover, these handcrafted features may not capture all intra-class variation and at the same time remain discriminative for an increasing number of faces to recognise.

2.1.2 The neural networks revolution for face recognition

Like many other pattern recognition and computer vision problems, automatic face recognition solutions benefited greatly from the leap of performance enabled by the neural networks and more especially the convolutional neural networks [LBB+98, Sch15] revived by ImageNet [KSH12]. Hence, even if neural networks had already been used for facial recognition [LGTB97] in the past, it is only much more recently that they achieved satisfying results on a large scale and that they draw attention again.

The face recognition challenge encompasses several different issues, like, face verification, classification, or clustering. Hence, the basic frameworks proposed for several image classification problems [KSH12, KH+09, SWT13] need to be expanded. The goal is to be able to compute a numerical representation of a face, depending only on the identity of that face, and independent of other factors such as luminosity, facial expressions, poses, etc. Some tried to learn such descriptors using directly the last features layers from face classifiers [TYRW14, OWJ18, CWW+13], the classification being an easier problem to solve. But such descriptors are most of the time unfit to measure distances, as descriptors coming from different faces might not be sufficiently distinct. Indeed, in comparison to other classification tasks, the face classification problem deals with a higher intra-class variance and a lower extra-class variance. To tackle this issue, others proposed to add a margin, i.e. stronger constraints, on the commonly-used cross-entropy loss [LWY+17, DGXZ18] that will allow to better differentiate them.

However, the best results have been achieved by using innovative loss functions that directly implement the goal to gather similar faces and distinguish dissimilar ones. The contrastive loss [CHL+05], for example, tries to maximize the distances between negative pairs of faces (belonging to different individuals), while minimizing the distances of positive pairs (belonging to the same person).
One of the more efficient loss function for that case is the widely-used triplet loss [SKP15]. It is also worth mentioning that it was first introduced for a face recognition problem. The triplet loss takes as input triplets, consisting of 3 elements forming a positive pair and a negative one. The condition described by the triplet loss is respected when the distance of the positive pair is lower than the one of the negative pair. Mathematically, the loss TL to be minimized, for a distance of the positive pair

d_p and a distance of the negative pair d_n, is calculated as:

$$TL = max(0, d_p - d_n + m) \tag{2.1}$$

where m is a margin value. The triplet loss principle is illustrated in Fig 2.3.

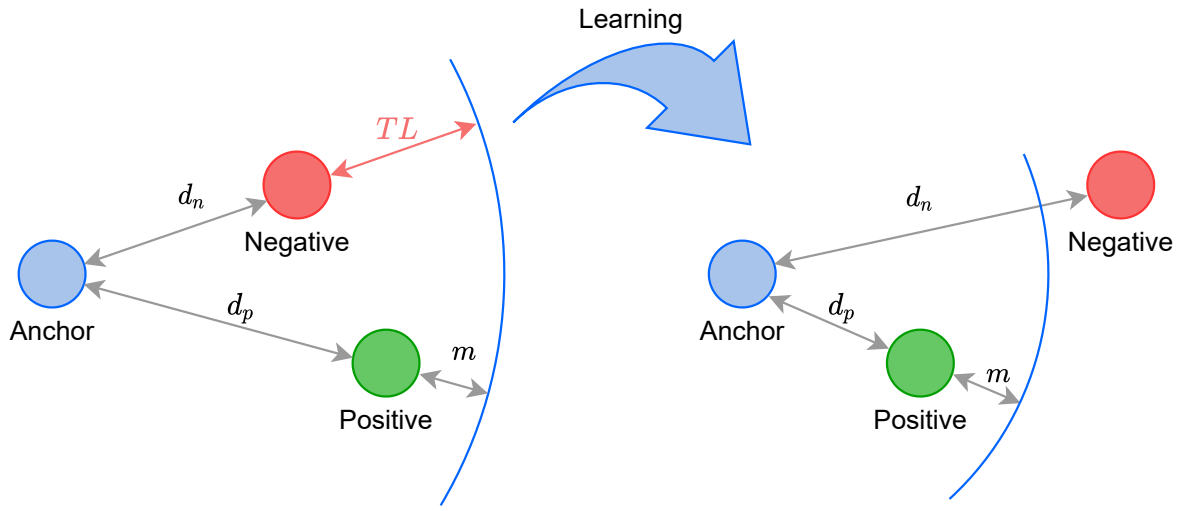

Figure 2.3: Principle of the triplet loss as explained in [SKP15].

Triplets selection is a critical aspect of the efficiency of the triplet loss. Sampling only easy triplets will mean that the condition $d_p + m < d_n$ will quickly be reached and the model will stop improving. To prevent this and allow the neural network to reach its full potential, triplets should be sampled in order to ensure that the condition over the positive and negative distances $d_p + m > d_n$ is respected.

Several triplets selection strategies have been proposed in order to solve this issue. In [SMN+17, WZL17], similar identity classes are identified and the negative element of each triplet is sampled only amongst the most similar classes. This way, the probability for the negative distances d_n to be inferior to the positive ones d_p is increased, leading to a faster optimization of the model. Another solution is to sample dissimilar positive pairs against similar negative ones [SMO+18]. This allows the very best triplets to be sampled at each iteration. However, this requires to track the embedding values of each element of the training set, which can be challenging depending on the training set size as well as the fact that the embedding values are constantly updated with the learning model. Moreover, using hard triplets early in the learning process can lead to converging in bad local minima. Hence, these hard triplet mining strategies are better used at the last stages of the model optimization and "semi-hard" triplets are favoured first.

2.2 Pre-processing methods

2.2.1 Face detection algorithms

Face detection is the very first step in the facial recognition pipeline and a problem on its own. The first solutions proposed were based on handcrafted features. Among them, the Viola-Jones face detector [JV03] has been considered the state-of-the-art for a long time. The current state-of-the-art is now unsurprisingly based on convolutional neural networks, like for most computer-vision related tasks. The problem of detecting faces in regular settings can now be considered solved [FSL15, LJL15] and the focus has shifted on detecting faces in challenging conditions in terms of pose, scale or occlusion [YLLT16].

2.2.2 Face normalisation

Normalisation is an important step in the facial recognition pipelines. The performances of neural networks depends greatly on the the quality of the input images. A good facial recognition model might be unable to recognise a face if it is tilted, occluded, or without enough light. In order to tackle this, the input images are most of the time transformed into a standardized form. The most common pre-processing steps consist simply in aligning the faces using a few facial landmarks [WJ19, BT17], like the eyes position. Some works tried to go further, by transforming the input images into new face images in a standardized setting. One method is to project the initial face image into a 3D face model [MRMN16, ZLY+15, HHPE15]. Another way is to use generative models to produce an aligned image to be fed to the facial recognition model itself [TYL17, HZLH17, DG07].
However, despite how innovative these approaches seem, they do not yield good results. Any artifact appearing in the output of the generative models or 3D face models will be conveyed to the face recognition model and amplified, making them not reliable. For this reason, the state-of-the-art methods for face recognition rely mostly on training using a large number of inputs selected in various conditions, the neural networks being good at scaling up and generalizing on diverse data [SKP15].

A generic face recognition pipeline, from the pre-processing steps to the final decision, is depicted in Fig. 2.4. The decision step can vary, from a classification layer in a CNN to a k-NN classification method.

2.3 Face recognition datasets and evaluation

With deep-learning methods becoming the leading methods for face recognition problems in the last decade, the amount and quality of available datasets appeared

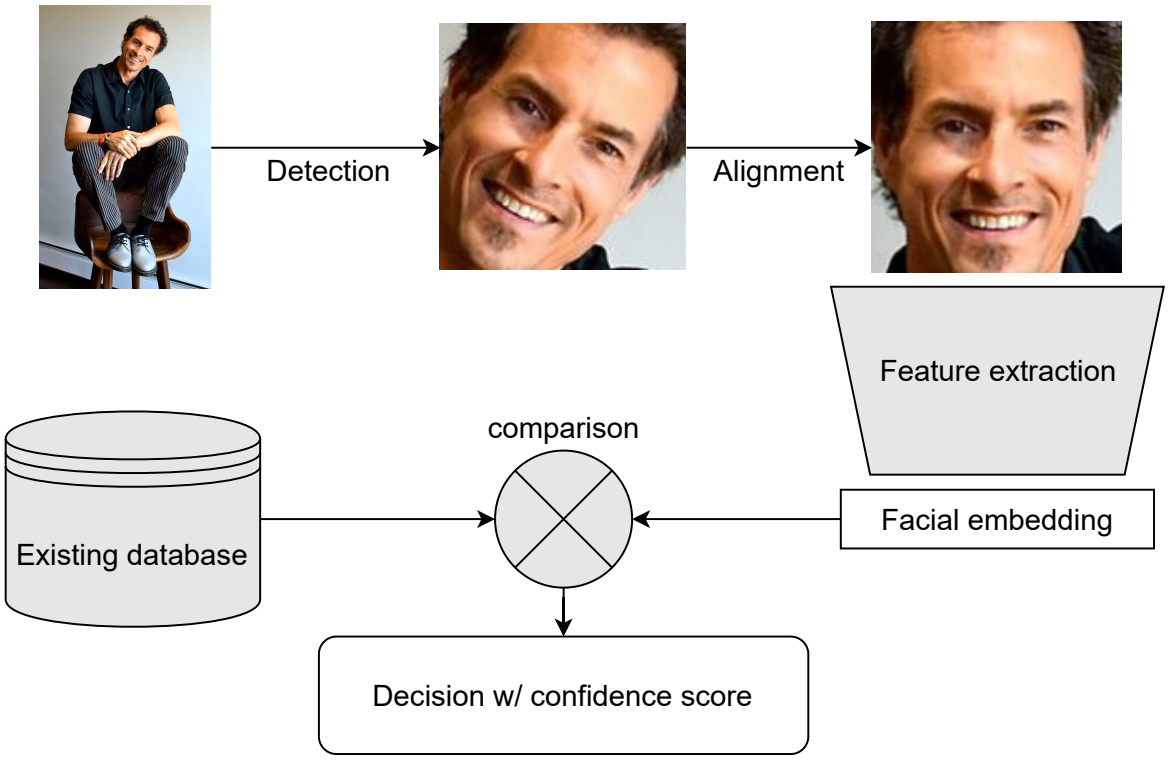

Figure 2.4: Common face recognition model pipeline.

to be critical in reaching optimal results, as the historical datasets led to a plateau of recognition performance. Many datasets have then been proposed with an ever increasing size.

Among these new larger-scale datasets, we can note the VGGFace dataset [PVZ15], containing 2.6M images equally distributed between 2,622 unique subjects. It was build by scrapping the images from search engines. Another dataset is the UMDFaces [BNC+17] dataset, that is smaller, with only about 400K images of 8,277 subjects, but contains both still images and video frames which are often more challenging. We can also note MS-Celeb-1M [GZH+16], which has impressive statistics with a total of 10M images of 100,000 subjects. However, it suffers from a lot of noise and duplicates. The VGGFace2 dataset [CSX+18] is an improved version of VGGFace with 3.3M images of 9,131 subjects. It is supposed to be more diverse but also less noisy than the original VGGFace thanks to a semi-automatic curation process. Though all of these datasets have been made public, it is also worth mentioning that some state-of-the-art results have been obtained on private sets that have not been published. Facebook has made use of a private dataset [TYRW14], but more notably the FaceNet model [SKP15] proposed by Google has been trained on a dataset of 200M images belonging to 8M different subjects. This is by far much more than any publicly available dataset has been able to offer, and this is also probably one of the reasons why the FaceNet model reached state-of-the-art performances.

In parallel to these various datasets, new benchmarks have also been published to keep up with the increasing performances of the facial recognition models and the growing size of the training sets. One early evaluation is the Labeled Faces in the Wild (LFW) benchmark [HMBLM08]. It is a face verification task, meaning that it evaluates the ability of a model to determine whether a pair of face images belong to same person or not. It is based on 13,233 images of 5,749 different people. Another benchmark is the YouTube Faces(YTF) benchmark [WHM11], which is similar to the LFW except that it is based on video sequences from Youtube and not still images. The MegaFace benchmark [KSSMB16] evaluates models both on a retrieval task (being able to retrieve images of one person in a large gallery set) and a verification task. It is much larger than other benchmarks as it uses 1M images of 690,572 subjects. Other benchmarks are the IARPA-Janus benchmarks: IJB-A [KKT+15], IJB-B [WTB+17] and IJB-C [MAD+18], which also evaluate the face recognition models on numerous and various protocols like retrieval, verification or clustering.

Chapter 3

Research Background on contextual information for person recognition

The context could be defined more generally as the set of available modalities that do not represent directly the information we are looking for, but still contain useful clues about it. In our case, we consider as contextual information any useful information that is not the face of a subject but that can help identify them. The context in general has of course been less studied than the facial recognition problem in itself. The main reason behind this is probably because a given type of context can be very application-specific, without being useful in the general case. Hence, even if there are in the literature many works considering contextual information, they do not necessarily refer to the same thing.

Among the useful contextual information for items or face recognition, we can mention the surrounding visual information, the co-occurrences of different items or people, temporal or location meta-data, and potentially any available information. This chapter is a review of how these various kind of contexts have been used in the literature in the objective of improving item or face recognition.

3.1 The context for person recognition in cognitive science

The fact that some contextual information is useful for human agents to recognize people, beside the sole facial appearances, is something that has been understood relatively early. In 1980, George Mandler introduced the *butcher on the bus* effect [Man80] after explaining how one could fail to recognize the familiar face of his butcher simply because he saw him on an unexpected context (namely the bus), while still being aware that he knew him from somewhere.

In reality, the *Context-shift decrement*, which refers to a decrease of recognition performances when the context differs from the learning to the recognition stages, has been observed not only for faces [HBTC10] but also for objects in general [HNR07]. This supports the fact that new faces or objects are systematically associated with contextual information when memorized, and that this contextual information plays a non-negligible role in recognizing people.
Similarly, in [KW82] was highlighted the fact that participants are more likely to later recognize a person if their picture comes with some textual personal information about them, than if their picture comes alone. Also, the information does not necessarily have to be repeated when the faces are shown again to be better recognized, even though it helps.
Other various contextual information such as the perceived wealth of a subject have been proven to impact how well they can be identified by other people [SYH+08]: on average, middle-class people will better recognize people they perceive as wealthy than people perceived as poor, highlighting how such an information can play a role in person recognition. The positive of negative emotions felt when introduced to a new face have also been shown to impact how it is memorized [Rai01, KM11].

Overall, it appears that the context plays a significant role for humans in the action of recognizing people, which can not be reduced to a simple face recognition task. We will detail below how such information has already been used to improve items or face recognition models.

3.2 Context for items recognition

Before being considered to improve face recognition, the contextual information has first been studied in the more generic scope of object recognition, of which face recognition can be seen as a specific case. For example, [DMS18] showed that the visual context itself contains a considerable amount of information about the objects to identify, and that this should be taken into account when performing data augmentation. In [ZBS+19, LSW20], the authors also showed that the visual context can be helpful to reach better predictions in the case of zero-shot learning.

A common approach is also to consider that if several objects are to be identified in a scene, they are likely to have some sort of semantic relationships [GB10, HGZ+18]. Likewise, in many methods proposed for the problem of action recognition in still images, the actions are not identified directly but are inferred *from the context* based on the objects identified on the image [LLX17, ZLSR17]; as we shall see in the following chapters, this can be compared to the way we can exploit the relationships between people to recognize them in our own problem of face recognition.

Some studies [GB10, Bie72, KH05] go further than focusing simply on the co-occurrences of different object categories to recognize them, but they also focus on their spatial relationships, meaning their relative positions. For example, the floor is not expected to be detected on the upper side of an image, but rather on the lower part. [GB10] also defines another type of contextual information useful in the case of object recognition: the scale context. It consists in taking into account the relative sizes of the different objects identified to make sure they are consistent. These two latter types of contextual information, if they are relevant to object recognition, are much more difficult to extend to the problem of face recognition.

3.3 Visual context

The first of the contextual modalities containing useful information for identifying a person is probably the visual context. This is something completely natural for humans: one will not expect seeing the same person when watching a football match or a political debate on TV. The visual context is so important for humans that we know that a human agent is able to achieve a score of 94.27% on the LFW protocol for face verification when all faces are masked, meaning that only the "visual context" (i.e. part of the hair, the body and the background) is left visible [KBBN11].
For comparison, when given only tight crops of the faces, the human agents reached a score of 97.53% on the same task according to that same paper, and when given the full original images containing both the faces and the visual contexts, they reached a score of 99.20%. Example of each one of these configurations are presented in Fig. 3.1. Given how limited the visual context is on the LFW dataset, where the original images are fairly centered on the faces, this shows how much information the visual context can hold about the subject's identity.

Figure 3.1: Example of a full, tightly cropped and masked pair from the LFW dataset (images from [KBBN11]).

The visual context recognition is not always studied in the scope of person recognition, but also and mostly for other purposes, like video genre classification or movie recommendation.

3.3.1 Handcrafted features

Some early studies tried to define the visual context using handcrafted features, based on lighting, color or motion to classify videos into a small amount of different categories [DMB19, Cho19] or to build a movie recommendation system based on visual similarity [DEC+16a]. In [DMB19], the proposed model aims at classifying videos as "*talkshow*" or "*other*". To do this, it uses heuristics such as the number of different shots, as well histograms of pixel colors to estimate the amount of movement. Similar heuristics are used in [Cho19, ESS10] to classify videos in a few different classes. In [DEC+16a], heuristics are based on features describing motion, lighting, color, and so on.

These handcrafted features can also be combined with automatically learned features from pre-trained neural networks [DEQC18, DCEZ+18]. In this case, numerous handcrafted features, similar to those mentioned above, and automatically learned features extracted from deep neural-networks are fused, for example with a Canonical Correlation Analysis (CCA) in order to maximize their correlation. The goal behind this is to reduce the number of dimensions by identifying correlations (i.e. redundant information) between handcrafted and automatic features. The same method can also be used to combine handcrafted visual features with categorical information [DEC+16b].

Handcrafted features require a good knowledge of the medium that is studied in order to be exploited; a heuristic designed to describe movies will probably not be suitable to categorize TV shows. Moreover, these features appear to be very limited compared to those extracted by more recent machine learning models.

3.3.2 Scenes classification with Deep Neural Networks

Later came attempts at describing the visual context of videos using neural networks trained to classify places or scenes. Several datasets have been released to that end: the SUN dataset [XHE+10] contains several hundreds of scene categories in various settings. Another similar dataset is the Places365 dataset [ZLK+17]. They both contain a large number of indoor and outdoor scene categories that are very diverse; Places365, for example, includes categories like "*igloo*", "*synagogue*" or even "*stables*". In [VN19], each frame of the input videos is classified using a classifier trained on the SUN dataset [XHE+10], and the predictions are aggregated to classify videos into 3 different video genres: "*news*", "*sports*" or "*entertainment*". To do this, the frequency of apparition of each SUN class in every one of these video genres is measured beforehand. At inference time, the distribution of SUN classes predicted by the model is compared to the expected distribution for each genre.

The large number of various categories available in scene classification datasets like SUN or Places365 [XHE+10, ZLK+17] is very interesting but does not make these datasets particularly suited to learn a visual descriptor of TV frames, as most categories are not related to TV programs. These datasets are more appropriate for identifying real world scenes, but the scenes commonly appearing in TV shows is very different: for example, there is a large number of TV studio sets, all quite similar except for some details.

The more specific approaches trying to classify videos into different video genres [VN19, DMB19, Cho19] are more related to what one can expect to see on TV; [SWBR16], for example, introduces a dataset of movie trailers for classification into 4 different genres, and shows that CNN-based approaches are more efficient than handcrafted features. However, classifying videos into only a few classes, that are not representative of the variety of TV shows, will not be precise enough when our goal is to identify people.

3.3.3 Multi-modal and joint feature space learning

The main flaw in these classification approaches is that they fail at capturing the semantics behind each visual context and the similarity and nuances between each class. A good way to access more subtle annotations, can be to make use of textual meta-data, as textual descriptions will always be richer than single class annotations. It is possible, for example, to learn a visual feature descriptor through a joint representation of handcrafted features and "tags" describing the entries [DEC+16b]. Conversely, a multi-modal approach using both text and images can be used to learn more precise words embeddings [ZPSG18]. In [CCP+18], a joint representation of input images with their corresponding caption is learned, ensuring that the visual descriptors convey semantic information. More specifically, two neural networks, one for each modality, are used and optimized jointly through a single loss function similar to the triplet loss [SKP15]. Hence, visual and textual representations for a matching pair are expected to match in the joint feature space.

In [FWL14, WALB15], a joint representation is learned for images and their corresponding tags through an objective function that maximizes the correlation between the embeddings of both modalities, but also minimizes the reconstruction loss from one modality to the other through the use of autoencoder-based methods or Canonical Correlation Analysis. This approached is illustrated in Fig. 3.2.

Another option is to use an adversarial loss function, similar to those used in Generative Adversarial Nets (GAN) [GPAM+14], coupled with another loss function aiming at minimizing the distances of the embeddings of both modality in the joint space [XHL+19, HXL+17, ZHWP19]. The adversarial loss function is here to guarantee

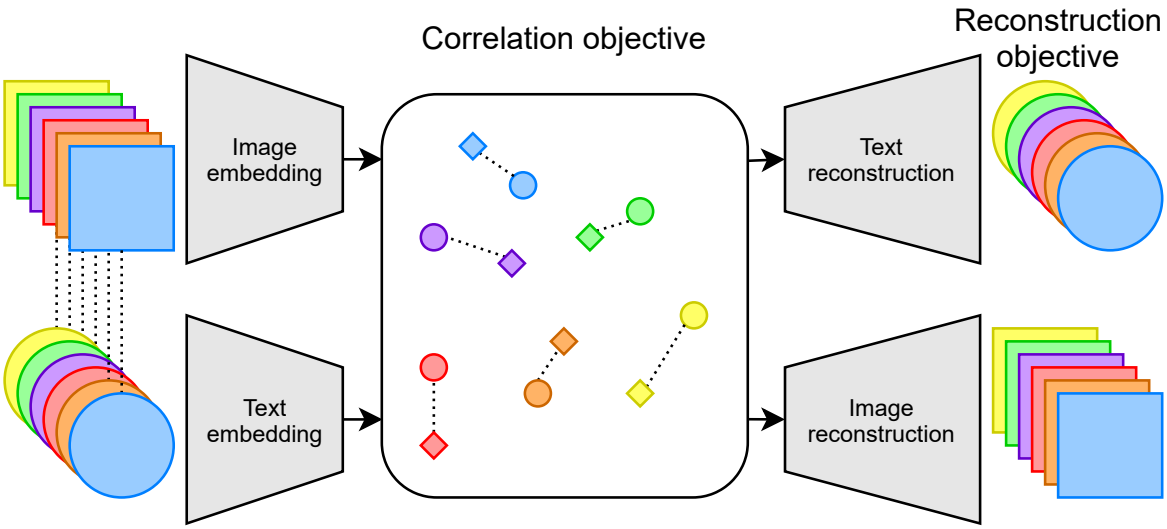

Figure 3.2: Joint representation learning for text and images as proposed in [FWL14, WALB15].

that the embeddings of both modalities joint feature space is homogeneous and that they do not lie in distinct regions of the space. These methods appear to be quite effective. Their only disadvantage is that they rely on a large amount of textual description that is not always available.

3.4 Social context

What we call the social context could be defined as the relationships between the different people we are trying to identify. This information has been much less studied than other more common modalities like the visual context. Hence, it has not been formalized clearly and this term can sometimes be used to describe different types of information. Nevertheless, the social context is an interesting source of information when trying to identify peoples on TV, where most individuals are likely to appear with their peers: politicians with other politicians, football players with their team mates, and so on.

3.4.1 Social context in social media

The first use of the social context to identify people in pictures has come quite naturally from the social media [SZD08, SZD10, MKT10]. The main advantages of social media, like Facebook, is that they can access a large amount of information to identify faces in pictures: some of them are labeled manually by the users, and can hence be considered reliable, and the relationships between the different users are also known. In [MKT10], it is shown that the co-occurrences of peoples in pictures is highly correlated with the users being "friends" on Facebook. This information is

used to identify faces in pictures where one face is already known (a "seed"). The prediction of identities for the other faces can then be limited to the first or second circle of friends of the seed, or be biased accordingly to their distance to the seed in the social network, as in the example depicted in Fig. 3.3.

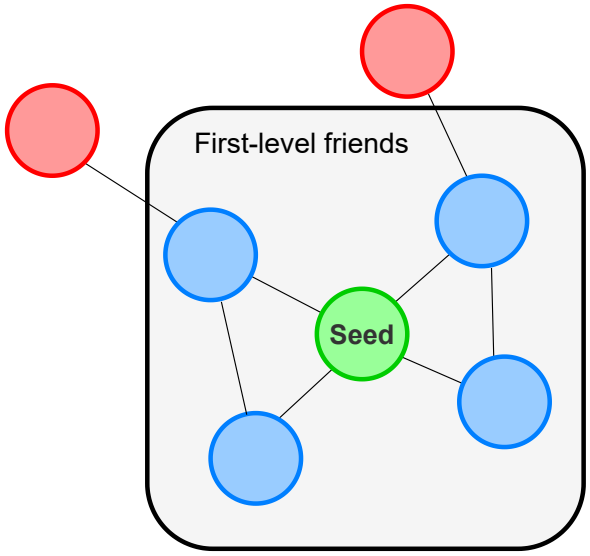

Figure 3.3: Illustration of the approach proposed in [MKT10].

However, identifying a reliable seed is crucial in this method as an erroneous one can deteriorate the results: if the results are satisfying when the seed is available beforehand, and bring significant improvement over a facial-features only model, this is no longer the case when the seed has to be estimated through an initial identity prediction on the unkown faces.

In [SZD08] the face recognition problem, is treated as a joint labelling problem and does not require to select a reliable seed. Given a picture uploaded on that social media containing n different faces $\{\boldsymbol{x_i}\}_{1\leq i\leq n}$, an energy function $E(\boldsymbol{y}|\boldsymbol{x})$ is defined for all possible joint labellings $\boldsymbol{y} = \{y_i\}_{1\leq i\leq n}$. The joint labelling problem for faces can then be described as an optimisation problem of the energy function $E(\boldsymbol{y}|\boldsymbol{x})$. This energy function itself is described as the sum of two energy functions $E_{faces}(\boldsymbol{y}|\boldsymbol{x})$ and $E_{social}(\boldsymbol{y}|\boldsymbol{x})$. The first term $E_{faces}(\boldsymbol{y}|\boldsymbol{x})$ describes the compatibility between each face $\boldsymbol{x_i}$ and its associated labeled y_i, as in a basic face recognition system. The second term $E_{social}(\boldsymbol{y}|\boldsymbol{x})$ describes the compatibility between the different identities y_i and y_j. This term represents what we call the social context.
In [SZD08], this social context term is computed based on the relationships of the identities on the social network (friends or not friends), and the number of pictures in which they have both been labeled previously. In the example depicted in Fig. 3.4, the energy of identities C, D and F is lower than the energy of A, C and E,

making them more likely to appear together.

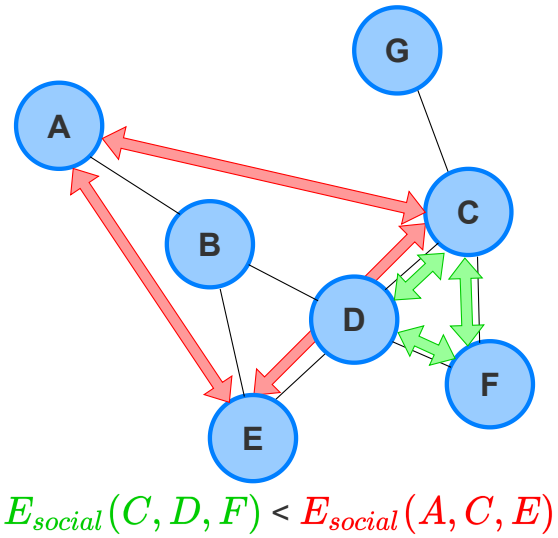

Figure 3.4: Illustration of the approach introduced in [SZD08].

These approaches achieve interesting results, but it should be noted that they take advantage from a lot of information available in social media that rarely have an equivalent in other real-world problems; in particular, pairs of peoples can rarely be described as "friend" or "not-friends" in the setting of a TV show. Most of the time, such an information can only be estimated, using for the example the number of times these people co-occur together.

3.4.2 Social context in photo albums and movies

The social context can also be exploited for identifying people in movies [HXL18] or in personal photo albums like the People In Photo Albums (PIPA) dataset [ZPT+15]. Several different contextual modalities are studied in [LBL+16] and applied on the PIPA dataset. Among them, the social context is integrated through an energy function describing whether two persons already co-occurred in the previous pictures, but giving no information about the number of times they did appear simultaneously. Hence are favored in the joint labelling of faces sets of identities that have already been seen together.

In [HXL18], the social context is two-fold: it considers both person-to-event relations and person-to-person relationships. The process here is iterative: first the identity of each face is estimated. Based on this first inference and the visual appearance of the images, each of these images are assigned to "events". In the case of movies, the events can be scenes, while in the case of personal photos, they can match real life events. Then, the people co-occurrences and the probability of each person to

appear in an event are updated using these previous estimations. These steps are iterated until convergence as shown in Fig. 3.5.

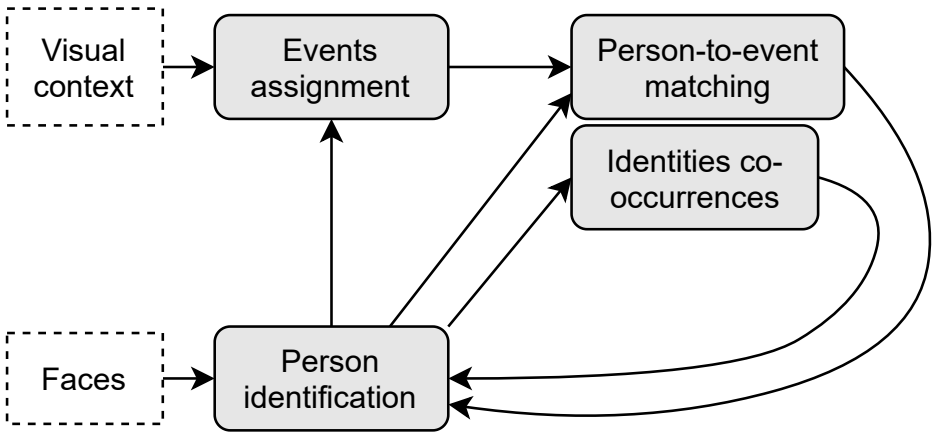

Figure 3.5: Iterative process for the social context learning derived in [HXL18].

Finally, the overall identity prediction is made similarly to what is done in [SZD08] by optimizing a unified objective function combining the visual similarities constraint, the person-to-event relations and the person-to-person relationships. One drawback of this approach is that the number of events expected to be discovered in the dataset has to be set beforehand.
A similar approach is performed in [BVS14]; all the faces appearing in the input image are analyzed simultaneously to make a first prediction based on the facial appearances only. Then, the prediction for each face is refined using the other predictions with the highest confidence scores and the co-occurrences previously observed in a gallery set.

These methods yield very good results. However, they are limited in their ability to be implemented in many problems. They can only be applied to identify a closed set of identities, and it is difficult to say how they will react to a new unknown face. Second, they have only be applied to relatively small datasets in comparison to our needs, with only a few thousands identities and less than 100K instances.

3.5 Temporal context

Temporal information and temporal pattern recognition is a domain that has attracted some attention for a wide variety of problems. However, the temporal context for face recognition is not one of them. Indeed, while the temporal information seems relevant to help identify faces appearing on TV, on which the different shows and programs follow a regular schedule, it is not the case for more common face

recognition problems. Some time analysis methods developed to solve different issues may nevertheless prove themselves useful in our case.
There are two kinds of temporal patterns that we are interested in. The first one corresponds to the periodic occurrences of people: most TV programs are broadcast regularly, on a daily or weekly basis. The second one constitute the one-time events, e.g. when one individual appears frequently over a short period of time due to some specific news event.

3.5.1 Time series analysis

Predicting future events based on past information is a common problem to which many solutions have been proposed. The most common way to deal with temporal information is through time series. In this case, we consider a series of synchronous input values. This means that the inputs are sampled at a fixed interval. For such problems considering synchronous inputs, probabilistic graphical models, like hidden Markov models [RJ86] or conditional random fields [LMP01], have been the state of the art for a long time. They make use of hidden states to propagate information from one timestep to the next.

More recently, recurrent neural networks like Long-Short Term Memory (LSTM) [HS97] or Gated Recurrent Units (GRU) [CGCB14] have become a common way to deal with sequences of synchronous inputs and progressively replaced probabilistic graphical models as the current state of the art. However, these methods are not able to process the time information itself, but only time series; they are of no use given isolated data points at a given timestamp nor can they deal with asynchronous series of data.

3.5.2 Time representation

What we are looking for is not predict future elements of time series but rather to gain information from single time points based on previous knowledge. We need to learn a time representation, or a function of time based on previous knowledge that can be interpolated or even extrapolated to new time points.

Gaussian processes [WR96, Ras03] offer an interesting solution to estimate the value of a time function for asynchronous inputs by estimating the probability density of the output values at different timestamps using the observed samples as priors. This approach can be efficient but is less reliable when fewer observations are available during some periods of time. Also, the behavior of the estimated output function depends on the choice of the kernel used. Hence, if it is possible to learn periodic functions through gaussian processes, the corresponding frequencies should be known beforehand. Other common regression techniques, like Support Vector Regression

[DBK+97], can be also be thought of. However, they will not allow to extrapolate periodic patterns that can be of importance in use cases like ours.

Some automatic methods have been proposed to learn a vector representation of time using neural networks [GA16, GG17, KGE+19]. These methods, that aims at learning a frequency decomposition of a temporal signal, are known as *Fourier Neural Networks*. In particular, [GG17] proposes a method called Neural Decompostion to decompose a unidimensional time signal into its periodic components, similarly to what a Fourier Transform can do, plus an additional non periodic component. One of its advantage is that contrary to time-series analysis method and classical recurrent neural networks like LSTM or GRU, training samples need not be synchronous. It uses a single hidden-layer neural network where each node corresponds to a sinusoid function. The optimization of the network is made by tuning the frequencies and offset parameters of these nodes.

This approach has been expanded in Time2Vec [KGE+19] for multi-dimensional output values. It is similar to [GG17] except that the output values are then fed to another neural network, possibly with additional inputs. The model takes as input a simple scalar value τ representing the time (a timestamp) and returns a multi-dimensional feature vector where each component except one is a periodic function of the input τ. More specifically, the $\boldsymbol{t2v}$ function is defined as follows for each component i:

$$\boldsymbol{t2v}(\tau)[i] = \begin{cases} \omega_i\tau + \phi_i & \text{if } i = 0 \\ \mathcal{F}(\omega_i\tau + \phi_i) & \text{if } 1 \leq i \leq k \end{cases}$$

The function $\mathcal{F}$ is a periodic function (typically, the sine or cosine functions) and the parameters ω_i and ϕ_i are learned automatically. They describe the input signal in the frequency domain. For this reason, Time2Vec could be seen as a machine learning equivalent of the Fourier Transform. Time2Vec is initially presented as a method for predicting events, by classifying timestamps, for example, through a classification layer on top of the Time2Vec model, or by combining it with a LSTM to classify sequences, as illustrated in Fig. 3.6. But many other uses can easily be thought of using the same simple idea.

3.6 Other contextual modalities

The different contextual modalities detailed here are the most frequently studied and the most relevant to our problem of face recognition in the setting of TV shows. There are of course many other modalities that can be of interest in our case or others. Here we present a few other contextual modalities that could also have been considered.

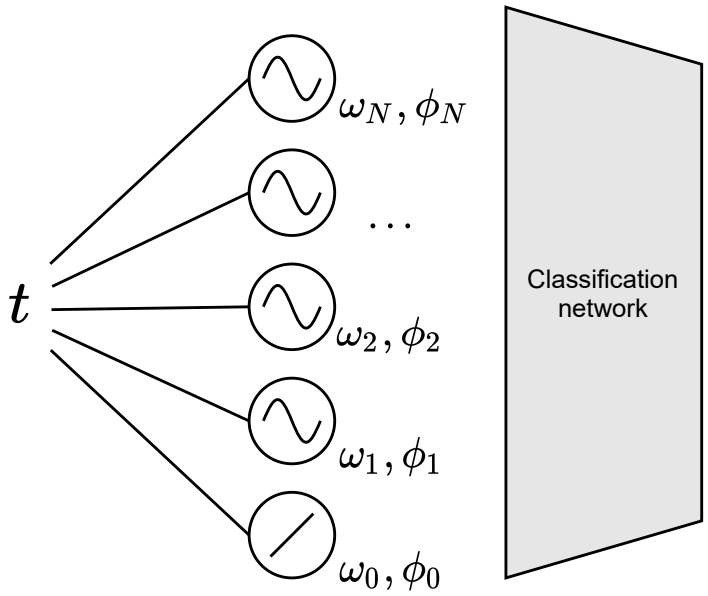

Figure 3.6: Scalar timestamps classification pipeline as proposed in [KGE+19]

3.6.1 Categorical contexts

Despite being much less frequent, there has also been a few works considering categorical contexts for face recognition. The natural intuition for such information is to deal with it using Bayesian methods like a Bayesian network. This is the approach developed in [dCN18]. The authors consider an unsupervised problem of face recognition that takes into account contextual information in the form of a finite set of discrete semantic labels. This categorical labels probabilities are assumed to be defined by a Dirichlet distribution, meaning that the probability to observe each of the possible context is proportional to their number of previous occurrences. Such an approach can be extended to infinite sets of discrete labels with a Dirichlet process [dCN], which promotes frequent previous observations but also allow for previously unseen identities to appear.

Another way to include categorical information can by transforming discrete categorical inputs into multi-dimensional embeddings, that can themselves be more easily fused with other feature vectors. This can be done in different ways depending on the kind of categorical data, from pre-trained Natural Language Processing models [DCLT18, RWC+19], to various matrix dimensionality reduction methods. The Correspondence Analysis [Ben73, Hil74] is a useful method for identifying the relationships between different categorical data. More specifically, it is usually applied to contingency tables and returns for both the rows and columns multi-dimensional embeddings in a common feature space. This is useful to identify which row inputs are important in describing a column and vice-versa.

3.6.2 Location context

In some cases, one of the interesting modalities to take into account when recognizing people can be the location. The location information can help reducing the search space for possible identities. This information can be used in photo albums [LKHB10] or in social media [TM, WYP+12], as previous locations of users can be known and some pictures are coming with meta-data such as the place and date they have been taken, or for user authentication [HE08]. The location can also be useful to identify people from video surveillance systems or for the problem of re-identification of people [MTC12] or even of vehicles [LDH+19].
Of course, in our problem of recognizing faces in TV shows, the utility of the location context is not obvious. Even if it could bring useful information in some cases like news coverage on the field, most TV shows are filmed on set and could not benefit from it.

3.6.3 Audio modality

The audio is another modality that can be rich in information about the people appearing on screen. The audio can convey information about the type of program being broadcast and hence about the people appearing in it. For example, in [DMB19], an audio classifier detects the presence of music to help classifying videos as "*talk-show*" or "*other*", jointly with other visual features. Features extracted from the audio signal have also been used for the problem of movie recommendation [DCEZ+18], here again jointly with other visual features.
We can go further and think also of speech analysis models. First, speaker recognition models [HH15, CNZ18] could be quite useful when combined with facial recognition models. Also, speech recognition [YD16] could give information about the subjects mentioned by the participants and hence about their identity.

3.6.4 Optical character recognition

Optical character recognition (OCR) can also be used to extract textual information from TV frames. Previous works showed that OCR allowed to match faces with potentially corresponding names [SV14, PBL+12]. Similarly, we can imagine a system extracting keywords (for example from overlaid banners in newscast) corresponding to the topics mentioned and containing information about who is likely or not to appear on screen.

3.7 Conclusion

While facial recognition models have achieved significant improvements over the last decade, they are still prone to suffer from some limitations affecting their per-

formances. We believe that, as some works already highlighted and similarly to how humans recognize the people they interact with, facial recognition models can greatly benefit from additional contextual information to identify people. We then identify two challenges:

- The first one is to identify which one among the contextual modalities available to us are likely to convey useful information for person recognition within TV shows.
- The second challenge is to find the more effective way to incorporate this contextual information in the model to improve our face recognition model.

In this thesis, we will study more especially the social relationships between the participants of TV shows in Chapter 5, the categorical information from the tags associated to TV shows in Chapter 6, the visual appearance of the shows in Chapter 7 and finally the time and date of broadcast in Chapter 8. Obviously, many more contexts could have been studied and are potentially useful for identifying people, such as the audio stream or optical character recognition. The contributions of this thesis are the result of an arbitrary choice, but other modalities could have been as useful and as relevant.

Part II

Contributions

Chapter 4

Base model training and datasets

As was already stated earlier, the works presented in this thesis do not aim at improving the face recognition model itself, but rather at enriching it with additional information. We need, however, to build upon an existing face recognition model. To this end, a neural network was trained to extract facial feature embeddings using state-of-the-art methods. This model, introduced in this chapter, will serve as the base model for the rest of this thesis; but the contributions introduced here could nonetheless be applied to any face descriptors extracted with another model. In this chapter, we also introduce a few datasets that will be used in the following chapter to exploit the contextual information of the archival collection of INA.

4.1 Face recognition model

4.1.1 Training

The neural network used for the facial feature extraction in Trombinos and in the following chapters is based on a Resnet18 architecture [HZRS16]. This architecture was chosen for its ease of training, which is due to both the reduced number of parameters in comparison to other networks of similar depth: the Resnet18 architecture has about 11M trainable parameters, while the one of VGG16 for example, has 138M. The simplicity of training of Resnet models is also due to the shortcut connections introduced in the residual learning building block, which is illustrated in Fig. 4.1. This building block helps mitigating the vanishing gradient issue and thus facilitates the training of deeper architectures.

The last layer of the model has been replaced with a 128-dimensional layer which is the desired output size of the facial descriptors. It takes as inputs 256×256 face images that have been centered and cropped using the OpenCV face detection network [Bra00] and aligned using the facial landmarks detector from the dlib library [Kin09]. This normalisation step consisting of centering and alignment is common

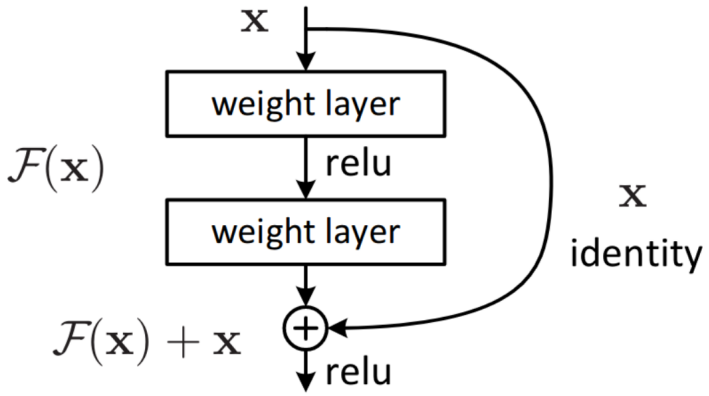

Figure 4.1: Residual learning building block (figure from [HZRS16])

and crucial for training and efficient face recognition model.

The dataset used for training is the VGGFace2 dataset [CSX+18]. It has been chosen due to its large size (3.3M images and more than 9k unique identities), but also because this is a dataset that offers an excellent trade-off in terms of diversity and data purity: the diversity in pose and expression is essential in order to be able to recognize faces in real-world situations (i.e. *in the wild*), while the data purity is required to obtain sufficiently consistent results at convergence.

The objective function for training is a triplet loss function [SKP15] with the hard triplet mining strategy as described in [SMN+17]: each batch is formed from the face images of one class and the faces images of other similar classes, also called "doppelgangers". This way, the formation of hard triplets is made easier and allows for a quicker convergence of the model.

After training and hyper-parameter optimisation, our 128-dimensional facial descriptors are evaluated on the *Labeled Faces in the Wild* (LFW) face verification protocol, on which they achieve a score of 98.98%. Though satisfying, this score remains lower than the state-of-the-art results claiming to reach an accuracy score of 99.63% [SKP15]. Such a score was however reached using more than 100M face images of about 8M identities, from a private dataset. The performance we obtained is close to the best we can achieve using available academical datasets and is satisfying enough to be used as a base model for our future experiments.

4.1.2 Inference

At inference, the face query is passed through the trained network and its facial feature descriptor is compared to a database of previously annotated descriptors.
The classification approach is based on a k-NN: first are retrieved the k most similar

face images in the gallery set. Then, each one of these nearest neighbors contributes with its own identity label through a voting score decreasing with its distance to the query according to a gaussian function. This way, a higher score is associated to the identity labels assigned repeatedly to instances similar to the query.
Because our gallery set suffers from a long-tail distribution, with a handful individual being over represented on the TV and the vast majority appearing only a few times, this method can lead to an imbalance and a bias in favor of high-profile identities. To avoid this, the score of each identity is then normalized by the total number of occurrences of this identity in the gallery set.

The score $\boldsymbol{s}_l(x)$ associated to the label l for a query x is computed as stated in Eq. 7.2, where $l(y)$ is the label associated to element y, $d_f(x, y)$ is the euclidean distance between facial feature descriptors of x and y and $delta_{l(y),l}$ is the Kronecker delta of $l(y)$ and l.

$$\boldsymbol{s}_l(x) = \frac{\sum_{y \in \text{kNN}(x)} \delta_{l(y),l} * e^{-\frac{d_f(x,y)}{2\sigma^2}}}{\text{card}(l)} \tag{4.1}$$

4.1.3 Trombinos

The facial recognition model, introduced above, and that has been used throughout these works, has been integrated into an application named *Trombinos*. The goal of *Trombinos* is to allow for a fast and easy annotation of the faces detected in the TV archival collection, through an automatic labelling joined to an efficient manual correction.

A few screenshots of Trombinos are displayed in Fig. 4.2 to 4.5. Images or video files can be uploaded for the faces to be identified and analyzed (see Fig.4.3). Opening one identity page allows to retrieve each frame (see Fig. 4.4) or even each TV show (see Fig. 4.5) in which the person appears.
The automatic annotations can be curated manually, one by one, grouped by show, or by clusters of similar faces discovered automatically. Because the classification of new inputs is made through a k-NN based approach, the manual correction of hard cases will help improve the performances on future queries, as will do the manual annotation of yet missing data, without the need to retrain the face recognition model and without recomputing the existing facial descriptors.

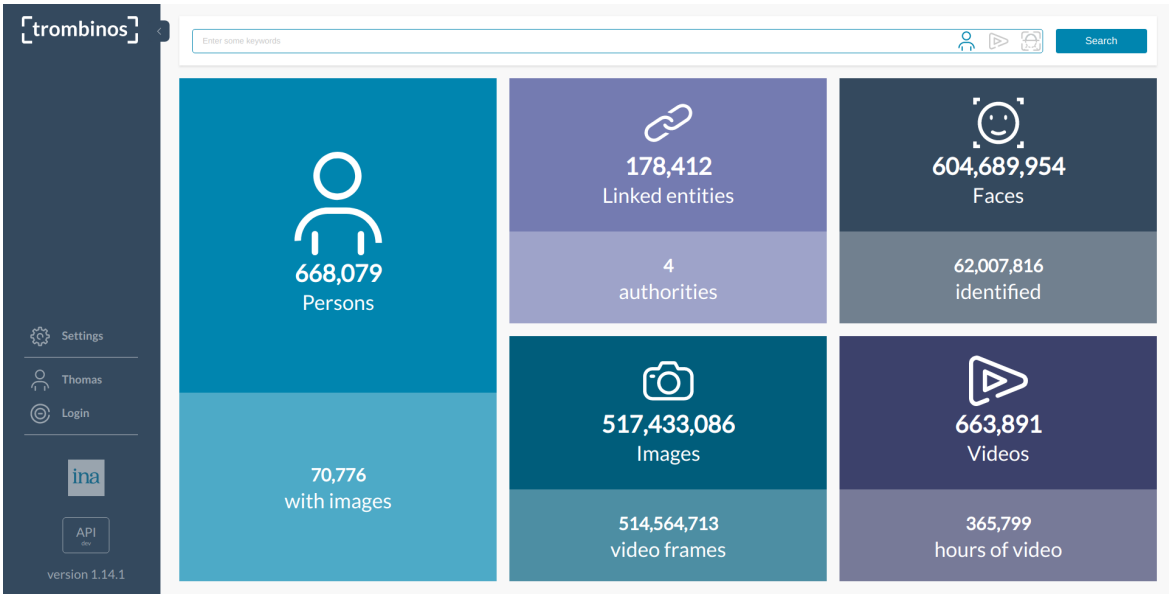

Figure 4.2: Trombinos dashboard.

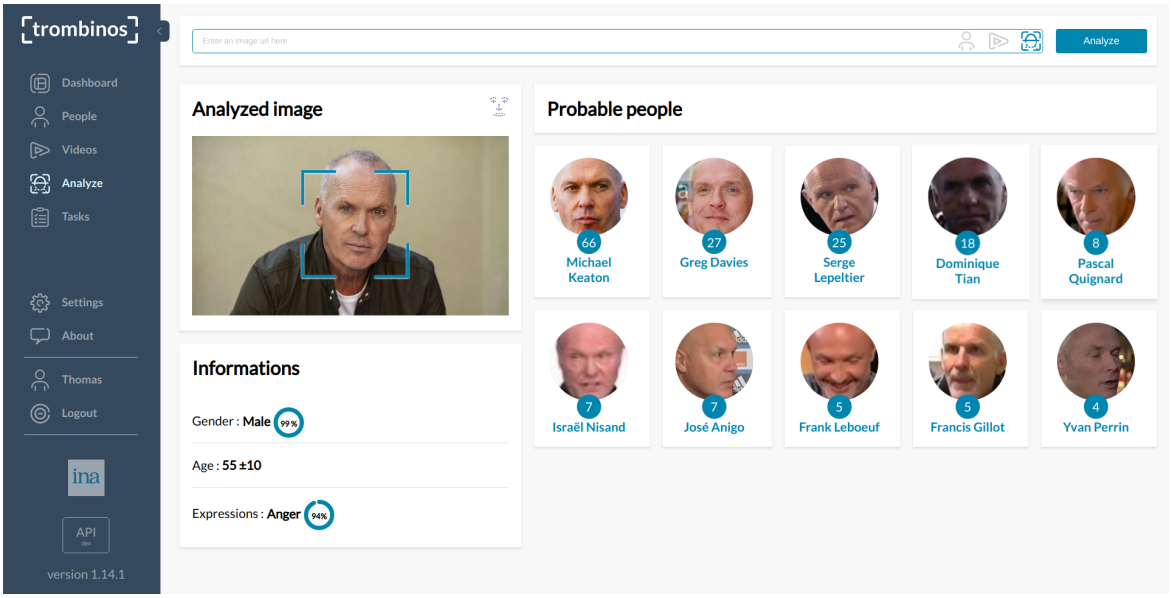

Figure 4.3: Trombinos image analysis results screen.

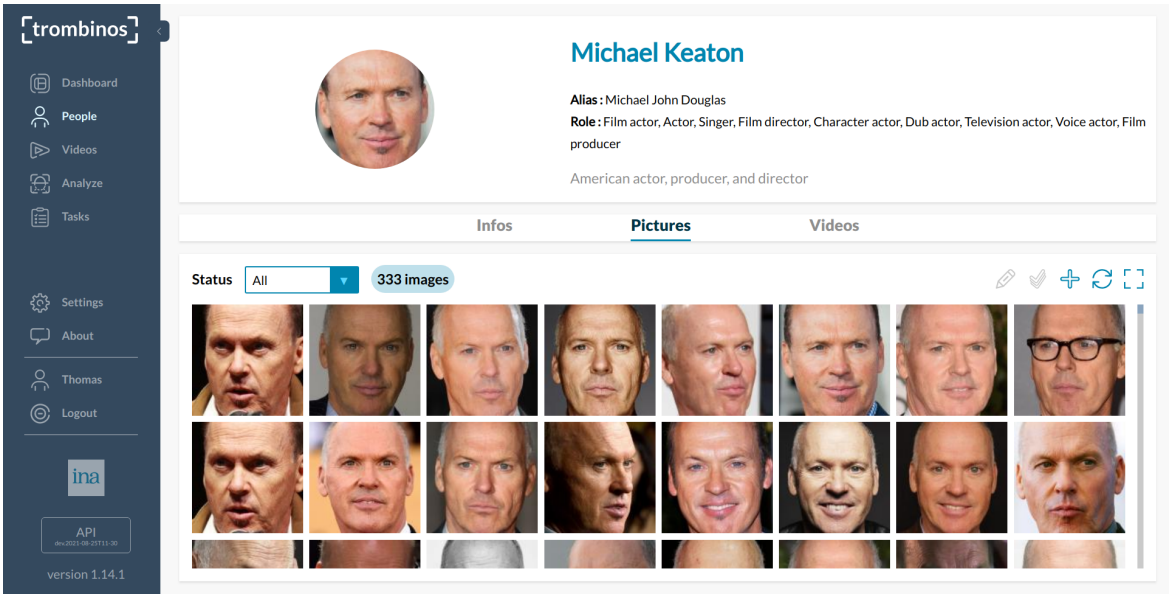

Figure 4.4: List of detected faces associated to Michael Keaton by Trombinos.

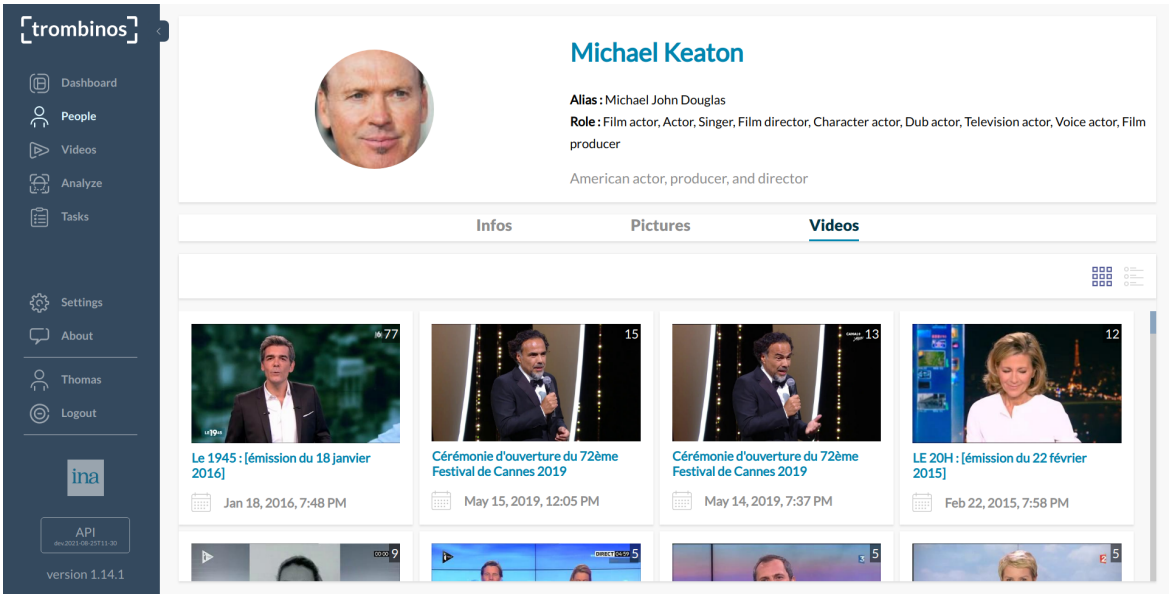

Figure 4.5: List of TV shows in which Michael Keaton has been identified by Trombinos.

4.2 Available data

Because we want to develop a model that takes advantage of the specificity of the archived data at INA (more especially the correlation between the appearing people, the visual appearance of the frames, the time of broadcast, and so on), we need a dataset, for both training and evaluating this model, displaying the same features.
The first and most severe limitation is that the faces in the INA television archives are not annotated directly at the frame level; at most, a few names can be assigned at the show level without further specification.

It is therefore necessary to build a satisfying dataset that will allow us to evaluate the impact of the various contextual modalities on the recognition of identities on actual data. In the context on my thesis, several datasets have been built and shared to this end.

4.2.1 Scrapped face image dataset

The first dataset that has been built is a dataset of faces of the most common identities to appear in the TV archives. The goal of this dataset was to be used later as a reference for annotating faces from the archives.

This dataset contains 2.8M images from 70K unique identities amongst the most common ones appearing in the archived TV shows. The images have been curated semi-automatically in order to ensure the purity of the dataset.

It has been collected as follows: The names of all of these identities have been searched on the search engine Qwant, and the 100 first returned images have been scrapped. All the faces have been detected in these images, and using the facial feature descriptors computed by the model described above in Section 4.1, a clustering is applied on the faces returned by the search engine to remove undesired results. The predominant clusters are conserved while the other ones are filtered out. In the case where there is no predominant cluster, or two predominant clusters of similar sizes, all the images are discarded in order to avoid incorporating noise into the dataset.

Finally, the faces assigned to different identities have been compared to each other in order to identify possible errors in the annotations: duplicates are removed and ambiguous pairs of classes, for which the distances between facial descriptors obtained with a pre-trained model was low, are curated manually. Overall, about 9K of the total 70K identity classes have been curated manually.
Some examples of instances from this dataset are displayed in Fig. 4.6.

Figure 4.6: Examples of some instances from the classes *Donald Trump* (left) and *Elise Lucet* (right) from the dataset described in Sec. 4.2.1.

4.2.2 Co-Occurring Faces in TV dataset

In order to study, among other modalities, the co-occurrences of people in TV shows, we selected a subset of the dataset of scrapped face images described above (Section 4.2.1) and assigned these face images to real TV shows and their corresponding meta-data. This dataset is consisting of 548,686 instances of 42,655 unique persons. All of those instances are distributed over 138,381 TV programs. These programs are organised in order to contain images of people who did appear in the same TV show at least once, particularly to capture the co-occurrence relationships between them.

The dataset is built from a subset of the first dataset described above (Section 4.2.1) and from a list of TV programs with their corresponding meta-data. More especially, this meta-data contains a non-exhaustive list of people appearing in these particular shows. Typically, a TV program will be assigned with none to 10 identities for the most detailed ones. We filtered out the TV programs with less than 2 assigned identities so that we are able to exploit this dataset for studying the people co-occurrences in common shows.
To each TV program in our list is assigned, when available, a face image of each identity that appears in it. No instance can be assigned to more than one program. If all instances of one person have been assigned, that person is no longer considered in the upcoming programs.

Two examples are displayed in Fig. 4.7: 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 program (six faces on the left). Another program 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 faces on the right).

Figure 4.7: Example of two programs with their associated face images from our dataset of Co-Occurring Faces in TV.

Finally, we obtain 548,686 instances (i.e. face images) of 42,655 unique identities distributed between 138,381 programs. Between 2 and 55 instances are assigned to each program, with an average of 3.96 instances per program. Every unique identity has on average 12.86 instances.

Because the TV programs used to build this dataset are real TV shows broadcast between January 1990 and January 2020 on French TV, it contains co-occurrences of persons typical of the French television. With each TV program in our dataset come not only face images of the person appearing in it, but also real meta-data, such as the date and time of broadcast, some categorical "tags" describing the program such as "News", "Sports", or "Entertainment", and other some other information.

To our knowledge, this is the largest public dataset of faces to contain such a large amount of information and meta-data. It is particularly suited to learn contextual information at the TV program level (co-occurrences of participants, descriptive tags, date and time of broadcast, etc.). It is freely available online[1].

This dataset is used for both training and evaluation in the Chapters 5 and 6.

4.2.3 INA archival collection

Using the facial feature descriptors introduced in Section 4.1, we analyzed about 366K hours of TV shows broadcast on various TV channels between 2010 and 2020. It is a total of 514,564,713 video frames that have been analyzed and over 600M faces that have been detected and for which facial features vectors have been computed. These TV shows have a large variability in their channel, thematics, type of

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

programs, in order to fully encompass the diversity of TV shows in general. Some example frames are displayed in Fig. 4.8.

Figure 4.8: Examples of frames from the 366K hours of processed TV shows.

The numerous faces of this large dataset are not annotated. Nevertheless, this dataset remains useful and rich in information: because it consists of the real data our final goal is to analyze, it also includes all of the available contextual modalities that can prove useful for person identification. Among them, we can mention once again descriptive tags, date and time of broadcast (at the frame level), but also the visual context.

Visual Context for TV Programs dataset

While studying the visual context for TV frames for person recognition, we used a subset of 10,684,217 of the analyzed frames from these 366K hours of TV shows. These frames are organised in triplets, with positive pairs and random negative elements. Examples of positive pairs, visual context-wise, are visible in Fig. 4.9. The dataset is also split into a training, a validation and a test set. More details and justifications on how this subset was built are given in the Chapter 7. The main purpose of this dataset was to allow for a visual context descriptor model to be learned. It is public and is completely available[2].

Figure 4.9: Examples of two positive pairs from our Visual Context for TV Programs dataset.

[2] https://dataset.ina.fr/

Doppelgangers face verification dataset

Another subset of frames from the INA archival collection has been sampled to form a face verification task: It consists of 2,922 pairs of faces, with an equal amount of positive and negative pairs. The positive pairs are made of two face images of the same person, while the negative pairs are made of face images of two distinct, though similar looking people (an example of a negative pair is given in Fig. 4.10). To achieve this, similar face images have been retrieved from a selection of queries and each one of the retrieved faces has been checked manually. This way, we ensure the sampling of adversarial negative pairs while making sure that no annotation error will impact the performance score. The set of frames used in this dataset is completely distinct to the frames used in the Visual Context in TV Programs dataset mentioned above.

Figure 4.10: Example of doppelgangers pair used for the face verification task.

Annotated faces from TV shows

We have been able to automatically annotate a subset of the INA archival collection through a comparison with the scrapped face images dataset (Section 4.2.1). It is a total of 62M (about 10% of the 600M faces detected in the analyzed frames) that have been annotated with the same 70K labels of the scrapped face images dataset.

4.2.4 Limitations of the available datasets

Because our "real-world" data are not labeled, the evaluation of our various methods proposed and introduced in the following chapters is made quite difficult.
Since the set of contextual data and meta-data we are trying to exploit is quite specific to the archival TV collection of INA, they can hardly be evaluated on other similar, publicly available, dataset.

For this reason, we use the reconstructed dataset of Co-Occurring Faces in TV for training and evaluation in the Chapters 5 and 6: because this dataset artificially

reconstructs real data and contains faces images that have been semi-automatically (meaning partially manually) curated, we assume the number of erroneous labeling to be low and we consider it to be safe enough to use for both training and evaluation. However, being reconstructed from the existing, non-exhaustive, annotations, this dataset can be seen as a simplified version of the actual data. More especially, being reconstructed from incomplete annotations, this dataset is more likely to focus mostly on the most frequent and famous people and not on the lesser known identities.

This reconstructed dataset, on the other hand, contains only cropped face images and for this reason can not be used when studying the impact of the visual context of TV frames. For this reason, the Visual Context for TV Programs dataset has been used to train a neural network able to learn a consistent visual context descriptor, using a process that does not rely on a proper and full annotation of each element.
For evaluating the impact of using the visual context for recognizing identities, on the other hand, a dataset with both and available visual context and labeled faces is required. To do this, we used the Doppelgangers face verification dataset introduced in Section 4.2.3 in both the Chapters 7 and 8. This face verification protocol uses manually annotated faces, which guarantees a low number of erroneous labelling. It is however of a limited size, and annotating manually enough faces to raise this number of an order of magnitude is highly impractical.
In order to evaluate our works on a task similar to our real-world problem and on a comparable scale, we used a subset of the annotated faces of the INA archival collection. 1,125,704 instances labeled with 13,032 unique labels are used for a k-NN based classification, in Chapters 7 and 8. This scale is much closer to the one we ultimately aim for. Nevertheless, despite using the scrapped face images dataset as a reference, which is itself partly manually annotated, and a conservative labelling to avoid erroneous annotation of ambiguous instances, we can not guarantee the absence of erroneous annotations on a dataset of this size. Results of this classification should hence be considered with a grain of salt and mostly for reference, while the doppelgangers face verification task is more reliable.

Chapter 5

Social context

5.1 Introduction

Among the contextual modalities available to us to identify people on TV, a very powerful one is the social context, meaning the social relationships between the subjects. These relationships can be described by the frequencies of co-occurrence of the 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 and that you have previous knowledge of their relationships, than if you see them alone. The same applies to many politicians, football players, actors... (see Fig. 5.1).

Figure 5.1: Examples of strong social contexts.

If we have indeed seen some works exploring this idea to improve face recognition systems (see Section 3.4), 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 program alongside other people, we want to retrieve all other instances of the same person in a large dataset of

TV programs.
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 into three main steps:

- We use a soft clustering on face descriptors to approximate the ground truth entities with clusters of faces.
- We build a probabilistic co-occurrence matrix to map the clusters to a contextual feature space capturing their co-occurrence relationships.
- 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 programs 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. The results presented in this chapter have been partly discussed in [PLDG21b].

5.2 Dataset

Our motivation in this chapter is to be able to retrieve faces in a large dataset of TV programs by exploiting both the visual features of the faces observed in them and their social context. 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, smaller dataset of face images, annotated semi-automatically and organized to reflect the co-occurrences of faces observed under real conditions.

This dataset and its building process have been detailed in the Section 4.2.2. It contains 548,686 instances of 42,655 unique identities distributed between 138,381 programs. Since each TV programs has been assigned with at least 2 face images, and about 4 on average, it makes this dataset particularly suited to learn the co-occurrences of the different identities in the same TV shows.

This is the dataset we use to evaluate our method on.

5.3 Methodology

5.3.1 Main approach

In order to exploit social relationships between entities to better recognize each face, we first need to identify those entities. In previous works, the set of entities is most of the time already known[HXL18, ZPT+15], as the ultimate goal is not to retrieve other faces, but to classify a query in a set of known and fixed identities, for which examples of social relationships are known. In this case, the frequencies of co-occurrences can easily be computed for each pair of identities and used directly. In our case, however, we do not dispose of this amount of information: the training set is assumed to be unknown, meaning that the faces are not labeled and that even the number of labels is unknown.

An overview of our approach is represented in Fig. 5.2. We call X the set of instances in our dataset, and S the list of programs, which are themselves subsets of X. For practical reasons and for readability, we will refer interchangeably as X for both the set of faces observed in all programs, and the set of facial features extracted from these faces with our model.

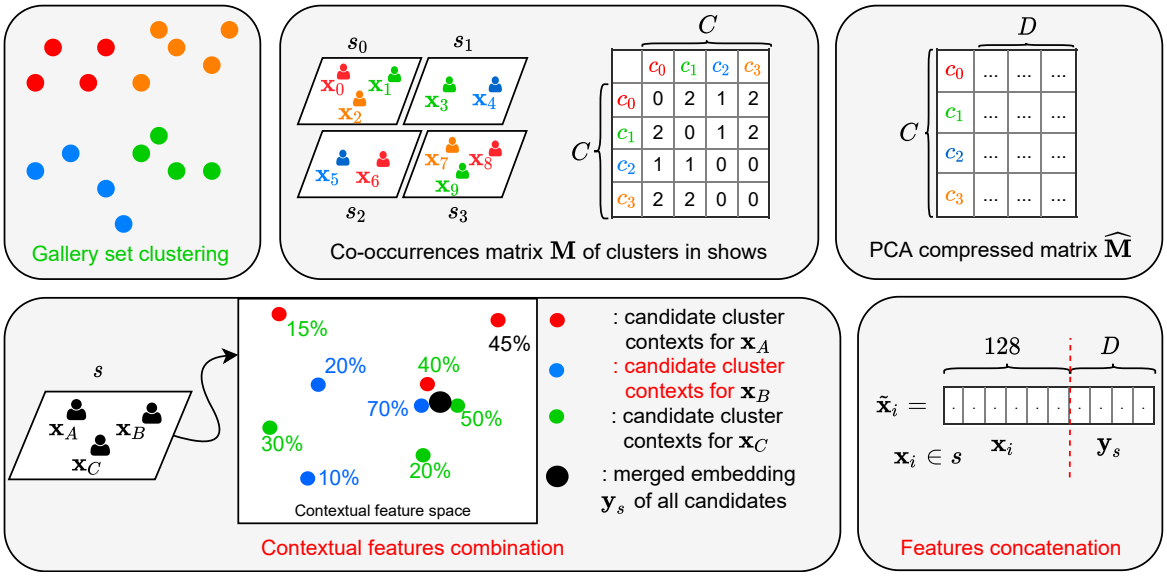

Figure 5.2: Our proposed pipeline for social context-aware face recognition. Similar faces appearing in the different programs s_i are first grouped together into C clusters c_0, c_1, etc., using our approximate soft-clustering approach operating on visual feature vectors $\boldsymbol{x}_i$. Cluster co-occurrences are registered in a specific co-occurrence matrix $\boldsymbol{M}$ which is used to learn a contextual descriptor matrix $\widehat{\boldsymbol{M}}$. For each program $s \in S$, the candidate contextual descriptors of all faces $\boldsymbol{x}_i \in s$ are then merged into a common descriptor $\boldsymbol{y}_s$ that best represent the whole show. Finally, the retrieval of faces is performed on the concatenation $\tilde{\boldsymbol{x}}_i$ of the facial descriptors $\boldsymbol{x}_i$ and the contextual descriptors $\boldsymbol{y}_s$.

As the real identities of all of our instances are 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 is then used as a proxy for the list of real identities, and we infer the social relationships from the co-occurrences of these identified clusters.
Different clustering methods for approximating the real identities are compared in section 5.3.2. We also show that using a soft-clustering approach can lead to interesting results; instead of assigning each instance with unique cluster labels, they are assigned with a probability distribution over the identified clusters. This allows to better handle outliers that are prone to bring noise in the upcoming steps.

Once the set of identities has been approximated with clusters, each one of them is mapped to a multidimensional feature space, so that clusters containing many co-occurring instances are associated to vectors close in the new feature space, while clusters with no co-occurring instances are mapped to distant feature vectors. This part is detailed in section 5.3.3.
To each program $s \in S$ we can now assign a contextual feature $\boldsymbol{y}_s$ that combines the contextual descriptors of all the clusters appearing in that program. The new combined contextual feature $\boldsymbol{y}_s$ should be representative of every instances of s while being robust to outliers. We compare two methods to combine the contextual features of the clusters appearing in a program in subsection 5.3.4. The first method is based on the average value, while the second one is based on the geometric median, which appears to be particularly efficient when used jointly with the soft-clustering approach.

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

The full pipeline is detailed in Algo. 1.

Algorithm 1 Unsupervised face-recognition with social context

1: Apply clustering algorithm on training dataset ▷ Sec. 5.3.2
2: Build co-occurrence matrix $\boldsymbol{M}$ based on the clustering results ▷ Sec. 5.3.3
3: Reduce the matrix $\boldsymbol{M}$ to $\widehat{\boldsymbol{M}}$
4: **for** query $\boldsymbol{x} \in$ program s **do**
5: Compute the probability distributions $[\boldsymbol{c}(\boldsymbol{x_i})]$ for all the faces $\boldsymbol{x_i} \in s$
6: Compute $\boldsymbol{y}_s$ based on $\widehat{\boldsymbol{M}}$ and $[\boldsymbol{c}(\boldsymbol{x_i})]$ for $\boldsymbol{x_i} \in s$ ▷ Sec. 5.3.4
7: Define $\tilde{\boldsymbol{x}}$ as the concatenation of $\boldsymbol{x}$ and $\boldsymbol{y}_s$
8: Make a new k-NN based prediction from $\tilde{\boldsymbol{x}}$
9: **end for**

5.3.2 Clustering methods

The choice of the clustering algorithm used to identify the possible set of identities in the gallery set is critical. Being the very first step in our pipeline, every error of clustering will become noise that will propagate throughout the following steps. Hence, we need to find the optimal balance between 1) having as many instances as possible assigned to a cluster, as we require as many information as we can get for our pipeline to be effective, and 2) reducing to the bare minimum the number of wrongly clustered instances.
In order to find that balance, we compare three different clustering approaches. The first one is a clustering method that assigns every instance to a cluster. The second one is a clustering method that leaves out outliers. Finally, the third one is a soft clustering approach that assigns each instance to several clusters with different probabilities.

These three approaches and the hypothesis we made are detailed below.

HDBSCAN

The first one is the HDBSCAN (for Hierarchical Density-Based Spatial Clustering of Applications with Noise) clustering method [MH17], which is based on DBSCAN [CMS13]. Its main advantage is that it does not require the number of expected clusters to be specified, as it is the case in many 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 $\boldsymbol{c}(\boldsymbol{x})$ can now be defined as one-hot encoders ($\boldsymbol{c}(\boldsymbol{x})_i = 1$ if $\boldsymbol{x}$ has been assigned to cluster i, 0 otherwise). If an instance $\boldsymbol{x}$ is considered as an outlier, then $\boldsymbol{c}(\boldsymbol{x}) = \mathbf{0}$. This means outliers are not accounted for when building the co-occurrence matrix $\boldsymbol{M}$ (see section 5.3.3).

Hard-clustering without outliers

In the second approach, we assign all points to a cluster. The HDBSCAN clustering algorithm is applied to the dataset and the data points identified as outliers are assigned to their nearest clusters identified. The vectors $\boldsymbol{c}(\boldsymbol{x})$ can once again be defined as one-hot encoders. Because there is no outlier, we have more available data to build the co-occurrence matrix $\boldsymbol{M}$ (see section 5.3.3) and to combine contextual feature vectors for each program (see section 5.3.4); however, these additional data points are more likely to be noisy.

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 distributions of all instances are not computed across all of the 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 $\boldsymbol{\Sigma} = \sigma \boldsymbol{I}$, 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: $P(c_i|\boldsymbol{x}) \propto P(\boldsymbol{x}|c_i)$. For an instance $\boldsymbol{x}$ and a cluster c_i with mean value $\boldsymbol{\mu_i}$, the estimated probability $P(c_i|\boldsymbol{x})$ is computed as follows:

$$P(c_i|\boldsymbol{x}) = \boldsymbol{c}(\boldsymbol{x})_i \propto \begin{cases} \exp\left(-\frac{\|\boldsymbol{x} - \mu_\mathbf{i}\|}{2\sigma^2}\right) & \text{if } \mathrm{c}_i \in k\text{-nearest clusters of } \boldsymbol{x} \\ 0 & \text{otherwise} \end{cases} \tag{5.1}$$

In practice, the probabilities we are dealing with are not normalized, nor do they sum up to one. Since we consider our problem to be an open problem, the real cluster to which $\boldsymbol{x}$ belongs may be unknown, so the probability distribution over the identified clusters does not have to sum up to one.

5.3.3 Co-occurrence matrix

After clustering X (detailed in subsection 5.3.2), C clusters are identified. For each instance $\boldsymbol{x} \in X$, we define a vector $\boldsymbol{c}(\boldsymbol{x}) \in \mathbb{R}^C$ that denotes the posterior probability of each identified cluster given $\boldsymbol{x}$:

$$\boldsymbol{c}(\boldsymbol{x})_i = P(c_i|\boldsymbol{x}) \text{ for } \boldsymbol{x} \in X \text{ and for } i \in \{1...C\} \tag{5.2}$$

that is, $\boldsymbol{c}(\boldsymbol{x})$ represents the probability mass function over all clusters for a given instance.
We then define a probabilistic co-occurrence matrix $\boldsymbol{M} \in \mathbb{R}^{C \times C}$ as follows:

$$\boldsymbol{M}_{i,j} = \begin{cases} \sum_{s \in S} \sum_{x_1, x_2 \in p, x_2 \neq x_1} \mathbf{c}(\mathbf{x_1})_i \mathbf{c}(\mathbf{x_2})_j & \text{if } i \neq j \\ 0 & \text{otherwise} \end{cases} \tag{5.3}$$

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

$$\begin{aligned} \boldsymbol{M}_{i,j} &= \textstyle\sum_{s \in S} \sum_{x_1, x_2 \in s, x_2 \neq x_1} P(c_i|\boldsymbol{x_1}) P(c_j|\boldsymbol{x_2}) \\ &= \textstyle\sum_{s \in S} P(c_i, c_j|s) \end{aligned} \tag{5.4}$$

In the case of hard clustering approaches, the $\boldsymbol{c}(\boldsymbol{x})$ vectors being one-hot encoders, the elements $\boldsymbol{M}_{i,j}$ for all i, j are simple counts of the number of co-occurrences of the clusters i and j.
When using a soft-clustering approach, however, the $\boldsymbol{M}_{i,j}$ element is the expected value of the number of programs in which an instance assigned to cluster i co-occurs with an instance assigned to cluster j, according to the probabilities returned by the soft-clustering algorithm.

As C can be quite large, the matrix $\boldsymbol{M}$ is rather sparse, and contains a great number of zeros as well as some redundancy. Further, each line/column of $\boldsymbol{M}$ represents a contextual description that is very long compared to the visual feature vectors. Thus the simple concatenation of both would lead to an imbalance that would degrade the final retrieval performance. Therefore, a Principal Component Analysis (PCA) is performed on $\boldsymbol{M}$ and the D most important components only are kept ($D < C$), transforming $\boldsymbol{M} \in \mathbb{R}^{C \times C}$ into a new matrix $\widehat{\boldsymbol{M}} \in \mathbb{R}^{C \times D}$. Each cluster c_i, for $i \in \{1...C\}$ is now associated with a (compressed) contextual feature vector $\widehat{\boldsymbol{M}}_i \in \mathbb{R}^D$.

Because the matrix $\boldsymbol{M}$ describes the number of co-occurrence of the identified clusters, it can also be seen as a weighted adjacency matrix of the graph describing the clusters relationships. Therefore, the matrix raised at the power p, $\boldsymbol{M}^p$ describes the co-occurrences relationships between the different clusters at the order p. This is easily visible for $p = 2$, where strong first-order relationships with common third clusters leads to a strong second order relationship:

$$M^2_{i,j} = \sum_k M_{i,k} M_{k,j}$$

Some experiments have been conducted by replacing the matrix $\boldsymbol{M}$ by $\boldsymbol{M}^2$ before reducing it, in order to see if the reduced matrix $\hat{\boldsymbol{M}^2}$ better captures the social relationships between the clusters. However, in our experiments, this did not give any sensible results. Thus we retained the approach using only first-order relationships.

In Fig. 5.3 are displayed the cosine distances between descriptors associated to different clusters obtained after reduction of the co-occurrence matrix. The clusters themselves are associated to different TV personalities; for convenience, the clusters are labeled with the name of the personalities they best describe. This does not mean that the clusters are pure (faces of other people might have been assigned to these clusters), nor that all faces of these personalities are assigned to them (some of their faces might have been assigned to other clusters). The clusters selected corresponds to different identities appearing on TV on similar occasions, in order to easily display the similarity between their social descriptors. We can hence distinguish amongst these examples 4 groups of social descriptors:

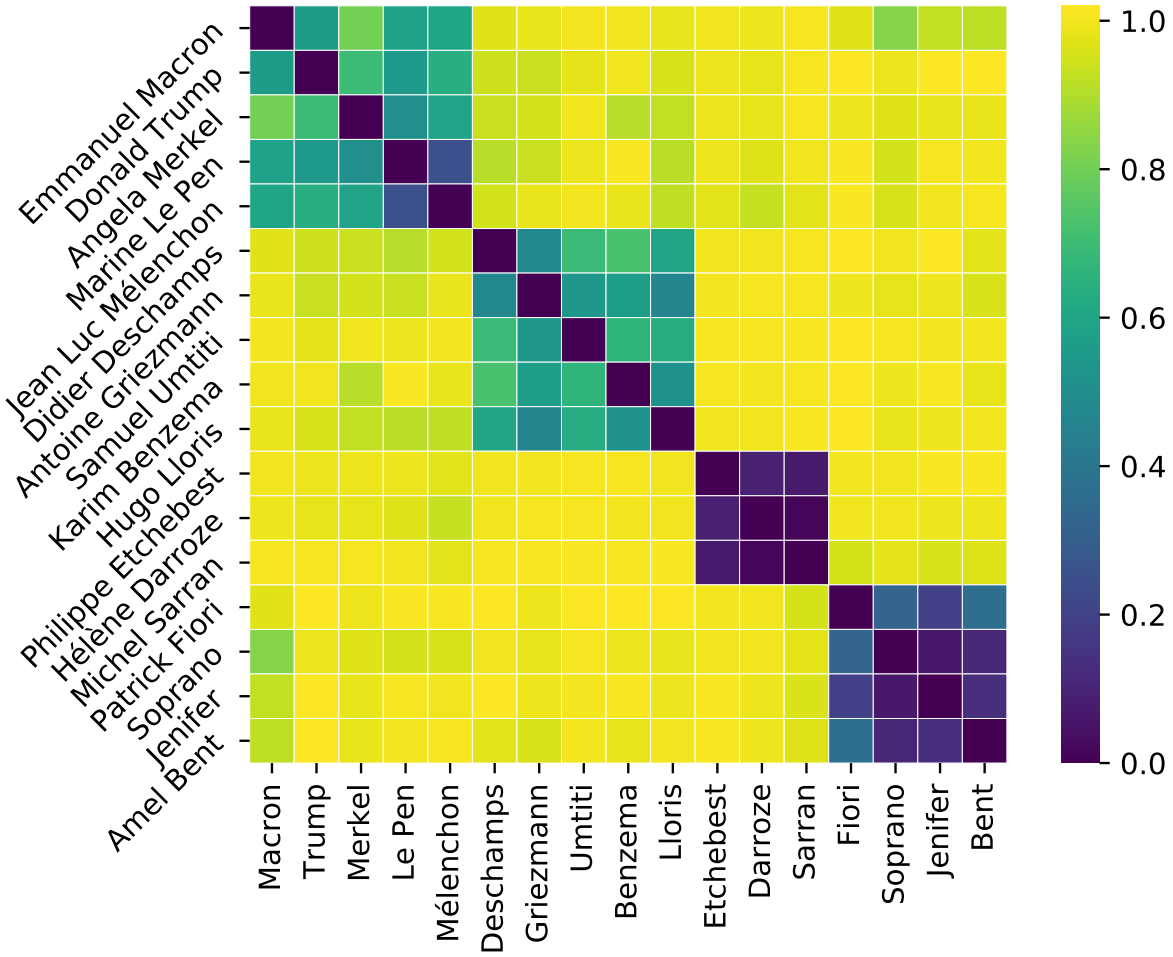

Figure 5.3: Cosine distance between clusters associated to TV personalities after reduction of the co-occurrence matrix.

The first one contains politicians (Macron, Trump, Merkel, and French opposition members). Even if their descriptors are not extremely close, because they do not appear often together, their descriptors still appear to be quite similar compared to other people.
The second cluster of contextual descriptors contains French footballers from the national team. The fact that they also play separately in their own clubs mitigate their proximity, but it is still quite clear that they belong to the same context.
The last two clusters are very close because they contain people that appear almost exclusively together: the first one contains jury members of a cooking TV show, while the second one contains jury members of the French edition of TV talent show "The Voice". Because they appear almost always together, their contextual descriptors are highly similar.

5.3.4 Combining contextual features for each program

For each program s, the list of clusters occurring in s is the list of clusters to which the instances of that program are assigned: $\{\boldsymbol{c}(\boldsymbol{x})\}_{\boldsymbol{x}\in s}$. These clusters have a contextual feature vector that embeds their co-occurrence relationships, as we defined just above. In order to define a contextual descriptor for the program itself, we propose to aggregate the contextual descriptors of its assigned clusters. Different approaches are possible to that end.

Average value (or center of mass)

The first, straightforward approach consists in simply computing the feature vector $\boldsymbol{y}_s$ of program s as the average feature vector of the clusters appearing in it:

$$\boldsymbol{y}_s = \frac{\sum_{\boldsymbol{x}\text{ in }s}\sum_{i=1}^{C}\boldsymbol{c}(\boldsymbol{x})_i\widehat{\boldsymbol{M_i}}}{\sum_{\boldsymbol{x}\text{ in }s}\sum_{i=1}^{C}\boldsymbol{c}(\boldsymbol{x})_i} \tag{5.5}$$

This value $\boldsymbol{y}_s$ is also the center of mass of the contextual descriptors of the clusters appearing in s.

Geometric median

The second approach consists in computing the feature vector $\boldsymbol{y}_s$ not as the average value, but as the geometric median of the feature vectors of the clusters occurring in s. This allow for $\boldsymbol{y}_s$ 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.

$$\boldsymbol{y}_s = \text{argmin}_{\mathbf{y}\in\mathbb{R}^D}\sum_{\boldsymbol{x}\text{ in }s}\sum_{i=1}^{C}\boldsymbol{c}(\boldsymbol{x})_i||\widehat{\boldsymbol{M}_i}-\boldsymbol{y}||_2 \tag{5.6}$$

The algorithm used for inferring the geometric median is the one detailed in [VZ00].

The difference between these two options is illustrated below in Fig. 5.4. In this example, the weighting is uniform across the data, which is not necessarily the case in practice. Most of the points seem to form a cluster, while 3 of them are really distant from the other ones and should probably be considered as outliers. We can see that the center of mass (in orange) is much more responsive to the outliers in comparison to the geometric median (in green), which is more robust and more representative of the actual cluster, once the noise has been filtered out.

Note that for both approaches, the feature vector of each cluster i is weighted by the probability $\boldsymbol{c}(\boldsymbol{x})_i = P(c_i|\boldsymbol{x})$ for $\boldsymbol{x}\in s$. This means that in the case of soft clustering, the k-nearest clusters identified in 5.3.2 are accounted for.

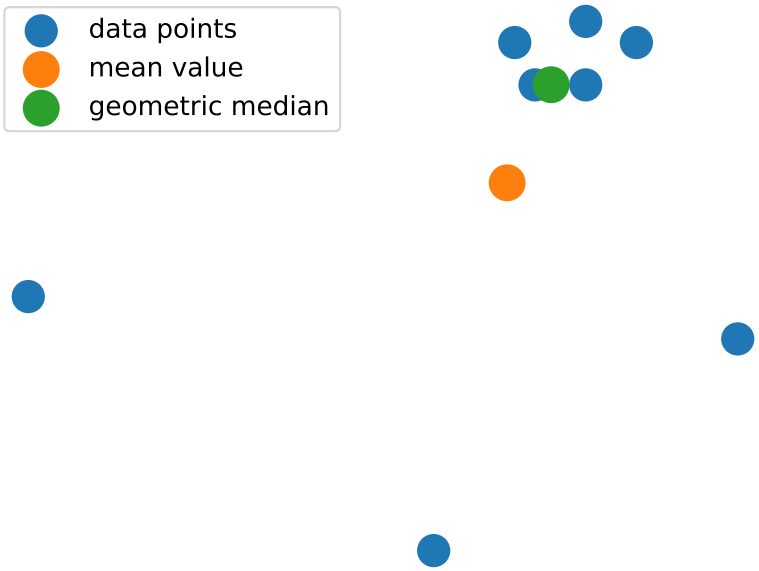

Figure 5.4: Example of the mean value (in orange) and geometric median (in green) for a set of data points (in blue) in 2 dimensions.

With the HDBSCAN clustering, it is possible that no instance of a program s has been assigned to any cluster: $\sum_{\boldsymbol{x} \text{ in } s} \sum_{i=1}^{C} \mathbf{c}(\mathbf{x})_i = 0$. In this case, $\boldsymbol{y}_s$ is set to the average value of all clusters contextual feature vectors.

5.4 Experiments

5.4.1 Experiments setup

Our dataset of faces distributed in TV programs is partitioned into two disjoint sets: a training set of 137,381 TV programs, containing 544,863 faces, and a test set of 1,000 TV programs of 3,823 faces, 3,770 of which belonging to people also appearing in the training set. We apply the pre-processing steps on the training set (i.e. clustering of its instances, computation and reduction of the co-occurrence matrix. See above section 5.3).
Since our approach is unsupervised, and requires no labeling of the training data, it can be used to identify faces from the training set (what we will call in the rest of this chapter internal queries), as well as new faces foreign to that training set (external queries). In the case of internal queries, this means that these queries belong to the set of instances clustered at the beginning of our pipeline and that they participate in the co-occurrence matrix. Differences of performances are then to be expected between these two cases.

Our method is hence 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.
We evaluate the performances of our approach on a retrieval task, and 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 5.3.2) applied on the training set, and the different feature vector combinations explained in 5.3.4.

5.4.2 Baseline

As a baseline, we use the model for facial features extraction described in section 4.1. 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.

5.4.3 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 5.1. 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.

Table 5.1: mAP over 9,820 internal queries under different configurations, for 300 additional components.

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

Figure 5.5 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 in-

creasing slightly for a contextual features dimension D higher than 300.

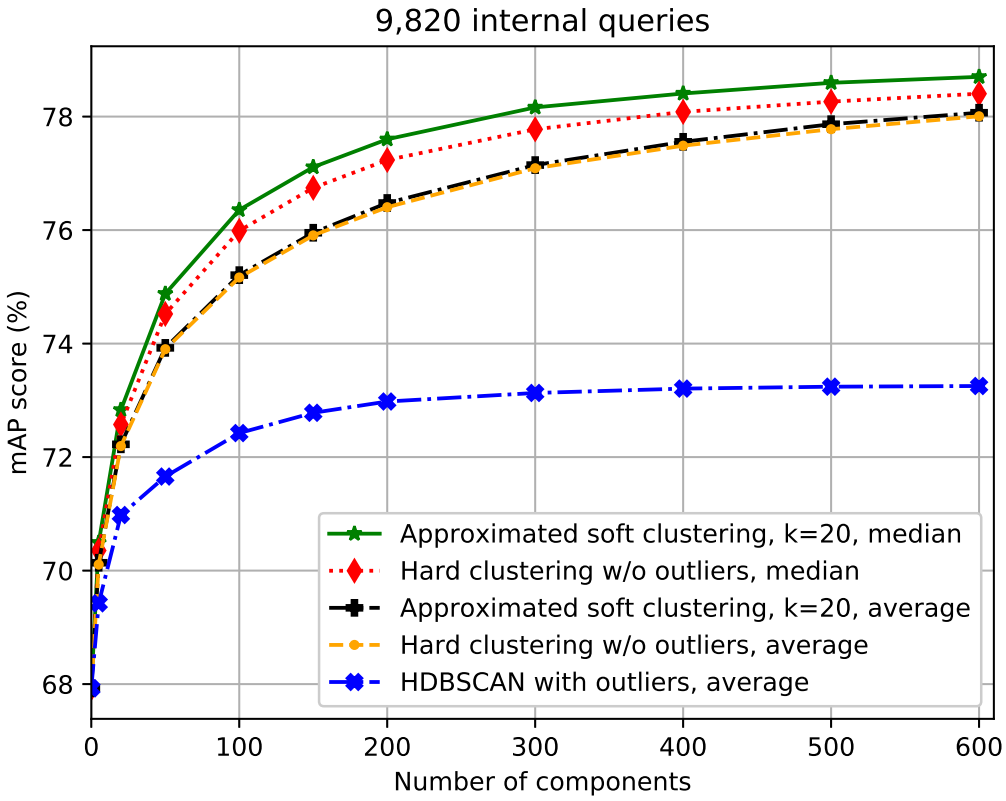

Figure 5.5: Mean Average Precision over the internal queries as a function of the number of additional components from the contextual features, for different clustering methods and configurations.

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 5.1) when computing a program 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 (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

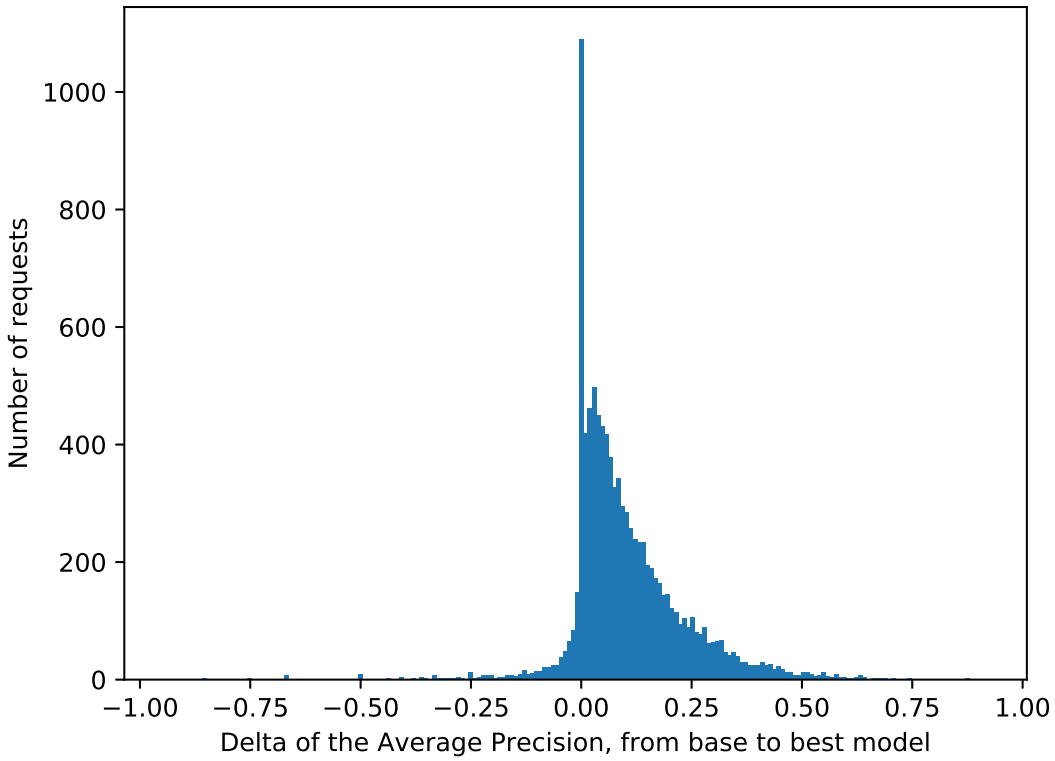

Figure 5.6: Profile of variation of the Average Precision between the baseline and the best model on the internal query set.

queries. It decreased for 782 other queries and remained unchanged for the 713 remaining ones. The profile of variation of the average precision of all queries can be seen in Fig. 5.6.

Impact of the number of instances co-occurring with the query.

We show that the variations of the Average Precision for a given query does not depend on the amount of people occurring in the program in which it is observed. The improvement is visible for programs with only 2 persons and remains stable for programs with more people (see Fig. 5.7).

Impact of the number of instances of the queried identity in the dataset.

As can be seen in Fig. 5.8, 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

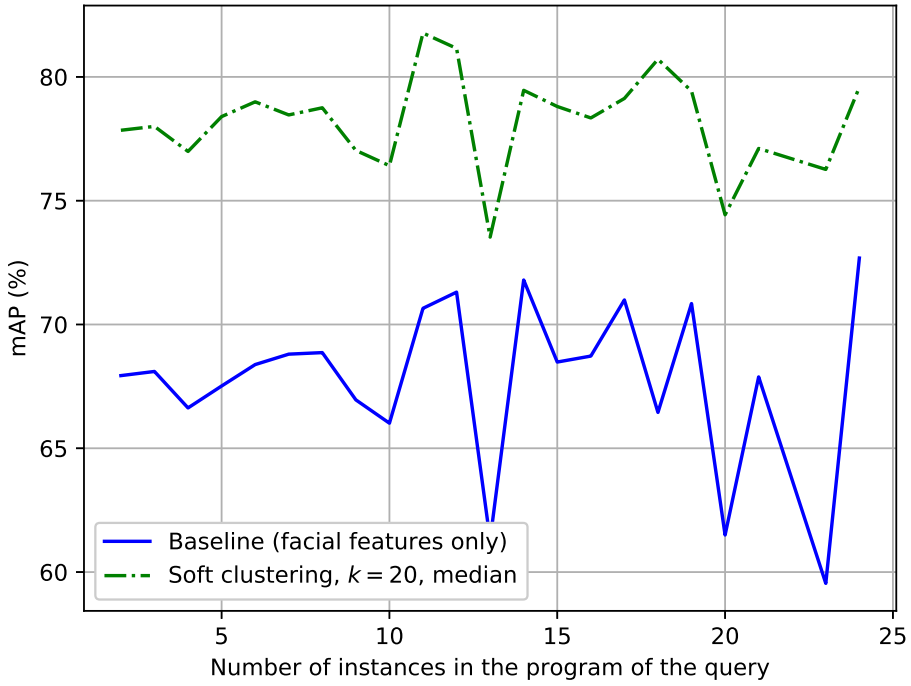

Figure 5.7: Mean Average Precision as a function of the number of people occurring in the program of the query, for the baseline and our best configuration.

is 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.

Impact of the parameter k for the approximated soft-clustering on the results.

In our soft clustering approach, the probability $P(c_i|\boldsymbol{x})$ of an instance $\boldsymbol{x}$ to belong to a cluster c_i is approximated as described in 5.3.2 and Eq. 5.1. 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 the facial recognition model introduced in 4.1), 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. The evolution of the mAP on our query set for different values of k is represented on Fig. 5.9. Except for that parameter, all results are obtained in the same configuration: the contextual feature vectors have 300 components, and the geometric median is used to combine each program contextual features. Even though the mAP seems to increase quickly for values of k ranging between 2 and 20 before reaching a plateau, the results obtained for all values of k remain very close.

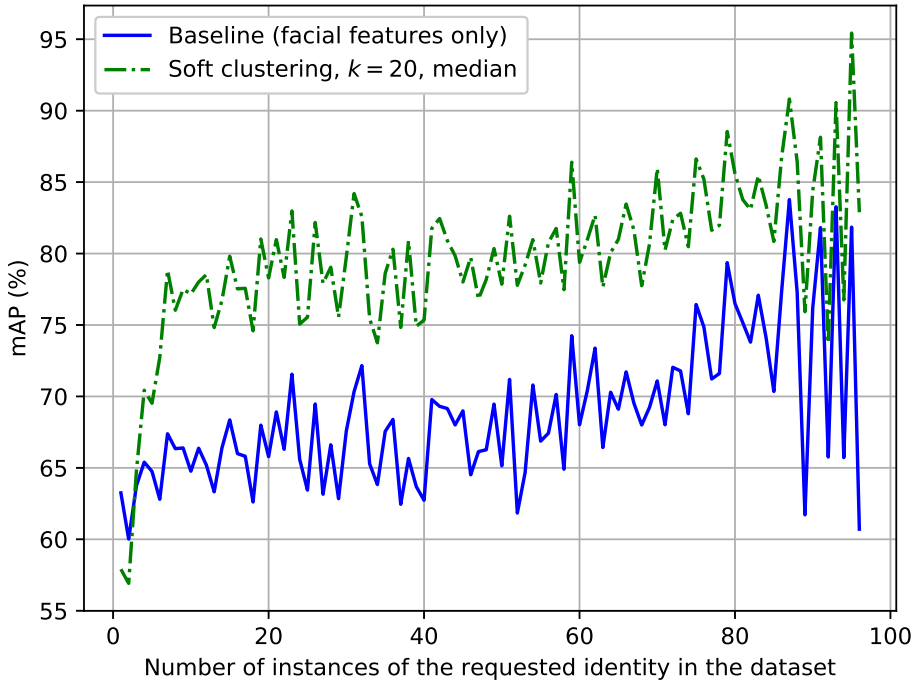

Figure 5.8: Mean Average Precision as a function of the number of instances of the requested identity in the dataset, for the baseline and our best configuration.

5.4.4 External queries results and analysis

External queries are also coming grouped into programs. Queries from a single program are identified simultaneously, so that they can leverage from each other. These queries need first to be assigned or soft-assigned to the nearest clusters identified in the gallery (training) set. The contextual features of the external programs 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 5.2. The mAP is computed on the retrieval of the combined vectors $\tilde{\boldsymbol{x}}$. Similarly to what has been done with internal queries, the dimensionality of the contextual features $\boldsymbol{y}_s$ 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. 5.10. 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.

5.4.5 Supervised setting

Because the whole pipeline described in Algo. 1 is quite long, we also decided to evaluate the performances obtained on our retrieval task in the case of a supervised

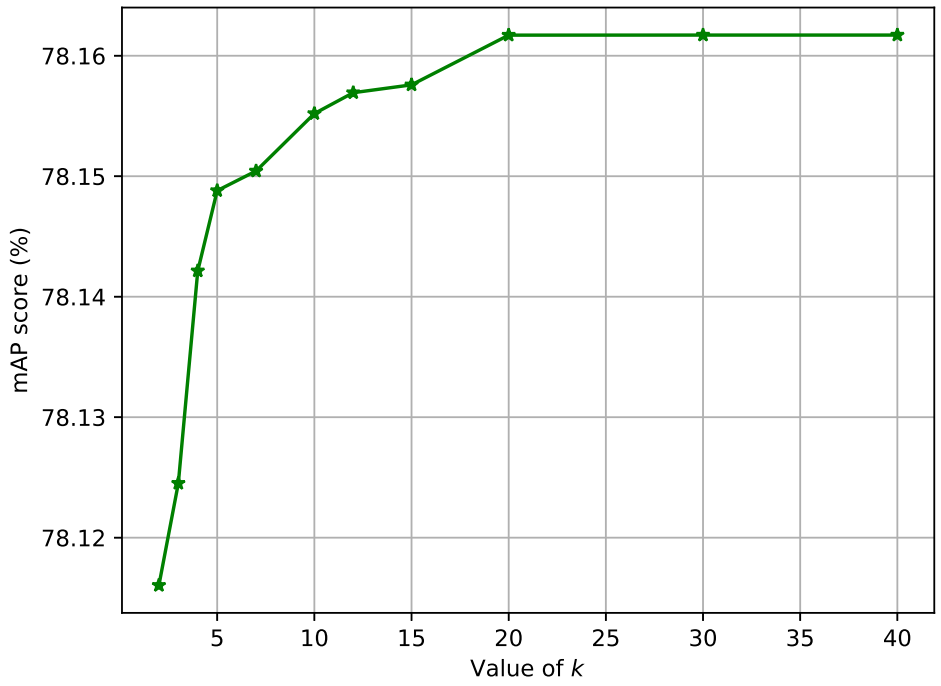

Figure 5.9: Mean Average Precision as a function of the number k of nearest clusters used for approximating the soft clustering. The number of contextual components is 300 and the features are combined with the geometric median.

setting.
In this case, we consider the identities of the training set to be known. This means that we can build the co-occurrence matrix $\boldsymbol{M}$ with its real value and not approximate it through a clustering (or soft-clustering) algorithm applied beforehand. This could be seen in fact as the particular case of a perfect clustering.

The rest of the pipeline is the same:

- The matrix $\boldsymbol{M}$ is reduced to $\widehat{\boldsymbol{M}}$
- at inference time, the probability distribution $[\boldsymbol{c}(\boldsymbol{x_i})]$ of each face $\boldsymbol{x_i}$ in the query program s to belong to the real clusters are computed
- the social descriptor $\boldsymbol{y_s}$ of s is computed
- the retrieval is computed on $\tilde{\boldsymbol{x}}$ the concatenation of $\boldsymbol{x}$ and $\boldsymbol{y_s}$

The evaluation is performed on the set of external queries also used in Section 5.4.4. The retrieval performances are measured as above through the mAP score. The results obtained for various number of components are displayed in Fig. 5.11.

Table 5.2: Mean Average Precision over 3,770 external queries under different configurations, for 300 additional components.

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

The performance gap between the supervised setting on the unsupervised setting is large: at about 300 additionnal components, the mAP score in the supervised setting is 10% higher than in the unsupervised setting. We can deduce from this that even though the unsupervised approach is useful, as it leads to proper improvement in comparison to the facial features only case, much information is lost while clustering the training set. On the other hand, this also means that sensible improvement is reachable if a more appropriated clustering algorithm, or more precise facial feature descriptors were to be used.

5.4.6 Computational cost

We here compare the computation times for retrieving external queries with facial features extracted with the model only, and with our approach for $k = 20$ and $D = 300$. The gallery set clustering and the computation of contextual features of all instances from this gallery is done offline. The facial features extracting time is not accounted for as it is the same for both approaches and as our model could be replaced with any other. We use the FAISS library for exhaustive similarity search [JDJ19] on a 2.1 GHz CPU.

When retrieving a new external query with the baseline model, the retrieval of the 3,770 128-dimensions facial features of the external queries in our gallery set (that is processed offline), takes 26.08s, which is about **145 queries per second** (for a mAP of 67.86%).
With our approach, we first need to identify the k-nearest clusters in the gallery of the query and of other instances co-occurring with it, which takes 2.80s; we then combine the contextual embeddings of all selected clusters and their probabilities with the geometric median, and concatenate them to the facial descriptors, which takes 7.88s. Finally, the retrieval of the combined feature vectors takes 34.70s. Overall, retrieving the 3,770 external queries in our dataset takes 45.38s, which is about 83 queries per second (for a mAP of 77.36%). Note that our method can be sped

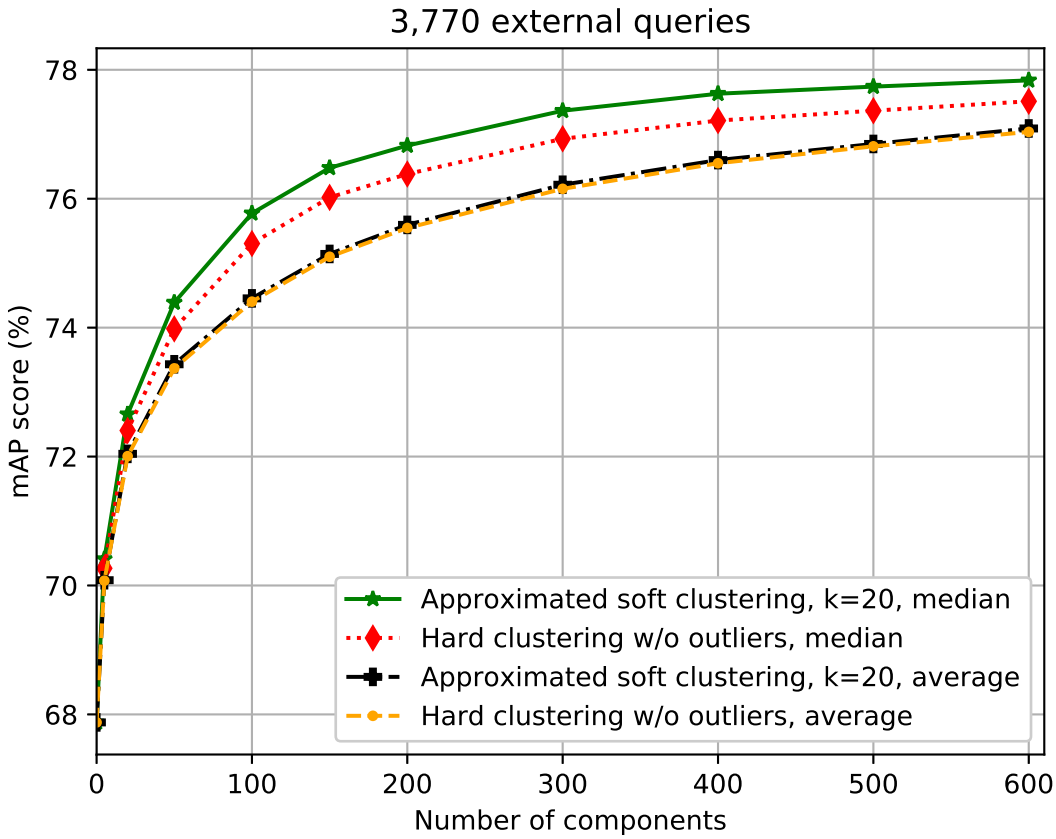

Figure 5.10: Mean Average Precision over the external queries as a function of the number of additional components from the contextual features, for different clustering methods and configurations.

up by reducing the values of the parameters k and D. For example, with $k = 5$ and $D = 200$, we can process about **104 queries per second** while maintaining a mAP of 76.80%.
Depending on the choice of the parameters values, our method is 1.3 to 1.8 times slower than the basic approach while increasing the mAP by about 10% points.

5.5 Conclusion

In this chapter, we described an unsupervised method aiming at exploiting the expected social relationships between the participants of TV shows to better recognize them simultaneously. Using a previously trained model to extract facial feature vectors, we identify for each face detected in a given TV show a set of possible identities with their corresponding probabilities. The social embeddings of these identities, learned beforehand from the gallery set, are then combined together into one that best represents all of the faces. This merged social embedding is then used alongside

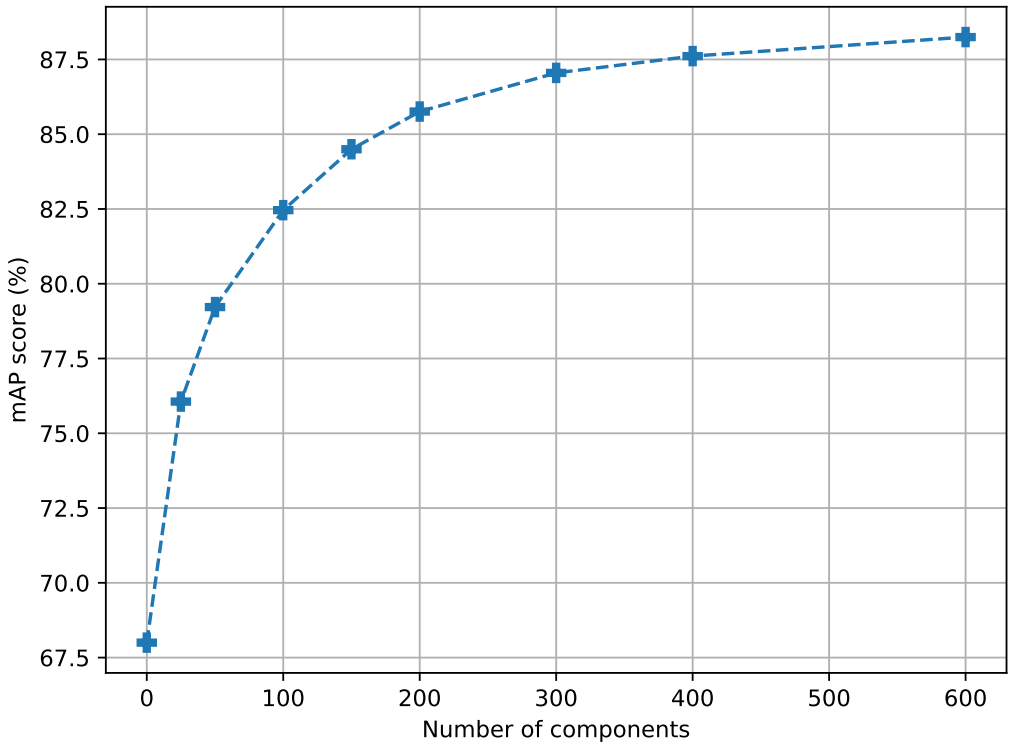

Figure 5.11: Mean Average Precision in the supervised setting as a function of the number of additional components from the contextual features.

the facial feature descriptors in order to identify the faces consistently with their facial appearances and the prediction of the other faces. 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 programs. 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.
Using additional data in the gallery set (assuming that the identities are known in this gallery set), we observed another leap of performance. This suggests that a more suitable clustering algorithm could possibly yield better results in the fully unsupervised case, or that more precise facial feature descriptors could also be beneficial.
The approach as it is proposed in this chapter is not able to deal with new unknown faces, as previous observations are needed for the social context to exist. We could however imagine an incremental approach, where a new unknown face contextual embedding could be approximated by its co-occurring faces. This way we would be able to recognize it in the future without applying again all the pre-processing steps. This idea could be investigated in future works.

Chapter 6

Categorical context

6.1 Introduction

Amongst the contextual information available to us, before trying to identify people appearing in TV shows, come textual metadata in the form of tags. They describe roughly the TV shows to which they are attached with topics (such as "Sports", "Politics", "Music", and so on) and genre (ranging from "TV news" and "Documentary" to "Talk show" or "Sketch"). Similarly, the different channels on which the shows are broadcast are also known and may contain useful information about the kind of people we would expect to observe on screen. Some examples of possible tag annotations are displayed in Fig. 6.1

In this chapter, we investigate how these categorical data coming in the form of tags could be used to improve the performance of our model for the task of face recognition or face retrieval. We compare these results with those obtained previously on the social context in Chapter 5 and try to evaluate how redundant both of these modalities are to determine if it is worth exploiting them simultaneously.

Figure 6.1: Examples frames from TV shows. The frame on the left comes from a show labeled with the tags "Magazine" and "Sports". The frame on the right come from a show labeled with the tag "Music".

6.2 Dataset

In order to experiment on the usefulness of the categorical metadata for the face retrieval task, we use the dataset introduced earlier in Section 4.2.2 and used for evaluation on the social context in Chapter 5. The same partitioning is used, with a training set of 137,381 unique TV programs among which are distributed 544,863 faces and a disjoint test set of 1,000 TV programs containing 3,770 query faces. Similarly as the experiments conducted for the social context, we evaluate the performances of our model on the retrieval task for faces, corresponding to the "external queries" experiment in Chapter 5.

The categorical metadata considered in this chapter are the following:

- "**Genre**" **tags.** They describe the format of the corresponding TV show without conveying any information about the subjects mentioned in it. After filtering out the least frequent tags appearing only a few times in the shows of the training test, we are left with 28 unique "genre" tags. Some of them are "Magazine", "Series", "Debate", "Game", or "Interview".
- "**Topics**" **tags.** Unlike the "Genre" tags, they do not describe the format of the show but its content. After also filtering the least frequent ones, remain 24 unique tags. Some of them are "Sports", "Sciences", "Society", "Fiction", or "Economy".
- **The TV channels.** The TV shows from our dataset have been broadcast on 19 different channels.
- **Temporal information.** We also introduce temporal information describing roughly the date and time of broadcast of the show. The time of broadcast are described through 24 classes ranging from "00H" to "23H". The year of broadcast are also similarly described through categorical classes. Temporal information are investigated in more depth in Chapter 8.

Each entry in the training or test set can be assigned with none, one, or several "Genre" or "Topics" tags. The distribution of the number of "Genre" and "Topics" tags across the programs of our dataset is displayed in Fig. 6.2. Most of the programs have been labeled with only 1 to 3 tags. A very small amount of them have none, while some have up to 7 tags. Every program is however assigned with exactly one TV channel, one time, and one year of broadcast.

6.3 Proposed method

Because all of the TV shows are not labeled with tags with an equal precision, we cannot expect the set of tags associated to each show to be exhaustive. This means

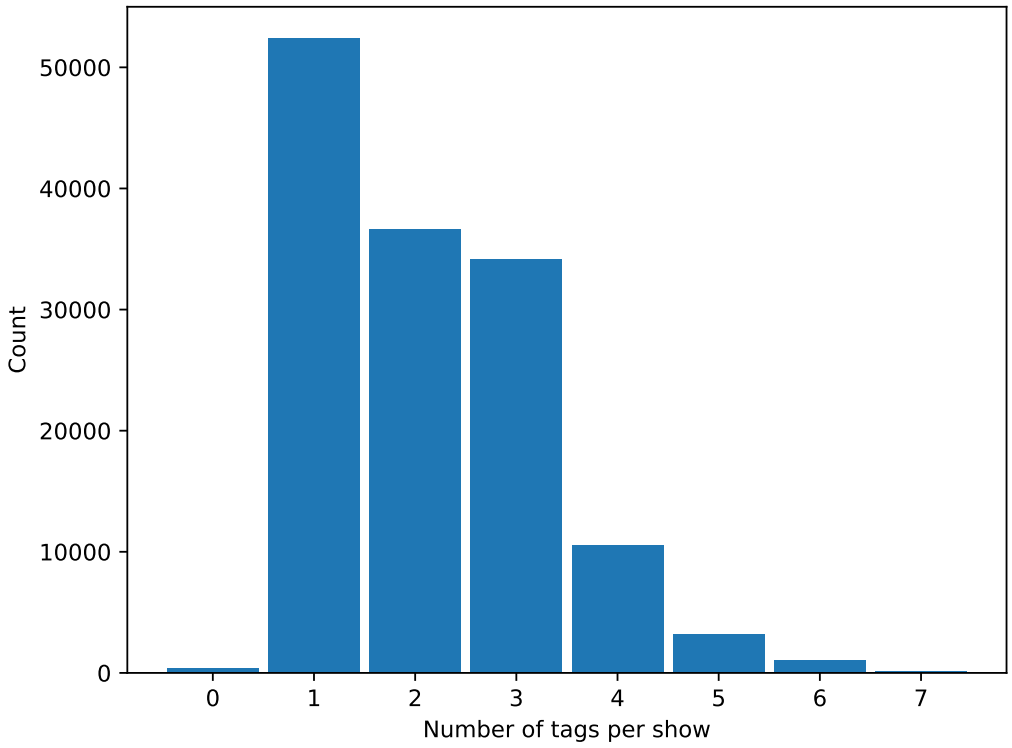

Figure 6.2: Distribution of the number of "Genre" and "Topics" tags per TV programs in the dataset.

that two similar shows in which appear potentially common people might not have been labeled with the same tags, and that the set of tags they have been assigned might also be completely disjoint.
For this reason, the categorical data cannot be compared directly the way they are and we need a way to determine which tags are semantically similar and which ones are not. We hence investigate how to learn embeddings to represent the tags in a multi-dimensional continuous vector space.

In this chapter, we make the assumption of a supervised setting. This means that the identity labels of the training set are expected to be known, as well as the tags describing the shows. We exploit the identity labels jointly with these descriptive tags to learn embeddings in a joint feature space through a Correspondence Analysis. This way, we will be able to compute a similarity score not only between the descriptive tags, but also between the identity labels. We will also be able to measure similarity scores between tags and identity labels.
At the inference time, only the tags are assumed to be known (besides the facial features).

6.3.1 Correspondence Analysis

Given the distribution of the number of tags associated to each show (see Fig. 6.2), and the low number of shows with more than 2 "Genre" or "Topics" tags, it is difficult to learn directly from them an embedding that would describe their similarity and semantics. Using a co-occurrence matrix of these tags like in the previous chapter would prove to be useless as it would be largely sparse.

We hence consider a supervised problem in which the identities of the training set are known. Because they are numerous (with 42,655 unique identities appearing on average 12.86 times each), they are more likely to bring the structure we need to learn which tags are supposed to be similar and which ones are not.

We use the Correspondence Analysis [Ben73, Hil74] (CA) to do this: a $N \times T$ matrix $\boldsymbol{M}$ is first built, where $N = 42,655$ is the number of identity in our training set and T is the total number of tags. This matrix describes the tags associated to the shows in which each identity has appeared: $\boldsymbol{M}_{(n,t)}$, for $n \in \{1...N\}, t \in \{1...T\}$ is equal to the number of shows labeled with the tag t in which the identity n has made an appearance.

Performing a Correspondence Analysis on the matrix $\boldsymbol{M}$ allows to learn in a common feature space embeddings of both the rows and the columns of the matrix, meaning in our case that we obtain embeddings of both the tags and the people identities in a common space. In Fig. 6.3 is shown a projection of some identity labels and tags in the common embedding space highlighting the similarities and dissimilarities between them. For readability purposes, only some "Genre" and "Topics" labels are displayed. The original space of the projected embeddings is 120-d. It appears that as we could expect, some tags have been assigned with very similar embeddings: it is the case of the tags "Cinema" and "Trailer", "Reality show" and "Game show", "Series" and "Fiction", or also "Politics" and "Debate".

Here are the five most similar genres and topics tag embeddings, in this order, for some identities according to the Correspondence Analysis:

- **Jean Dujardin** (actor): "Humour", "Trailer", "Best-of", "Cinema", "Interview".
- **Clarisse Agbegnenou** (athlete): "Sports", "Report", "Time slot", "Magazine", "News".
- **Michel Drucker** (TV host): "Talk show", "Best-of", "Entertainement", "Variété music", "Archives".
- **Angela Merkel** (politician): "News", "Information", "Atypical program", "On set", "Debate".

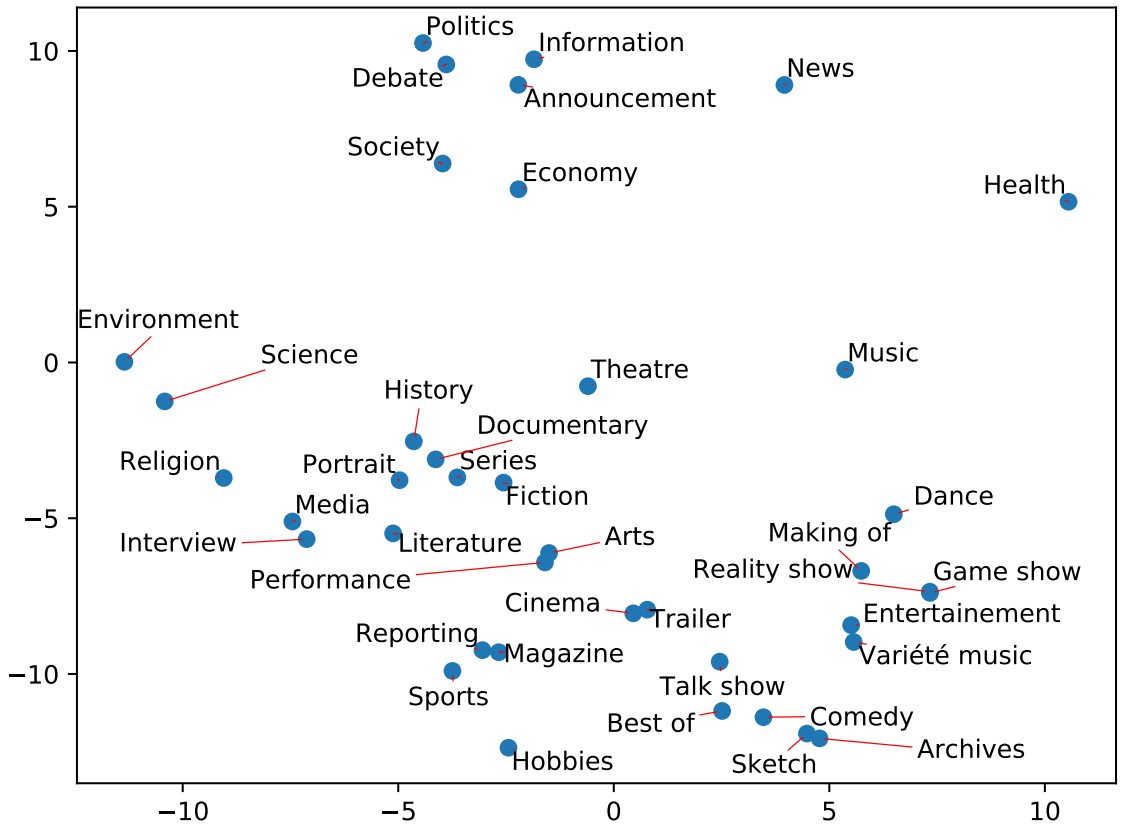

Figure 6.3: T-SNE projection of some tags embeddings.

- **Christine and the Queens** (singer): "Entertainment show", "Variété music", "Dance", "Music", "Talk show".

After learning these embeddings, the labeled facial descriptors of the training set are enriched with the embeddings of their corresponding identity labels. Because the similarity between CA embeddings is mainly determined through the angles, they are first normalized before being concatenated to the facial descriptors.
During the inference, the mean embeddings of the tags used to describe the query programs are used similarly to enrich the facial descriptors of the queries. The embeddings of the tags are compared directly to the embeddings of the identity labels from the training/gallery set.

6.3.2 Results

The results obtained on the retrieval task are displayed on Fig. 6.4. The performances are measured for various dimensions of the tags embeddings. The higher dimension evaluated is 146, which corresponds to the total number of tags.

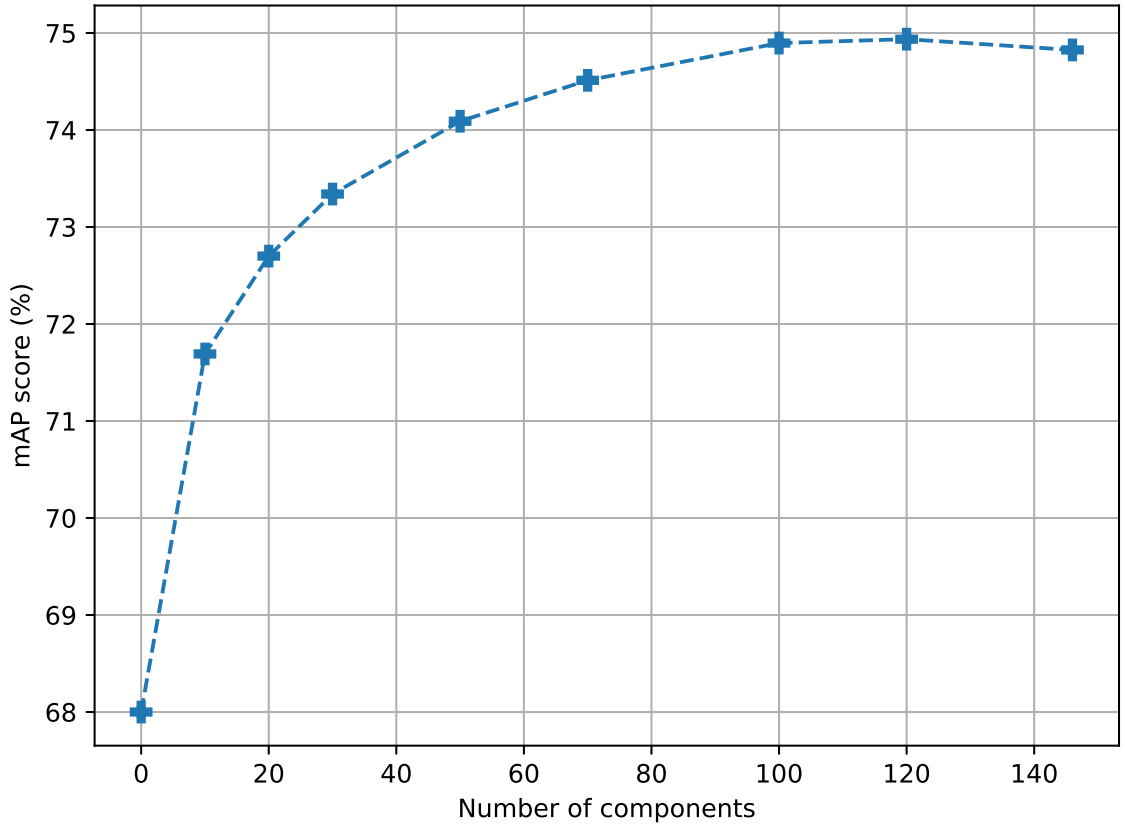

Figure 6.4: Mean Average Precision on the query set as a function of the number of additional components from the tags embeddings.

The performance observed on the retrieval task increase noticeably with the number of additional components from the tags embeddings before reaching a plateau at around 100 components. In the following section, we compare these results with the results yielded by the exploitation of the social context.

6.4 Co-occurrences embeddings and CA embeddings fusion

The results introduced above and obtained using the descriptive tags can be compared to those obtained in Section 5.4.5 for the social context in supervised setting. The only additional information that is used here are the descriptive tags for both the training/gallery set and the query set. The gain of performance obtained with tags are however much lower than those obtained with the co-occurrences information. It is in fact even lower than the gain of performance observed in the unsupervised setting of the social context, which uses much less information. The question that arises then is to know whether those modalities are complementary, in which case it

could be useful to exploit them simultaneously, or if they convey in fact redundant information, in which case we could drop one of them.

6.4.1 Embeddings concatenation and ablation study

To evaluate how complementary both embeddings can be, we first fuse them simply by concatenating both of them to the facial feature descriptors. The social context descriptor is computed in the supervised context as described in Section 5.4.5. The tags embedding is computed as described above in this chapter.

Because all configurations cannot be investigated, we use here as many components of both modalities before concatenating them. The performance observed is compared to those obtained with each modality separately in Table 6.1.

Table 6.1: mAP in the retrieval task in both modality and for their concatenation for various number of components.

Nb of comp. for each modality	20	50	100	120
Tags embedding only	72.70%	74.09%	74.90%	74.94%
Social embedding only	75.36%	79.22%	82.46%	83.40%
Concatenation	**77.50%**	**80.89%**	**83.49%**	**84.17%**

We can conclude that most of the increase in performance is brought by the social embedding, and that the tags embedding only conveys few additional information. Exploiting 120 components from the social context lift the mAP score from 67.88% with facial features only to 83.40%, while adding 100 more components from the tags embeddings only brings us to 84.17%. Knowing this, it can then be more useful given a fixed number of contextual component, to exploit solely the social context. The mAP scores for a fixed amount of contextual components in different configurations are compared in Table 6.2.

Note that there is no value for 200 components of the tags embeddings as the maximum dimension of these embeddings is 146, which is the total number of unique tags. For each total number of components considered, the social context outperforms the tags embeddings and the fusion of both.

Table 6.2: mAP in the retrieval task in both modality and for their concatenation for various number of components.

Total nb of comp.	40	100	200
Tags embedding only	73.73%	74.90%	-
Social embedding only	**78.14%**	**82.46%**	**85.76%**
Concatenation	77.50%	80.89%	83.49%

6.4.2 Inter-modality embeddings reconstruction and information redundancy

Since we know that both modalities contain redundant information, we can try to quantify that amount of redundant information. In order to do this, we apply a Canonical Correlation Analysis (CCA). The CCA allows us to project data points from two different feature spaces into a common feature space. Applying the reverse projections, we can also reconstruct one modality given the other one, assuming that they are indeed correlated.

We learn a CCA model on the training/gallery set for being able to project one embedding in the feature space of the other modality. Using this, we evaluate the performances obtained when using one modality in the gallery set, and a different one for the query set. The contextual embedding of the query is projected using the CCA onto the feature space of the training set. The results obtained for the different combinations of modalities in the gallery and query set are presented in Table 6.3.

Query modality \ Gallery modality	Tags emb.	Social emb.
Tags emb.	74.90%	72.46%
Social emb.	**76.36%**	**82.46%**

Table 6.3: mAP precision using different modalities for the gallery set and the query set. When the modalities do not match, the modality of the gallery set is reconstructed through CCA. All the results are for 100 contextual components.

What we observe is that even when using the tags embeddings in the gallery set, the best performances are achieved by reconstructing the tags embeddings through the social embeddings for the queries. The tags embeddings, on the other hand, are not enough to reconstruct faithfully the social embeddings. We can hence conclude that

for a given embeddings size, all of the useful information of the tags embeddings is contained in the social embeddings, while the opposite is not true.

The amount of noise in the tags embeddings can be explained by the fact that we use no more than 146 unique tags. Since the number of tags per show is often quite low (see the distribution of "Genre" and "Topics" tags in Fig. 6.2), the fusion of all the tags embeddings of one show can hardly contain enough information to describe it. On the other hand, the social embedding of one show is based on the fusion of identities co-occurrences descriptors. Given the number of identities in our dataset is much larger (42,655 unique identities), this allows for a lot more nuances in the social embeddings.

6.5 Conclusion and perspectives

In this chapter, we investigated how the available categorical meta-data, also referred to as "tags", can be used to improve the retrieval of one person's faces. We showed that this information can lead to some increase in terms of performance, in comparison to the case where the facial appearances only are considered. Also, we have been able to observe that both the tags and the social embeddings (introduced in the previous chapter 5) convey partly redundant information. The social embeddings remain however more precise than the tags descriptors, to the point that projecting the social embeddings onto the tags descriptors space still yields higher retrieval scores than using the tags descriptors directly.

We believe the gap of useful information between the social embeddings and the tags embeddings is due to both the few amount of unique tags combined with the low number of tags associated with each show. Also the fact that many shows have few to no descriptive tags (except for the temporal and channel descriptors) makes them more difficult to exploit.

The method described in this chapter is supervised. It makes it useful to investigate to what extent this modality and the social context descriptors are redundant, without having to deal with the noise due to the choice of a clustering algorithm or other parameters. We could, however, turn this approach into an unsupervised one, either through a clustering of the training set faces, like in the previous chapter before applying the CA, or by directly learning embeddings for each tag. As it was the case for the social context, this would very likely lead to a drop of performance. Moreover, this would not solve the redundancy problem with the social context (which already yield better performances in the unsupervised setting than this modality does in the supervised setting).

Some other approaches could be experimented to tackle the issues mentionned above: for example, applying Natural Language Processing (NLP) models on the shows titles could yield interesting results, as we could expect the information contained in the titles to be more diverse than that from the tags.

Chapter 7

Visual context

7.1 Introduction

Among the useful information that the human brain uses to identify people, the visual context, i.e. all the visual information except for the face to identify is particularly useful. We know, for example, that a human agent is able to achieve an accuracy score of 94.27% on the LFW protocol [HMBLM08], which is a face verification protocol, when all faces have been masked [KBBN11], meaning using only the visual context. Furthermore, the fact that human brain takes advantage of visual scenes to recognizes objects has also been studied [Bar04].

We also know that the television is a very standardised media; given only a few static frames, a human agent is most of the time able to say from which kind of show they have been sampled, whether it is a sport match, a newscast, a political debate, and so on. We believe that this knowledge carries much information about the people possibly appearing in that show and could be used to disambiguate the cases in which facial recognition is difficult. Some examples illustrating this are presented in Fig. 7.1. Our goal is to exploit the visual context to improve face retrieval and face

Figure 7.1: Examples of strong visual contexts.

verification in a dataset of TV shows.

In this chapter, we explain how we built a dataset of over 10M images from TV shows aired between 2010 and 2020 and how we used it in order to learn a visual context descriptor specific to TV. We show that it can be used jointly with face descriptors to improve the performance over a face verification task and a face classification task when such a visual context is available. We also discuss about the optimal strategy for fusing the visual context descriptor and the facial features to that end. The works presented in this chapter have been partially published in [PLDG21a].

7.2 Dataset

7.2.1 Motivation

Our goal is to be able to exploit the visual information in a video frame other than the facial appearances to help identify the people appearing on TV shows. We want to be able to extract continuous feature descriptors from video frames that carry information about the context of the show, and hence about the people appearing in those frames.

Existing datasets oriented towards context recognition have been designed in order to classify the images. Some datasets of locations images like SUN [XHE+10] or Places365 [ZLK+17] are quite exhaustive in terms of labels, however they do not allow to capture the semantic proximity that can exist between two different classes, like between an airfield and an airport terminal, or between the ocean and a harbour. Also their distribution is quite different from what is to be expected in a dataset of TV frames. Some other datasets which are more TV specific only focus on classifying TV programs in a few classes like news, sports, music, and so on, but they fail at capturing the diversity within each of these classes and at recognizing the relations that can exist between them. However, there are loss functions for similarity metric learning that allow to learn a continuous descriptor (which is richer in information than simple classes), like the triplet loss [SKP15] or the contrastive loss [CHL+05], which require triplets as inputs, or at least pairs of similar elements.

7.2.2 Dataset structure

The dataset built for this task is the Visual Context for TV programs introduced earlier in Section 4.2.3. It contains 10,684,217 frames of TV shows aired between 2010 and 2020, mostly on the French TV, but not only. This dataset covers the diversity of visual contexts prone to appear on television with frames selected from a large number of TV shows of all sorts, such as news, sports, entertainment, talk shows,

and so on. For practical reasons that will be detailed below, all the faces have been blurred in this dataset, so as to be unrecognizable.
This dataset is unlabeled. However, it comes with a list of 4,362,818 pairs of frames where at least one individual appears on both frames.

It is divided into three subsets:

- a training set, containing 4,357,969 positive pairs and 4,331,132 more elements to form the negative pairs of the triplets during training
- a validation set, with 2,456 triplets
- and a test set containing 2,393 triplets

Figure 7.2 displays a few examples of frames from this dataset.

Figure 7.2: Sample of the frames from our dataset. The visual contexts are various and reflect the diversity of the shows one can see on TV: news, entertainment, sport, weather forecast, and so on.

7.3 Methodology

The dataset has been built in order to be able to compute a continuous feature descriptor from a static image that captures its visual context and that can help

identify the people depicted in it. We believe that the best way to do so is to organize the images of our dataset in triplets including one anchor, one positive element and a third negative element, so that a model can be learned using the triplet loss or a similar loss function. This approach has been proved to be efficient for visual descriptor learning [SKP15].

7.3.1 Triplet formation

The difficulty lies in being able to define what makes a positive or a negative pair of frames. Most previous works could rely on external information, like categorical [SWBR16] or textual meta-data [DEC+16b, ZHWP19]. However, we do not dispose of such an information. The purpose of this dataset is to help identify the people appearing in the frames. Thus, what would be ideal would be to be able to rely on the identities of these people to build our triplets. Since their identities are unknown, we rather decided to use their facial appearances to build our positive and negative pairs of frames, and hence to form the (anchor, positive, negative) triplets.

Positive pairs

We consider one pair of frames as positive context-wise if we are able to identify at least one person that co-occurs in both frames. Given the impossibility to label manually all of the faces appearing in over 10M frames, we performed this automatically. We first detected all of the faces in the original frames of our dataset (not blurred) and computed the corresponding facial feature descriptors. We then formed our pairs using a selective distance threshold between the faces to assert they do in fact belong to the same person. Also, to make sure we do not build pairs based on reruns of a unique show broadcast twice, we also introduce a minimum distance threshold between the faces. Hence we ensure the pairs are not made of duplicates. The facial features model we used to do build these pairs is a ResNet18 architecture trained on the VGGFace2 dataset and presented in Section 4.1. It achieves a 98.98% score on the LFW protocol [HMBLM08].
Examples of thus formed pairs, after blurring the faces, are depicted in Fig. 4.9 (Section 4.2.3).

Negative pairs

A common and effective strategy in similarity metric learning setting is to focus on hard negative examples during training. For example, for facial features learning, comparing similar identities helps differentiating them [SMO+18, WZL17, SMN+17, SKP15]. This often implies selecting negative pairs with similar embeddings.
Our problem, however, differs. If we can consider a pair as being positive when a common person appears in both frames, the absence of such a person is not enough

to consider a pair as being negative. We might for example, sample two frames from the same show where no one appears on both frames. We do not want this pair to be considered as negative, since the context is not expected to be any different within a unique show from one scene to another. The same can apply to two different shows with a very similar context but no common participant.
Moreover, we face the difficulty of defining precisely what a negative pair is, *context-wise*. But a binary classification in "positive" and "negative" is not enough to describe the various level of proximity in terms of context.

This makes such an adversarial sampling very difficult to apply in our case as it could lead to sampling too many false negative pairs. For this reason, we decided to not use a hard example mining strategy, but to simply sample the negative elements of our triplets randomly from the whole dataset, and to rely on the large size and on the diversity of our dataset to make false negative pairs highly unlikely.

7.3.2 Model learning

Architecture

To train our visual context model, we use a Resnet50 architecture [HZRS16] pre-trained on Places365 [ZLK+17][1]. The last classification layer, with 365 output nodes, is replaced with a 16-dimensional layer. This network is fed with 256×256 images.

Loss function

The pretrained model is fine-tuned using a loss function inspired by the Triangular Similarity Metric Learning (TSML) [ZIG+15]. This loss function is similar to the widely used triplet loss introduced in [SKP15], except that it exploits the triangular inequality. For a triplet (X_0, X_p, X_n) where X_0 is the anchor, X_p the positive element and X_n the negative element, our loss function is defined as:

$$L_{tsml}(X_0, X_p, X_n) = \|X_0\|^2 + \frac{1}{2}\|X_p\|^2 + \frac{1}{2}\|X_n\|^2 - \|X_0 + X_p\| - \|X_0 - X_n\| \quad (7.1)$$

We also made a few experiments with the triplet loss. However, we observed that the performances using this TSML-inspired loss are slightly better. Hence will be reported here the results obtained with it.

About the blurring of faces

We mentioned earlier that all faces in our dataset have been blurred, in particular for legal reasons. It appears that this is also a practical choice. A model has first been

[1] `https://github.com/CSAILVision/places365`

Figure 7.3: When using a variant of our dataset where faces are not blurred, the trained model focused mainly on the actual faces and not on the surrounding context (left). This is no longer the case when trained on a dataset with faces blurred (right).

trained similarly, using the same architecture, loss function and dataset, with the only difference being that the faces have not been blurred. Its performances were satisfying; however, they decreased when the model was applied on frames where faces were blurred, which proves that it learned to recognize the faces more than it learned to recognize the visual context. This is easily explainable as our positive pairs have been build in order to contain faces of the same person, as described above in subsection 7.3.1.
Moreover, we could confirm using some visualization techniques like the Smoothgrad algorithm [STK+17] that this model focused primarily on the faces appearing on the images, and not on the background as was desired (see Fig. 7.3). Blurring the dataset helped to largely avoid this issue: the model trained with blurred faces no longer focuses on the faces, but on the visual context and background as expected.

7.4 Evaluation and comparison of the visual context descriptor alone

After training our model on our training set, and using the validation set for early stopping, we evaluated it on our test set and compared it to other existing models.

7.4.1 Evaluation and comparison on our test set

This first evaluation is a verification task on the visual context only (meaning the faces are not taken into account here). Given pairs of frames, we want to predict whether these pairs are to be considered similar or dissimilar.

To evaluate our model, we split our test set into 5 and use it to perform a 5-fold cross-validation. This way is determined the optimal distance threshold to classify a pair as being positive or negative. The overall accuracy displayed in Table 7.1 is the average accuracy obtained over the 5 folds $\pm$ the standard deviation.

Model	Accuracy
Places365 Resnet50	$75.34 \pm 0.68\%$
Places365 Resnet50 penultimate layer	$76.72 \pm 0.65\%$
Places365 Densenet161	$74.70 \pm 0.68\%$
Ours	$\mathbf{85.17 \pm 1.46\%}$

Table 7.1: Average accuracy $\pm$ standard deviation with 5-fold cross-validation over our test set.

We compare our model with the pre-trained Places365 models. For these pre-trained models, we use either the 365-dimensional outputs of the classification layer or the 2048-dimensional output of the previous layer (which is the input of the classification layer itself).
Unsurprisingly, we observe that our model is more suited to the verification task on a dataset of TV frames, whereas Places365 models are trained on more various and general settings.

Figure 7.4 shows the distribution of the cosine distances of the positive and negative pairs of the test set using our model. We can observe that the distribution of the negative pairs matches what would be expected from a random distribution uniformly distributed on the hyper-space, as described in subsection 7.3.1 (note that because we use angular distances and not euclidean distances, the density is almost zero at cosine distances 0 and 2 and peaks around 1).
The distribution of the positive pairs, however, is much more concentrated towards low distance values, meaning that our model has indeed been able to learn useful information from the visual context of the frames. A long tail still remains with some longer distances. This is more difficult to interpret: as we mentioned above, similarity between visual context is not binary and could be expressed on different levels. Hence, longer distances are not completely impossible in positive pairs, even though we expect them to be quite rare.

7.4.2 Qualitative results

In order to illustrate how our model performs, Fig. 7.5 displays a few sample images and their nearest neighbors in our test set, according to our visual context descriptor. As a reminder, our test set is made of 2,393 triplets of unique frames, meaning 7,179

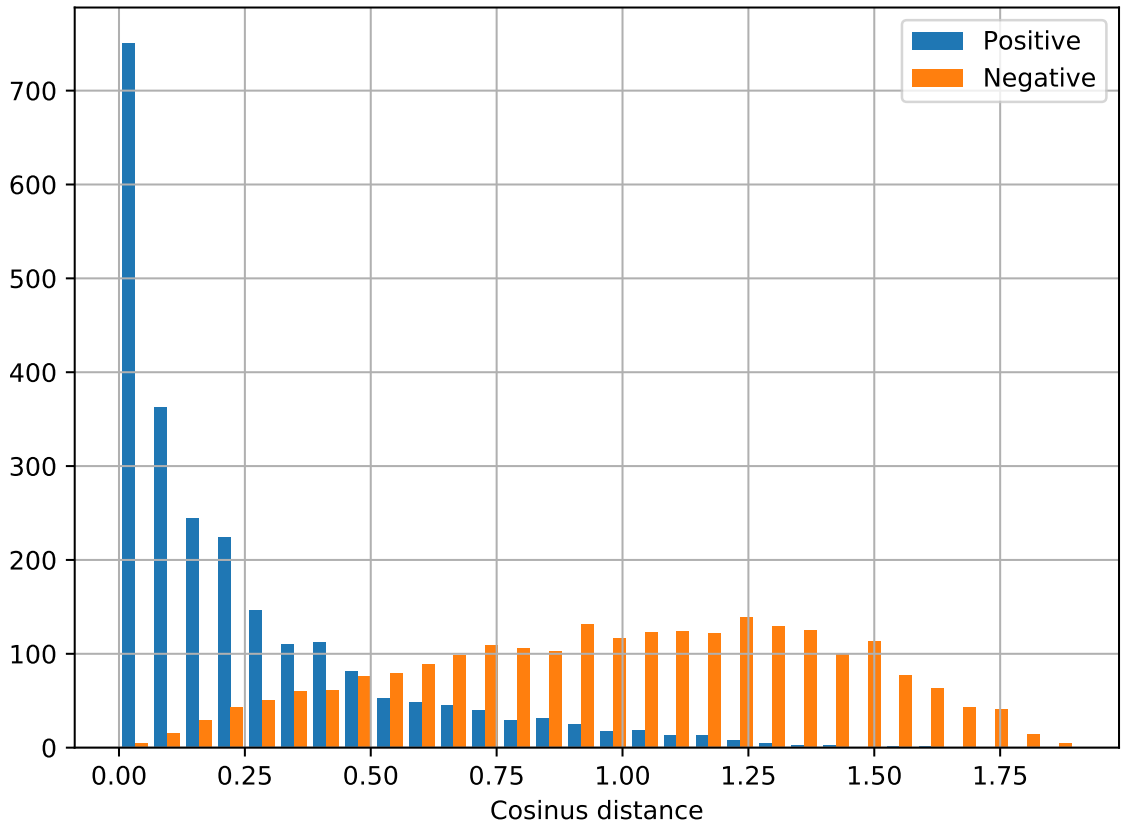

Figure 7.4: Distributions of the distances of the positive and negative pairs of the test set computed with our visual context model.

frames sampled randomly from real TV shows. The distribution of the frames from our test is hence expected to be representative of "real world data".
From these examples, we can see that frames from newscast, cartoons or weather forecasts are close to each other, respectively, even when coming from different TV channels.

7.5 Evaluation of the visual context descriptor in the face recognition setting

7.5.1 Evaluation on a face verification task of doppelgangers

We also evaluate the ability of our model to help recognize people when used jointly with facial feature descriptors. In order to do this, we perform a face verification task on the Doppelgangers face verification dataset introduced in Section 4.2.3. The positive pairs are sampled in the same way as for the training set, i.e. with two face images of the same person. In this evaluation, however, the negative pairs have not

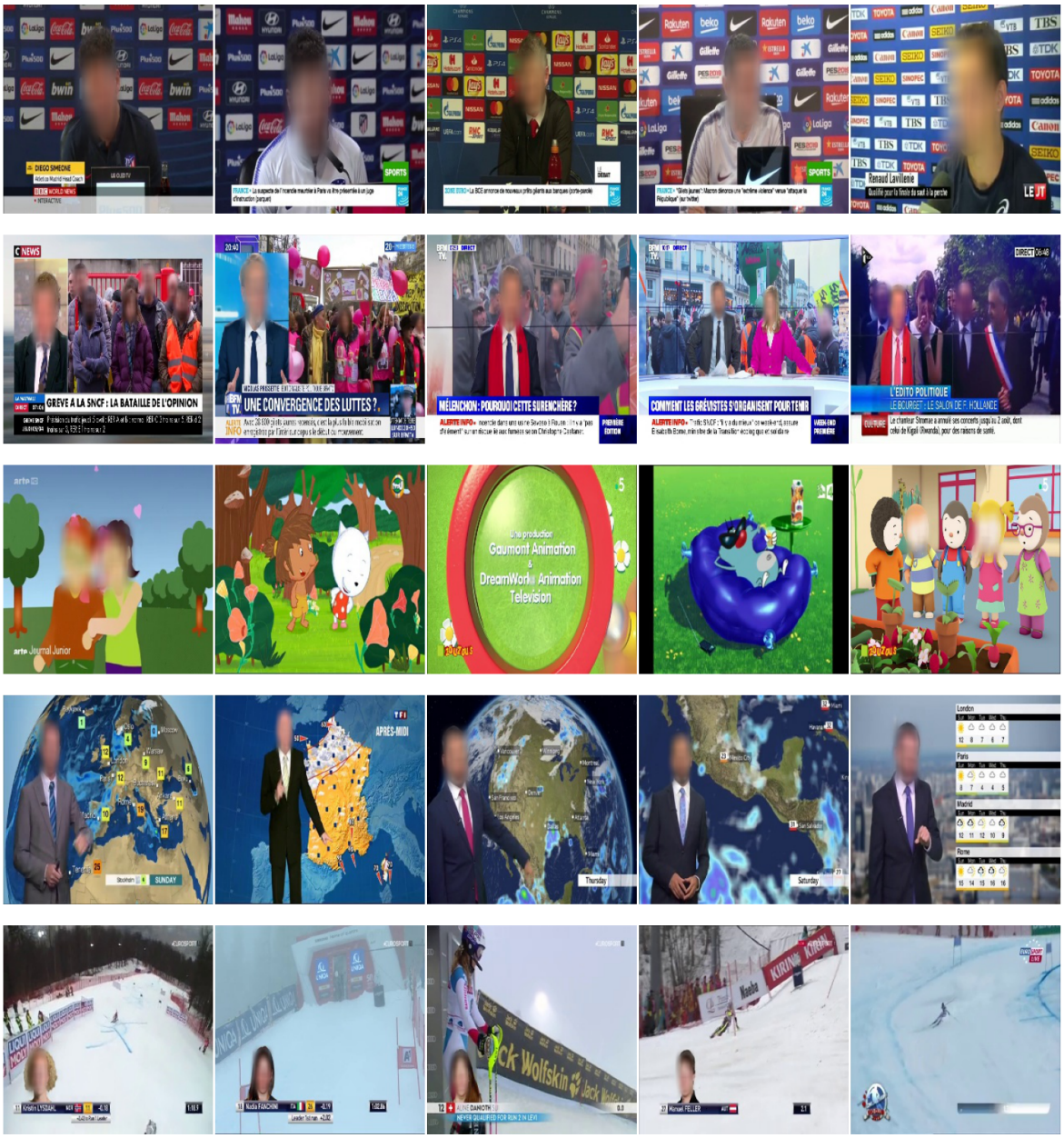

Figure 7.5: Sample images and their nearest-neighbors in the test set. The queries are on the left column, and the nearest-neighbors are then displayed from the closest ones to the furthest.

been sampled randomly; we selected hard examples where both members of a negative pair are visually similar but are not the same person (see Fig 4.10 in Section 4.2.3). The pairs of faces of this verification task have all been checked manually to avoid any annotation error that would nullify the performance score.

All images have been sampled from TV shows that do not appear in the training, validation or test set introduced above. These face verification images are available

alongside our dataset.
The pairs are distributed into 2 splits. The first split is used to learn the best weight to combine the facial descriptors distances and the contextual descriptors distances and to learn the best threshold to classify pairs as positive or negative. These parameters are applied on the second split to get an accuracy score. The two splits are then swapped and this operation is repeated.

In Table 7.2 are displayed the average accuracy scores over both of the splits. We observe that the combination of both facial and visual context descriptors achieves a better performance than the facial descriptors alone.

Table 7.2: Average face verification score over both splits

Input	Faces only	Context only	Faces + context
Accuracy	85.87 $\pm$ 0.03 %	65.86 $\pm$ 0.47%	**87.10** $\pm$ **0.17%**

We also notice that the context is sufficient to recognize that a same person is appearing in two frames or not in 65% of the time; this is more than a random classifier that would be right 50% of the time. It is however less than the performances presented in section 7.4.1.
If we cannot conclude with certainty on why is that, one hypothesis that seems likely is that people that look similar, like it is the case in this verification task (same age range, same sex, same ethnicity) are more likely to evolve in the same social environment. If politicians look more like other politicians, and athletes like other athletes, it might explain why the visual context is less good at distinguishing similar looking people than it is at distinguishing completely random pairs of people.

7.5.2 Evaluation on a classification task

After assessing the performances of the visual context descriptor in a face verification task, we also tried to evaluate it on a classification task, which is much closer to its intended use in real conditions.
In order to do this, we use 4,826 queries, consisting of faces observed in TV frames, sampled from shows distinct from the training, validation and test set introduced above. Both faces from the query set and the training set have been labeled semi-automatically; 13,032 unique labels appear in the training set. The classification method is a k-Nearest Neighbors (k-NN) based classification through a comparison with a subset of 1,125,704 labeled faces from the training set.

The inference is performed in the same way as in the Trombinos prototype (see Section 4.1.2): in the case of the facial feature descriptors alone, we retrieve a set of

labeled k-NN (with k=1000), and each one of these nearest neighbors votes for its own label, with a weight decreasing with their distance to the query, so that distant neighbors do not weight as much as very close ones.
The score $\boldsymbol{s}_l(x)$ associated to the label l for a query x is computed as stated in Eq. 7.2, where $l(y)$ is the label associated to element y, $d_f(x,y)$ is the euclidean distance between facial feature descriptors of x and y and $delta_{l(y),l}$ is the Kronecker delta of $l(y)$ and l.

$$\boldsymbol{s}_l(x) = \sum_{y \in \text{kNN}(x)} \delta_{l(y),l} * e^{-\frac{d_f(x,y)}{2\sigma^2}} \tag{7.2}$$

In the case of the combination of both descriptors, the k-NN are determined with the facial descriptors again, but they are associated with a linear combination of the distances in the facial space and the visual context space, as shown in Eq. 7.3, where $d_c(x,y)$ is the distance between the visual context descriptors of x and y. We use the optimal weighting scheme determined in subsection 7.5.1 as the data used in both evaluation are distinct.

$$\boldsymbol{s}_l(x) = \sum_{y \in \text{kNN}(x)} \delta_{l(y),l} * e^{-\frac{d_f(x,y) + \alpha * d_c(x,y)}{2\sigma^2}} \tag{7.3}$$

Due to the unequal number of appearance of each identity in TV shows, the most frequent people in the dataset tend to be over favoured over less frequent people by computing the identity scores $[s(x)]_l$ this way. This imbalance is prevented by normalizing these scores by the frequency of each identity in the gallery set.

The complete algorithms for predicting the identity labels are detailed below in Algo. 2 and 3.

Algorithm 2 (1) Faces only

Query x
$\boldsymbol{s}(x) = \mathbf{0}$
for y in k-NN(x) **do** ▷ $k = 1000$ in practice
    $\boldsymbol{s}_l(x) = \boldsymbol{s}_l(x) + e^{-\frac{d_f(x,y)}{2\sigma^2}}$
end for
for l in candidates labels **do**
    $\boldsymbol{s}_l(x) = \boldsymbol{s}_l(x)/\text{card}(l)$ ▷ To mitigate the long-tail distribution
end for
label(x) $\leftarrow$ Argmax($s(x)$)

Algorithm 3 (2) Faces + visual context (static)

Query x
$s(x) = \mathbf{0}$
for y in k-NN(x_q) **do**
    $\boldsymbol{s}_l(x) = \boldsymbol{s}_l(x) + e^{-\frac{d_f(x,y) + \alpha * d_c(x,y)}{2\sigma^2}}$ ▷ α learned on the verification set
end for
for l in candidates labels **do**
    $\boldsymbol{s}_l(x) = \boldsymbol{s}_l(x)/\text{card}(l)$
end for
label(x) $\leftarrow$ Argmax($s(x)$)

Results

In Fig. 7.3 below are the results obtained on the classification task, with the 128-dimensional facial features vectors alone, and with a combination of the facial features vectors and the 16-dimensional visual context descriptors.

Table 7.3: Classification score over 4,826 queries

Input	Faces only	Faces + context
Accuracy	87.84%	**87.94%**

The improvement gained from the visual context is very small. In comparison to the gain in the doppelgangers faces verification task, it seems that the visual context is not as useful in the general case than it is at disambiguating similar faces. Observing the results in more details, it appears that while some wrongly classified queries are indeed corrected thanks to the use of the visual context, some previously correctly classified queries are also misclassified due to the visual context contribution.

7.6 Identifying difficult faces

The disappointing gain of performance due to the visual context in the classification task, compared to the interesting results observed in the verification task of doppelgangers, motivated us to try and find the optimal way to fuse facial descriptors and contextual information. More especially, it seems that the additional information brought by the visual context is useful when disambiguating similar looking faces, when the facial descriptor is not precise enough. On the other hand, merging systematically facial descriptors with contextual information might bring noisy data

that could lead to erroneous prediction, that could have been correctly identified using the facial descriptors alone.
Hence, we would like to be able to tell beforehand when the facial descriptor is apparently good enough to identify the face by itself, and when the information carried by the visual context is necessary to identify the face.

7.6.1 Intuition

The model we use for extracting the facial feature descriptors, despite achieving good performances, is not flawless. While using it, we could observe that it was less efficient at recognizing certain types of faces; for example, it recognizes young and white people correctly, but appears to make errors more regularly on black or elderly people.
This can be explained by the fact that this model has been trained on the VGGFace2 dataset [CSX+18], which comprises primarily western celebrities. A variation in the distribution of the facial appearances observed in the training set VGGFace2 and the faces observed in our use case could explain these varying performances: a population with common traits that are more frequent in our use case than in the training set might result in very close distances in the features space, and hence in higher density zone in that feature space.

This can be observed by applying a clustering algorithm to the labeled faces of the training set. To neutralize the imbalance of the dataset due to hyper-frequent people, we only consider one data point per identity and call the resulting dataset the "1-instances" dataset. The clustering of these 13,032 data points with the DBSCAN [EKS+96] algorithm highlights the existence of a few region of higher density in the facial features space. A T-SNE visualisation [VdMH08] of the clusters is displayed in Fig. 7.6. Visualising some elements from these clusters is helpful in understanding what kind of population our model has more difficulties to recognize.

Some random elements belonging to each one of these clusters are displayed in Fig. 7.7. Based on these examples, we can identify common traits for each higher density zone in the feature space of the "1-instances" dataset: one contains mostly older white males, another one contains black people of both genders. There seem to also be a group of bearded men and one of blonde women. The two largest high-density zones, respectively the one of older white men and the one of black people, correspond to the populations we identified based on our experience as more difficult to recognize for our model.

We can furthermore confirm our intuition by estimating the local density around faces that have been correctly and incorrectly identified in the classification task (subsection 7.5.2). To make things simple, the local density around one data point

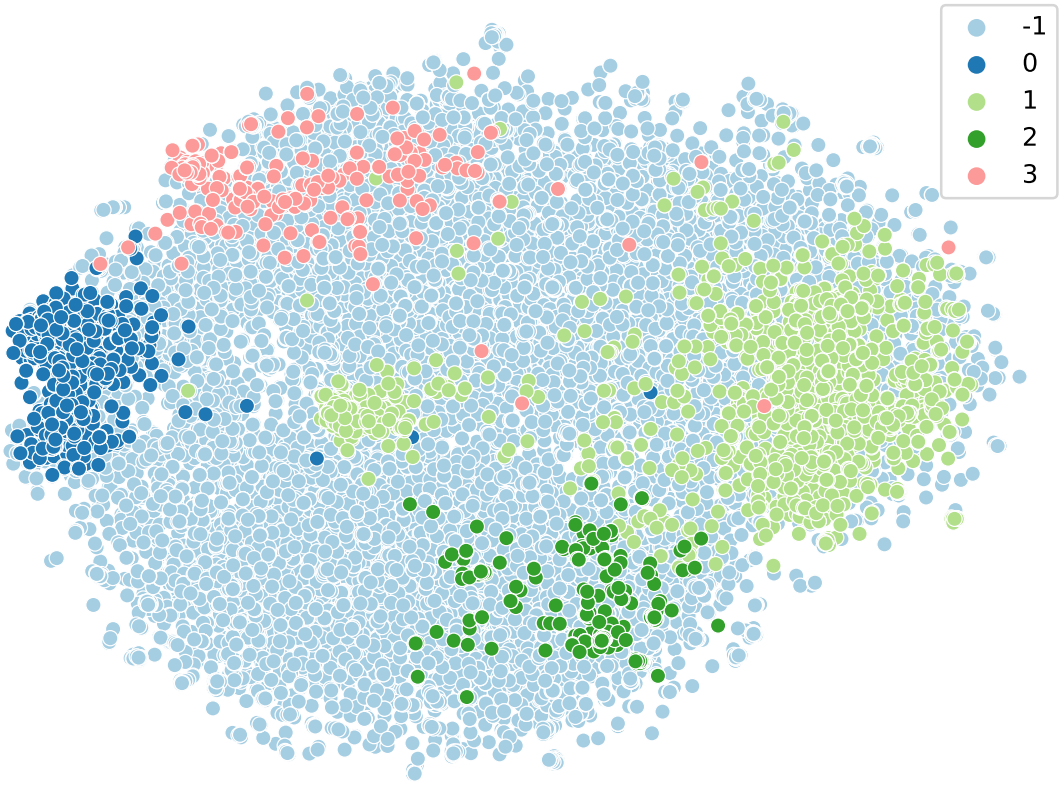

Figure 7.6: T-SNE 2-d projection of the higher density regions in the facial features space. Lower density regions are in light blue (label -1).

can be estimated through the distance to the fifth nearest neighbor. The distribution of distances to the fifth nearest neighbors (among the 13,032 unique data points described above, so that each identity is represented only once) for correct and incorrect classification in displayed in Fig. 7.8. Once again, it appears that the distribution for the incorrectly classified queries is visibly offset towards shorter distances, compared to the distribution for correctly classified queries, even though the overlap is not negligible. This strengthen our hypothesis that faces that are difficult to identify lie in denser regions of the features space.

7.6.2 Proposed approach

We want to improve the classification score observed in subsection 7.5.2. We propose, for each query, to first estimate the local density around that query in the facial feature descriptors space, and to use that density estimation to weight the amount of information from the visual context to be added to the facial descriptors. Since there are several ways to estimate the local density around one data point, we here decided to use inverse value of the distance to the fifth nearest neighbors as a proxy.

Figure 7.7: Samples from the high density clusters in the "1-instances" dataset as displayed in Fig. 7.6

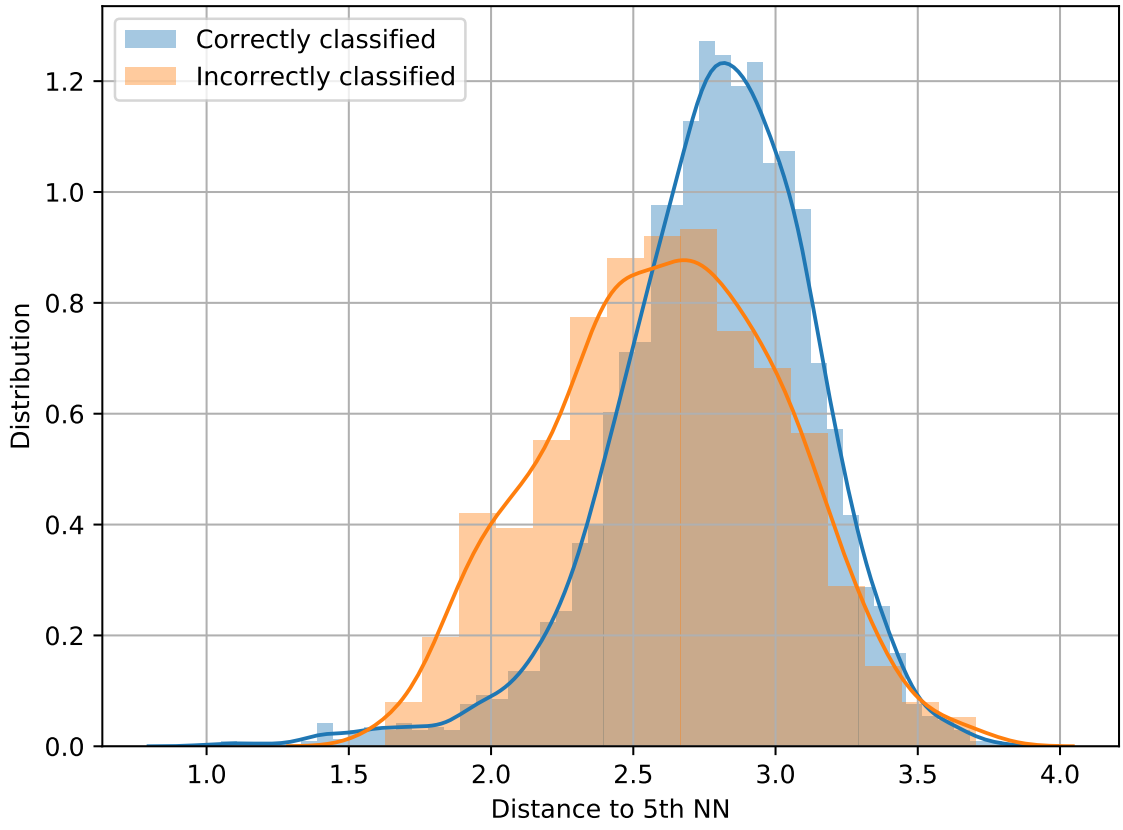

Figure 7.8: Distribution of distances to the 5th nearest neighbors for queries correctly and incorrectly classified.

Other sophisticated estimations could also be investigated.

The difficulty lies in being able to find the optimal function to map the density estimation to the visual context weight. In order to do this, different approaches have been experimented and compared.

Algorithm 4 (3) Faces + visual context (ambiguous)

Query x
$s(x) = \mathbf{0}$
for y in k-NN(x_q) **do**
    $\boldsymbol{s}_l(x) = \boldsymbol{s}_l(x) + e^{-\frac{d_f(x,y)}{2\sigma^2}}$
end for
for l in candidates labels **do**
    $\boldsymbol{s}_l(x) = \boldsymbol{s}_l(x)/\text{card}(l)$
end for
if $s(x)$ second highest value $> 0.1 * s(x)$ highest value: **then**
    $s(x) = \mathbf{0}$
    for y in k-NN(x) **do**
        $\boldsymbol{s}_l(x) = \boldsymbol{s}_l(x) + e^{-\frac{d_f(x,y) + \alpha * d_c(x,y)}{2\sigma^2}}$
    end for
    for l in candidates labels **do**
        $\boldsymbol{s}_l(x) = \boldsymbol{s}_l(x)/\text{card}(l)$
    end for
end if
label(x) $\leftarrow$ Argmax($s(x)$)

The first and simpler approach consists in making a first prediction using solely the facial features vectors. If the output is ambiguous (meaning if the confidence level of the prediction is low compared to the confidence level of the second rank prediction), then a new prediction is made based on a combination of both facial feature descriptors and visual context descriptors. The different steps are detailed in Algo. 4.

The second approach consists in weighting the contribution of the visual context dynamically. Both the facial features and the visual context are considered, and a dynamic weight is used for the visual context, increasing proportionally with our density approximation. This method is detailed in Algo. 5.

Algorithm 5 (4) Faces + visual context (dynamic)

Query x
$\alpha \leftarrow \beta /$ distance to 10th-NN in the "1-instances" dataset
$s(x) = \mathbf{0}$
for y in k-NN(x_q) **do**
    $\boldsymbol{s}_l(x) = \boldsymbol{s}_l(x) + e^{-\frac{d_f(x,y) + \alpha * d_c(x,y)}{2\sigma^2}}$
end for
for l in candidates labels **do**
    $\boldsymbol{s}_l(x) = \boldsymbol{s}_l(x)/\text{card}(l)$
end for
label(x) $\leftarrow$ Argmax($s(x)$)

7.6.3 Results

The classification results using the strategies detailed above are presented in Table 7.4.

Table 7.4: Classification score over 4,826 queries

Input	(1) Faces	(2) F + V (static)	(3) F + V (ambiguous)	(4) F + V (dynamic)
Accuracy	87.84%	87.94%	**88.33%**	88.13%

The results displayed for the facial features only (1) and the static weight for merging facial features and visual context descriptors (2) are the same as those displayed in Table 7.3.
The results obtained while weighting the contribution of the visual context proportionally to the estimated local density around the query (4) show a slight improvement compared to the case where the weighting scheme is constant (2).
The improvement is a little bit more important in the other case (3). Here, a first prediction is made exactly like in the case where faces alone are used (1). When the confidence score of the second identity label is at least 1% of the confidence score of the first label returned, we consider this result to be "ambiguous" and we make a new prediction using both the facial and the contextual descriptors with a constant weight, like in the case (2). Since the results hence obtained are better than both cases (1) and (2), it appears that the visual context indeed lead to a decay in the non-ambiguous queries.
Overall, however, the improvement is still very small. Many other fusion strategies for the facial and contextual information have been experimented but none led to a substantial improvement.

7.7 Conclusion

In this chapter, we explored how the visual context, being any visual cue except for faces, appearing in the TV shows can be described semantically and used to identify the people appearing on screen. To this end, we built a new dataset of over 10M frames from TV shows broadcast over an entire decade and that can be used in order to learn a descriptor embedding the visual context of the show. It is, to our knowledge, the largest dataset available to learn a visual context descriptor that is specific to TV shows.
We show that this dataset can be used to build a model able to retrieve frames from semantically similar TV shows, and we show that this model can also be used jointly with facial feature descriptors to improve the performances of a face verification task when such a visual context is available.

It appears that this additional information is mainly useful in ambiguous cases to distinguish similar looking faces. In the general case, however, where most faces can already be identified without additional data, this modality is likely to also convey noisy information, and then to lead to mixed results. It is hence necessary to devise a strategy to identify the cases in which the visual context descriptors are to be used, and the cases in which the facial feature descriptors alone are sufficient. We explored different strategies for this in the last part of this chapter.
One strategy that has been experimented was to weight the visual context information based on the local density of the facial feature descriptors space, due to evidences suggesting that "hardest" faces lie in denser regions. The results on the classification task could not confirm this hypothesis. It does not however invalidate the hypothesis either; it is not impossible that the visual context is unable to disambiguate between similar faces because they might tend to appear in similar contexts (for example, most politicians are white old men while athletes tend to be much younger with more diverse ethnicity).
We believe that the performances on both the face verification and classification tasks could be further improved with a suitable feature fusion strategy for facial descriptors and visual context descriptors.

Chapter 8

Temporal context

8.1 Introduction

Another contextual information we considered is the temporal modality. The TV programs are indeed not only visually standardised, but they also often follow a program schedule that can be rich in information about the kind of show being broadcast, and hence about who is likely or not to appear in it. The information conveyed by this temporal modality is two-fold:

- one-time events, due to people making the news for a short period of time. This is the case of many news stories, media controversy revolving around a specific matter, and so on.

- periodic events, and periodic occurrences of people. This encompasses many daily, weekly or even yearly TV shows, like entertainment shows, newscast, sport events and many more.

Examples of such periodic events are displayed in Fig. 8.1: the news are broadcast daily at the same hour, while sport events like Roland Garros happen every year at

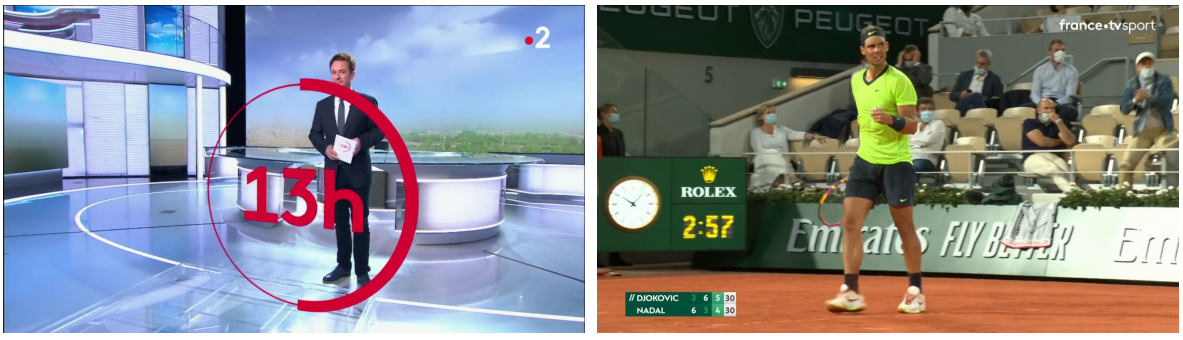

Figure 8.1: Examples of events occurring periodically: newscast (left) and Roland Garros (right)

the same period. In both cases, some people (like the news host or the athletes) are likely to occur again at each new iteration of these events.
In this chapter, we investigate several approaches in order to highlight the usefulness of this temporal information for the problem of facial recognition, and we experiment different approaches for exploiting this modality.

8.2 Neural networks for time representation

8.2.1 Methodology

Again, we focus on machine learning-based methods because these rich temporal relations are difficult to extract with more conventional techniques, or with handcrafted features. As for the other contextual information, we would like to automatically learn an embedding that best represents temporal similarity in terms of reappearing faces, and thus allowing the resulting descriptors to be easily combined with the other modalities.

Our approach is a Deep Fourier Transform method based on the Time2Vec [KGE+19] method and inspired by others similar works [GG17]. The idea is to automatically identify a set of frequencies at which people are likely to occur, with their relative importance. We make, however, the assumption that these frequencies are independent of the subjects. Indeed, we do not estimate a set of frequencies for each separate identity, but instead we focus on identifying a common set of appearance frequencies for all of the identities in our database.
It would indeed probably be more interesting to have separate frequency distributions for each identity, as some appear daily, while others will appear only for a short time frame every year, and so on. Unfortunately, this is not possible in our case. The reason behind that is because our set of identities suffers from a long-tail distribution; many different people appear in our dataset, but only a fraction of them appear regularly enough that we can compute meaningful frequencies for them. A large part of the identities in our dataset appear only a few times and do not allow us to recognize specific patterns.

8.2.2 Model and learning

The Time2Vec framework returns a multi-dimensional output where each component is a periodic function of the uni-dimensional input. In the paper introducing Time2Vec [KGE+19], these outputs are not used by themselves but jointly with other neural networks and potentially other inputs for tasks such as classification.

We face here the same problem as in the study of the visual context. that is, we are not interested in classifying timestamps, but rather to estimate a proximity between several of them, not in a linear way but as a probability of the same people to appear at these timestamps. Following the same reasoning, we decided to adopt a similarity metric learning approach using triplets and the Time2Vec framework. As we will detail below, this leads to some difficulties in the example sampling.

As mentioned earlier in subsection 3.5.2, the i^{th} component of the Time2Vec function applied to input τ is:

$$\boldsymbol{t2v}(\tau)[i] = \begin{cases} \omega_i\tau + \phi_i & \text{if } i = 0 \\ \mathcal{F}(\omega_i\tau + \phi_i) & \text{if } 1 \leq i \leq k \end{cases}$$

The function $\mathcal{F}$ is a periodic function (a sine in our case), while the parameters ω_i and ϕ_i, being the frequencies and the phase offset, are the values to be learned.
In our case, since we want to directly use these outputs to measure distances between them, we also introduce a set of scalar values α_i to weight each components. These values are representative of the respective importance of each periodic component.

Finally, our updated temporal embedding can be written as:

$$\boldsymbol{t2v*}(\tau)[i] = \begin{cases} \alpha_i * \omega_i\tau + \phi_i & \text{if } i = 0 \\ \alpha_i * \mathcal{F}(\omega_i\tau + \phi_i) & \text{if } 1 \leq i \leq k \end{cases} \tag{8.1}$$

The loss function used for the parameters optimisation is derived from the triplet loss and aims at minimizing the distances of the embeddings of positive pairs in comparison to the embeddings of negative pairs. For a triplet (τ, t_p, t_n) where τ is the triplet anchor, t_p the positive element and t_n the negative one, the loss function L is defined as:

$$L(\boldsymbol{t2v*}, \tau, t_p, t_n) = \max(0, \|\boldsymbol{t2v*}(\tau) - \boldsymbol{t2v*}(t_p)\|_1 - \|\boldsymbol{t2v*}(\tau) - \boldsymbol{t2v*}(t_n)\|_1 + m)$$

where m is the margin value.

8.2.3 Dataset, limitation and sampling

We use a dataset of 250,000 timestamps triplets for training. The positive pairs are sampled from appearances of the same people occurring in different TV shows broadcast between 2010 and 2019, while the negative pairs are sampled uniformly randomly along the same time frame (the way we would expect erroneous matches to be distributed). The distribution of distances between elements of positive and negative pairs is displayed in Fig. 8.2.

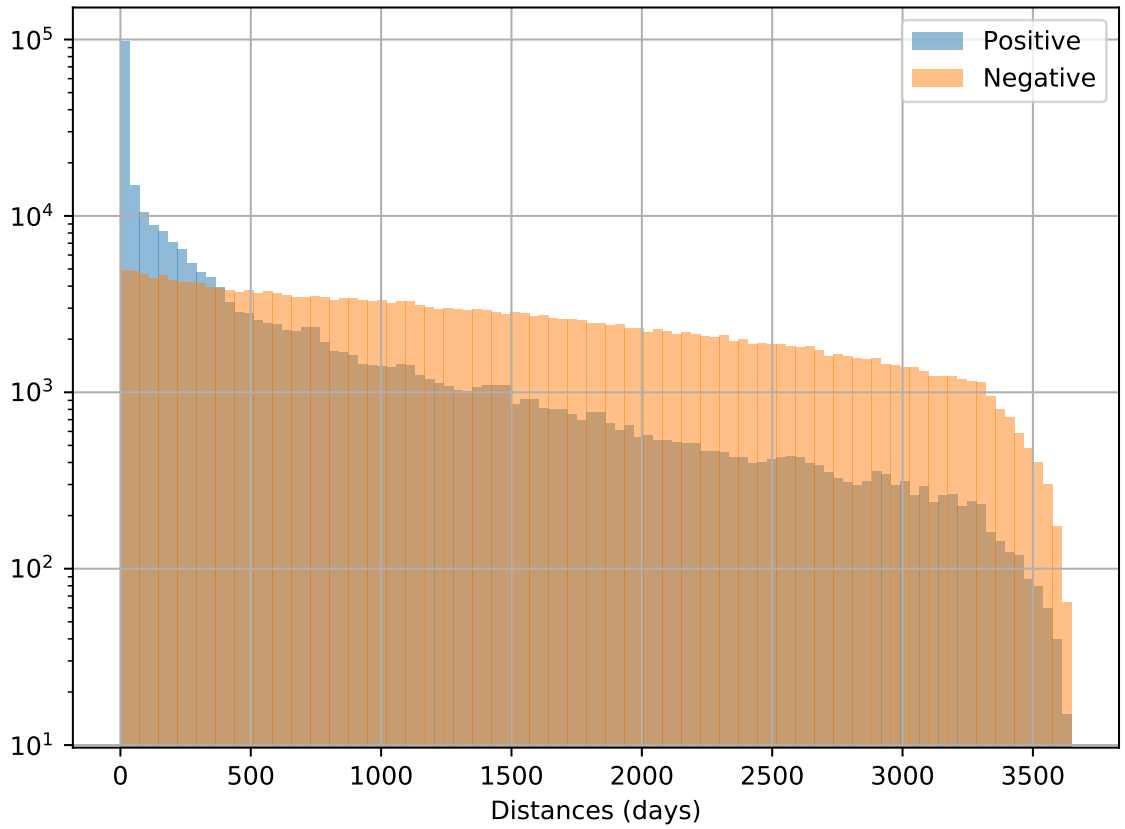

Figure 8.2: Distances distribution for positive and negative pairs of timestamps (the Y-axis is logarithmic).

The distribution of the positive timestamp pairs is far from being uniform: as is depicted in Fig. 8.2, most people, even if they appear following a periodical pattern, also appear several times at relatively short intervals.
This is because the temporal information is more likely to convey information through one-time events stretching over a few days or weeks than through periodical patterns only. Hence, short temporal distances are more common than longer ones among positive time pairs (relatively to the total time frame considered in this experiment, which is ten years): in our dataset, where positive pairs are sampled randomly, half of the positive pairs corresponds to some people appearing less than 115 days apart (less than 4 months). For timestamps sampled uniformly on a 10 years time frame, this number rises to 1200 days (almost 4 years), and would of course be even higher for a larger time frame.
This imbalance means that sampling negative pairs uniformly along the total time frame might favour long time periods (i.e. low frequencies) when differentiating positive and negative timestamps pairs.

The distribution of *(frequency, weight)* pairs obtained when learning a 450-dimensional

descriptor is displayed in Fig. 8.3. As detailed above, the positive pairs are sampled without any particular precaution, and the negative pairs are sampled randomly along a time frame of ten years. The learned distribution is very noisy as no time period really stands out, except at around 365 days, or one year. Long time periods also seem to be over-represented and keep increasing with each training iteration, without reflecting any real periodic pattern. A linear component is already considered during training to prevent this issue (see Eq. 8.1), but it is obviously not sufficient: because of the distribution imbalance between positive and random timestamps pairs, long periods (or low-frequencies) are favored.

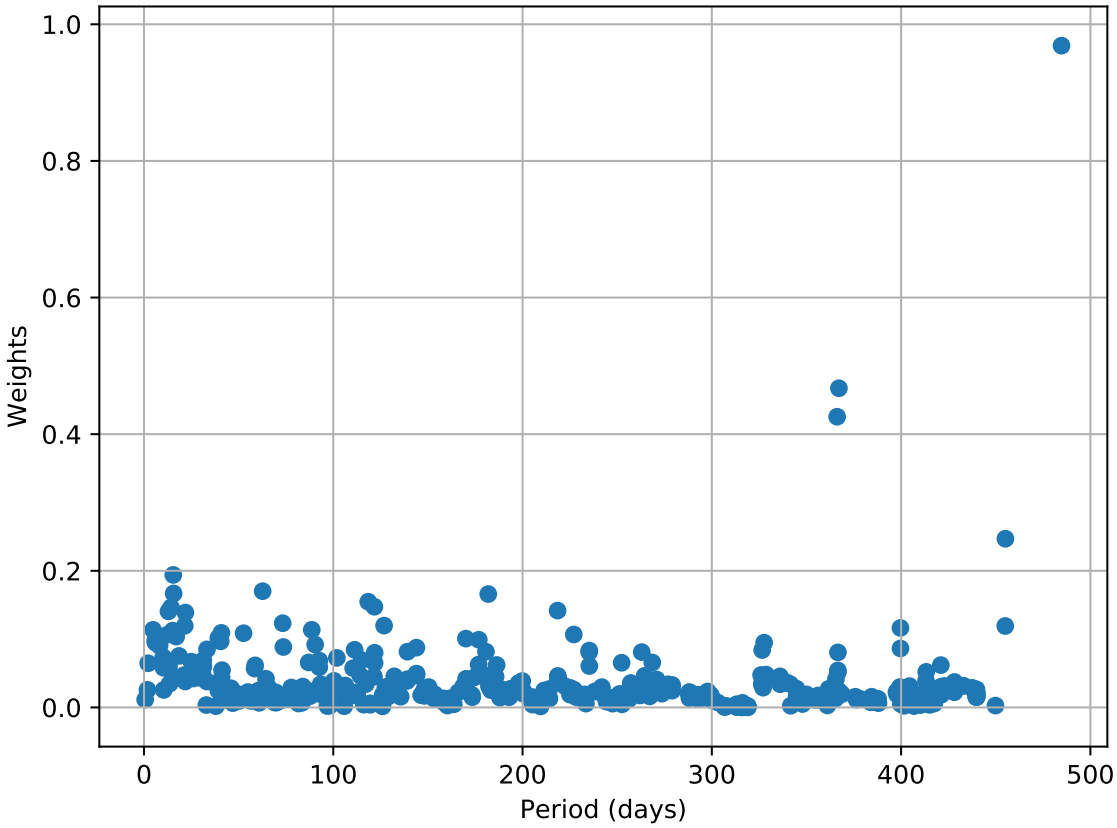

Figure 8.3: Period-weights pairs learned using the adapted Time2Vec approach.

To check the relevance of this approach, and its ability to learn sensible information, we decided to simplify the training problem by fixing the frequency values ω_i and optimizing at training time only the corresponding weights α_i and phase offsets ϕ_i. The frequencies are chosen in order to match with time periods ranging from one day to 450 days. This way, we expect to identify time periods ranging from 24H (the shortest time period we consider of interest) to one year, with some margin. The weights corresponding to these 450 time periods learned after training are displayed in Fig. 8.4 below. For better readability, the periods associated to the highest

weights are annotated next to their points.

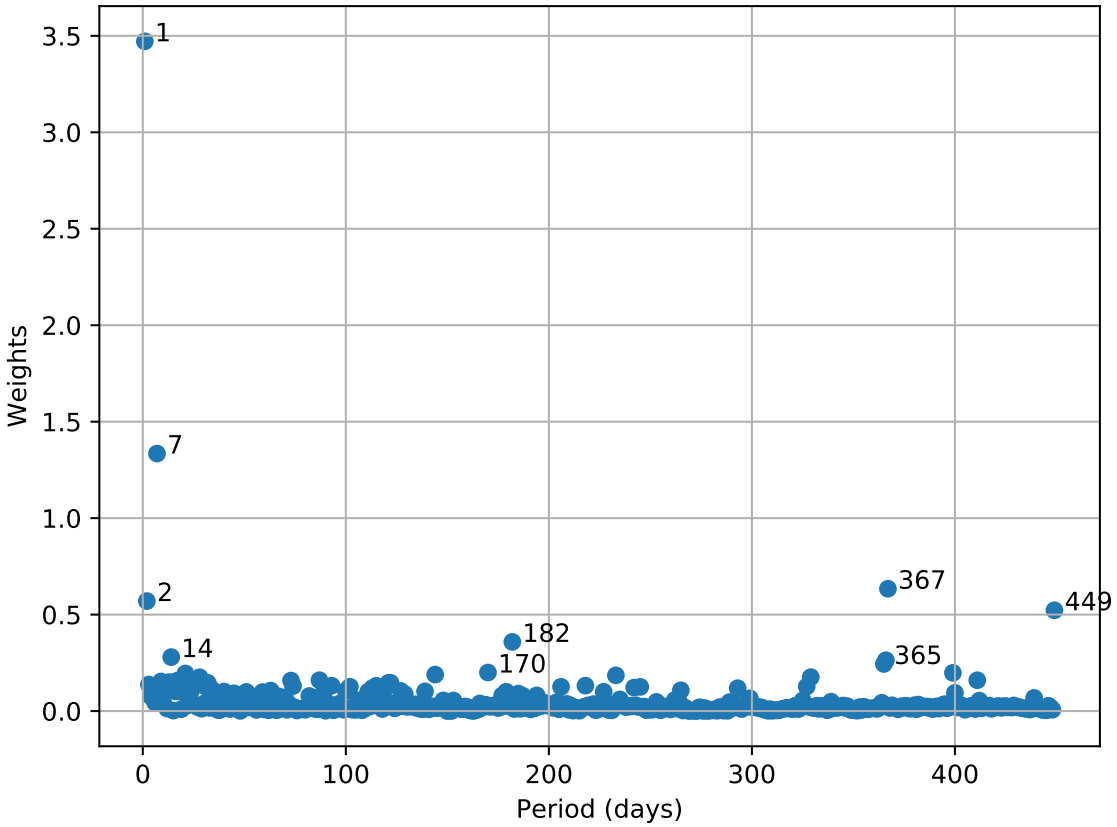

Figure 8.4: Weights associated to fixed time periods learned using the adapted Time2Vec approach.

Most of these time periods are assigned with null or almost null weights. However, a handful of them stand out by being assigned with more important weights. These time periods can easily be identified as "natural" time periods for a TV program schedule:

- First come a period of 24h, or one-day. This corresponds to the people appearing on different days but at the same time. It is by far the most important periodical pattern.
- Second come a time period of 7 days. Quite naturally, this corresponds to the people appearing in weekly shows. The weight assigned to the 14 days time period follows.
- Third come a non-negligible weight for the period of one year. The second harmonic for that frequency also stands out with a noticeable weight assigned to the time period 182 days.

Finally, it is also to be noted that the longest time-period, 450 days in this case has also been assigned to a non-negligible weight. This is again due to the imbalance in the distribution of positive and negative time pairs. Despite learning the periodic components jointly with a linear component (as described in Eq. 8.1) to prevent this, longer time periods patterns are still identified wrongly, even if this effect has been mitigated in comparison to the previous experiment (see Fig. 8.3).

In order to measure how efficient this model is at identifying temporal patterns in the appearances of people, we first evaluate its ability to distinguish positive and negative time pairs using a separate test, built in the same way as our training set, meaning it will have the same imbalance, which we also expect to observe in real world data. This test set contains 50,000 positive timestamps pairs and as many negative pairs.
Table 8.1 displays the area under the ROC curve obtained when distinguishing positive and negative timestamp pairs under different configurations. The first configuration uses simple scalar timestamps. The second configuration correspond to the periodic components of the model learned above without the linear component. Finally, the last configuration is the full model containing both the periodic components and the linear one.

Model	Timestamps only	Periodic components	Full model
AUC-ROC	82.01%	70.49%	81.82%

Table 8.1: AUC-ROC on the test set of timestamps under different configurations

As was already visible in Fig. 8.2, the "timestamps only" score highlights the amount of information contained in simple distances of scalar timestamps. This score is much higher than the result obtained using the learned periodical components alone. We can hence confirm that the "one-time events" seem to be predominant over the regular periodic appearances in terms of temporal information.
Moreover, the joint use of both the linear component and the periodical components does not lead to an increase of performance in comparison to the linear component used alone.

8.3 Experiments and results

8.3.1 Doppelgangers verification task

Because our goal is to use the contextual information to recognize people, we evaluate the usefulness of the temporal information in identifying people using the face

verification task introduced in the visual context chapter, in section 7.5.1.

As a reminder, this face verification task uses similar looking negative face pairs. It contains 2,923 face pairs sampled from TV shows and divided into two sets, one used for training, to identify optimal weights and threshold value, and a second one used for the evaluation. The sets are then swapped and the average classification score is returned. The results under different configurations are displayed in Table 8.2.

Input	Accuracy
Faces only	85.87 $\pm$ 0.03 %
Timestamps only	59.89 $\pm$ 0.04%
Periodic time embeddings	57.77 $\pm$ 0.24 %
Full time embeddings	62.15 $\pm$ 0.71 %
Faces + Timestamps	85.67 $\pm$ 0.06%
Faces + Periodic embeddings	**86.62 $\pm$ 0.37 %**
Faces + Full embeddings	86.34 $\pm$ 0.39 %

Table 8.2: Average face verification score over both splits

From these observations, it appears that the scalar timestamps values contain few information and few discriminative power. The time embedding learned as described above is however more interesting as it lead to a small improvement in the accuracy score when used jointly with the facial feature descriptors. More especially, the periodic components of the time embeddings seem to be conveying most of the useful information as they lead to the biggest improvement.

8.3.2 Classification task

We also evaluate the improvement brought by the temporal information over the classification task introduced in section 7.5.2. It consists or classifying 4,826 face queries sampled from TV shows by comparing them against 1,125,704 faces labeled with 13,032 unique labels. As in section 7.5.2, the classification strategy is a k-Nearest Neighbors based approach.

For each entry in the training set and the query set, we also have the exact timestamps at which the corresponding face has been spotted on TV. As for the face verification task, we use this information and combine it with the facial feature descriptors to observe the evolution of performance in the classification task.
Given the doppelganger face verification dataset is completely disjoint from the training and query set of the classification task, we use it in order to first learn

the optimal weighting scheme for the fusion of facial and temporal information.

Input	Accuracy
Faces only	**87.84%**
Faces + Timestamps	**87.98%**
Faces + Timestamps (ambiguous)	87.98%
Faces + Embeddings periodic components	87.46%
Faces + Embeddings periodic components (ambiguous)	87.53%
Faces + Full embeddings	87.75%
Faces + Full embeddings (ambiguous)	87.73%
Faces + 2-d embedding	**87.94%**
Faces + 2-d embedding (ambiguous)	87.92%

Table 8.3: Classification score over 4,826 queries

The results under the different configurations are displayed in Table 8.3. Different merging strategies are explored as in section 7.5.2: the first strategy is once again to simply always merge facial feature descriptor with the temporal descriptor (be it a scalar timestamp or a multi-dimensional embedding) together. The second strategy is to resort to the temporal information only in the case that the results returned by the facial feature descriptors alone are deemed ambiguous. The criterion used to define ambiguous classification is the same as in Algo. 4 in Sec. 7.6.2: a ratio of the prediction scores of the two first returned classes higher than 1%.
Additionally, because the 450 periodic components are numerous and cannot all be expected to convey useful information, we also consider a stripped-down 2-dimensional embedding, consisting of the 24-hour periodic component and the linear component only.

Unfortunately, the results obtained in each one of these configurations are quite disappointing, as they never lead not a noticeable improvement in comparison to the "vanilla" case of using facial feature descriptors only. More especially, it appears that our learned embeddings returns decayed performances in comparison to the scalar timestamps only. This shows that even though our temporal embedding model visibly learned to identify meaningful time periods, they may also bring a lot of noise. This is what we investigate in the following section.

8.3.3 Interpretation

The failure to achieve satisfying performance using the learned multi-dimensional embedding is probably due to the difficulty to embed in a common descriptor linear

components as well as periodic components; indeed, for each positive pairs of timestamps, corresponding to two apparition of the same person, they will match only on very few identified time periods. However, they will very rarely match all time periods simultaneously, leading to relatively large distances nonetheless between the learned embeddings.

In order to observe the influence of each component separately, and their ability to discriminate positive and negative timestamps pairs, we visualize in Fig. 8.5 the distribution of distances of positive pairs of timestamps from our training set modulo the first learned periods (being 1, 2, 7 and 365 days). Because many positive pairs appear almost simultaneously as depicted in Fig. 8.2, we filter out the timestamps distances inferior to length of considered period to focus solely on the periodic patterns and not "one-time events" stretching over one period. The distribution of random timestamps modulo the same time periods are also displayed as a basis for comparison.

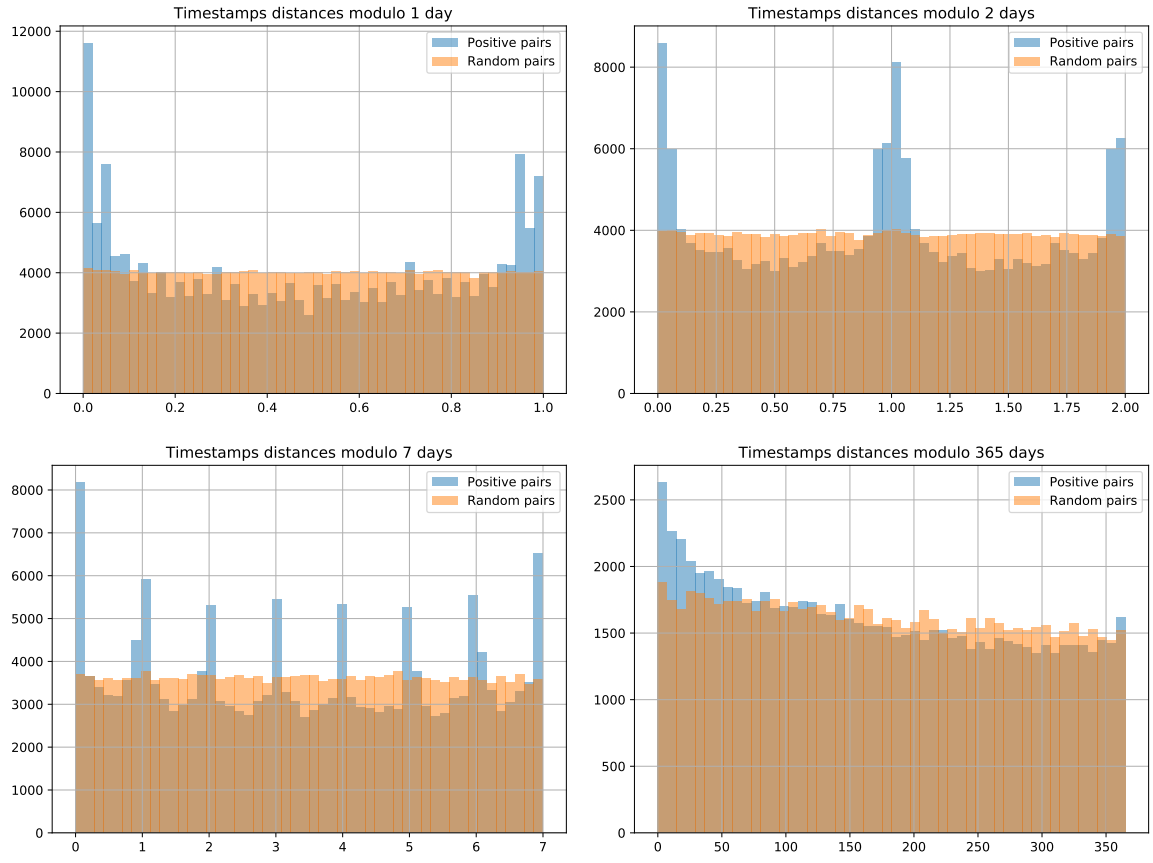

Figure 8.5: Distribution of timestamps pairs distances from the training set modulo different time periods

The convexity of the distribution profile of positive distances modulo 1 day, high-

light the periodic pattern at this time period: a noticeable fraction of the positive time pairs do appear at roughly the same time on different day. However, it is also visible that the vast majority of the positive pairs distances are scattered randomly and do not follow this periodic pattern.
The same is visible with the distribution of timestamps distance modulo 2 days which mostly conveys the same information and displays nothing more than the 1-day time period.
This 1-day time period is still very much visible on the distance distribution modulo 7 days. However, the 7 days time period is here also visible. Indeed, the peaks at 0 and 7 days (modulo 7 days) are more important than the 6 other peaks corresponding to the same time on other days of the week. But once again, this periodic pattern covers only a fraction of all timestamps pairs and most of the positive timestamp pairs do not match on this period.
The convex distribution highlighting the periodic pattern is also present for the timestamps distances distribution modulo 365 days, even though it is even less noticeable in this case.

Overall, the periodic patterns identified while learning our time embedding model are clearly present, though submerged within the noise. Thus, these periodic patterns are noticeable enough to be picked up by our model during training, but they are still not discriminative enough to allow for an improvement in face recognition tasks when used jointly with facial feature descriptors.

This fact is highlighted in Fig. 8.6, which displays the ROC-AUC score over the timestamps test set introduced above for different time periods. Once again, we filter out the time pairs whose distance is inferior to the considered period to mitigate the effect of one-time events stretching over a few days. The highest ROC-AUC score is once again reached for a 24-hours time period, but only reaches 57%. As a reminder, a ROC-AUC score of 50% is characteristic of a fully random model. The two best time periods are also 7 and 365 days and only reach a ROC-AUC score of 53% and 52% respectively, which shows a discriminative power barely better than that of a random model.

Given these insights, one possible way to mitigate this effect and to use the temporal modality to improve the person recognition could be to use unique models for each identity, in order to learn their own time patterns. Unfortunately, as has been explained earlier in this chapter, this would require to have a sufficient amount of information for every one of the possible identities, which is not our cases. We dispose of much information for a small amount of very publicised and frequent people, but a large proportion of our database consist of people with too few appearances to learn a real pattern.

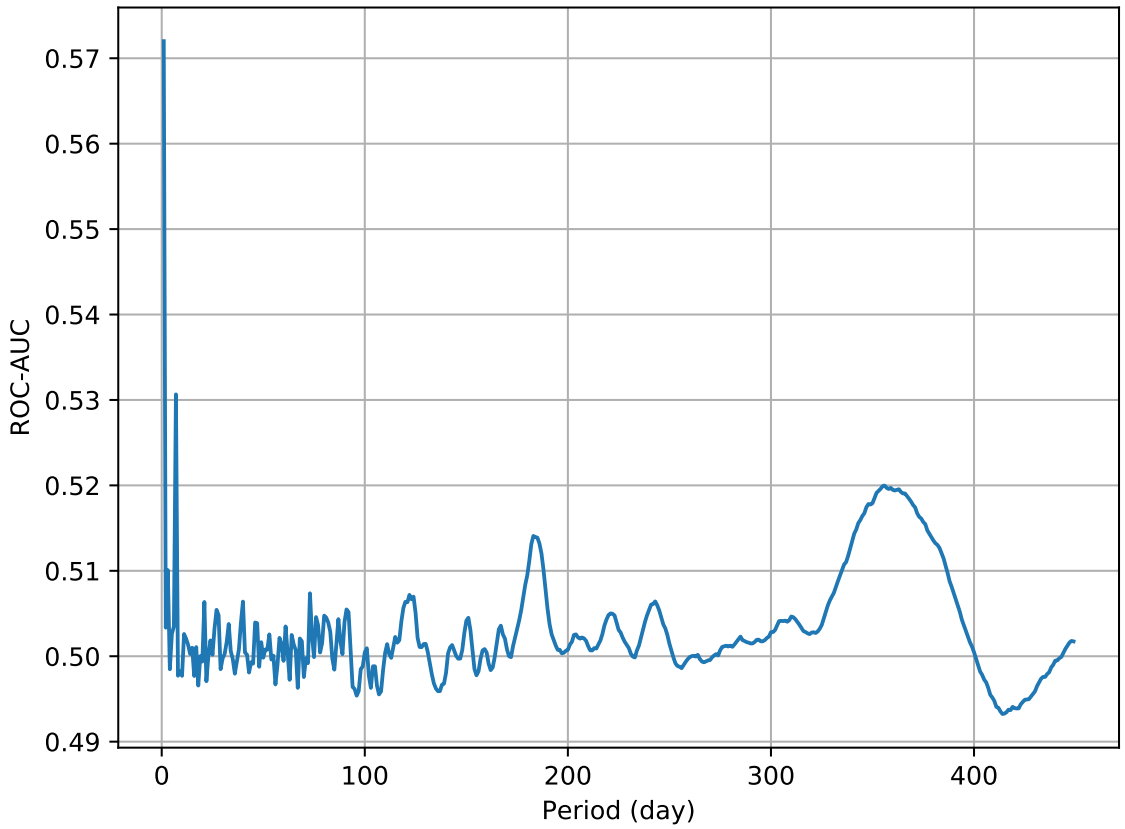

Figure 8.6: ROC-AUC score over the timestamps test set for various time periods.

8.4 Conclusion

In this chapter, we investigated how information such as the time and date at which a particular show has been broadcast can be used to recognize the faces that appear in it. The temporal information can be described as being two-fold: first, the linear representation of time, that can be used to describe one time-events on a timeline. This linear component is not to be under-estimated as most appearances of the same identities tend to follow each other closely. Second, a periodic representation of time, that is useful to describe daily, weekly, or monthly TV shows in which appears the same hosts or participants.
Using a neural network, we have been able to identify the most common periods of appearance of people on TV with their relative importance in a sample of real data, and hence learn a vector representation of scalar timestamps. However, after evaluating this vector representation, it appears that the linear representation of timestamps brings limited improvement when identifying people, while the periodic representation brings no improvement at all.
After analyzing the data, we can conclude that despite these periodic patterns being effectively apparent in the timestamps of people occurrences, the amount of noise

simply makes them unusable for a prediction purpose. A possible solution to this issue could be to learn one periodic pattern per identity; however, this is hardly feasible in practice, as it would require many data for each identity, which is not available.

Chapter 9

Conclusion

9.1 Contributions

The first contribution of this thesis is the datasets that have been built for the purpose of studying context-aware face recognition. If part of these data are for internal use only, we made available to the community some datasets:

- The first one is a "synthetic" dataset of faces organised in a reconstructed but realistic set of shows, in order to study mainly the social context and co-occurrences of the people in it. It can also be used for studying other contextual modalities like we did in this thesis for the categorical descriptive tags of shows. This dataset is named *Co-Occurring Faces in TV* and is available online[1].
- The second one is a dataset of real data sampled from the INA archival collection. Its main goal was to study the relevance of the visual information like the scenes and backgrounds for recognizing people. It is named *Visual Context for TV Programs* and is also available online[2]. It comes with two other datasets designed for evaluating performances, the first one being a dataset of doppelgangers for face verification evaluation, and the second one being a dataset of annotated faces from TV frames for retrieval and classification evaluation.

Building on these datasets, we have been able to investigate on various contextual modalities available to us and that we considered to be possibly relevant for identifying participants in TV shows:

- We first studied the social relationships between the participants of TV shows and observed that some people tend to frequently appear together, making them easier to identify all together than independently. By learning the co-occurrences of participants on a gallery set, we have been able to learn a social embedding for each one of them. At inference time, we make the assumption

[1] `https://github.com/ina-foss/co-occurring_faces_in_tv`
[2] `https://dataset.ina.fr/`

that all the participants appearing in the same show have a similar social embedding, hence reducing the occasional erroneous predictions. We also showed that prior information is not necessary on the gallery set for the social relationships between participants to be inferred; while it does bring better results, a non negligible improvement is still possible through a clustering of the unannotated gallery set.

- The second modality we have been focusing on is the categorical descriptive tags associated with the TV shows. We have been able to prove that they can indeed be useful in retrieving instances of faces. However, we have also shown that the information they convey is redundant with the one derived from the social context studied earlier.

- The information contained in the visual context also showed interesting results. Based on the dataset mentioned above, we built a model for visual context embedding. We showed that these visual context embeddings can describe the TV shows and that they proved effective in disambiguating between similar looking faces.

- In order to exploit the temporal information that is the date and time of broadcast of the TV shows, we considered both the linear and the periodic aspects to learn a temporal embedding. Despite learning sensible time periods to describe the frequencies of the participants on TV, the resulting embedding thus obtained gave no significant improvement leading us to conclude that this modality is too scarce in terms of information for the goal of person recognition.

9.2 Ethical considerations

Because the works presented in this thesis focus on the identification of individuals, some ethical concerns arise and should be addressed.
Indeed, even though we tried to develop an efficient model for person recognition solely to the purpose of building a better archival research system at the disposal of journalists, historians, sociologists and various researchers, the tools developed for innocent purposes are made available to all after they have been published, and could be re-used for more questionable goals. Separating possible applications from pure research is naive. For this reason, it is important to put things in perspective and see our contributions through a critical eye.

The ethical concerns to be considered for such a person recognition model are numerous. The first issues can arise during the collection of the data. While the early

datasets for person recognition (and specifically face recognition) used to be obtained with the cooperation of volunteers, the amount of data necessary for modern deep neural networks, as well as amount of data available *de facto* on the internet has led many researchers to exploit personal data with an implied agreement (through the terms of use of social networks, for example), or even without agreement. Most datasets of images used today have been scrapped from the web and some people end up unknowingly on datasets shared publicly. This is the case, for example of the VGGFace dataset [PVZ15] (which we used for our facial features extraction model). The PIPA dataset [ZPT+15], which was designed for the recognition of people in unconstrained settings, and contains contextual information similar to the social context an the visual context studied in this thesis, contains personal pictures shared online. After being exposed by the project `Exposing.ai` [Har21], the dataset has been taken down but keeps being shared amongst researchers. It has been used and cited, amongst others, by a Chinese military research university, unbeknownst to the people appearing in them.

This leads us to the others concerns, raised by how one contribution can be used by others. We know, for example, that facial recognition solutions have been developed in China to recognize ethnicities, and more especially the Uyghur minority in Xinjiang. Also, if we know that facial recognition models are used by some US police departments, we do not know how widely it is used as they do not have to declare it. However, uninformed users might be unaware of the limits and bias of such technologies which are often sold as magical solutions. This has already led to some wrongful arrests in the US. Some legal frameworks have been implemented to regulate the use of such technologies with large disparities between countries [ASL21].

Here are some points worth mentioning about this thesis specifically:
We did use VGGFace2 for training our facial recognition model, despite the criticisms made by the project `Exposing.ai`. Indeed, the only datasets available for learning facial feature descriptors have been scrapped from the web. This leads to the dilemma of either refusing to use any scrapped dataset, and letting companies such as Facebook and Google being the only ones with enough data to build facial recognition models, or accepting to use such a dataset.
We are also fully aware that the contributions we made to exploit the social and temporal contexts could be used for a surveillance tool, by identifying for example that some individuals tend to appear together on surveillance cameras or that they often visit some places at the same time. We can obviously imagine something similar being used for repression in some countries. Even though this is something we stand strongly against, we are also fully aware of the possibilities.
Finally, all of the data that we published have been approved by the legal department of INA to ensure that it complies fully with the European GPDR. In the most

open of the datasets that we published, containing over 10M TV frames, we made sure that every face appearing in it had been blurred so as to be unrecognizable.

9.3 Perspectives and future works

As a conclusion to this thesis, we will go over a few of the possible perspectives of the works and contributions presented in this manuscript.

One obvious perspective would be to study and incorporate additional modalities, like voice analysis, but also optical character recognition (OCR) to identify names or keywords displayed on screen. We can also think of a pipeline coupling a speech-to-text tool with a natural language processing to identify the names but also the topics discussed on screen; this could be used similarly to the categorical tags studied in this thesis, but also provide a more precise and detailed semantic information, that better captures the possible identities of the participants.

Another point to be explored is the fusion of more contextual modalities together. Similarly to how we highlighted how the categorical tags and the social context contain redundant information in this thesis, identifying the redundancy in all of the different contextual modalities is a requirement to efficiently merge them, more especially if the number of contexts considered is expected to increase.

Finally, thanks to the newly developed Trombinos prototype and the contributions of this thesis, we should now be able to annotate a large amount of the TV archival collections manually, thanks to an efficient label suggestion and similar face retrieval. This new amount of manually annotated data could be used for extended training set and as a new reference for future evaluations, and could in turn lead to improving the face recognition model.

Bibliography

[ASL21] Denise Almeida, Konstantin Shmarko, and Elizabeth Lomas. The ethics of facial recognition technologies, surveillance, and accountability in an age of artificial intelligence: a comparative analysis of US, EU, and UK regulatory frameworks. *AI and Ethics*, pages 1–11, 2021. pages 121

[Bar04] Moshe Bar. Visual objects in context. *Nature Reviews Neuroscience*, 5(8):617–629, 2004. pages 87

[Ben73] JP Benzécri. Data Analyses. Volume II. Correspondence Analysis. *Dunod: Paris*, 1973. pages 40, 80

[BHK97] Peter N Belhumeur, João P Hespanha, and David J Kriegman. Eigenfaces vs. fisherfaces: Recognition using class specific linear projection. *IEEE Transactions on Pattern Analysis & Machine Intelligence*, (7):711–720, 1997. pages 21

[Bie72] Irving Biederman. Perceiving real-world scenes. *Science*, 177(4043):77–80, 1972. pages 31

[BLGT06] Manuele Bicego, Andrea Lagorio, Enrico Grosso, and Massimo Tistarelli. On the use of SIFT features for face authentication. In *2006 Conference on Computer Vision and Pattern Recognition Workshop (CVPRW'06)*, pages 35–35. IEEE, 2006. pages 22

[BNC+17] Ankan Bansal, Anirudh Nanduri, Carlos D Castillo, Rajeev Ranjan, and Rama Chellappa. Umdfaces: An annotated face dataset for training deep networks. In *2017 IEEE International Joint Conference on Biometrics (IJCB)*, pages 464–473. IEEE, 2017. pages 26

[Bra00] G. Bradski. The OpenCV Library. *Dr. Dobb's Journal of Software Tools*, 2000. pages 45

[BT17] Adrian Bulat and Georgios Tzimiropoulos. How far are we from solving the 2d & 3d face alignment problem?(and a dataset of 230,000 3d facial landmarks). In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1021–1030, 2017. pages 25

[BVS14] Samarth Bharadwaj, Mayank Vatsa, and Richa Singh. Aiding face recognition with social context association rule based re-ranking. In *IEEE International Joint Conference on Biometrics*, pages 1–8. IEEE, 2014. pages 37

[CCP+18] Micael Carvalho, Rémi Cadène, David Picard, Laure Soulier, Nicolas Thome, and Matthieu Cord. Cross-modal retrieval in the cooking context: Learning semantic text-image embeddings. In *The 41st International ACM SIGIR Conference on Research & Development in Information Retrieval*, pages 35–44, 2018. pages 33

[CGCB14] Junyoung Chung, Caglar Gulcehre, KyungHyun Cho, and Yoshua Bengio. Empirical evaluation of gated recurrent neural networks on sequence modeling. *arXiv preprint arXiv:1412.3555*, 2014. pages 38

[CHL+05] Sumit Chopra, Raia Hadsell, Yann LeCun, et al. Learning a similarity metric discriminatively, with application to face verification. In *CVPR (1)*, pages 539–546, 2005. pages 23, 88

[Cho19] Kazimierz Choroś. Fast method of video genre categorization for temporally aggregated broadcast videos. *Journal of Intelligent & Fuzzy Systems*, 37(6):7657–7667, 2019. pages 32, 33

[CLY+92] Yong-Qing Cheng, Ke Liu, Jingyu Yang, Yong-Ming Zhuang, and Nian-Chun Gu. Human face recognition method based on the statistical model of small sample size. In *Intelligent Robots and Computer Vision X: Algorithms and Techniques*, volume 1607, pages 85–95. International Society for Optics and Photonics, 1992. pages 21

[CMS13] Ricardo JGB Campello, Davoud Moulavi, and Jörg Sander. Density-based clustering based on hierarchical density estimates. In *Pacific-Asia conference on knowledge discovery and data mining*, pages 160–172. Springer, 2013. pages 61

[CNZ18] Joon Son Chung, Arsha Nagrani, and Andrew Zisserman. Voxceleb2: Deep speaker recognition. *arXiv preprint arXiv:1806.05622*, 2018. pages 41

[CSX+18] Q. Cao, L. Shen, W. Xie, O. M. Parkhi, and A. Zisserman. VGGFace2: A dataset for recognising faces across pose and age. In *International Conference on Automatic Face and Gesture Recognition*, 2018. pages 26, 46, 99

[CWW+13] Xudong Cao, David Wipf, Fang Wen, Genquan Duan, and Jian Sun. A practical transfer learning algorithm for face verification. In *Proceedings of the IEEE international conference on computer vision*, pages 3208–3215, 2013. pages 23

[DBK+97] Harris Drucker, Chris JC Burges, Linda Kaufman, Alex Smola, Vladimir Vapnik, et al. Support vector regression machines. *Advances in neural information processing systems*, 9:155–161, 1997. pages 39

[DCEZ+18] Yashar Deldjoo, Mihai Gabriel Constantin, Hamid Eghbal-Zadeh, Bogdan Ionescu, Markus Schedl, and Paolo Cremonesi. Audio-visual encoding of multimedia content for enhancing movie recommendations. In *Proceedings of the 12th ACM Conference on Recommender Systems*, pages 455–459, 2018. pages 32, 41

[DCLT18] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018. pages 40

[dCN] Daniel Coelho de Castro and Sebastian Nowozin. Contextual face recognition with a nested-hierarchical nonparametric identity model. pages 40

[dCN18] Daniel Coelho de Castro and Sebastian Nowozin. 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)*, pages 745–761, 2018. pages 40

[DEC+16a] Yashar Deldjoo, Mehdi Elahi, Paolo Cremonesi, Franca Garzotto, Pietro Piazzolla, and Massimo Quadrana. Content-based video recommendation system based on stylistic visual features. *Journal on Data Semantics*, 5(2):99–113, 2016. pages 32

[DEC+16b] Yashar Deldjoo, Mehdi Elahi, Paolo Cremonesi, Farshad Bakhshandegan Moghaddam, and Andrea Luigi Edoardo Caielli. How to combine visual features with tags to improve movie recommendation accuracy? In *International conference on electronic commerce and web technologies*, pages 34–45. Springer, 2016. pages 32, 33, 90

[DEQC18] Yashar Deldjoo, Mehdi Elahi, Massimo Quadrana, and Paolo Cremonesi. Using visual features based on MPEG-7 and deep learning for movie recommendation. *International journal of multimedia information retrieval*, 7(4):207–219, 2018. pages 32

[DG07] Stefan Duffner and Christophe Garcia. Face recognition using non-linear image reconstruction. In *2007 IEEE Conference on Advanced Video and Signal Based Surveillance*, pages 459–464. IEEE, 2007. pages 25

[DGXZ18] Jiankang Deng, Jia Guo, Niannan Xue, and Stefanos Zafeiriou. Arcface: Additive angular margin loss for deep face recognition. *arXiv preprint arXiv:1801.07698*, 2018. pages 23

[DMB19] Sher Muhammad Daudpota, Atta Muhammad, and Junaid Baber. Video genre identification using clustering-based shot detection algorithm. *Signal, Image and Video Processing*, 13(7):1413–1420, 2019. pages 32, 33, 41

[DMS18] Nikita Dvornik, Julien Mairal, and Cordelia Schmid. Modeling visual context is key to augmenting object detection datasets. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 364–380, 2018. pages 30

[EKS+96] Martin Ester, Hans-Peter Kriegel, Jörg Sander, Xiaowei Xu, et al. A density-based algorithm for discovering clusters in large spatial databases with noise. In *kdd*, volume 96, pages 226–231, 1996. pages 99

[ESS10] Hazim Kemal Ekenel, Tomas Semela, and Rainer Stiefelhagen. Content-based video genre classification using multiple cues. In *Proceedings of the 3rd international workshop on Automated information extraction in media production*, pages 21–26, 2010. pages 32

[FSL15] Sachin Sudhakar Farfade, Mohammad J Saberian, and Li-Jia Li. Multi-view face detection using deep convolutional neural networks. In *Proceedings of the 5th ACM on International Conference on Multimedia Retrieval*, pages 643–650, 2015. pages 25

[FWL14] Fangxiang Feng, Xiaojie Wang, and Ruifan Li. Cross-modal retrieval with correspondence autoencoder. In *Proceedings of the 22nd ACM international conference on Multimedia*, pages 7–16, 2014. pages 33, 34

[GA16] Michael S Gashler and Stephen C Ashmore. Modeling time series data with deep fourier neural networks. *Neurocomputing*, 188:3–11, 2016. pages 39

[GB10] Carolina Galleguillos and Serge Belongie. Context based object categorization: A critical survey. *Computer vision and image understanding*, 114(6):712–722, 2010. pages 30, 31

[GG17] Luke B Godfrey and Michael S Gashler. Neural decomposition of time-series data for effective generalization. *IEEE transactions on neural networks and learning systems*, 29(7):2973–2985, 2017. pages 39, 106

[GJ09] Cong Geng and Xudong Jiang. Face recognition using SIFT features. In *2009 16th IEEE international conference on image processing (ICIP)*, pages 3313–3316. IEEE, 2009. pages 22

[GPAM+14] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. *Advances in neural information processing systems*, 27, 2014. pages 33

[GZH+16] Yandong Guo, Lei Zhang, Yuxiao Hu, Xiaodong He, and Jianfeng Gao. Ms-celeb-1m: A dataset and benchmark for large-scale face recognition. In *European Conference on Computer Vision*, pages 87–102. Springer, 2016. pages 26

[Har21] Jules. Harvey, Adam. LaPlace. Exposing.ai, 2021. pages 121

[HBTC10] Scott M Hayes, Elsa Baena, Trong-Kha Truong, and Roberto Cabeza. Neural mechanisms of context effects on face recognition: automatic binding and context shift decrements. *Journal of cognitive neuroscience*, 22(11):2541–2554, 2010. pages 30

[HE08] RJ Hulsebosch and PWG Ebben. Enhancing face recognition with location information. In *2008 third international conference on availability, reliability and security*, pages 397–403. IEEE, 2008. pages 41

[HGZ+18] Han Hu, Jiayuan Gu, Zheng Zhang, Jifeng Dai, and Yichen Wei. Relation networks for object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3588–3597, 2018. pages 30

[HH15] John HL Hansen and Taufiq Hasan. Speaker recognition by machines and humans: A tutorial review. *IEEE Signal processing magazine*, 32(6):74–99, 2015. pages 41

[HHPE15] Tal Hassner, Shai Harel, Eran Paz, and Roee Enbar. Effective face frontalization in unconstrained images. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4295–4304, 2015. pages 25

[Hil74] Mark O Hill. Correspondence analysis: a neglected multivariate method. *Journal of the Royal Statistical Society: Series C (Applied Statistics)*, 23(3):340–354, 1974. pages 40, 80

[HMBLM08] Gary B Huang, Marwan Mattar, Tamara Berg, and Eric Learned-Miller. 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*, 2008. pages 27, 87, 90

[HNR07] Scott M Hayes, Lynn Nadel, and Lee Ryan. The effect of scene context on episodic object recognition: parahippocampal cortex mediates memory encoding and retrieval success. *Hippocampus*, 17(9):873–889, 2007. pages 30

[HS97] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997. pages 38

[HXL+17] Li He, Xing Xu, Huimin Lu, Yang Yang, Fumin Shen, and Heng Tao Shen. Unsupervised cross-modal retrieval through adversarial learning. In *2017 IEEE International Conference on Multimedia and Expo (ICME)*, pages 1153–1158. IEEE, 2017. pages 33

[HXL18] Qingqiu Huang, Yu Xiong, and Dahua Lin. Unifying identification and context learning for person recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2217–2225, 2018. pages 36, 37, 59

[HZLH17] Rui Huang, Shu Zhang, Tianyu Li, and Ran He. Beyond face rotation: Global and local perception gan for photorealistic and identity preserving frontal view synthesis. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 2439–2448, 2017. pages 25

[HZRS16] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016. pages 45, 46, 91

[JDJ19] Jeff Johnson, Matthijs Douze, and Hervé Jégou. Billion-scale similarity search with gpus. *IEEE Transactions on Big Data*, 2019. pages 73

[JV03] Michael Jones and Paul Viola. Fast multi-view face detection. *Mitsubishi Electric Research Lab TR-20003-96*, 3(14):2, 2003. pages 25

[Kan74] Takeo Kanade. Picture processing system by computer complex and recognition of human faces. 1974. pages 21, 22

[KBBN11] Neeraj Kumar, Alexander Berg, Peter N Belhumeur, and Shree Nayar. Describable visual attributes for face verification and image search. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 33(10):1962–1977, 2011. pages 31, 87

[KGE+19] Seyed Mehran Kazemi, Rishab Goel, Sepehr Eghbali, Janahan Ramanan, Jaspreet Sahota, Sanjay Thakur, Stella Wu, Cathal Smyth, Pascal Poupart, and Marcus Brubaker. Time2vec: Learning a vector representation of time. *arXiv preprint arXiv:1907.05321*, 2019. pages 39, 40, 106

[KH05] Sanjiv Kumar and Martial Hebert. A hierarchical field framework for unified context-based classification. In *Tenth IEEE International Conference on Computer Vision (ICCV'05) Volume 1*, volume 2, pages 1284–1291. IEEE, 2005. pages 31

[KH+09] Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. 2009. pages 23

[Kin09] Davis E. King. Dlib-ml: A Machine Learning Toolkit. *Journal of Machine Learning Research*, 10:1755–1758, 2009. pages 45

[KKT+15] Brendan F Klare, Ben Klein, Emma Taborsky, Austin Blanton, Jordan Cheney, Kristen Allen, Patrick Grother, Alan Mah, and Anil K Jain. Pushing the frontiers of unconstrained face detection and recognition: Iarpa janus benchmark a. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1931–1939, 2015. pages 27

[KM11] Eva G Krumhuber and Antony SR Manstead. When memory is better for out-group faces: On negative emotions and gender roles. *Journal of Nonverbal Behavior*, 35(1):51–61, 2011. pages 30

[KSH12] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pages 1097–1105, 2012. pages 23

[KSSMB16] Ira Kemelmacher-Shlizerman, Steven M Seitz, Daniel Miller, and Evan Brossard. The megaface benchmark: 1 million faces for recognition at scale. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4873–4882, 2016. pages 27

[KW82] Nancy H Kerr and Eugene Winograd. Effects of contextual elaboration on face recognition. *Memory & Cognition*, 10(6):603–609, 1982. pages 30

[LBB+98] Yann LeCun, Léon Bottou, Yoshua Bengio, Patrick Haffner, et al. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998. pages 23

[LBL+16] Haoxiang Li, Jonathan Brandt, Zhe Lin, Xiaohui Shen, and Gang Hua. A multi-level contextual model for person recognition in photo albums. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1297–1305, 2016. pages 36

[LDH+19] Kai Lv, Heming Du, Yunzhong Hou, Weijian Deng, Hao Sheng, Jianbin Jiao, and Liang Zheng. Vehicle Re-Identification with Location and Time Stamps. In *CVPR Workshops*, pages 399–406, 2019. pages 41

[LGTB97] Steve Lawrence, C Lee Giles, Ah Chung Tsoi, and Andrew D Back. Face recognition: A convolutional neural-network approach. *IEEE transactions on neural networks*, 8(1):98–113, 1997. pages 23

[LJL15] Shengcai Liao, Anil K Jain, and Stan Z Li. A fast and accurate unconstrained face detector. *IEEE transactions on pattern analysis and machine intelligence*, 38(2):211–223, 2015. pages 25

[LKHB10] Dahua Lin, Ashish Kapoor, Gang Hua, and Simon Baker. Joint people, event, and location recognition in personal photo collections using cross-domain context. In *European Conference on Computer Vision*, pages 243–256. Springer, 2010. pages 41

[LLX17] Xiaodan Liang, Lisa Lee, and Eric P Xing. Deep variation-structured reinforcement learning for visual relationship and attribute detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 848–857, 2017. pages 30

[LMP01] John Lafferty, Andrew McCallum, and Fernando CN Pereira. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. 2001. pages 38

[LMT+07] Jun Luo, Yong Ma, Erina Takikawa, Shihong Lao, Masato Kawade, and Bao-Liang Lu. Person-specific SIFT features for face recognition. In *2007 IEEE International Conference on Acoustics, Speech and Signal Processing-ICASSP'07*, volume 2, pages II–593. IEEE, 2007. pages 22

[Low04] David G Lowe. Distinctive image features from scale-invariant keypoints. *International journal of computer vision*, 60(2):91–110, 2004. pages 22

[LSW20] Yanan Li, Yilan Shao, and Donghui Wang. Context-guided super-class inference for zero-shot detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, pages 944–945, 2020. pages 30

[LWY+17] Weiyang Liu, Yandong Wen, Zhiding Yu, Ming Li, Bhiksha Raj, and Le Song. Sphereface: Deep hypersphere embedding for face recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 212–220, 2017. pages 23

[MAD+18] Brianna Maze, Jocelyn Adams, James A Duncan, Nathan Kalka, Tim Miller, Charles Otto, Anil K Jain, W Tyler Niggel, Janet Anderson, Jordan Cheney, et al. Iarpa janus benchmark-c: Face dataset and protocol. In *2018 International Conference on Biometrics (ICB)*, pages 158–165. IEEE, 2018. pages 27

[Man80] George Mandler. Recognizing: The judgment of previous occurrence. *Psychological review*, 87(3):252, 1980. pages 29

[MH17] Leland McInnes and John Healy. Accelerated hierarchical density based clustering. In *2017 IEEE International Conference on Data Mining Workshops (ICDMW)*, pages 33–42. IEEE, 2017. pages 61

[MKT10] Nikolaos Mavridis, Wajahat Kazmi, and Panos Toulis. Friends with faces: How social networks can enhance face recognition and vice versa. In *Computational Social Network Analysis*, pages 453–482. Springer, 2010. pages 34, 35

[MRMN16] Iacopo Masi, Stephen Rawls, Gérard Medioni, and Prem Natarajan. Pose-aware face recognition in the wild. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4838–4846, 2016. pages 25

[MTC12] Riccardo Mazzon, Syed Fahad Tahir, and Andrea Cavallaro. Person re-identification in crowd. *Pattern Recognition Letters*, 33(14):1828–1837, 2012. pages 41

[OWJ18] Charles Otto, Dayong Wang, and Anil K Jain. Clustering millions of faces by identity. *IEEE transactions on pattern analysis and machine intelligence*, 40(2):289–303, 2018. pages 23

[PBL+12] Johann Poignant, Hervé Bredin, Viet-Bac Le, Laurent Besacier, Claude Barras, and Georges Quénot. Unsupervised speaker identification using overlaid texts in tv broadcast. In *Interspeech 2012-Conference of the International Speech Communication Association*, page 4p, 2012. pages 41

[PLDG21a] Thomas Petit, Pierre Letessier, Stefan Duffner, and Christophe Garcia. Exploiting Visual Context to Identify People in TV Programs. In *International Conference on Computer Analysis of Images and Patterns*, pages 220–230. Springer, 2021. pages 88

[PLDG21b] Thomas Petit, Pierre Letessier, Stefan Duffner, and Christophe Garcia. Unsupervised learning of co-occurrences for face images retrieval. In *Proceedings of the 2nd ACM International Conference on Multimedia in Asia*, pages 1–7, 2021. pages 58

[PVZ15] Omkar M Parkhi, Andrea Vedaldi, and Andrew Zisserman. Deep face recognition. 2015. pages 26, 121

[Rai01] Natascha Rainis. Semantic contexts and face recognition. *Applied Cognitive Psychology: The Official Journal of the Society for Applied Research in Memory and Cognition*, 15(2):173–186, 2001. pages 30

[Ras03] Carl Edward Rasmussen. Gaussian processes in machine learning. In *Summer school on machine learning*, pages 63–71. Springer, 2003. pages 38

[RJ86] Lawrence Rabiner and Biinghwang Juang. An introduction to hidden markov models. *ieee assp magazine*, 3(1):4–16, 1986. pages 38

[RWC+19] Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9, 2019. pages 40

[Sch15] Jürgen Schmidhuber. Deep learning in neural networks: An overview. *Neural networks*, 61:85–117, 2015. pages 23

[SKP15] Florian Schroff, Dmitry Kalenichenko, and James Philbin. Facenet: A unified embedding for face recognition and clustering. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 815–823, 2015. pages 15, 23, 24, 25, 26, 33, 46, 88, 90, 91

[SMN+17] Evgeny Smirnov, Aleksandr Melnikov, Sergey Novoselov, Eugene Luckyanets, and Galina Lavrentyeva. Doppelganger mining for face representation learning. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1916–1923, 2017. pages 24, 46, 90

[SMO+18] Evgeny Smirnov, Aleksandr Melnikov, Andrei Oleinik, Elizaveta Ivanova, Ilya Kalinovskiy, and Eugene Luckyanets. Hard example mining with auxiliary embeddings. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pages 37–46, 2018. pages 24, 90

[STK+17] Daniel Smilkov, Nikhil Thorat, Been Kim, Fernanda Viégas, and Martin Wattenberg. Smoothgrad: removing noise by adding noise. *arXiv preprint arXiv:1706.03825*, 2017. pages 92

[SV14] François Salmon and Félicien Vallet. An effortless way to create large-scale datasets for famous speakers. In *LREC*, pages 348–352, 2014. pages 41

[SWBR16] Gabriel S Simões, Jônatas Wehrmann, Rodrigo C Barros, and Duncan D Ruiz. Movie genre classification with convolutional neural networks. In *2016 International Joint Conference on Neural Networks (IJCNN)*, pages 259–266. IEEE, 2016. pages 33, 90

[SWT13] Yi Sun, Xiaogang Wang, and Xiaoou Tang. Hybrid deep learning for face verification. In *Proceedings of the IEEE international conference on computer vision*, pages 1489–1496, 2013. pages 23

[SYH+08] Edwin R Shriver, Steven G Young, Kurt Hugenberg, Michael J Bernstein, and Jason R Lanter. Class, race, and the face: Social context modulates the cross-race effect in face recognition. *Personality and Social Psychology Bulletin*, 34(2):260–274, 2008. pages 30

[SZD08] Zak Stone, Todd Zickler, and Trevor Darrell. Autotagging facebook: Social network context improves photo annotation. In *2008 IEEE computer society conference on computer vision and pattern recognition workshops*, pages 1–8. IEEE, 2008. pages 34, 35, 36, 37

[SZD10] Zak Stone, Todd Zickler, and Trevor Darrell. Toward large-scale face recognition using social network context. *Proceedings of the IEEE*, 98(8):1408–1415, 2010. pages 34

[TM] Nina Taherimakhsousi and Hausi A Müller. Location-based face recognition using smart mobile device sensors. In *Proceedings of the International Conference on Computer and Information Science and Technology (CIST)*, page 111. pages 41

[TP91] Matthew A Turk and Alex P Pentland. Face recognition using eigenfaces. In *Proceedings. 1991 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 586–591. IEEE, 1991. pages 21, 22

[TYL17] Luan Tran, Xi Yin, and Xiaoming Liu. Disentangled representation learning gan for pose-invariant face recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1415–1424, 2017. pages 25

[TYRW14] Yaniv Taigman, Ming Yang, Marc'Aurelio Ranzato, and Lior Wolf. Deepface: Closing the gap to human-level performance in face verification. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1701–1708, 2014. pages 23, 26

[VdMH08] Laurens Van der Maaten and Geoffrey Hinton. Visualizing data using t-SNE. *Journal of machine learning research*, 9(11), 2008. pages 99

[VN19] Jina Varghese and KN Ramachandran Nair. A novel video genre classification algorithm by keyframe relevance. In *Information and Communication Technology for Intelligent Systems*, pages 685–696. Springer, 2019. pages 32, 33

[VZ00] Yehuda Vardi and Cun-Hui Zhang. The multivariate L1-median and associated data depth. *Proceedings of the National Academy of Sciences*, 97(4):1423–1426, 2000. pages 65

[WALB15] Weiran Wang, Raman Arora, Karen Livescu, and Jeff Bilmes. On deep multi-view representation learning. In *International conference on machine learning*, pages 1083–1092. PMLR, 2015. pages 33, 34

[WHM11] Lior Wolf, Tal Hassner, and Itay Maoz. *Face recognition in unconstrained videos with matched background similarity*. IEEE, 2011. pages 27

[WJ19] Yue Wu and Qiang Ji. Facial landmark detection: A literature survey. *International Journal of Computer Vision*, 127(2):115–142, 2019. pages 25

[WR96] Christopher KI Williams and Carl Edward Rasmussen. Gaussian processes for regression. 1996. pages 38

[WTB+17] Cameron Whitelam, Emma Taborsky, Austin Blanton, Brianna Maze, Jocelyn Adams, Tim Miller, Nathan Kalka, Anil K Jain, James A Duncan, Kristen Allen, et al. Iarpa janus benchmark-b face dataset. In *proceedings of the IEEE conference on computer vision and pattern recognition workshops*, pages 90–98, 2017. pages 27

[WYP+12] Zixuan Wang, Jinyun Yan, Cong Pang, David Chu, and Hamid Aghajan. Who is here: Location aware face recognition. In *Proceedings of the Third International Workshop on Sensing Applications on Mobile Phones*, pages 1–5, 2012. pages 41

[WZL17] Chong Wang, Xue Zhang, and Xipeng Lan. How to train triplet networks with 100k identities? In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1907–1915, 2017. pages 24, 90

[XHE+10] Jianxiong Xiao, James Hays, Krista A Ehinger, Aude Oliva, and Antonio Torralba. Sun database: Large-scale scene recognition from abbey to zoo. In *2010 IEEE computer society conference on computer vision and pattern recognition*, pages 3485–3492. IEEE, 2010. pages 32, 33, 88

[XHL+19] Xing Xu, Li He, Huimin Lu, Lianli Gao, and Yanli Ji. Deep adversarial metric learning for cross-modal retrieval. *World Wide Web*, 22(2):657–672, 2019. pages 33

[YD16] Dong Yu and Li Deng. *Automatic Speech Recognition*. Springer, 2016. pages 41

[YLLT16] Shuo Yang, Ping Luo, Chen-Change Loy, and Xiaoou Tang. Wider face: A face detection benchmark. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5525–5533, 2016. pages 25

[ZBS+19] Eloi Zablocki, Patrick Bordes, Laure Soulier, Benjamin Piwowarski, and Patrick Gallinari. Context-aware zero-shot learning for object recognition. In *International Conference on Machine Learning*, pages 7292–7303. PMLR, 2019. pages 30

[ZHWP19] Liangli Zhen, Peng Hu, Xu Wang, and Dezhong Peng. Deep supervised cross-modal retrieval. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10394–10403, 2019. pages 33, 90

[ZIG+15] Lilei Zheng, Khalid Idrissi, Christophe Garcia, Stefan Duffner, and Atilla Baskurt. Triangular similarity metric learning for face verification. In *2015 11th IEEE International Conference and Workshops on Automatic Face and Gesture Recognition (FG)*, volume 1, pages 1–7. IEEE, 2015. pages 91

[ZLK+17] Bolei Zhou, Agata Lapedriza, Aditya Khosla, Aude Oliva, and Antonio Torralba. Places: A 10 million image database for scene recognition. *Transactions on Pattern Analysis and Machine Intelligence*, 2017. pages 32, 33, 88, 91

[ZLSR17] Bohan Zhuang, Lingqiao Liu, Chunhua Shen, and Ian Reid. Towards context-aware interaction recognition for visual relationship detection. In *Proceedings of the IEEE international conference on computer vision*, pages 589–598, 2017. pages 30

[ZLY+15] Xiangyu Zhu, Zhen Lei, Junjie Yan, Dong Yi, and Stan Z Li. High-fidelity pose and expression normalization for face recognition in the wild. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 787–796, 2015. pages 25

[ZPSG18] Eloi Zablocki, Benjamin Piwowarski, Laure Soulier, and Patrick Gallinari. Learning multi-modal word representation grounded in visual

context. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 32, 2018. pages 33

[ZPT+15] Ning Zhang, Manohar Paluri, Yaniv Taigman, Rob Fergus, and Lubomir Bourdev. Beyond frontal faces: Improving person recognition using multiple cues. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4804–4813, 2015. pages 36, 59, 121

FOLIO ADMINISTRATIF

THESE DE L'UNIVERSITE DE LYON OPEREE AU SEIN DE L'INSA LYON

NOM : PETIT DATE de SOUTENANCE : 10/06/2022

Prénoms : Thomas, Guillaume, Laurent

TITRE : CONTEXT AWARE PERSON RECOGNITION IN TV PROGRAMS

NATURE : Doctorat Numéro d'ordre : 2022LYSEI049

Ecole doctorale : EDA512 : INFORMATIQUE ET MATHEMATIQUES DE LYON

Spécialité : Informatique

RESUME :
L'identification automatique et la recherche par similarité des visages peut s'avérer être un outil utile pour la fouille de grandes bases de données telles que les archives télévisuelles de l'INA. Bien que les outils de reconnaissance faciale aient grandement progressé récemment, ils ne sont pas pour autant exempts d'erreurs, notamment lorsque la quantité de visages et le nombre de personnalités à reconnaître deviennent trop grands.
En revanche, les programmes télévisés sont généralement très codifiés, de telle manière qu'il est aisé pour chacun de dire en quelques secondes d'une émission s'il s'agit d'une émission sportive, de divertissement ou d'actualité. Cette codification des programmes, bien qu'implicite, peut s'étendre de l'apparence visuelle du plateau au choix du créneau horaire.
L'objectif de cette thèse est ainsi d'exploiter l'ensemble des informations contextuelles disponibles et potentiellement utiles pour l'identification des personnalités apparaissant dans les programmes télévisés. Pour chacune de ces modalités, nous en extrayons l'information, qui combinée aux descripteurs faciaux des sujets à reconnaître, permettra d'améliorer la recherche de nouvelles instances ou la classification des visages.
Nous nous intéressons notamment aux relations sociales entre les différents participants faisant que certains sont plus susceptibles d'apparaître ensemble à la télévision que d'autres. Nous proposons ainsi une méthode non-supervisée pour identifier simultanément l'ensemble des participants à un programme télévisé, en estimant leur probabilité d'apparaître conjointement.
Dans une seconde partie, nous nous intéressons aux informations contenues dans le contexte visuel des programmes télévisé et montrons que les arrière-plans visibles à l'écran peuvent aider à d'identifier avec succès les visages ambigus.
Nous explorons aussi les modalités contextuelles telles que les heures de diffusion ou les catégorisations thématiques des programmes, pour lesquelles nous évaluons l'apport d'informations utiles à la reconnaissance des participants ainsi que leur redondance avec les autres modalités étudiées.

MOTS-CLÉS :
Retrieval, similarity measure, dataset, television, visual context, person recognition, facial recognition

Laboratoire(s) de recherche : LIRIS - IMAGINE

Directeur de thèse: Christophe GARCIA, Professeur des Universités, INSA-LYON
Co-directeur de thèse : Stefan DUFFNER, Maître de Conférences (HDR), INSA-LYON

Président de jury :

Composition du jury :
Céline HUDELOT, Maître de Conférences (HDR), CENTRALE-SUPELEC, Rapporteure
Lynda TAMINE-LECHANI, Professeur des Universités, PAUL-SABATIER, Rapporteure
Guillaume GRAVIER, Directeur de Recherche, IRISA, Examinateur
Laure SOULIER, Maître de Conférences, SORBONNE UNIVERSITE, Examinatrice
Grégoire LEFEBVRE, Docteur, Chercheur, ORANGE LABS, Examinateur
Jenny BENOIS-PINEAU, Professeure des Universités, UNIVERSITE DE BORDEAUX, Examinatrice
Christophe GARCIA, Professeur des Universités, INSA-LYON, Directeur de thèse
Stefan DUFFNER, Maître de Conférences (HDR), INSA-LYON, Co-directeur de thèse