Face Analysis: State of the Art and Ethical Challenges

Domingo Mery[✉]

Department of Computer Science, Pontificia Universidad Catolica de Chile,
Santiago, Chile
domingo.mery@uc.cl
http://domingomery.ing.uc.cl

Abstract. In face analysis, the task is to identify a subject appearing in an image as a unique individual and to extract facial attributes like age, gender, and expressions from the face image. Over the last years, we have witnessed tremendous improvements in face analysis algorithms developed by the industry and by academia as well. Some applications, that might have been considered science fiction in the past, have become reality now. We can observe that nowadays tools are far from perfect, however, they can deal with very challenging images such as pictures taken in an unconstrained environment. In this paper, we show how easy is to build very effective applications with open source tools. For instance, it is possible to analyze the facial expressions of a public figure and his/her interactions in the last 24 h by processing images from Twitter given a hashtag. Obviously, the same analysis can be performed using images from a surveillance camera or from a family photo album. The recognition rate is now comparable to human vision, but computer vision can process thousands of images in a couple of hours. For these applications, it is not necessary to train complex deep learning networks, because they are already trained and available in public repositories. In our work, we show that anyone with certain computer skills can use (or misuse) this technology. The increased performance of facial analysis and its easy implementation have enormous potential for good, and –unfortunately– for ill too. For these reasons, we believe that our community should discuss the scope and limitations of this technology in terms of ethical issues such as definition of good practices, standards, and restrictions when using and teaching facial analysis.

Keywords: Face analysis · Social networks · Social and ethical challenges

1 Introduction

Nowadays, it is very easy to download thousands of images from social networks and build a database with information extracted from all faces that are present in the images as illustrated in Fig. 1. Thus, we can build a relational database of
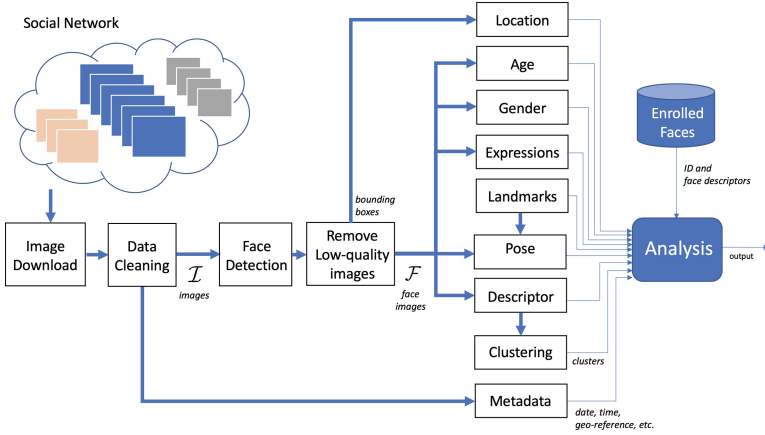
Fig. 1. Proposed approach for facial analysis in social networks.

the images with their faces and facial attributes. In this database, we can store for all detected faces: the bounding box, size, quality, location, age, gender, expressions, landmarks, pose, face descriptor and face cluster. With a simple query on this database, we can retrieve very useful and accurate information. Having this powerful database and a query image of a person, for example from a woman called Emily, some questions could naturally arise:

1. Is it possible to find Emily in the majority of the images (even in unconstrained environments with different poses, expressions and some degree of occlusion)?
2. Is it possible to extract the age, gender and facial expressions of Emily?
3. Using metadata of the pictures of the database, is it possible to establish when and where Emily was present (or absent)?
4. Is it possible to analyze the gender, age, and expressions of James and Louise that appears in the same picture with Emily?
5. Is it possible to search in the whole database those pictures in which Emily appears with other persons and select the person that most frequently co-occurred with Emily? And can we add a constraint to this person (it must be a man, or a woman, or a boy, or a girl, etc.)?
6. Is it possible to use the head poses of Emily and Gabriel (present in the same picture) and find if they are looking to each other?
7. Is it possible to build a graph of connections of Emily with other subjects that co-occurred in the pictures of the database?
8. Is it possible to determine from the face of James if he is criminal? or part of the LGTBQ community?

This is the wrong paper, if the reader is looking for the answer of the last question[1], however, for the remaining questions (#1 to #7), the answer is: yes, it

---

[1] We hope that our community will not research on fields related to question #8.

is possible. Over the last decade, the focus of face recognition algorithms shifted to deal with unconstrained conditions. In recent years, we have witnessed tremendous improvements in face recognition by using complex deep neural network architectures trained with millions of face images (see for example advances in face recognition [4,9,13] and in face detection [19,21]), and in many cases, algorithms are better at recognizing faces than human beings. In addition, there are very impressive advances in face clustering [12,18], and in the recognition of age [16], gender [20], (FER) facial expressions [2] and facial landmarks [23].

In this field, many works deal with applications that can be developed using the face analysis tools. Here, some examples, just to cite a few. In [15], social networks are built by detecting and tracking faces in news videos, the idea is to establish how much and with whom a politician appears on TV news. In [3], for example, facial behavior analysis is presented. The method can extract expressions and action units (facial movements such as 'inner portion of the brows is raised' or 'lips are relaxed and closed'), that could be used to build interactive applications. In [22], 'social relations' (defined as the association like warm, friendliness and dominance, between two or more persons) are detected in face images in the wild. In [7], face-based group-level emotion recognition is proposed to study the behavior of people participating in social events. Similar advances have been made in video analysis using the information of the location and actions of people in videos. See for example [6], where a 'context aware' configuration model is proposed for detecting groups of persons and their interactions (*e.g.*, handshake, hug, etc.) in TV material. Nowadays, some applications, that might have been considered science fiction in the past, have become reality now. Nevertheless, it is worthwhile to note that we are able to develop applications for a 'good cause' (*e.g.*, personal applications like searching the happiest faces in a family photo album; applications for history research like searching people in old archives of pictures; forensic applications like detection of pornographic material with children, etc.) and applications for a 'bad cause' (*e.g.*, security applications that collect privacy-sensitive information about the persons that appears in a surveillance video) as well.

In this paper, our main contribution is to show that anyone with certain computer skills can use (or misuse) this technology. The open-source tools are available on public repositories, and it is not necessary to train complex deep learning networks, because they are already trained. We will show that the state of the art is able now to do very accurate facial analysis in very complex scenarios (like mentioned in the first seven questions) with outstanding results as shown in Figs. 3, 4, 5, 6, 7, 8, 9. We believe that these two results, accurate and easy implementation of facial analysis, should challenge us to discuss possible restrictions and good practices. For this reason, in this paper, we give some ethical principles that can be considered when using this technology.

2   Open Source Tools

2.1   Tools for Social Networks

1. Image download: Images from social networks can be downloaded in a very simple way by using Application Program Interfaces (API's) or dedicated software. For example, there are API's for Twitter[2], Instagram[3] and Flickr[4]. For YouTube, there are some websites that offer the download service in an easy way. On the other hand, GitHub is a repository for code and datasets in which the datasets can be downloaded directly.

2. Data cleaning: Data cleaning is very relevant in these kind of problems. In our experiments, it has been mandatory to eliminate duplicated images when dealing with images that have been downloaded from twitter with a common hashtag (because there are many retweeted or copied images). In order to eliminate the duplicate images, we follow a simple strategy with very good results as follows: For a set of K images, $\{\mathbf{I}_k\}$, for $k = 1 \cdots K$, we convert each image $\mathbf{I}_k$ to a grayscale image $\mathbf{Y}_k$ and we resize it to a 64×64-pixel image $\mathbf{Z}_k$ using bicubic interpolation [5]. In addition, the gray values of the resized image is linearly scaled from 0 to 255. The resulting image is converted into a column vector $\mathbf{z}_k$ of $64^2 = 4096$ elements with uni-norm. Thus, we remove from the set of images those duplicated images that have a dot product $\mathbf{z}_i^{\mathsf{T}}\mathbf{z}_j > 0.999$ for all $i \neq j$. In our experiments, from 1/4 to 1/3 of the images were eliminated because they were duplicated. This method removes efficiently and quickly those duplicated images that have been scaled, however, this method does not remove those rotated or translated images (approx. $1 \sim 2\%$ of the images). In case it is necessary to remove rotated or translated images, a strategy using SIFT points can be used [10].

3. Metadata extraction: Usually, the downloaded images have associated metadata, *e.g.*, date and time of capture, or date and time of the tweet, or GPS information that can be used as geo-reference. In many images, the metadata information is stored in the same image file as EXIF data (Exchangeable Image File Format).
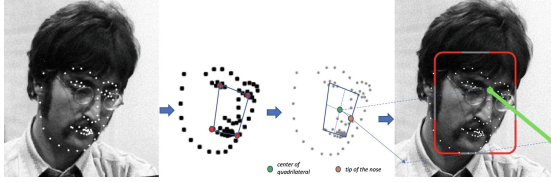
2.2   Computer Vision Tools

1. Face Detection: Face detection identifies faces in an image. In our work, the goal is to detect all faces that are present in an image independent on the expression, pose and size. For this end, we use the method called Multi-task Cascaded Convolutional Networks (MTCNN) [21][5] that has been demonstrated to be very robust in unconstrained environments against poses, illuminations, expressions and occlusions. The output of the face detection function (h) of a

---

[2] https://developer.twitter.com.
[3] https://www.instagram.com/developer/.
[4] https://www.flickr.com/services/api/ or software Bulkr.
[5] https://github.com/kpzhang93/MTCNN_face_detection_alignment.

Fig. 2. Landmarks of a face and estimation of its pose vector.

given image $\mathbf{I}$ is bounding box $\mathbf{B}$ which defines a rectangle that contains the face:

$$\{\mathbf{B}_k\}_{k=1}^N = h(\mathbf{I}). \tag{1}$$

For N faces detected in image $\mathbf{I}$, we define the founding box $\mathbf{B}_k = (x_1, y_1, x_2, y_2)_k$, where $(x_1, y_1)_k$ are the coordinates of the top-left corner and $(x_2, y_2)_k$ the coordinates of the bottom-right corner of detected face image k. From these coordinates, it is possible to extract face image $\mathbf{F}_k$, *i.e.*, the rectangular window of $\mathbf{I}$ defined by the mentioned two corners.

2. Face Location and size: From the bounding box of the face detected in previous step, it is possible to establish the location and the size of the face image. Typically, the center of mass of the bounding box is used: $\bar{\mathbf{m}}_k = (\bar{x}_k, \bar{y}_k)$, with $\bar{x}_k = (x_1 + x_2)_k/2$ and $\bar{y}_k = (y_1 + y_2)_k/2$. This information can be used to establish the closeness between two faces i and j as $d_{ij} = ||(\mathbf{m}_i - \mathbf{m}_j)||$. In addition, the size of an image can be computed as the geometric mean of the length of the sides of the rectangle: $s_k = \sqrt{(x_2 - x_1)_k(y_2 - y_1)_k}$.

3. Quality: Typically, face images that are smaller than 25×25 pixels, *i.e.*, $s_k < 25$, are not so confident because of the low quality and low resolution. In addition, for the measurement of quality of a face image, we use a score based on the ratio between the high-frequency coefficients and the low-frequency coefficients of the wavelet transform of the image [14]. We call this quality measurement q_k for face image k. Low score values indicate low quality. For this end, we resized all face images to 64×64 pixels before the blurriness score is computed. It is recommended to remove those face images that are too small or too blur.

4. Age, gender, expressions: The age, gender and facial expressions of a person can be estimated from the face image. Many models based on convolutional neural networks have been trained in the last years with promising results. The library py-agender[6] offers very good results for age and gender estimation. The age, given in years, is estimated as a real number and can be stored in variable a_k for face k. On the other hand, the gender is estimated as a real number between 0 and 1 (greater than 0.5 means female, otherwise male). The gender value for face k can be stored in variable g_k. Finally, the facial expressions are

---

[6] https://pypi.org/project/py-agender/.

typically defined as a vector of seven probabilities for the seven main expressions [2]: angry, disgust, scared, happy, sad, surprised, and neutral. Thus, $\mathbf{e}_k$ can be defined as the 7-element vector of expression for face k. It can be established, for example, that if the fourth element of vector $\mathbf{e}_k$ is maximal, then face image k shows a smily face.

5. Face Landmarks: In the same way, using a large dataset of face images with different poses, models have been trained to extract landmarks in a face image. Typically, 68 facial landmarks can be extracted from a face image. They give the coordinates (x, y) of the eyebrows (left and right), eyes (left and right), nose, mouth and jawline. For each of them several salient points are given (see Fig. 2). For this end, we use the library Dlib[7] with very good results. The landmarks of image k are stored in the 68-element vector $\mathbf{l}_k$.

6. Face Pose: We use a simple and fast method to establish the pose of the face given its 68 landmarks as follows (see Fig. 2): we define a quadrilateral with the four corners defined by the center of mass of each eye and the extrema points of the mouth, we compute the center of this quadrilateral and we define the vector that starts at this central points and goes through the point of the tip of the nose. The vector is shifted and located between the eyes. We call this vector $\mathbf{v}_k$ for face image k. The direction of the vector indicates approximately the direction the face is looking to.

7. Face Descriptor: Face recognition by using complex deep neural network architectures trained with millions of face images has achieved a tremendous improvement in the last years. The models have been trained with thousands of identities, and each of them with thousands of face images. The idea is to use these models and extract the descriptor embedded in one of the last layers. These kind of descriptors are very discriminative for faces that have not been used in the training. That means descriptors extracted from face images of same/different subjects are similar/different. Thus, the idea is to extract a descriptor $\mathbf{x}$, a column vector of d elements, for every face image:

$$\mathbf{x}_k = f(\mathbf{F}_k) \tag{2}$$

We use descriptor with uni-norm, *i.e.*, $||\mathbf{x}_k|| = 1$. In our experiments, we used many trained models (like VGG [13], FaceNet [17], OpenFace [1], Dlib [8] and ArcFace [4]). Our conclusion is that ArcFace, that computes an embedding of $d = 512$ elements, has achieved outstanding results comparing its performance to human vision in many complex scenarios. Thus, we can establish that for face images i and j of the same person the dot product $\mathbf{x}_i^\mathsf{T}\mathbf{x}_j$ is greater than a threshold. For ArcFace, in our experiments we set the threshold to 0.4.

8. Face Clustering: The idea of face clustering is to build subsets (clusters) of faces that belong to the same identity. Typically, face clustering works using a similarity metric of the face descriptors, because face images of the same identity

---

[7] http://dlib.net/face_landmark_detection.py.html.

should have similar face descriptors. Thus, the task is to assign –in an unsupervised way– all similar face faces to one cluster considering that different faces must belong to different clusters. For a set of m face images, in which face image $\mathbf{F}_k$ has a face descriptor $\mathbf{x}_k$ computed by (2), face clustering assigns each descriptor $\mathbf{x}_k$ to a cluster c_k, for $i = 1 \cdots m$. Thus, face images of the same identity have the same cluster number (*e.g.*, if face images 10, 35 and 221 are from the same subject, then $c_{10} = c_{35} = c_{221}$). For this end, we use an agglomerative hierarchical clustering [12]. Since our descriptors has unit norm, we use cosine similarity as metric, the closer to one is the dot product $\mathbf{x}_k^\mathsf{T}\mathbf{x}_j$, the more similar are the faces $\mathbf{F}_k$ and $\mathbf{F}_j$. The algorithms of face clustering is as follows: *(i)* each face image starts in its own cluster, *(ii)* we merge cluster i with cluster j if the maximal cosine similarity of all combinations of members of both clusters is maximal for all $i \neq j$ and $i < j$, *(iii)* last step is repeated until the maximal cosine similarity is below to a threshold.

2.3   Facial Analysis

In this section, we present our proposed facial analysis. We assume that the images have been downloaded from the social network, the duplicated images have been removed and the existing metadata has been stored as explained in Sect. 2.1. Before we perform the analysis, it is necessary to do some preliminary computations as explained in Sect. 2.2 to generate a relational database of two tables, one for the images and one for the faces. For each face image of all images we have following information, bounding box, size, quality, location, age, gender, expressions, landmarks, pose, face descriptor and face cluster.

0. Preliminary Computations: The idea of our approach is to analyze a set $\mathcal{I}$ of n images $\{\mathbf{I}_i\}$, for $i = 1 \cdots n$. The images of set $\mathcal{I}$ should not be duplicated. It is recommended to follow the procedure explained in sub-section 2.1.2 for images downloaded from Twitter to avoid duplicate ones. We detect all faces of $\mathcal{I}$ using function h of (1) explained in sub-section 2.2.1. All detected faces are stored as set $\mathcal{F}$ of m face images $\{\mathbf{F}_k\}$, for $k = 1 \cdots m$. In addition, we store in vector $\mathbf{z}$ of m elements the image index of the detected face image, *i.e.*, $z_k = i$, if face image $\mathbf{F}_k$ was detected in image $\mathbf{I}_i$. Furthermore, the m bounding boxes of the detected faces are stored in matrix $\mathbf{B}$ of $m \times 4$ elements with coordinates $\mathbf{b}_k = (x_1, y_1, x_2, y_2)_k$ for face image k.

After face detection is performed, for each face image k, we compute the size (s_k) and the quality (q_k) as explained in sub-sections 2.2.2 and 2.2.3. It is highly recommended to remove from $\mathcal{F}$ those images that are too small of too blur. Afterwards, we compute for the remaining face images the age (a_k), the gender (g_k), the seven expressions $(\mathbf{e}_k)$, the 68 landmarks $(\mathbf{l}_k)$, the pose vector $(\mathbf{v}_k)$ the face descriptor of d elements $(\mathbf{x}_k)$ as explained in sub-sections from 2.2.4 to 2.2.7. It is very useful to define matrix $\mathbf{X}$ as a matrix of $d \times m$ elements (one column per face descriptor), in which column k stores descriptor $\mathbf{x}_k$. Finally, we compute the cluster of each face image (c_k) following the face clustering algorithm explained in sub-section 2.2.8.

1. Search for subjects (recognition): There are two typical ways to search a person in the set of m face images (with m face descriptors stored in matrix $\mathbf{X}$ of $d \times m$ elements). The first one is using an enrolled picture $\mathbf{E}$ and its corresponding descriptor computed by (2) as $\mathbf{x}_e = f(\mathbf{E})$. The second one is using a detected face of the group of images. For instance, we find a face in an image of the set and we want to know where is this person in the rest of images. In this case, we define $\mathbf{x}_e = \mathbf{x}_j$, where j is the number of the detected face in the group, and we delete this face from the gallery by setting column j of $\mathbf{X}$ to zero. There are three main approaches that can be used to find the enrolled person in the images of set $\mathcal{I}$. (a) Similar faces: It is necessary to compute the similarity between 'enrolled image' and 'gallery images' as $\mathbf{y} = \mathbf{X}^\mathsf{T}\mathbf{x}_e$. Thus, we find all elements $y_k > \theta$, that means, images $\mathbf{F}_k$, that are located in bounding box $\mathbf{B}_k$ in image $\mathbf{I}_{z_k}$. (b) Clustered face images: using last approach (a), we look for the most similar face image in the gallery as $k = \mathrm{argmax}(y_k)$ and we find all face images that belong to the cluster of face image k, that means the subset of images that have cluster number $c_i = c_k$ for $i = 1 \cdots m$. c) Refine: In addition, we could find those face images that are similar enough to those already selected face images in previous steps (a) or (b). The output is a list $\mathbf{k} = (k_1 \cdots k_p)$ of the indices of p face images that belong to the person being searched.

2. Analysis of expressions: From list $\mathbf{k}$ of face images (that belong to the same person) we could analyze the expression of each face image of the list. There are two simple ways to analyze them: a) Average: we compute the average of the expressions: $\bar{\mathbf{e}} = (\mathbf{e}_{k_1} + \cdots \mathbf{e}_{k_p})/p$. An histogram of $\bar{\mathbf{e}}$ show the distribution of expressions across the p face images. b) Predominant expression: we can define vector $\hat{\mathbf{e}}$, in which the element j of this vector is the ratio of face images of $\mathbf{k}$ that have the expression j maximal. For instance, if we have $p = 20$ face images, and in 5 of them the expression 'happy' (the fourth expression) is maximal, then $\hat{e}_4 = 5/20 = 25\%$. Obviously, we could find the *happiest picture*, by looking for the face image that have the maximal value in the fourth element of vector $\mathbf{e}$.

3. Analysis of age: Similarly to the analysis of expression, we can compute the average of the age, we can select the *oldest* one, or we can sort the face image according to the estimated ages.

4. Co-occurrences: Using the clustering information we could analyze the other faces that are present in the images where the person being searched appears. It is easy to count the number of co-occurrences. For instance, if the person being searched belongs to cluster c_i, it is easy to count the number of images in which faces from cluster c_i and faces from cluster c_j are present. We can find the pair (c_i, c_j) that has the maximal co-occurrence. In our experiments in family albums, this pair corresponds typically to couples. In addition, it is very simple to add some constraints to person c_j in the co-occurrence, for example the gender of c_j must be female or male, or the age must be older or younger than a certain age, or we can select the happiest pictures of persons c_i and c_j. Moreover, we can select co-occurrence pairs of face images that are very close to each other, *e.g.*, $||\bar{\mathbf{m}}_i - \bar{\mathbf{m}}_j|| < 3(s_i + s_j)/2$, and in order to avoid perspective problem, both face images should have similar size, *e.g.*, $|1 - s_i/s_j| < 0.15$.

5. Connections: We can use the pose information in a picture as follows: if we have a picture with two faces (face i and face j), it is possible to analyze the face poses (vectors $\mathbf{v}_i$ and $\mathbf{v}_i$) by estimating if the intersection of both vectors are between of in front of the faces. The distance of the intersection point to the faces can be used to determine how connected are to each other. In addition, if the vectors are parallel to each other it can be established that both persons are looking at the same direction.

6. Attendance: If we have pictures of the same place in different days, and we have the metadata of the date of the images, it is easy to establish if a person was present across the days. This is very typical in a student attendance system.

3   Experimental Results

In this section, we report the experiments that we used to validate the proposed approaches. For this end, we used sets of images download from Twitter, YouTube, Flickr and GitHub. On these sets of images, we tested the following facial analysis techniques: recognition, expressions, ages, co-occurrences, connections and attendance.

3.1   Datasets

In order to test our algorithms, we used the following datasets:

1. Twitter - The Beatles: On July 9th, 2019, we downloaded images from Twitter given the hashtags `#TheBeatles` and `#Beatles`, and from the accounts `@TheBeatles` and `@BeatleHeadlines`. In these images, we can observe many pictures of the famous English rock band 'The Beatles' and its members (Paul McCartney, John Lennon, George Harrison and Ringo Starr) in many poses, facial expressions and with different ages. This dataset has 1266 images, 452 were removed because they were duplicated, and 2228 faces were detected.

2. Twitter - Donald Trump: On July 19th, 2019, we downloaded images from Twitter given the hashtags `#Trump` and `#DonaldTrump`. In those days, Trump suggested on Twitter that the legislators that "originally came from countries whose governments are a complete and total catastrophe" should "go back" to those "totally broken and crime infested places"[8]. In the downloaded images, we can observe many reactions for and against the four mentioned Democratic congresswomen. This dataset has 494 images, 126 were removed because they were duplicated, and 677 faces were detected.

3. Flickr - Family Album: On July 18th, 2019, we downloaded from Flickr twelve different family albums of pictures taken by Sandra Donoso (username sandrli)[9]. The pictures are licensed under a Creative Commons "Attribution-NonCommercial-NoDerivs 2.0 Generic". In these pictures, we can observe the

---

[8] https://time.com/5630316/trump-tweets-ilhan-omar-racist-conspiracies/.
[9] https://www.flickr.com/photos/sandreli/albums.

members of the family in celebrations and visiting vacation places in the last 5 years. This dataset has 1478 images, and 2838 faces were detected.

4. Flickr - Volleyball Game: On July 2nd, 2019, we downloaded from Flickr the album "VBVBA RVC 2 2010" of pictures taken by Bruce A Stockwell (username bas68)[10]. The pictures are licensed under a Creative Commons "Attribution-NonCommercial-NoDerivs 2.0 Generic". In these pictures, we observe pictures of different volleyball games played on April 2010 by teenage players. This dataset has 1131 images, and 4550 faces were detected.

5. YouTube - Films of the 90s: We downloaded the summaries done by WatchMojo.com of the "Top 10 Most Memorable Movies of 199" and the "Top 10 Movies of the 1990s" (12 min each)[11] and took one frame per second to build the set of images. In these images, we can observe movies like 'Matrix', 'Schindler List', 'Pulp Fiction', 'Ghost',etc. This dataset has 1492 images, and 1449 faces were detected.

6. GitHub - Classroom: A dataset for student attendance system in crowded classrooms was built in [11] with pictures taken in 25 sessions. The dataset contains pictures of a classroom with around 70 students[12]. In each dataset, approx. 6 pictures have been taken per session. Very useful for this dataset is the metadata of the dates on which each picture was taken. With this information, it is possible to establish the attendance record of each student previously enrolled (an enrolled face image is available for all students). This dataset has 153 images, and 5805 faces were detected.

3.2 Experiments

For all datasets mentioned in previous section, we performed the preliminary computations of section 2.3.0. For each analysis mentioned in Sect. 2.3, we show in this section at least one example.

1. Search for subjects: Given a face image of a volleyball player, in this example we show how this person was searched in all images of dataset 'Flickr-Volleyball Game'. The person was found in 170 images, twelve of them are shown in Fig. 3. We can see that the person was perfectly found in very complex scenarios with different facial expressions, poses and occlusion. The reader can check the effectiveness of the method by recognizing the number '15' in her T-shirt.

2. Analysis of expression: Given a face image of Paul McCartney, in this example we show how he was searched in all images of dataset 'Twitter-The Beatles', his facial expressions were analyzed and the happiest pictures was displayed. 100 face images were sorted in a descending way from more to less happy (see Fig. 4). We can see that after this analysis, in 21% of the pictures in which he appears, the expression 'Happy' was maximal.

---

[10] https://www.flickr.com/photos/bas68/albums/72157624234584197.

[11] https://youtu.be/5GvBPtlb-Ms and https://youtu.be/dA-KcQ5BzUw.

[12] https://github.com/mackenney/attendance-system-wacv19.

3. Analysis of age: Given a face image of each member of The Beatles, in this example we show how they were searched in all images of dataset 'Twitter-The Beatles', their ages were analyzed and 100 face images of each one were sorted in ascending way from younger to older (see Fig. 5). We can see that the method is able to recognize and sort face images of Ringo Starr and Paul McCartney when they were very young (around 20 years old) and how they are now (older than 75 years old).

4. Co-occurrence: Given a face image of a young man, in this example, we search pictures in which he appears with other persons in all images of dataset 'Flickr-Family Album'. We select from them the most present woman and show the pictures in which he and she appears together. The result is shown in Fig. 6. It is very impressive to see that the pictures correspond to a couple in different moments of its life.

5. Connections: Given pictures extracted from dataset 'YouTube - Films of the 90s', it is possible to analyze the pose vectors of the faces as shown in Fig. 7. In another experiment, given a face of Donald Trump, in this example we show how he was searched in all images of dataset 'Twitter-Trump'. We select one picture and we analyze the connections, that means which persons are close to each other, and which pairs are looking to each other (see Fig. 8). In addition, we can cluster them by closeness and compute a graph of connections: 'A → B' means person A is looking to person B, 'A - - - B' means the intersection of pose vectors of A and B is close to the faces of A and B. Moreover, the expressions of each person can be estimated.

6. Student Attendance: Given pictures of enrolled students, we can establish the attendance record of each student in dataset 'GitHub-Classroom'. In this

Fig. 3. Searching a volleyball player in dataset 'Flickr-Volleyball'.

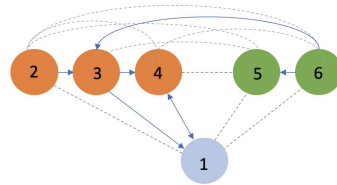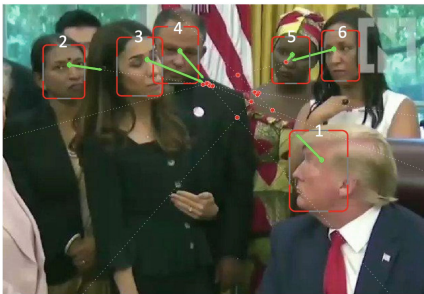**Fig. 4.** The happiest pictures of Paul McCartney, and his expression analysis.

Fig. 5. 100 face images of each member of The Beatles sorted from younger to older.

Fig. 6. Female co-occurrences of young man in dataset 'Flickr-Family Album'.

Fig. 7. Connections in face images of dataset 'YouTube - Films of the 90s'.

Person	Angry	Disgust	Scared	Happy	Sad	Surprised	Neutral
1	8%	0%	19%	10%	19%	7%	36%
2	16%	1%	21%	8%	16%	2%	37%
3	9%	1%	14%	42%	10%	4%	20%
4	5%	0%	6%	55%	5%	2%	27%
5	11%	1%	13%	34%	6%	2%	34%
6	7%	0%	6%	14%	14%	3%	54%

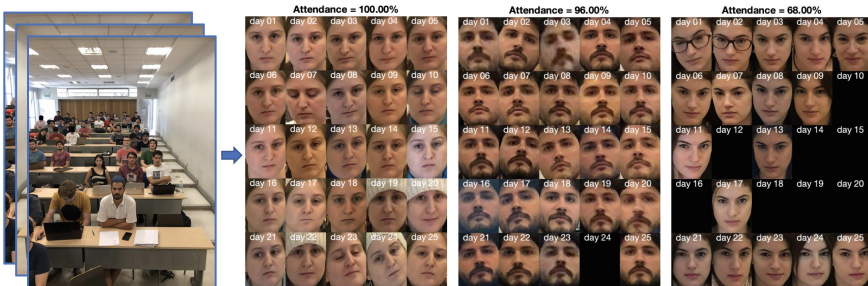**Fig. 8.** Connections and expressions in a picture of dataset 'Twitter-Trump' (see text).

Fig. 9. Student attendance record of three students in 25 sessions.

example, we search three students in 25 sessions. The results are shown in Fig. 9, in which the attendance was 100%, 96% and 68%. It is very easy to see when students 2 and 3 were absent.

4   Final Discussion

In this paper, we presented how easy is to develop very effective applications with open source tools. From a group of pictures (downloaded for example from social networks), we can build a relational database of the images with their faces and facial attributes. With a simple query on this database we can retrieve very accurate information, *e.g.*, we can search very quickly a person, extract age, gender, and facial expressions, find the person that most frequently co-occurred with him/her, the connections and the other persons that he/she is watching, etc. Surprisingly, no training is necessary, because the required deep learning models are already trained and available in public repositories. Thus, anyone with certain computer skills can use (or misuse) this technology.

Face analysis has been assimilating into our society with surprising speed. However, privacy concerns and false identification problems in facial recognition software have gathered an anti-surveillance movement[13]. The city of San Francisco, for example, recently banned facial recognition technology by the police and other agencies[14]. We think that the warnings are clear and it is time to discuss the social and ethical challenges in facial analysis technologies. In this way, we can reduce errors that have severe social and personal consequences.

In this direction, some ethical principles that can be considered when using and teaching a technology based on facial analysis are the following:

- It must respect human and civil rights such as privacy and non-discrimination.
- It must not decide autonomously in cases that require human analysis/criteria.
- It must be developed and implemented as a trustworthy system[15].
- Its pros and cons, such as recognition rates and false matching rates, must be rigorously evaluated before operational use.
- It must be lawful, that means capturing, processing, analyzing and storing of images must be regulated and accepted by the individuals.

Since there is no clear regulation in this field, we believe that our community should discuss the scope and limitations of this technology in terms of the definition of good practices, standards, and restrictions when using facial analysis. It is time to deepen our understanding of the ethical impact of facial analysis systems, in order to regulate and audit these processes.

---

[13] https://www.biometricsinstitute.org/facial-recognition-systems-and-error-rates-is-this-a-concern/.

[14] https://edition.cnn.com/2019/07/17/tech/cities-ban-facial-recognition.

[15] https://ec.europa.eu/digital-single-market/en/news/ethics-guidelines-trustworthy-ai.

References

1. Amos, B., Ludwiczuk, B., Satyanarayanan, M.: Openface: A general-purpose face recognition library with mobile applications. Tech. rep., CMU-CS-16-118, CMU School of Computer Science (2016)
2. Arriaga, O., Valdenegro-Toro, M., Plöger, P.: Real-time convolutional neural networks for emotion and gender classification. arXiv preprint arXiv:1710.07557 (2017)
3. Baltrusaitis, T., Zadeh, A., Lim, Y.C., Morency, L.P.: Openface 2.0: facial behavior analysis toolkit. In: 2018 13th IEEE International Conference on Automatic Face & Gesture Recognition (FG 2018), pp. 59–66. IEEE (2018)
4. Deng, J., Guo, J., Xue, N., Zafeiriou, S.: Arcface: additive angular margin loss for deep face recognition. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 4690–4699 (2019)
5. Gonzalez, R., Woods, R.: Digital Image Processing. Prentice Hall, third edn, Pearson (2008)
6. Hoai, M., Zisserman, A.: Talking heads: Detecting humans and recognizing their interactions. In: Computer Vision and Pattern Recognition (CVPR), pp. 875–882 (2014)
7. Huang, X., Dhall, A., Goecke, R., Pietikäinen, M., Zhao, G.: Multimodal framework for analyzing the affect of a group of people. IEEE Trans. Multimedia 20(10), 2706–2721 (2018)
8. King, D.E.: Dlib-ml: A machine learning toolkit. J. Mach. Learn. Res. 10, 1755–1758 (2009)
9. Li, P., Prieto, L., Mery, D., Flynn, P.J.: On low-resolution face recognition in the wild: comparisons and new techniques. IEEE Trans. Inf. Forensics Secur. 14(8), 2000–2012 (2019)
10. Lowe, D.G.: Distinctive image features from scale-invariant keypoints. Int. J. Comput. Vis. 60(2), 91–110 (2004)
11. Mery, D., Mackenney, I., Villalobos, E.: Student attendance system in crowded classrooms using a smartphone camera. In: 2019 IEEE Winter Conference on Applications of Computer Vision (WACV), pp. 857–866. IEEE (2019)
12. Otto, C., Wang, D., Jain, A.K.: Clustering millions of faces by identity. IEEE Trans. Pattern Anal. Mach. Intell. 40(2), 289–303 (2017)
13. Parkhi, O.M., Vedaldi, A., Zisserman, A.: Deep face recognition. In: British Machine Vision Conference (BMVC2015). vol. 1, p. 6 (2015)
14. Pertuz, S., Puig, D., Garcia, M.: Analysis of focus measure operators for shape-from-focus. Pattern Recogn. 46(5), 1415–1432 (2013)
15. Renoust, B., Kobayashi, T., Ngo, T.D., Le, D.D., Satoh, S.: When face-tracking meets social networks: a story of politics in news videos. Appl. Netw. Sci. 1(1), 4 (2016)
16. Rothe, R., Timofte, R., Van Gool, L.: Deep expectation of real and apparent age from a single image without facial landmarks. Int. J. Comput. Vis. 126(2–4), 144–157 (2018)
17. Schroff, F., Kalenichenko, D., Philbin, J.: FaceNet: a unified embedding for face recognition and clustering. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 815–823 (2015)
18. Shi, Y., Otto, C., Jain, A.K.: Face clustering: representation and pairwise constraints. IEEE Trans. Inf. Forensics Secur. 13(7), 1626–1640 (2018)

19. Sun, X., Wu, P., Hoi, S.C.: Face detection using deep learning: An improved faster rcnn approach. Neurocomputing **299**, 42–50 (2018)
20. Zavan, F.H.D.B., Bellon, O.R., Silva, L., Medioni, G.G.: Benchmarking parts based face processing in-the-wild for gender recognition and head pose estimation. Pattern Recogn. **123**, 104–110 (2019)
21. Zhang, K., Zhang, Z., Li, Z., Qiao, Y.: Joint face detection and alignment using multitask cascaded convolutional networks. IEEE Signal Process. Lett. **23**(10), 1499–1503 (2016)
22. Zhang, Z., Luo, P., Loy, C.C., Tang, X.: Learning social relation traits from face images. In: Proceedings of the IEEE International Conference on Computer Vision, pp. 3631–3639 (2015)
23. Zhu, M., Shi, D., Zheng, M., Sadiq, M.: Robust facial landmark detection via occlusion-adaptive deep networks. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 3486–3496 (2019)