Student Attendance System in Crowded Classrooms using a Smartphone Camera

Domingo Mery  
Universidad Católica de Chile  
domingo.mery@uc.cl

Ignacio Mackenney  
Universidad Católica de Chile  
imackenney@uc.cl

Esteban Villalobos  
Universidad Católica de Chile  
egvillalobos@uc.cl

Abstract
To follow the attendance of students is a major concern in many educational institutions. The manual management of the attendance sheets is laborious for crowded classrooms. In this paper, we propose and evaluate a general methodology for the automated student attendance system that can be used in crowded classrooms, in which the session images are taken by a smartphone camera. We release a realistic full-annotated dataset of images of a classroom with around 70 students in 25 sessions, taken during 15 weeks. Ten face recognition algorithms based on learned and handcrafted features are evaluated using a protocol that takes into account the number of face images per subject used in the gallery. In our experiments, the best one has been FaceNet, a method based on deep learning features, achieving around 95% of accuracy with only one enrollment image per subject. We believe that our automated student attendance system based on face recognition can be used to save time for both teacher and students and to prevent fake attendance.

1. Introduction
To follow the attendance of students is a major concern in many educational institutions. The manual management of the attendance sheets is laborious and tedious for crowded classrooms. In our experience, the time invested for this task in a 70-student classroom is about 4 minutes, i.e. in the whole semester the total invested time could be longer than the duration of one lecture of 80 minutes. We believe that an automated student attendance system—based on face recognition—can be used to save time for both teacher and students, and to prevent fake attendance. This system could be part of a Next Generation Smart Classrooms[55], in order to improve teaching and learning experience in the classroom.

In face recognition, the task is to identify a subject appearing in an image as a unique individual. Over the last decade, we have witnessed tremendous improvements in face recognition algorithms. Some applications, that might have been considered science fiction in the past, have become reality now. However, it is clear that face recognition, is far from perfect when tackling more challenging images such as faces taken in unconstrained environments e.g. face images acquired by long-distance cameras. Although innovative methods in computer vision have improved the state of the art, the performance obtained in low-quality images remains unsatisfactory for many applications. This has been the case of face recognition in crowded classrooms for a student attendance system as illustrated in Fig. 1.

In this paper, we propose an automated student attendance system that can be used in crowded (and small) classrooms. In this application, after an enrollment stage, in which a face image of each student is acquired and the corresponding ID is registered, the user, e.g. the teacher, can take one or several pictures of the classroom using his/her smartphone in order to capture all students that are present. The proposed algorithm detects the faces in the picture(s) and recognize which students are present or absent in order to record the attendance of the class.

The main contributions of the paper are the following:

- A full-annotated dataset of images of a classroom with 67 students in 25 sessions, taken by a smartphone camera during 15 weeks. An example is shown in Fig. 1.
- A simple method based on known deep learning models implemented in Python that can be used as Student Attendance System.
- An evaluation protocol that takes into account the number of face images per subject in the gallery to compute the average accuracy in 25 sessions.
- A comparison of ten different face recognition methods in this task.

All enrollment images, session images, cropped face images, extracted descriptors and code are available at our webpage.
Figure 1: Results of our Student Attendance System. In this example of 25 sessions, the selected student was not present in 5 sessions (see black squares in the right image), i.e., the attendance for him is 20/25 = 80.0%. In the middle image, the session of March 29th is shown, where the student was recognized in the last row of the classroom (see zoom of the red square in the left panel). In these experiments, the images were acquired using a smartphone camera.

The rest of the paper is organized as follows. In Section 2, a literature review in face recognition in low-quality images and in student attendance systems is presented. In Section 3, the proposed method is explained in further detail. In Section 4, the experiments, results and implementation are presented. Finally, in Section 5, concluding remarks are provided.

2. Related Work

In this Section, the literature review is focused on face recognition in low-quality images and student attendance systems.

2.1. Face Recognition in Low-quality Images

Over the last decade, 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 [53][43][50][63][33][45]. In this Section, however, the focus is on face recognition in low-quality images [36].

It is clear that image degradation, like blurriness, affects the performance of face recognition algorithms based on deep learning [30][14][13]. Image enhancement or considering of degraded samples in training dataset could lead to better deep learning models [20]. In order to cast the problem of face recognition in low-quality images, a straightforward approach is to estimate a high-quality image from one that is low-quality. Face recognition is then performed as normal with high-quality face images. This process involves image restoration techniques in the case of blurred images, and super-resolution techniques in the case of low-resolution images. Among image restoration techniques, we can identify blind deconvolution [35], non-blind deconvolution [65], regularization methods on total variation [48], and Tikhonov regularization [54]. In addition, there are more direct methods based on features of the image that are invariant to blurriness, such as processing images in spatial and frequency domains [15][18]. Nevertheless, the level of restoration is not satisfactory enough for severe blurriness. Conversely, super-resolution techniques, known as face-hallucination for low-resolution face images [4], attempt to estimate a high-resolution face image from one that is low-resolution. We can identify techniques based on sparse representations [62], patch-oriented strategies [7] and deep learning features [58], among others. Unfortunately, these methods do not obtain an adequate reconstruction of the high-quality face image when the resolution of the input image is very low, e.g. less than 22×15 pixels.

Novel methods that do not follow the above mentioned straightforward approach have been proposed in recent years. Some of these have attempted to perform face recognition by simultaneously computing super-resolution and feature extraction so as to measure the low and high frequencies of the face images [24]. Other methods extract features from face images in resized formats [25][34]. Finally, there are also methods that construct a common feature space (called inter-feature space [59]) for matching between low- and high-resolution features [42][5][47][60][23]. Although these innovative methods have improved the state of the art, the performance obtained in low-quality images remains unsatisfactory for many applications such as forensics and video surveillance.
In the last three years, novel methods based on deep learning for low-quality face images have been developed. In [57], Partially Coupled Networks are proposed for unsupervised super-resolution pre-training. The classification is by fine-tuning on a different dataset for specific domain super-resolution and recognition simultaneously. In [27], an attention model that shifts the network’s attention during training by blurring the images with various percentage of blurriness is presented for gender recognition. In [41], three obfuscation techniques are proposed to restore face images that have been degraded by mosaicing (or pixilation) and blurring processes. In [6], a multitask deep model is proposed to simultaneously learn face super-resolution and facial landmark localization. The face super-resolution subnet is trained using a Generative Adversarial Network (GAN) [11]. In [26], inspired by the traditional wavelet that can depict the contextual and textual information of an image at different levels, a deep architecture is proposed. In [8], a network that contains a coarse super-resolution network to recover a coarse high-resolution image is presented. It is the first deep face super-resolution network utilizing facial geometry prior to end-to-end training and testing. In [10], a deblurring network for deblurring facial images is proposed using a ResNet-based non-maxpooling architecture. In [64], a face hallucination method based on an upsampling network and a discriminative network is proposed. In [51], global semantic priors of the faces are exploited in order to restore blurred face images. In all these methods, we see that computer face recognition is far from perfect when tackling more challenging such as faces taken in unconstrained environments, surveillance, forensics, etc.

In the literature review presented in this Section, we concluded that finding techniques to improve face recognition in low-quality images is an important contemporary research topic. A very challenging application is to manage a student attendance record in a crowded classroom using a smartphone camera as illustrated in Fig. 1.

### 2.2. Student Attendance System

In the literature, there are some works that reported student attendance systems. A summary of them (since 2012) is presented in Table 1. The most relevant are discussed in the following. In [46], a face recognition approach based on eigen-faces is presented for a classroom of 15 students. In this approach, the testing images are individual face images instead of images of the whole classroom, that means, no face detection is required. In [38], not only the students are recognized, but also their behavior (e.g., activities like entering or leaving the classroom). The experiments are conducted in a 9-student classroom. In [31], an observation camera with fisheye lens is used on the ceiling of the classroom to detect where the students are sitting, and another camera is directed to the selected seat using the pan/tilt/zoom. In [49], a method based on deep learning is presented. The size of trained model is small enough to be installed in simple microprocessors. The method achieves excellent results in small classrooms using a high-resolution reflex camera. In [9], a method based on LBP and SVM is presented for recognition of faces at the entrance of the classroom. In the query images, few students are present and the resolution of the face images is 100×100. In [32], a comparison is given between controlled and uncontrolled environments. Unsurprisingly, the uncontrolled environments are prone to error. In [22], experiments were conducted with different illuminations. They concluded that the more the quality of the images the more the accuracy of the recognition.

In all these methods, the evaluation protocol used to estimate the accuracy is not clear enough to be reproduced.

<table>
<thead>
<tr>
<th>Reference</th>
<th>Year</th>
<th>Approach</th>
<th>Input</th>
<th>Number of subjects in the query image</th>
<th>Number of enrolled subjects</th>
<th>Size of query face image</th>
<th>Accuracy†</th>
<th>Sessions</th>
</tr>
</thead>
<tbody>
<tr>
<td>Kar et al. [29]</td>
<td>2012</td>
<td>Eigen-faces</td>
<td>Still images</td>
<td>1</td>
<td>10</td>
<td>50×50</td>
<td>95.0%</td>
<td>–</td>
</tr>
<tr>
<td>Chintalapati et al. [9]</td>
<td>2013</td>
<td>LBP, PCA, SVM</td>
<td>Still images</td>
<td>2</td>
<td>80</td>
<td>100×100</td>
<td>78.0%</td>
<td>–</td>
</tr>
<tr>
<td>Wagh et al. [56]</td>
<td>2015</td>
<td>Eigen-faces</td>
<td>Still images</td>
<td>–</td>
<td>–</td>
<td>–</td>
<td>–</td>
<td>–</td>
</tr>
<tr>
<td>Lukas et al. [60]</td>
<td>2016</td>
<td>DWT+DCT</td>
<td>Still images</td>
<td>13</td>
<td>16</td>
<td>64×64*</td>
<td>82.0%</td>
<td>–</td>
</tr>
<tr>
<td>Assarasee [3]</td>
<td>2017</td>
<td>Microsoft API</td>
<td>Still images</td>
<td>5</td>
<td>5</td>
<td>–</td>
<td>80.0%</td>
<td>1</td>
</tr>
<tr>
<td>Fu et al. [16]</td>
<td>2017</td>
<td>Deep learning</td>
<td>Still images</td>
<td>5</td>
<td>7</td>
<td>80×60*</td>
<td>–</td>
<td>–</td>
</tr>
<tr>
<td>He et al. [22]</td>
<td>2017</td>
<td>Deep learning</td>
<td>Still images</td>
<td>1</td>
<td>64</td>
<td>–</td>
<td>100.0%</td>
<td>–</td>
</tr>
<tr>
<td>Kawaguchi [31]</td>
<td>2017</td>
<td>–</td>
<td>Still images</td>
<td>2</td>
<td>15</td>
<td>–</td>
<td>80.0%</td>
<td>1</td>
</tr>
<tr>
<td>Lim et al. [58]</td>
<td>2017</td>
<td>Fisher-faces</td>
<td>Video</td>
<td>9</td>
<td>9</td>
<td>45×34*</td>
<td>81.9%</td>
<td>1</td>
</tr>
<tr>
<td>Rekha et al. [46]</td>
<td>2017</td>
<td>Eigen-faces</td>
<td>Still images</td>
<td>1</td>
<td>15</td>
<td>64×48*</td>
<td>–</td>
<td>–</td>
</tr>
<tr>
<td>Surekha et al. [52]</td>
<td>2017</td>
<td>MKD-SRC</td>
<td>Video</td>
<td>20</td>
<td>20</td>
<td>–</td>
<td>60.0%</td>
<td>1</td>
</tr>
<tr>
<td>Polamarasetty et al. [44]</td>
<td>2018</td>
<td>HOG</td>
<td>Still images</td>
<td>3</td>
<td>14</td>
<td>112×92</td>
<td>–</td>
<td>–</td>
</tr>
<tr>
<td>Sarkar et al. [49]</td>
<td>2018</td>
<td>Deep learning</td>
<td>Still images</td>
<td>14</td>
<td>14</td>
<td>120×117</td>
<td>100.0%</td>
<td>1</td>
</tr>
<tr>
<td>Ours</td>
<td>2018</td>
<td>Deep learning</td>
<td>Still images</td>
<td>45</td>
<td>67</td>
<td>50×40</td>
<td>95.0%</td>
<td>25</td>
</tr>
</tbody>
</table>

† Accuracies are not comparable, because experiments are different. * Estimated size from test images presented in the published paper. ‘–’ means ‘not given’.

Table 1: Related works of automated attendance systems
That means, number of sessions, number of enrolled face images per subject, days between enrollment and testing, etc. are not clearly reported. It seems, that many of these recognition experiments have been conducted on images taken in only one session. Moreover, the subjects that are present in the query images are no more than 20, in many cases less than 10. In addition, the datasets and the implemented codes are not public, i.e. comparisons with the proposed methods is not possible. It is worthwhile to note that an attendance system must be used in a long period (sometimes a semester or a year), so the enrolled face images and the query face images can differ significantly in the time as shown in our dataset presented in Fig. 1. Thus, a dataset and a protocol that include more realistic scenarios with more sessions are required to design a robust attendance system.

3. Proposed Method

In this work, we explain the proposed algorithm for Student Attendance System. It is presented in Fig. 2. It consists of five steps: ① enrollment, ② capture of classroom images, ③ face detection and description, ④ query database and ⑤ matching algorithm. They will be addressed in the following five sub-sections in further details (one sub-section and one panel in Fig. 2 per step):

3.1. Enrollment

The enrollment of the participants is the first step in a face recognition system of a Student Attendance System. In this step, the biometric information of every subject of the classroom is captured and stored. As illustrated in Fig. 2, panel ①, we build the Enrollment Database by storing the face image, a description of the image (descriptor) and the ID of the subject. The last one is built using the identity information, such as name or ID number, given by the enrolled subject. The descriptor is a discriminative vector of \(d\) elements, e.g. for VGG-face model [43], \(d = 4.096\). The information of enrollment is necessary for the recognition. In the recognition stage, face images that belong to the same/different subject might have similar/different vectors. Thus, Euclidean distance or cosine similarity can be used to match the image faces [37].

In our work, we asked for a selfie of each student that was sent per e-mail to the teacher\(^2\). Usually, only one face image is required by a recognition system, however, in our experiments, due to the low image quality of the face images of the students that are sitting in the last rows of the classrooms (see Fig. 1), we improved the accuracy of the system by adding more face images to the gallery. The new face images can be those images that are detected in the classroom images and added manually to the database.

\(^2\)In our classroom, there were 74 attendees, 67 of them gave their permission to be part of this research.
In the enrollment database there are \(n\) subjects. In our experiments, \(n = 67\). Thus, the ID of the enrolled students will be a number \(i\) for \(i = 0, \cdots , n - 1\). The enrolled color face images, detected by a face detection algorithm, are defined as \(X_{ij}\), for \(j = 1 \cdots n_i\), where \(n_i\) is the number of enrolled face images of subject \(i\). The corresponding descriptions are defined as:

\[
x_{ij} = f(X_{ij})
\]

where \(f(\cdot )\) is the function that resizes the image if necessary and extracts the descriptor (a column vector of \(d\) elements) from the face image. In order to use the cosine similarity, function \(f\) returns a descriptor that has norm one. Thus, the similarity is computed by a simple dot product that corresponds to a normalized scalar product (cosine of angle). In our case, all enrolled face images \(\{X_{ij}\}\) have the same size: \(165 \times 120\) pixels. The size of the original face image was changed using bicubic interpolation [17].

### 3.2. Capture of Classroom Images

In each session, we take several images of the classroom in order to capture all students that are present in the classroom. The capture is a collaborative process because the students that are present want to be recognized by the attendance system. The attendees were asked to look at the camera and make a neutral expression. If the classroom is small enough, only one image is required, however, in our classroom, the angle of view of the camera did not cover the whole classroom, for this reason we had to take several images as shown in Fig. 2, panel 2. For this end, we used an smartphone (iPhone-8, iOS 11.2.6) with a resolution of \(4.032 \times 3.024\) pixels without flash. The color images were stored in HEIC format and converted to PNG using iMazing HEIC Converter 1.06, with 97% quality. Each converted image is stored in a file of 10MB approximately.

We define each captured session image as \(S_{pq}\), for \(p = 1 \cdots M\) and \(q = 1 \cdots n_p\), where \(M\) is the number of sessions (in our case, \(M = 25\) sessions) and \(n_p\) is the number of images captured in session \(p\). The number of images of each session is given in Table 2. Totally, we captured 153 images. In average, there were 6.1 images per session. All session images were captured in a period of 105 days (15 weeks). The days between consecutive sessions, \(t_{p+1} - t_p\), were in average 4.4 (that correspond to two sessions per week with some exceptions). The number of attendees that are present in each session is given as \(s_p\) in Table 2, in average, there are 51 subjects per session.

### 3.3. Face Detection and Description

In each image session, faces are detected automatically with a face detection algorithm (e.g. Dlib\(^3\)) and then manually checked as illustrated in Fig. 2, panel 3. The number of detected faces in session image \(S_{pq}\), i.e. image \(q\) of session \(p\), is defined as \(n_{pq}\). The detected faces in image \(S_{pq}\) are stored as \(Y_{pq}\), for \(r = 1 \cdots n_{pq}\). For the description, we use the same function defined in (1):

\[
y_{pq} = f(Y_{pq}).
\]

After this step, for each detected face defined, we have the location of its bounding box and its descriptor.

### 3.4. Query Database

In this step, we build the Query Database (see Fig. 2, panel 4) by storing all detected faces and their corresponding information. In the database, we store the following information for each detected face:
1. The detected image face.
2. The ground truth ID \((i)\), that is the ID of the face.
3. The number of the session \((p)\).
4. The number of the image of the session \((q)\).
5. The number of the detected face image in the image of the session \((r)\).
6. The location of the detected bounding box (coordinates \(x_1, y_1\) and \(x_2, y_2\) – the left-top and right-bottom pixel of the bounding box).
7. The descriptor \((y)\), a \(d\)-element vector computed by (2).

The ground truth ID of the Query Database is used for two purposes: i) For evaluation purposes, that is to determine the correctness of a matching; and ii) For testing purposes in which certain detected faces can be included in the gallery. For example, we can add to the Enrollment Database all face images of the first \(p\) sessions of the Query Database, and test the accuracy in the recognition with the rest, i.e. sessions \(p + 1 \cdots M\). The ground truth can be established manually or by a semi-supervised algorithm that checks for special consistency.

---

\(^3\)See http://dlib.net.

### Table 2: Details of the dataset

| Session \((p)\) | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 | 21 | 22 | 23 | 24 | 25 |
|----------------|---|---|---|---|---|---|---|---|---|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|
| Images \((m_p)\) | 6 | 6 | 5 | 4 | 8 | 6 | 5 | 11 | 10 | 7 | 5 | 8 | 4 | 8 | 4 | 6 | 6 | 4 | 6 | 4 | 6 | 4 | 6 | 4 |
| Day \((t_p)\) | 0 | 5 | 7 | 12 | 11 | 19 | 21 | 28 | 33 | 35 | 47 | 49 | 56 | 61 | 63 | 66 | 75 | 77 | 77 | 84 | 89 | 91 | 96 | 98 | 103 | 105 |
| Subjects \((s_p)\) | 60 | 60 | 61 | 62 | 56 | 58 | 52 | 54 | 56 | 47 | 55 | 49 | 49 | 45 | 47 | 48 | 56 | 48 | 52 | 46 | 46 | 49 | 50 | 39 | 39 | 50 |

\[^{3}\text{See http://dlib.net.}\]
3.5. Matching

As shown in Fig. 2, panel 5, a matching algorithm is used to establish if a subject is present in a session. A simple algorithm –based on cosine similarity– can be used to determine if subject $i$ is present in session $p$ as follows: we find if the maximal value of the dot product of all combinations $\langle x_{i,j}, y_{pq} \rangle$ is greater than a threshold $\theta$. We recall the reader that both vectors have norm one. This can be easily implemented using two matrices: $X_i$ and $Y_p$ defined by:

$$X_i = [x_{i,1} \cdots x_{i,j} \cdots x_{i,n_i}],$$

where $x_{i,j}$ is a $d \times 1$ vector defined in (1), and

$$Y_p = [\{y_{p,1}\} \cdots \{y_{p,q}\} \cdots \{y_{p,m_p}\}],$$

where $\{y_{p,q}\}$ is a matrix of $d \times n_{pq}$ elements defined as:

$$\{y_{p,q}\} = [y_{p,q,1} \cdots y_{p,q,r} \cdots y_{p,q,n_{pq}}].$$

where $y_{p,q,r}$ is a $d \times 1$ vector defined in (2). In this case, $X_i$ is a matrix of size $d \times n_i$, in which the $n_i$ descriptors of enrolled face images of subject $i$ are stored in columns of $d$ elements; and matrix $Y_p$ is of size $d \times n_p$, in which the descriptors of $n_p$ detected faces in session $p$ are stored in columns of $d$ elements, where $n_p$ is defined as $n_p = n_{p1} + n_{p2} + \cdots + n_{p_{mp}}$. All combinations of the dot product can be computed by

$$Z_{ip} = X_i^T Y_p,$$

where the result is a matrix of $n_i \times n_p$ elements. Thus, if

$$\max (Z_{ip}) > \theta,$$

we could say that subject $i$ is present in session $p$, where parameter $\theta$ is a threshold given in the algorithm. In our experiments, $\theta = 0.5$ achieves good results.

3.6. Attendance Algorithm

The attendance algorithm is based on the method described in Section 3.5. The key-idea of the approach is not to recognize the identity of every detected face in the images of a session, but to recognize if an specific subject is present in the session, that is to find the most similar detected face in the session given an ID. Thus, given the descriptors of the enrolled face images of subject ID, we have to look for the most similar descriptor of the detected faces in the session. The algorithm that computes the attendance percentage of subject ID is presented in Algorithm 1. For example, the attendance percentage of subject #02 in sessions 8, 9, 16, can be estimated by setting in the input of Algorithm 1:

ID = 02 and $p = [8 \ 9 \ \cdots \ 16]$ (see for example the result for the same subject for $p = [1 \ \cdots \ 25]$ in Fig. 1). If we want to compute the attendance sheet in all sessions for the whole class, we have to set $p = [1 \ \cdots \ M]$, and repeat Algorithm 1 for every enrolled student, that is for $i = 1 \ \cdots \ n$.

**input**: ID and sessions $p$

**output**: Attendance percentage of ID

begin

\[
\begin{align*}
\text{i} & \leftarrow \text{ID} \\
\text{Compute } X_i \text{ using (3)} \\
\text{s} & \leftarrow 0 \% \text{ number of sessions} \\
\text{a} & \leftarrow 0 \% \text{ number of attended sessions} \\
\text{for } p \text{ in } p \text{ do} \\
& \text{Compute } Y_p \text{ using (4)} \\
& Z_{ip} \leftarrow X_i^T Y_p \\
& \text{if } \max (Z_{ip}) > \theta \text{ then} \\
& \quad a \leftarrow a + 1 \\
\text{end} \\
\text{AttendancePercentage} \leftarrow a/s \times 100 \\
\end{align*}
\]

end

Algorithm 1: Algorithm that computes the attendance percentage of subject ID in the sessions given in array $p$.

4. Experimental Results

In order to present the experiments used in our work, in this Section, we give further details of the dataset, the experimental protocol, the obtained results with analysis and the implementation.

4.1. Dataset

The dataset consists of 3 sets of images, the database of detected faces, the set of descriptors for the enrolled and query faces and the attendance sheet:

- Enrollment Images Database: The Enrollment Database contains the face images of 67 subjects and their IDs.
- Session Images: For each session there are several color images $$(4.032 \times 3.024 \text{ pixels})$$ of the classroom. The number of images per session are given in Table 2. Totally, there are 153 session images in 25 sessions.
- Query Images Database: For each detected face, a crop is saved for the purpose of descriptor extraction. There is a total of 4,898 detected face images.
- Query Database: The Query Database has one entry per detected face in the session images. The descriptor vectors are stored separately. See more details in Section 3.4.
- Descriptor Database: for each face the Enrollment and Query Databases there is a descriptor vector. To test each one of the ten different descriptors this database is loaded accordingly. For further details see Section 4.3.
- Attendance sheet: For this end, we define a binary array $A$ of size $n \times M$, where $n = 67$ is the number of enrolled subjects and $M = 25$ is the number of sessions. Thus, $A(i, p)$ is 1 or 0 if subject $i$ was present or absent in session $p$. 
4.2. Experimental Protocol

In order to understand Algorithm 2, we start with the explanation of the accuracy of a session: In each session, there are positive and negative ‘samples’, that means subjects that are present and subjects that are absent. Thus, the accuracy in a session will be defined as:

$$\eta = \frac{TP + TN}{n}$$  \hspace{1cm} (8)

where $TP$ (true positives) is the number of present subjects correctly detected, $TN$ (true negatives) is the number of absent subjects correctly detected, and $n$ is the number of ‘samples’, that means number of enrolled subjects (in our case $n = 67$). Ideally, $TP + TN = n$, i.e. $\eta = 1$.

In our protocol of Algorithm 2, we define the average accuracy $\bar{\eta}(s)$, that is the average of the accuracies $\eta$ according to (8) computed for sessions $s + 1 \cdot \cdot \cdot M$ when in the gallery we have the original enrolled face with the faces of the first $s$ sessions. For example, $\bar{\eta}(0)$ means the average accuracy of all sessions $(1 \cdot \cdot \cdot M)$ when in the gallery we have only the original face image used in the enrollment. In addition, $\bar{\eta}(3)$ means the average accuracy of sessions $4 \cdot \cdot \cdot M$, when in the gallery we have the original face image used in the enrollment and the face images of the first three sessions. The idea is to establish, how many face images are necessary in the gallery to achieve robust results.

4.3. Experiments

We execute Algorithm 2 for ten handcrafted and learned descriptors as follows:

- **Handcrafted descriptors**: i) **LBP**: descriptor based on Local Binary Patterns [1]. As pre-processing, the images were resized to $224 \times 224$ pixels and converted to grayscale. Afterwards, the grayscale images were divided into 4x4 partitions. Thus, we extracted 16 LBP features of 59 elements each yielding a descriptor of 944 elements. ii) **HOG**: descriptor based on Histogram of Gradients [12]. We followed the pre-processing mentioned in LBP. Afterwards, HOG features were extracted from grayscale images using a cell size of $18 \times 18$ pixels. Thus, the HOG-descriptor has 4,356 elements. iii) **Gabor**: descriptor based on Gabor Transform [21]. We followed the pre-processing mentioned in LBP. Afterwards, Gabor features were extracted from grayscale images using a factor of 16 for downsampling along rows and columns, with 5 scales and 8 orientations. Thus, the Gabor-descriptor has 7,840 elements.

- **Learned descriptors**: i) **VGG-Face**: a deep learning model with a descriptor of 4,096 elements [43]. ii) **Dlib**: a deep learning model based on the ResNet architecture with a descriptor of 128 elements [32]. iii) **FaceNet**: a deep learning model with a descriptor of 128 elements [50]. iv) **OpenFace**: a deep learning model with a descriptor of 128 elements [2]. v) **SqueezeNet**: a deep learning model with a low number of layers with a 2,048-element descriptor [19]. vi) **GoogleNet-F**: a known model (GoogleNet) trained for faces with a 2,048-element descriptor [19]. vii) **AlexNet-F**: a known model (AlexNet) trained for faces with a 4,096-element descriptor [19].

For each descriptor, we choose the best parameter $\theta$ by maximizing the accuracy using exhaustive search. The idea is to report the maximal accuracy achieved by each method. In our experiments, we decided to focus on face recognition and not in face detection, because manual and automated face detection (using for example Dlib [32]) achieved very similar results in our session images. The most likely reason of this is the collaborative nature of the capture process in which all faces to be recognized were frontal with neutral face expressions.
Figure 3: Average accuracy according to Algorithm 2.

Table 3: Average accuracy for $s = 0, 3, 5, 7$

<table>
<thead>
<tr>
<th>Descriptor</th>
<th>$\theta$</th>
<th>$\bar{\eta}(0)$</th>
<th>$\bar{\eta}(3)$</th>
<th>$\bar{\eta}(5)$</th>
<th>$\bar{\eta}(7)$</th>
</tr>
</thead>
<tbody>
<tr>
<td>FaceNet</td>
<td>0.55</td>
<td>0.9445</td>
<td>0.9674</td>
<td>0.9709</td>
<td>0.9751</td>
</tr>
<tr>
<td>Dlib</td>
<td>0.50</td>
<td>0.9116</td>
<td>0.9518</td>
<td>0.9575</td>
<td>0.9602</td>
</tr>
<tr>
<td>VGG-Face</td>
<td>0.50</td>
<td>0.6657</td>
<td>0.9450</td>
<td>0.9493</td>
<td>0.9610</td>
</tr>
<tr>
<td>GoogleNet-F</td>
<td>0.35</td>
<td>0.5009</td>
<td>0.7972</td>
<td>0.8187</td>
<td>0.8449</td>
</tr>
<tr>
<td>AlexNet-F</td>
<td>0.50</td>
<td>0.4537</td>
<td>0.8005</td>
<td>0.8299</td>
<td>0.8400</td>
</tr>
<tr>
<td>HOG</td>
<td>0.50</td>
<td>0.4251</td>
<td>0.7992</td>
<td>0.8134</td>
<td>0.8234</td>
</tr>
<tr>
<td>LBP</td>
<td>0.51</td>
<td>0.4269</td>
<td>0.7354</td>
<td>0.7463</td>
<td>0.7703</td>
</tr>
<tr>
<td>SqueezeNet</td>
<td>0.50</td>
<td>0.5875</td>
<td>0.7341</td>
<td>0.7373</td>
<td>0.7413</td>
</tr>
<tr>
<td>OpenFace</td>
<td>0.65</td>
<td>0.6358</td>
<td>0.7341</td>
<td>0.7440</td>
<td>0.7570</td>
</tr>
<tr>
<td>Gabor</td>
<td>0.24</td>
<td>0.5594</td>
<td>0.7137</td>
<td>0.7381</td>
<td>0.7604</td>
</tr>
</tbody>
</table>

4.4. Results and Implementation

Results are summarized in Fig. 3. On the one hand, we observe that the best deep learning method in this experiment was FaceNet achieving 97% using as gallery the enrollment face image and the (labeled) face images of the first three sessions (see Fig. 1 for a perfect attendee record of ID #02 in all 25 sessions using our method). On the other hand, all handcrafted methods achieve low accuracies, however, the best one was HOG (more than 80% after the first three sessions). In order to compare numerically all methods, a summary is presented in Table 3. Here, the average accuracy after using in the gallery the enrollment faces and the face images of the first 0, 3, 5 and 7 sessions is shown.

We implemented our methods in Matlab (VGG-face and handcrafted features) and in Python (the rest of features and Algorithm 2)\(^4\).

5. Conclusions

In this paper, we propose an automated student attendance system based on deep learning that can be used in crowded classrooms, where the session images are taken by a smartphone camera. To the best knowledge of the authors, this is the first work that presents a realistic solution in a crowded classroom (around 70 attendees) in so many sessions (25 sessions with images taken during 15 weeks). Ten well known face recognition algorithms based on learned and handcrafted features were evaluated using a protocol that takes into account the number of face images per subject used in the gallery. In our experiments, the best one has been FaceNet, a method based on deep learning features, achieving around 95% of accuracy using only one enrollment face image per subject. Both full annotated databases and codes are available on our webpage. We believe that our automated student attendance system based on face recognition, can be used to save time for both teacher and students, and to prevent fake attendance. As future work, we will implement more sophisticated algorithms based on reduction of dimensionality checking spatial consistency of the attendees.

Acknowledgments

This work was supported by Fondecyt Grant No. 1161314 from CONICYT, Chile.

References


\(^4\)The code is available on [http://dmery.ing.puc.cl/index.php/material/](http://dmery.ing.puc.cl/index.php/material/) (available after publication).


