Pattern Recognition 61 (2017) 610–628

Contents lists available at ScienceDirect

Pattern Recognition

journal homepage: www.elsevier.com/locate/pr

Facial expression recognition with Convolutional Neural Networks: Coping with few data and the training sample order

André Teixeira Lopes [a], Edilson de Aguiar [b], Alberto F. De Souza [a], Thiago Oliveira-Santos [a,*]

[a] Department of Informatics, Universidade Federal do Espírito Santo (Campus Vitória), 514 Fernando Ferrari Avenue, 29075910 Goiabeiras, Vitória, Espírito Santo, Brazil

[b] Department of Computing and Electronics, Universidade Federal do Espírito Santo (Campus São Mateus), BR 101 North highway, km 60, 29932540 Bairro Litorâneo, São Mateus, Espírito Santo, Brazil

ARTICLE INFO

Article history:
Received 30 January 2016
Received in revised form
15 July 2016
Accepted 16 July 2016
Available online 19 July 2016

Keywords:
Facial expression recognition
Convolutional Neural Networks
Computer vision
Machine learning
Expression specific features

ABSTRACT

Facial expression recognition has been an active research area in the past 10 years, with growing application areas including avatar animation, neuromarketing and sociable robots. The recognition of facial expressions is not an easy problem for machine learning methods, since people can vary significantly in the way they show their expressions. Even images of the same person in the same facial expression can vary in brightness, background and pose, and these variations are emphasized if considering different subjects (because of variations in shape, ethnicity among others). Although facial expression recognition is very studied in the literature, few works perform fair evaluation avoiding mixing subjects while training and testing the proposed algorithms. Hence, facial expression recognition is still a challenging problem in computer vision. In this work, we propose a simple solution for facial expression recognition that uses a combination of Convolutional Neural Network and specific image pre-processing steps. Convolutional Neural Networks achieve better accuracy with big data. However, there are no publicly available datasets with sufficient data for facial expression recognition with deep architectures. Therefore, to tackle the problem, we apply some pre-processing techniques to extract only expression specific features from a face image and explore the presentation order of the samples during training. The experiments employed to evaluate our technique were carried out using three largely used public databases (CK+, JAFFE and BU-3DFE). A study of the impact of each image pre-processing operation in the accuracy rate is presented. The proposed method: achieves competitive results when compared with other facial expression recognition methods – 96.76% of accuracy in the CK+ database – it is fast to train, and it allows for real time facial expression recognition with standard computers.

1. Introduction

Facial expression is one of the most important features of human emotion recognition [1]. It was introduced as a research field by Darwin in his book "The Expression of the Emotions in Man and Animals' [2]. According to Li and Jain [3], it can be defined as the facial changes in response to a person's internal emotional state, intentions, or social communication. Nowadays, automated facial expression recognition has a large variety of applications, such as data-driven animation, neuromarketing, interactive games, sociable robotics and many other human–computer interaction systems.

Expression recognition is a task that humans perform daily and effortlessly [3], but it is not yet easily performed by computers, despite recent methods have presented with accuracies larger than 95% in some conditions (frontal face, controlled environments, and high-resolution images). Many works in the literature do not perform a consistent evaluation methodology (e.g. without subject overlap in training and testing) and therefore present a misleading high-accuracy, but do not represent most of the face expression recognition problems real scenarios. On the other hand, low accuracy has been reported on databases with uncontrolled environments and in cross-database evaluations. Trying to cope with these limitations, several research works have tried to make computers reach the same accuracy of humans, and some examples of these works are highlighted below. This problem is still a challenge for computers because it is very hard to separate the expressions' feature space, i.e. facial features from one subject in two different expressions may be very close in the feature space, while facial features from two subjects with the same expression

* Correspondence to: Departamento de Informática, Centro Tecnológico, Universidade Federal do Espírito Santo, Campus Universitário de Goiabeiras, Prédio: CT-VII, Sala 28, Av. Fernando Ferrari, 541, 29075910 Vitória – ES, Brazil.

E-mail addresses: andreteixeiralopes@gmail.com (A.T. Lopes), edilson.de.aguiar@gmail.com (E. de Aguiar), alberto@lcad.inf.ufes.br (A.F. De Souza), todsantos@inf.ufes.br (T. Oliveira-Santos).

http://dx.doi.org/10.1016/j.patcog.2016.07.026
0031-3203/

Fig. 1. Three different subjects with the happy expression. As it can be seen, the images vary a lot from each other not only in the way that the subjects show their expression but also in light, brightness, position and background. The images are from the following databases: CK+ database [4], JAFFE database [5] and BU-3DFE database [6], in this order.

may be very far from each other. In addition, some expressions like "sad" and "fear", for example, are, in some cases, very similar.

Fig. 1 shows three subjects with a happy expression. As it can be seen in the figure, the images vary a lot from each other not only in the way that the subjects show their expression, but also in lighting, brightness, pose and background. This figure also exemplifies another challenge related to the facial expression recognition that is the uncontrolled training–testing scenarios (training images can be very different in terms of environmental conditions and subject ethnicity from the testing images). One approach to evaluate the facial expression recognition under these scenarios is to train the method with one database and to test it with another (possibly from different ethnic groups). We present results following this approach.

Facial expression recognition systems can be divided into two main categories: those that work with static images [7–13] and those that work with dynamic image sequences [14–17]. Static-based methods do not use temporal information, i.e. the feature vector comprises information about the current input image only. Sequence based methods, in the other hand, use temporal information of images to recognize the expression captured from one or more frames. Automated systems for facial expression recognition receive the expected input (static image or image sequence) and typically give as output one of six basic expressions (anger, sad, surprise, happy, disgust and fear, for example); some systems also recognize the neutral expression. This work will focus on methods based on static images and it will consider the six and seven expressions sets (six basic plus neutral), for controlled and uncontrolled scenarios.

As described by Li and Jain [3], automatic facial expression analysis comprises three steps: face acquisition, facial data extraction and representation, and facial expression recognition. Face acquisition can be split in two major steps: face detection [18–21] and head pose estimation [22–24]. After the face acquisition, the facial changes caused by facial expressions need to be extracted. These changes are usually extracted using geometric feature-based methods [25–27,21] or appearance-based methods [25,8–11,28,13]. The extracted features are often represented in vectors, referred as feature vectors. Geometric feature-based methods work with shape and location of facial components like mouth, eyes, nose and eyebrows. The feature vector that represents the face geometry is composed of facial components or facial feature points. Appearance-based methods work with feature vectors extracted from the whole face, or from specific regions; these feature vectors are acquired using image filters applied to the whole face image [3].

Once feature vectors related to the facial expression are available, expression recognition can be performed. According to Liu et al. [7], expression recognition systems basically use a three-stage training procedure: feature learning, feature selection and classifier construction, in this order. The feature learning stage is responsible for the extraction of all features related to the facial expression. The feature selection selects the best features to represent the facial expression. They should minimize the intra-class variation of expressions while maximizing the inter-class variation [8]. Minimizing the intra-class variation of expressions is a problem because images of different individuals with the same expression are far from each other in the pixel's space. Maximizing the inter-class variation is also difficult because images of the same person in different expressions may be very close to one another in the pixel's space [29]. At the end of the whole process, a classifier (or a set of classifiers, with one for each expression) is used to infer the facial expression, given the selected features.

One of the techniques that has been successfully applied to the facial expression recognition problem was the deep multi-layer neural network [30–32,14,10,11,13]. This technique comprises the three steps of facial expression recognition (learning and selection of features and classification) in one single step. In the last decade, neural network researches were motivated to find a way to train deep multi-layer neural networks (i.e. networks with more than one or two hidden layers) in order to increase their accuracy [33,34]. According to Bengio [35], until 2006, many new attempts have shown little success. Although somewhat old, the Convolutional Neural Networks (CNNs) proposed in 1998 by Lecun et al. [36] has shown to be very effective in learning features with a high level of abstraction when using deeper architectures (i.e. with a lot of layers) and new training techniques. In general, this type of hierarchical network has alternating types of layers, including convolutional layers, sub-sampling layers and fully connected layers. Convolutional layers are characterized by the kernel's size and the number of generated maps. The kernel is shifted over the valid region of the input image generating one map. Sub-sampling layers are used to increase the position invariance of the kernels by reducing the map size [37]. The main types of sub-sampling layers are maximum-pooling and average pooling [37]. Fully connected layers on CNN's are similar to the ones in general neural networks, its neurons are fully connected with the previous layer (generally: convolution layer, sub-sampling layer or even a fully connected layer). The learning procedure of CNNs consists of finding the best synapses' weights. Supervised learning can be performed using a gradient descent method, like the one proposed

by Lecun et al. [36]. One of the main advantages of CNN, is that the models' input is a raw image rather than a set of hand-coded features.

Besides the methods using deep architecture, there are many others in the literature, but some aspects of the evaluation of these methods still deserve attention. For example, validation methods could be improved in [38–42,30,10] in order to consider situations where the subject in the test set is not in the training set (i.e. test without subject overlap), accuracy is somewhat low in [38,1,43,44], and the recognition time in [7,43,16] could be improved so as to perform real time evaluations.

Trying to cope with some of these limitations while keeping a simple solution, in this paper, we present a deep learning approach combining standard methods, like image normalizations, synthetic training-samples generation (for example, real images with artificial rotations, translation and scaling) and Convolutional Neural Network, into a simple solution that is able to achieve a very high accuracy rate of 96.76% in the CK+ database for 6 expressions, which is the state-of -the-art. The training time is significantly smaller if compared with other methods in the literature and the whole facial expression recognition system can operate in real time in standard computers. We have examined the performance of our system using the Extensive Cohn–Kanade (CK+) database [4], the Japanese Female Facial Expression (JAFFE) database [5] and the Binghamton University 3D Facial Expression (BU-3DFE) database [6], achieving a better accuracy in the CK+ database, which contains more samples (important for deep learning techniques) than the JAFFE and BU-3DFE databases. In addition, we have performed an extensive validation, with cross-database tests (i.e. training the method using one database and evaluating its accuracy using another one). In summary, the main contributions of this work are:

i. an efficient method for facial expression recognition that operates in real time;
ii. a study of the effects of image pre-processing operations in the facial expression recognition problem;
iii. a set of pre-processing operations for face normalization (spatial and intensity) in order to decrease the need of controlled environments and to cope with the lack of data;
iv. a study to handle the variability in the accuracy caused by the presentation order of the samples during training; and
v. a study of the performance of the proposed system with different cultures and environments (cross-database evaluation).

This work extends the one presented in the 28th SIBGRAPI (Conference on Graphics, Patterns and Images) [45] as follows:

i. it presents a deeper literature review which were used to enlarge the comparisons of the results;
ii. it presents the results using a new implementation based on a different framework (from ConvNet, a Matlab based implementation [46], to Caffe, a C++ based implementation [47]), which consequently reduced the total training time by almost a factor of four;
iii. it presents results showing a reduced recognition time, which is now real time;
iv. it presents results showing better accuracy due to longer training and small changes (see below) in the method;
v. it includes improved experimental methodology, as for example, using training, validation and test sets, instead of training and test sets only; and
vi. it presents a more complete evaluation including cross-database tests.

The changes in the method that allowed better accuracy were as follows:

a. The synthetic samples have a slight different generation process, allowing larger variation among them (now synthetic samples can be the original image rotated, scaled or translated, instead of only rotated);
b. We increased the number of synthetic samples, from 30 to 70 (motivated by the previous item); and
c. The logistic regression loss function was replaced by a SoftmaxWithLoss function (described in Section 3).

The remainder of this paper is organized as follows: the next section presents the most recent related work, while Section 3 describes the proposed approach. In Section 4, the experiments we have performed to evaluate our system are presented and compared with several recent facial expression recognition methods. Finally, we conclude in Section 5.

2. Related work

Several facial expression recognition approaches were developed in the last decades with an increasing progress in recognition performance. An important part of this recent progress was achieved thanks to the emergence of deep learning methods [7,10,12] and more specifically with Convolutional Neural Networks [14,11], which is one of the deep learning approaches. These approaches became feasible due to: the larger amount of data available nowadays to train learning methods and the advances in GPU technology. The former is crucial for training networks with deep architectures, whereas the latter is crucial for the low cost high-performance numerical computations required for the training procedure. Surveys of the facial expression recognition research can be found in [3,48].

Some recent approaches for facial expression recognition have focused on uncontrolled environments (e.g. not frontal face, images partially overlapped, spontaneous expressions and others), which is still a challenging problem [12,49,50]. This work will focus on more controlled environments and evaluation among different ethnic groups, the latter is a more challenging scenario in facial expression recognition. This section discusses recent methods that achieve high accuracy in facial expression recognition using a comparable experimental methodology (as explained in Section 4) or methods that are based on deep neural networks.

Liu et al. [7] proposed a novel approach called boosted deep belief network (BDBN). The BDBN is composed by a set of classifiers, named by the authors as weak classifiers. Each weak classifier is responsible for classifying one expression. Their approach performs the three learning stages (feature learning, feature selection and classifier construction) iteratively in a unique framework. Their experiments were conducted using two public databases of static images, Cohn–Kanade [4] and JAFFE [51], and achieved an accuracy of 96.7% and 91.8%, respectively. They also performed experiments on less controlled scenarios, using a cross-database configuration (training with the CK+ and testing in JAFFE) and achieved an accuracy of 68.0%. All images were firstly preprocessed based on the given eye coordinates, i.e. performing alignment and crop. The training and the test adopted a one-versus-all classification strategy, i.e. they used a binary classifier for each expression. The time required to train the network was about 8 days. The recognition was calculated as a function of the weak classifiers. In their method, they use six or seven classifiers, depending on the amount of expressions to be recognized (one for each expression). Each classifier took 30 ms to recognize each expression, with a total recognition time of about 0.21 s. The recognition time was reported by the authors using a 6-core 2.4 GHz PC.

Song et al. [10], developed a facial expression recognition system that uses a deep Convolutional Neural Network and runs on a smartphone. The proposed network is composed of five layers and

65,000 neurons. According to the authors, it is common to have an overfitting when using a small amount of training data and such a big network. Therefore, the authors applied data augmentation techniques to increase the amount of training data and used the drop-out [52] during the network training. The experiments were performed using the CK+ [4] dataset, and other three datasets created by the authors. The images of the CK+ dataset were firstly cropped to focus on regions that contains facial changes caused by an expression. The experiments performed by the authors follow a 10-fold cross validation, but they do not mention if there were images of the same subject in more than one fold. Therefore, we assumed that there was an overlap among subjects in the training and test sets. An accuracy of 99.2% was achieved in the CK+ database while recognizing only five expressions (anger, happy, sad, surprise and neutral).

Burkert et al. [11] also proposed a method based on Convolutional Neural Networks. The authors claim that their method is independent of any hand-crafted feature extraction (i.e. uses the raw image as input). Their network architecture consists of four parts. The first part is responsible for the automatic data preprocessing, while the remaining parts carried out the feature extraction process. The extracted features are classified into a given expression by a fully connected layer at the end of the network. The proposed architecture comprises 15 layers (7 convolutions, 5 poolings, 2 concatenations and 1 normalization layer). They evaluated their method with the CK+ database and the MMI database, achieving an accuracy of 99.6% and 98.63%, respectively. Despite the high accuracy, in their experimental methodology they did not guarantee that subjects used in training were not used in test. As will be discussed in Section 4, this is an important restriction that should be enforced in order to perform a fair evaluation of facial expression recognition methods [44,53].

Liu et al. [12] proposed an action unit (AU) inspired deep networks (AUDN) in order to explore a psychological theory that expressions can be decomposed into multiple facial expression action units. The authors claim that the method is able to learn: (i) informative local appearance variation; (ii) an optimal way to combine local variations; (iii) and a high-level representation for the final expression recognition. Experiments were performed in the CK+ [4], MMI [54] and SFEW [55] datasets. The latter contains images captured from various movies under uncontrolled scenarios, representing the real-world environment. Experiments were performed using a cross-validation approach without subject overlap between training and test groups, and evaluating the six basic expressions. The method achieves an accuracy of 93.70% in the CK+ database, 75.85% in the MMI database and 30.14% in SFEW database.

Ali et al. [13] proposed a collection of boosted neural network ensembles for multiethnic facial expression recognition. The proposed model is composed by three main steps: firstly a set of binary neural networks are trained, secondly the predictions of these neural networks are combined to compose the ensemble's collection and finally these collections are used to detect the presence of an expression. The multicultural facial expression database was created by the authors with images from three different databases, which contains images from Japanese (JAFFE), Taiwanese (TFEID), Caucasians (RaFD), and Moroccans subjects. The authors reported the result of recognizing five expressions (anger, happy, sad, surprise and fear) in two different experimental approaches. In the first, they trained and evaluated the system in the multicultural database achieving an accuracy of 93.75%. The second experiment was performed to evaluate the proposed method in a less controlled environment. The method was trained with two databases (TFEID and RaFD) and evaluated in the JAFFE database, achieving an accuracy of 48.67%.

Shan et al. [8] performed a study using local binary patterns (LBP) as feature extractor. They combined and compared different machine learning techniques like template matching, support vector machine (SVM), linear discriminant analysis and linear programming to recognize facial expressions. The authors also conducted a study to analyze the impact of image resolution in the accuracy result and concluded that methods based on geometric features do not handle low-resolution images very well, whereas those based on appearance, like Gabor wavelets and LBP, are not so sensitive to the image resolution. The best result achieved in their work was an accuracy of 95.1% using SVM and LBP in the CK+ database. Using a cross-database validation (training with the CK+ and testing with JAFFE) to evaluate the proposed system in a less controlled scenario, the authors achieved an accuracy of 41.3%. The images were firstly cropped using the eye positions. The experimental setup used was a 10-fold cross validation scheme without subject overlap. The training and the recognition times were not mentioned by the authors.

A video-based facial expression recognition system was proposed by Byeon and Kwak [14]. They developed a 3D-CNN having an image sequence (from neutral to final expression) using 5 successive frames as 3D input. Therefore, the CNN input is $H*W*5$ (where H and W are the image height and width, respectively, and 5 is the number of frames). The authors claim that the 3D CNN method can handle some degrees of shift and deformation invariance. With this approach, they achieved an accuracy of 95%, but the method relies on a sequence containing the full movement from the neutral to the expression. The experiments were carried out with 10 persons only on a non-usual dataset. The training and recognition times were not mentioned by the authors.

Another video-based approach, proposed by Fan and Tjahjadi [16], used a spatial–temporal framework based on histogram of gradients and optical flow. Their method comprises three phases: pre-processing, feature extraction and classification. In the pre-processing phase, the detection of facial landmarks was performed and a face alignment was carried out (in order to reduce variations in the head pose). In the feature extraction phase, a framework that integrates dynamic information extracted from the variation in the facial shape caused by the expressions was employed. In the last phase, the classification, a SVM classifier with a RBF kernel was used. The experiments were carried out using the CK+ and the MMI databases. The accuracy achieved by the authors in the CK+ database for seven expressions was 83.7% and in the MMI database was 74.3%. The training time was not mentioned, while the recognition time was about 350 ms per image in the CK+ database and 520 ms in the MMI database.

In comparison with the methods above, this work: presents a higher accuracy in the CK+ and JAFFE databases (including the cross-database validation) and a smaller training and evaluation time than Liu et al. [7,12], Shan et al. [8] and Fan and Tjahjadi [16]; a more robust evaluation methodology (without subject overlap between training and test) than Song et al. [10] and Bukert et al. [11]; recognition of six and seven expression, instead of only five or six as done by Song et al. [10], Bukert et al. [11] and Ali et al. [13]; validates the proposed method on three largely used databases, to allow for a fair comparison with other methods in the literature, instead of using unusual non-public databases like Byeon and Kwak [14]. Many of the works mentioned here present a very high accuracy that cannot be fairly compared with our method because they allow for subject overlap in the training and test sets. Preliminary experiments performed with our method considering such overlapping scenarios also showed accuracies closer to 100% without much effort.

3. Facial expression recognition system

Our system for facial expression recognition performs the three learning stages in just one classifier (CNN). The proposed system

operates in two main phases: training and test. During training, the system receives a training data comprising grayscale images of faces with their respective expression id and eye center locations and learns a set of weights for the network. To ensure that the training performance is not affected by the order of presentation of the examples, a few images are separated as validation and are used to choose the final best set of weights out of a set of trainings performed with samples presented in different orders. During test, the system receives a grayscale image of a face along with its respective eye center locations, and outputs the predicted expression by using the final network weights learned during training.

An overview of the system is illustrated in Fig. 2. In the training phase, new images are synthetically generated to increase the database size. After that, a rotation correction is carried out to align the eyes with the horizontal axis. Subsequently, the image is cropped to remove background information and to keep only expression specific features. A down-sampling procedure is carried out to get the features in different images in the same location. Thereafter, the image intensity is normalized. The normalized images are used to train the Convolutional Neural Network. The output of the training phase is the set of weights of the round that achieved the best result with the validation data after a few training rounds considering data in different orders. The testing phase use the same methodology as the training phase: spatial normalization, cropping, down-sampling and intensity normalization. Its output is a single number – the id – of one of the six basic expressions. The expressions are represented as integer numbers (0 – angry, 1 – disgust, 2 – fear, 3 – happy, 4 – sad and 5 – surprise).

3.1. Synthetic sample generation

Unfortunately, the spatial normalization employed is not enough to ensure that all faces will have the eyes correctly aligned due to imperfections in the eye detection procedure. Fortunately, CNN's are very good at learning transformation invariant functions (i.e. they can handle distorted images [56]). However, one of the main problems of deep learning methods is that they need a lot of data in the training phase to perform this task properly [56]. Unfortunately, the amount of data available in public datasets is not enough to achieve such behavior in our application.

To address this problem Simard et al. [56] proposed the generation of synthetic images (i.e. real images with artificial rotations, translations and skewing) to increase the database, this process is referred as data augmentation. The authors show the benefits of applying combinations of translations, rotations and skewing for increasing the database. Following this idea, in this paper, we use a 2D Gaussian distribution ($\sigma = 3$ pixels and $\mu = 0$) to introduce random noise in the locations of the center of the eyes. Synthetic images are generated by considering the normalized versions of noisy eye locations. For each image, 70 additional synthetic images were generated.

As it can be seen in Fig. 3, points are generated for both eyes following a Gaussian distribution centered in their original location. The new eye center position is therefore equivalent to the original one but disturbed by a Gaussian noise. As the new values given to the normalization procedure are not the real eye center, the resulting images will be either disturbed by a translation, a rotation and/or a scale, or not disturbed at all. The specific value of

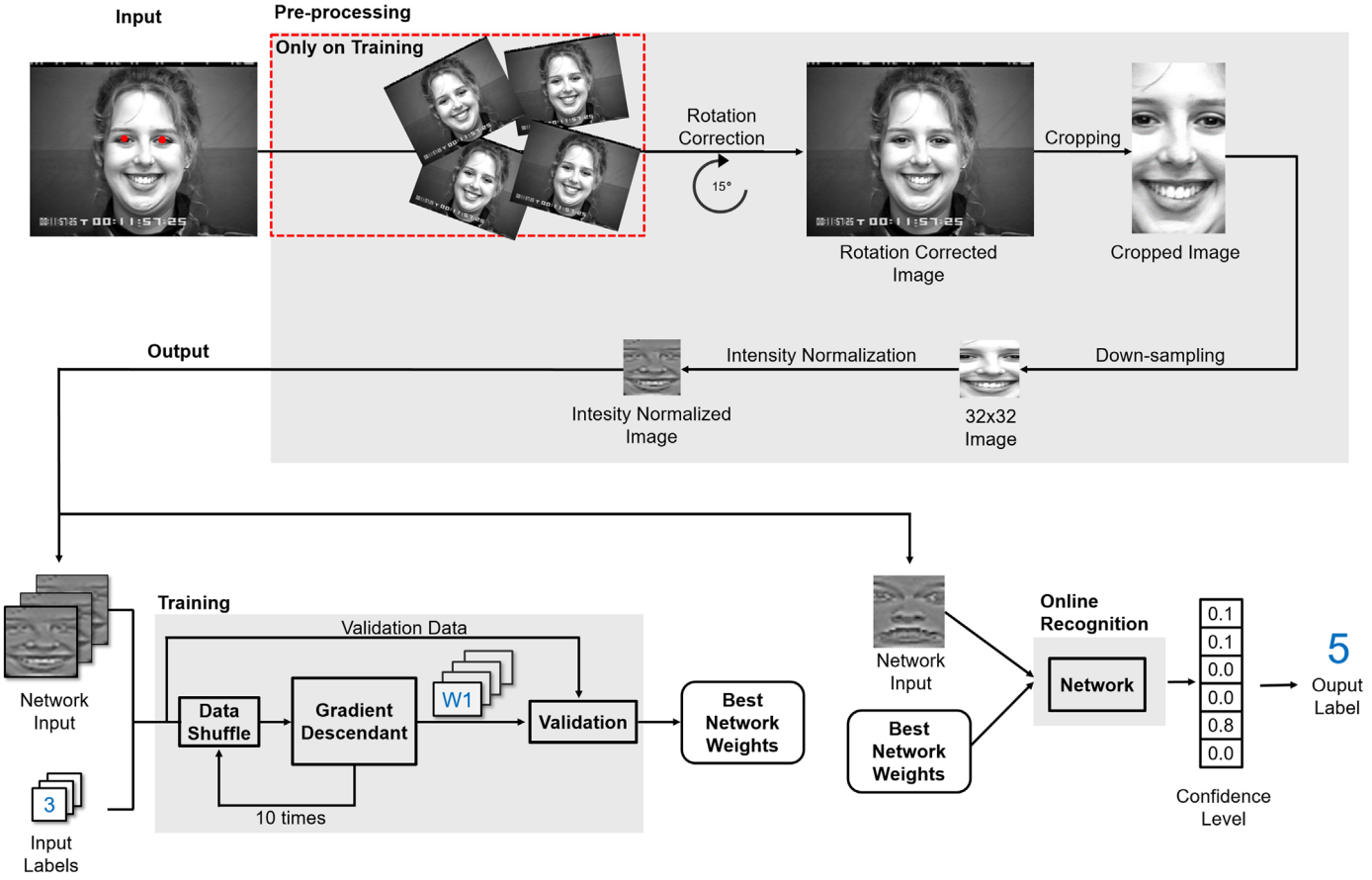


Fig. 2. Overview of the proposed facial expression recognition system. The system operates in two main phases: training and test. The training phase takes as input a set of images with a face, its eyes locations and expression id, and outputs the set of weights of the round that achieved the best result with the validation data after a few training rounds considering the data in different orders. The test phase receives the weight set from the learning step and a face image with its eyes locations, and outputs the expression id of the image.

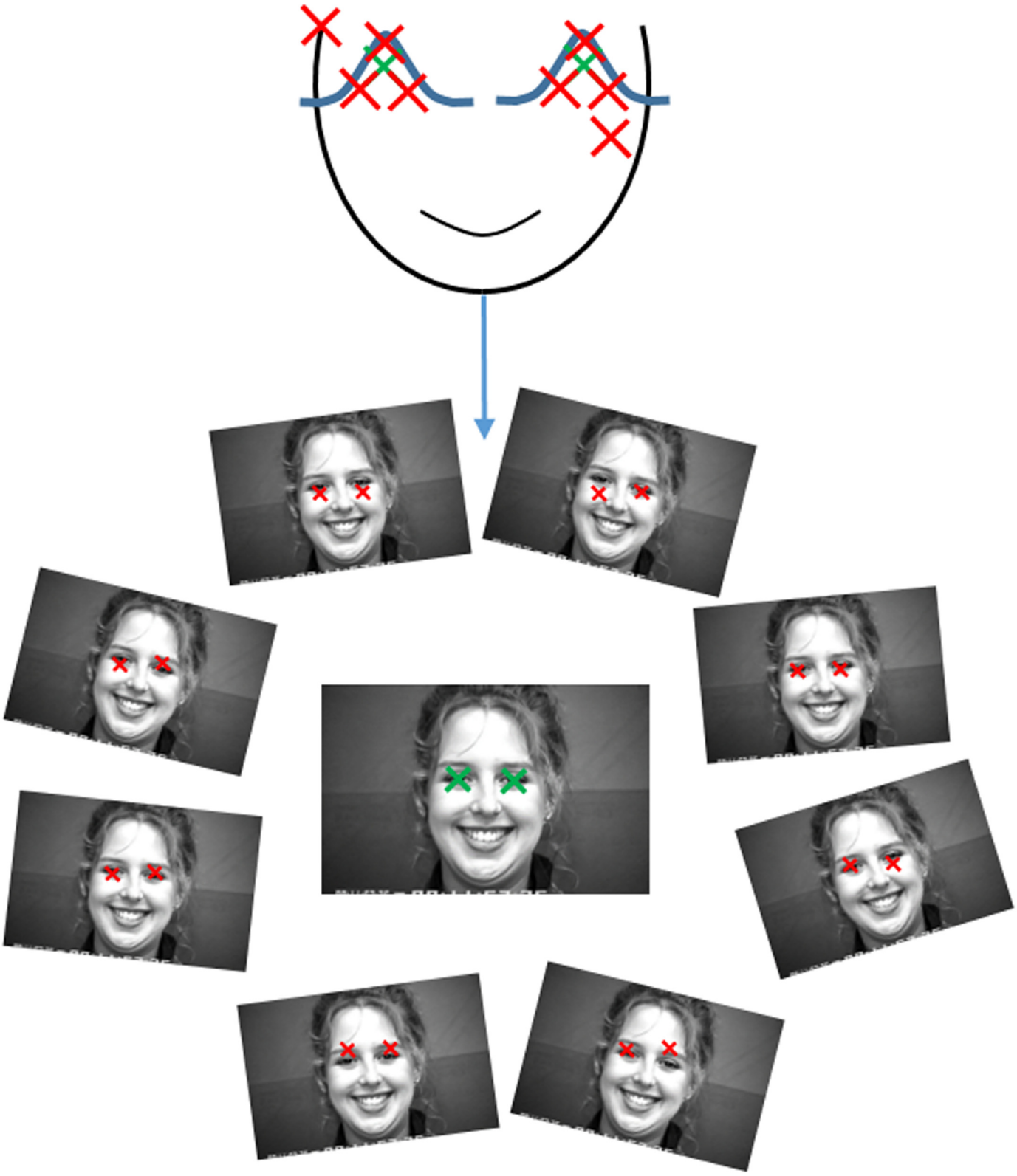


Fig. 3. Illustration of the synthetic sample generation. The Gaussian synthetic sample generation procedure increases the database size and variation, adding a noise (Gaussian with $\sigma = 3$ pixels) in the images considering a controlled environment. The new images are generated using the normalization step and the new eye points. The green crosses are the original eye points, while the red ones are the synthetic points. (For interpretation of the references to color in this figure caption, the reader is referred to the web version of this paper.)

the Gaussian standard deviation needs to be carefully chosen. A very small deviation could cause no variation in the original data and generate a lot of useless equal images. On the other hand, a big deviation for each eye could introduce too much translation, rotation and/or scale noise in the images making the scenario more complex for the classifier to learn the expression features. A

standard deviation of $\sigma = 3$ pixels was empirically chosen. It is important to note that the synthetic data is only used in the training.

3.2. Rotation correction

The images in the databases, and also in real environments, vary in rotation, brightness and size even for images of the same subject. These variations are not related to the face expression and can affect the accuracy rate of the system. To address this problem, the face region is aligned (with rotation normalization) with the horizon and a center point in order to correct possible geometric issues like rotations and translations. To perform this alignment two information are needed, the facial image and the center of both eyes. There are already a lot of methods in the literature that are able to find the eyes, and the others facial points, with high precision [57–61], and this is not the focus of this work. Cheng et al. [61] developed a CUDA version of DRMF [60], allowing for real-time facial keypoints detection, while keeping an accurate detection of these keypoint even with the face partially occluded (a shape RMSE below 0.05 fraction of inter-ocular distance).

To perform the face alignment, a rotation transformation is applied to align the eyes with the horizontal axis of the image and an affine translation defined by the locations of the eyes to centralize the face in a specific point of the image. The rotation makes the angle formed by the line segment going from one eye center to the other, and the horizontal axis to be zero. Rotations and translations in the images are not related to the facial expression and therefore should be removed to avoid negatively affecting the accuracy rate of the system. The rotation correction procedure is shown in Fig. 4.

The input to this procedure can be an original or a synthetic image. The rotation correction, for the synthetic generated images, may not carry out a perfect align with the horizontal axis because the eye center is the real position disturbed by a random Gaussian noise. Therefore, it will generate images disturbed by rotations and translations, which increases the variation in the training examples.

3.3. Image cropping

As shown in Fig. 2, the original image has a lot of background information that is not important to the expression classification procedure. This information could decrease the accuracy of the classification because the classifier has one more problem to solve, i.e. discriminating between background and foreground. After the cropping, all image parts that do not have expression specific information are removed. The cropping region also tries to remove facial parts that do not contribute for the expression (e.g. ears, part of the forehead, etc.). Therefore, the region of interest is defined based on a ratio of the inter-eyes distance. Consequently, our method is able to handle different persons and image sizes without human intervention. The cropping region is delimited by a vertical factor of 4.5 (considering 1.3 for the region above the eyes and 3.2 for the region below) applied to the distance between the eyes middle point and the right eye center. The horizontal cropping region is delimited by a factor of 2.4 applied to this same distance. These factor values were determined empirically. An example of this procedure is illustrated in Fig. 5.

3.4. Down-sampling

The down-sampling operation is performed to reduce the image size for the network and to ensure scale normalization, i.e. the same location for the face components (eyes, mouth, eyebrow, etc.) in all images. The down-sampling uses a linear interpolation approach. After this re-sampling, one can guarantee that the eye center will be approximately in the same position. This procedure helps the CNN to learn which regions are related to each specific expression. The down-sampling also enables the convolutions to be performed in the GPU since most of the graphics card nowadays have limited memory. The final image is down-sampled, using a linear interpolation, to 32 $\times$32 pixels.

3.5. Intensity normalization

The image brightness and contrast can vary even in images of the same person in the same expression, therefore, increasing the variation in the feature vector. Such variations increase the complexity of the problem that the classifier has to solve for each expression. In order to reduce these issues an intensity normalization was applied. A method adapted from a bio-inspired technique described in [62], called contrastive equalization, was used. Basically, the normalization is a two step procedure: firstly a subtractive local contrast normalization is performed; and secondly, a divisive local contrast normalization is applied. In the first step, the value of every pixel is subtracted from a Gaussian-weighted average of its neighbors. In the second step, every pixel is divided by the standard deviation of its neighborhood. The neighborhood for both procedures uses a kernel of 7 $\times$7 pixels (empirically chosen). An example of this procedure is illustrated in Fig. 6.

Eq. (1) shows how each new pixel value is calculated in the intensity normalization procedure:

$$x' = \frac{x - \mu_{nhgx}}{\sigma_{nhgx}} \tag{1}$$

where x' is the new pixel value, x is the original pixel value, μ_{nhgx} is the Gaussian-weighted average of the neighbors of x, and σ_{nhgx} is the standard deviation of the neighbors of x.

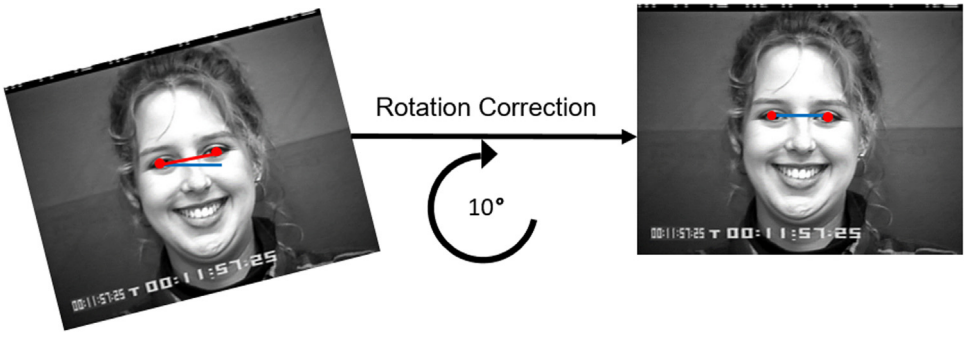


Fig. 4. Rotation correction example. The non-corrected input image (left) is rotated (right) using the line segment going from one eye center to the other (red line) and the horizontal axis (blue line). The 10° is just an example of a possible correction. (For interpretation of the references to color in this figure caption, the reader is referred to the web version of this paper.)

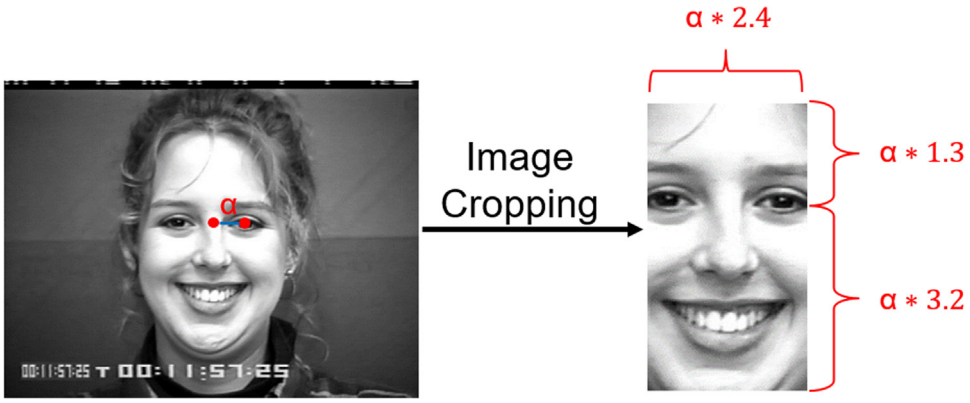


Fig. 5. Image cropping example. The spatial normalization is carried out using half of the inter-eyes distance (α). The input image (left) is cropped (right) using a horizontal factor (α*2.4) to crop in the horizontal and a vertical factor (α*4.5 considering 1.3 for the region above the eyes and 3.2 for the region below) to crop in the vertical. This operation aims to remove all non-expression features, such as background and hair.

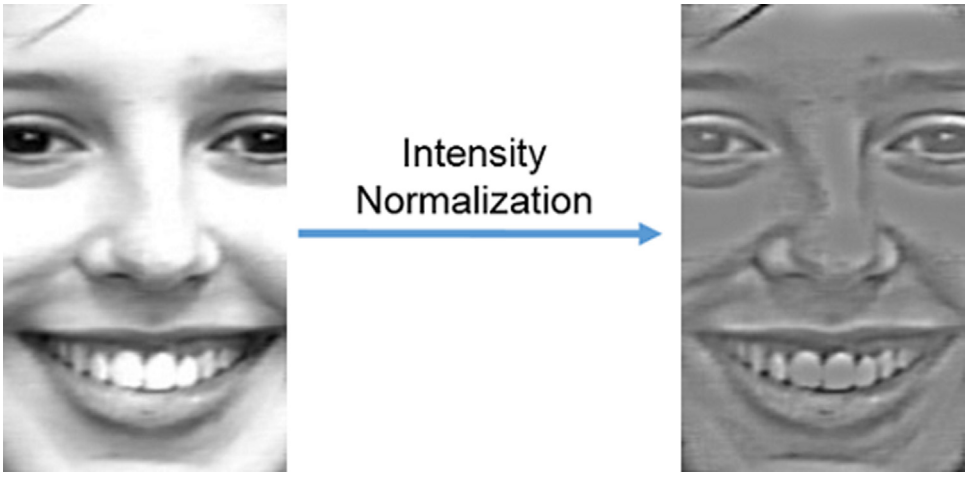


Fig. 6. Illustration of the intensity normalization. The figure shows the image with the original intensity (left) and its intensity normalized version (right).

3.6. *Convolutional Neural Network*

The architecture of our Convolutional Neural Network is represented in Fig. 7. The network receives as input a 32x32 grayscale image and outputs the confidence of each expression. The class with the maximum value is used as the expression in the image. Our CNN architecture comprises 2 convolutional layers, 2 sub-sampling layers and one fully connected layer. The first layer of the CNN is a convolution layer, that applies a convolution kernel of 5 ×5 and outputs 32 images of 28 ×28 pixels. This layer is followed by a sub-sampling layer that uses max-pooling (with kernel size 2 ×2) to reduce the image to half of its size. Subsequently, a new convolution layer performs 64 convolutions with a 7 ×7 kernel to map of the previous layer and is followed by another sub-sampling, again with a 2 ×2 kernel. The outputs are given to a fully connected hidden layer that has 256 neurons. Finally, the network has six or seven output nodes (one for each expression that outputs their confidence level) that are fully connected to the previous layer.

The first layer of the network (a convolution layer) aims to extract elementary visual features, like oriented edges, end-point, corners and shapes in general, like described by Lecun et al. [36]. In the facial expression recognition problem, the features detected are mainly the shapes, corners and edges of eyes, eyebrow and lips. Once the features are detected, its exact location is not so important, just its relative position compared to the other features. For example, the absolute position of the eyebrows is not important, but their distances from the eyes are, because a big

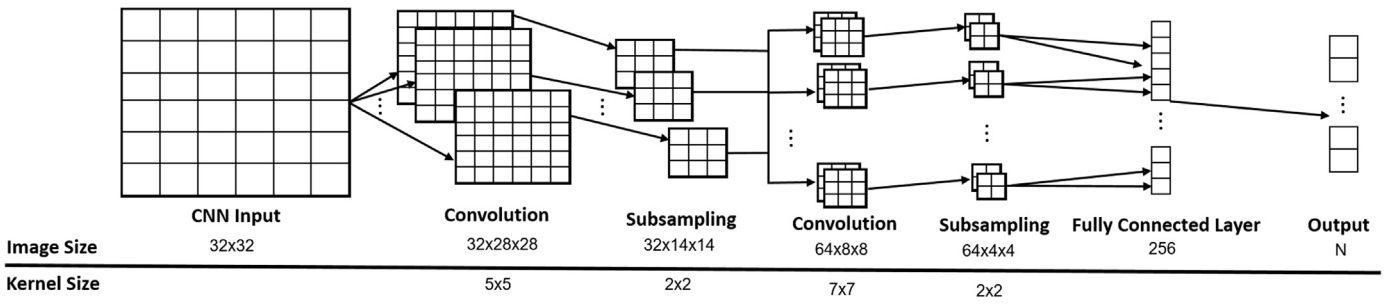


Fig. 7. Architecture of the proposed Convolutional Neutral Network. It comprises of five layers: two convolutional layers, two sub-sampling layers and one fully connected layer.

distance may indicate, for instance, the surprise expression. This precise position is not only irrelevant but it can also pose a problem, because it can naturally vary for different subjects in the same expression. The second layer (a sub-sampling layer) reduces the spatial resolution of the feature map. According to Lecun et al. [36], this operation aims to reduce the precision with which the position of the features extracted by the previous layer are encoded in the new map. The next two layers, one convolutional and one sub-sampling, aim to do the same operations that the first ones, but handling features in a lower level, recognizing contextual elements (face elements) instead of simple shapes, edges and corners. The concatenation of sets of convolution and sub-sampling layers achieve a high degree of invariance to geometric transformation of the input. The last hidden layer (a fully connected layer) receives the set of features learned and outputs the confidence level of the given features in each one of the considered expressions.

This network uses the stochastic gradient descent method to calculate the synaptic weights between the neurons, this method was proposed by Buttou [63]. The initial value of these synapses for the convolutions and for the fully connected layer are generated using the Xavier filler, proposed by Glorot et al. [64], that automatically determines the scale of the initialization based on the number of input and output neurons. The loss is calculated using a logistic function of the soft-max output (known as *SoftMaxWithLoss*). The activation function of the neurons is a ReLu (rectified linear unit), defined as $f(z) = \max(z, 0)$. The ReLu function generally learns much faster in deep architectures [65].

4. Experiments and discussions

The experiments were performed using three publicly available databases in the facial expression recognition research field: The Extended Cohn–Kanade (CK+) database [4], the Japanese Female Facial Expressions (JAFFE) database [5] and the Binghamton University 3D Facial Expression (BU-3DFE) database [6]. Accuracy is computed considering one classifier to classify all learned expressions. In addition, to allow for a fair comparison with some methods in the literature, accuracy is also computed considering one binary classifier for each expression, as used in [7].

The implementation of the pre-processing steps was done inhouse using OpenCV, C++ and a GPU based CNN library (Caffe [47]). All the experiments were carried out using an Intel Core i7 3.4 GHz with a NVIDA GeForce GTX 660 CUDA Capable that has 1.5 Gb of memory in the GPU. The environment of the experiments was Linux Ubuntu 12.04, with the NVIDIA CUDA Framework 6.5 and the cuDNN library installed. The preprocessing step (rotation correction, cropping, down-sampling and intensity normalization) took only 0.02 s and the network recognition (classification step) took in average 0.01 s.

In this section, a study is carried out showing the impact of every normalization step in the accuracy of the method. Firstly, we describe the databases used for the experiments. Secondly, the metrics used to evaluate the system accuracy are explained. Thirdly, the results of the tuning experiments and the influence of each pre-processing step is presented. Fourthly, the results with different databases are shown and discussed in details. Finally, a comparison with several recent facial expression recognition methods that uses the same evaluation methodology is presented and the limitations of our method are discussed.

4.1. Database

The presented system was trained and tested using the CK+ database [4], the JAFFE database [5] and the BU-3DFE database [6]. The CK+ database comprises 100 university students with age between 18 and 30 years old. The subjects in the database are 65% female, 15% are African-American and 3% are Asian or south American. The images were captured from a camera located directly in front of the subject. The students were instructed to perform a series of expressions. Each sequence begins and ends with the neutral expression. All images in the database are 640 by 480 pixel arrays with 8-bit precision for grayscale values. Each image has a descriptor file with its facial points, these points were used to normalize the facial expression image. The facial points in the database are coded using the facial action coding system (FACS) [66]. The database creators used the active appearance models (AAMs) to automatically extract the facial points. The database contains images for the following expressions: neutral, angry, contempt, disgust, fear, happy, sad and surprise. To do a fair comparison with the major part of the recent methods [16,7,44,43,67,8], in our experiments the contempt expression images were not used. Some examples of the CK+ database images are shown in Fig. 8.

To perform a fair evaluation of the proposed method, the database was separated in 8 groups without subject overlap between groups (i.e. if an image of one subject is in one group, no image of the same subject will be in any other group). Each group contains about 12 subjects. This methodology ensures that testing groups do not have subjects from the training group and is also used by many methods in the literature [16,44,43,67–69,8,7,70]. As discussed by Girard et al. [53], this methodology (different subjects in training/testing groups and cross-validation) ensures the generalizability of classifiers. Zavaschi et al. [44] also discuss and support this data separation procedure. They conduct two experiments using the same methodology, just changing the way that the data groups are separated. In one experiment the groups contain images of the same subject (not the same images), while in the other experiment they guarantee that images of the same subject are not in the training and testing groups at same time. In the first experiment an accuracy of 99.40% was achieved, while in

Fig. 8. Example of the images in the CK+ database. In (1), the subject is in the neutral expression. In (2), the subject is in the surprise expression. In (3), the subject is in the disgust expression. In (4), the subject is in the fear expression.

the second the accuracy goes down to 88.90%. This result shows that methods which are evaluated without the same subjects in training/testing groups (which we believe to be a fairer evaluation) generally presents a lower accuracy than those that do not guarantee this constraint. We also confirmed these results with preliminary experiments using our method.

To verify the generalization of the proposed method some cross-database experiments were also performed. These experiments used the JAFFE and the BU-3DFE databases. The JAFFE database consists of 213 images from 10 Japanese female subjects. In this database, there are about 4 images in each one of the six basic expressions and one image of the neutral expression from each subject. All images in the dataset are 256 by 256 pixel arrays with 8-bit precision for grayscale values. As this database is smaller, the groups are separated by subject, i.e. each group contains images of just one subject, so 10 groups were formed. Some examples of the JAFFE database images are shown in Fig. 9.

Another database used to evaluate the proposed method is the Binghamton University 3D Facial Expression (BU-3DFE) [6]. The BU-3DFE database contains 64 subjects (56% female and 44% male), ranging age between 18 years to 70 years old, with a variety of ethnic/racial ancestries, including White, Black, East-Asian, Middle-east Asian, Indian and Hispanic Latino. All images in the dataset are 156 by 209 pixel arrays. This database was also separated in 8 groups, without subject overlap between groups. Each group contains about 8 subjects. Some examples of the BU-3DFE database images are shown in Fig. 10.

All databases contain a lot of images of the same subjects for each expression and these images are very similar to each other. So, only 3 frames of each expression (i.e. most expressive frames) and 1 frame for the neutral expression (i.e. least expressive frame), of each subject, were used in this work. Following this methodology, the resulting database sizes are as follows, for the CK+ database 2100 samples (without the synthetic samples) and 147,000 samples (with the synthetic samples), for the JAFFE database 213 samples (without the synthetic samples) and 14,910 samples (with the synthetic samples) and for the BU-3DFE database 1344 samples (without the synthetic samples) and 94,080 samples (with the synthetic samples).

4.2. Metrics

To allow for a fair comparison of the presented method with the literature, the accuracy was computed in two different ways. In the first, one classifier for all basic expression is used. The accuracy is computed simply using the average, C_{nclass}, of the n-class classifier accuracy per expression, $C_{nclassE}$, i.e. number of hits of an expression per amount of data of that expression, see the following equation:

$$C_{nclass} = \frac{\sum_1^n C_{nclassE}}{n}, \quad C_{nclassE} = \frac{Hit_E}{T_E} \tag{2}$$

where Hit_E is the number of hits in the expression E, T_E is total number of samples of that expression and n is the number of expressions to be considered.

In the second, one binary classifier for each expression performs a one-versus-all classification, as proposed in [7]. Using this approach, the images are presented to n binary classifiers, where n is the number of expressions being classified. Each classifier aims to answer "yes" if the image contains one specific expression, or "no" otherwise. For example, if one image contains the surprise expression, the surprise classifier should answer "yes" and all the other five classifiers should answer "no". The only difference for this classifier from the architecture presented in Section 3 is that only two outputs are required for each classifier. The accuracy is computed using the average, C_{bin}, of the binary classifier accuracy per expression, C_{binE}, i.e. the number of hits of an expression plus the number of hits of a non-expression divided per total amount of data, see the following equation:

$$C_{bin} = \frac{\sum_1^n C_{binE}}{n}, \quad C_{binE} = \frac{Hit_E + Hit_{NE}}{T} \tag{3}$$

where Hit_E is the number of hits in the expression E, i.e. number of times the classifier E responded "yes" and the tested image was of the expression E. Hit_{NE} is the number of times the classifier E responded "no" and the tested image was not the expression E. T is the total number of tested images and n is the number of expressions to be considered.

4.3. Pre-processing tuning

As described earlier, the proposed method combines a preprocessing step, that aims to remove non-expression specific features of a facial image and a Convolutional Neural Network to classify this preprocessed image in one of the six (or seven) expressions. In this section, we present the impact in the classification accuracy of each operation in the preprocessing step. As these experiments aim only to show the impact of the operations, a simplified version of our methodology was employed. Here, we randomly generate the order of the samples to the network and use a simple k-fold cross validation between the 8 groups of the CK+ database. The database was divided into two sets, training (with 7 of the groups) and test (with 1 of the groups). The training was performed 8 times using only 2000 epochs for each of them. The accuracy was computed per expression ($C_{6classE}$) and overall average for all expressions (C_{6class}).

(a) *No pre-processing* . This first experiment was carried out using the original database, without any intervention or image

Fig. 9. Example of the images in the JAFFE database. In (1), the subject is in the surprise expression. In (2), the subject is in the happy expression. In (3), the subject is in the sad expression. In (4), the subject is in the sad expression.

Fig. 10. Example of the images in the BU-3DFE database. In (1) the subject is in the fear expression. In (2) the subject is in the neutral expression. In (3) the subject is in the fear expression. In (4) the subject is in the fear expression.

pre-processing, just a down-sampling to the image to be of the same size as the input of the CNN. In this experiment, the average accuracy for all expressions was $C_{6class} = 53.57\%$. The accuracy per expression is shown in Table 1. The accuracy shown is an average of Eq. (2) for all runs.

As it can be seen in Table 1, using only the CNN without any image pre-processing, the recognition rate is very low compared with methods in the literature. We believe the variation and amount of samples in the CK+ database was small which did not allow the Convolutional Neural Network learn how to deal with pose, environment and subject variance. Additionally, it does not use the full range of the image space available in the network input to represent the face.

(*b*) *Image cropping*. In order to increase our method performance, as explained in Section 3, the image is automatically cropped to remove non-expression specific regions, in both training and testing steps. As a results, the average accuracy for all expressions increased to $C_{6class} = 71.67\%$. The accuracy per expression is shown in Table 1. Here, the down-sampling is also performed, because the input of the proposed network is a fixed 32×32 pixels image.

Compared with the result shown before, we can note a significant increase of the recognition rate by adding only the cropping process. The main reason for the accuracy increase is that with the cropping, we remove a lot of unnecessary information that the classifier will need to handle to determine the subject expression and use better the image space available in the network input.

(*c*)*Rotation correction*. A rotation correction (and the down-sampling) is performed in the image to remove rotations that are not related to expression facial changes (that can be pose-specific or caused by a camera movement), in both training and testing steps. The average accuracy for all expressions was $C_{6class} = 61.55\%$.

Note that, this result is applying just the rotation correction, but not the cropping. Compared with the result of no pre-processing we can note an increase of the accuracy in about 8.00%. This increase might be caused by the lower variation that the network needs to handle. With the rotation correction, the facial elements (eyes, mouth, and eyebrows) stay mostly in the same pixel space, but still has the influence of the background and does not use the full range of the image space available in the network input to represent the face, as in a).

(*d*) *Spatial normalization*. As seen before, the image cropping and the rotation correction applied separately, increase the classifier accuracy. This happens because both procedures reduce the problem complexity. Here, we discuss the full spatial normalization, composed by the image cropping, rotation correction and down-sampling. With the operations combined, the average accuracy for all expressions was $C_{6class} = 87.86\%$.

As expected, joining both procedures in the pre-processing step increase the accuracy. This happens because a lot of variation not related to the expression was removed from the image. Although the Convolutional Neural Network could handle these variations, we would need a bigger database (that we do not have) and maybe a more complex architecture.

(*e*) *Intensity normalization*. The spatial normalization procedure significantly increases the overall accuracy of the system. The intensity normalization is used to remove brightness variation in the images in both steps, training and testing. This experiment was performed using just the intensity normalization. It uses the same methodology described before. The average accuracy for all expressions was $C_{6class} = 57.00\%$.

As it can be seen, by just applying the intensity normalization the classifier accuracy was also slightly increased.

(*f*) *Spatial and intensity normalization*. Combining the spatial (rotation correction, cropping and down-sampling) and intensity

Table 1
Preprocessing steps tuning for the CK+ database: (a) no pre-processing; (b) just cropping; (c) just rotation correction; (d) cropping and rotation correction; (e) only intensity normalization; (f) both normalizations; (g) spatial normalization and synthetic samples; and (h) both normalizations and synthetic samples.

Preprocessing Step	**Angry** (%)	**Disgust** (%)	**Fear** (%)	**Happy** (%)	**Sad** (%)	**Surprise** (%)	**Average** (%)
(a)	28.10	51.23	17.91	70.68	20.99	77.52	53.57
(b)	68.60	79.01	23.37	86.39	23.46	87.16	71.67
(c)	17.17	79.09	00.00	48.92	05.05	91.25	61.55
(d)	81.82	90.74	73.13	95.81	66.67	94.50	87.86
(e)	27.27	52.94	08.22	79.10	18.29	85.54	57.00
(f)	78.51	93.21	53.73	95.29	75.31	93.12	86.67
(g)	86.05	88.30	69.33	96.60	77.11	95.34	87.10
(h)	**79.34**	**94.44**	**73.13**	**99.48**	**72.84**	**94.94**	**89.76**

Table 2
Impact of the presentation order in the accuracy.

Presentation order	Accuracy (%)
1	**88.18**
2	86.36
3	86.36
4	86.36
5	88.18
6	**84.55**
7	85.45
8	84.55
9	87.27
10	88.18
Average: 86.71%	
Standard deviation: 1.43%	

normalizations, we remove a big part of the variations unrelated to the facial expression and leave just the expression specific variation that is not related to the pose or environment. This experiment was done using the same methodology described before. The average accuracy for all expression was C_{6class} = 86.67%. The accuracy per expression is shown in Table 1.

As it can be seen, the accuracy of applying both normalization procedures is lower than the one that uses only the spatial normalization. The result of the fear expression has a very low accuracy, which reduces the overall recognition average.

(*g*) *Spatial normalization and synthetic samples*. The result of the spatial and intensity normalization and only the spatial normalization gives a false impression that the intensity normalization might decrease the accuracy of the method. To verify this assumption, a new experiment was conducted using only the spatial normalization procedure and the additional synthetic samples for training. For the synthetic samples generation, thirty more samples were generated for each image using a Gaussian standard deviation of 3 pixels ($\theta = 3$). The average accuracy for all expression was C_{6class} = 89.11%. The accuracy per expression is shown in Table 1.

This result increases the accuracy of applying just the spatial normalization (87.86%). However, this accuracy is outperformed by the next experiment, that apply both normalizations and the synthetic sample generation, showing that the synthetic sample generation procedure indeed increases the robustness of the classifier (motivated by the increase of the samples and its variation).

(*h*) *Spatial normalization, intensity normalization and synthetic samples*. The best result achieved in our method applies the three image pre-processing steps: spatial normalization, intensity normalization and synthetic samples generation. This experiment is performed using the same methodology described before. The average accuracy for all expression was C_{6class} = 89.79%. The accuracy of this experiment shows that combining the three techniques (spatial normalization, intensity normalization and synthetic samples) is better than using them individually.

Table 1 shows the mean accuracy for each expression using all the preprocessing steps already discussed for the CK+ database: (a) no preprocessing, (b) cropping, (c) rotation correction, (d) spatial normalization (cropping and rotation correction), (e) intensity normalization, (f) both normalizations (spatial and intensity), (g) spatial normalization using the synthetic samples and (h) both normalizations and the synthetic samples. The accuracy is computed using the six class classifier ($C_{6classE}$).. The best accuracies achieved are highlighted in bold.

4.4. Results

As discussed before, a simplified training/testing methodology was used to evaluate the impact of the pre-processing steps. In contrast to the tuning experiments that used only training and test sets, this section separates the databases for each experiment in three main sets: training set, validation set and test set.

In addition, it was also mentioned that the gradient descent method was used for training the network. Such methods might be influenced by the order of presentation of the samples to the network during training, which causes a variation in accuracy. As can be seen in Table 2 the accuracy increases (or decreases) in about 4.00% only with the order change of the training samples (as it can be seen in the values highlighted in bold). Table 2 shows the accuracy for the same training, performed 10 times, each one with a random presenting order of the training samples. To be less affected by this accuracy variation, we propose a training methodology that uses a validation set to choose the best network weights based on different trainings, illustrated in Fig. 2. Therefore, the final accuracy result of our experiments is computed using the network weights of the best run out of 10 runs, having a validation set for accuracy measurement. Each run has a random presentation order of the training samples. The weights of the best run in the validation set are later used to evaluate the test set and compute the final accuracy.

To verify that the proposed approach handles well other databases and even in unknown environments, some experiments with the BU-3DFE database, the JAFFE database and cross-database experiments were performed. The experiments with other databases follow the same approach that the CK+ database. In the cross-database experiments, the network was trained with the CK+ database and the accuracy evaluation was calculated using the BU-3DFE database or in the JAFFE database. In the cross-database experiments, no images from the BU-3DFE database or JAFFE database was used during the network training.

CK+ database experiment. The database was separated into 8 groups of non-overlapping subjects (with about 12 subjects each). Seven groups were used to compose the training set, whereas the eighth group (also with about 12 subjects) was shared between the validation set and test set, where 11 subjects were used for validation and 1 subject for testing. Our experiments follow a *k*-fold cross-validation configuration, in which, each time, one group is separated for validation/test and the other seven for training. When a group is selected for validation/test, a leave-one-out procedure is performed within the 12 subjects (having 11 for validation and 1 for test) for each of the training runs. For each configuration of validation and test subjects, the training is carried out 10 times, changing the presentation order of the training images. The validation group is used to select the best epoch for each run during training and to select the run with the best presentation order. Based on these information (best epoch and presentation order), the best network weights are selected and used to compute the accuracy of the test set. With this experimental configuration, the training of the network is performed about 960 times (8 groups * 12 subjects per group * 10 runs with different presenting order). The training of each run took only 2 min, therefore the total training of the system takes about 20 min (2 min * 10 runs). The overall experiment time was about 32 hs (including all combinations of training, validation and test). The results of this experiment are computed as an average of the 96 runs (8 folds * 12 subjects per group * 1 best configuration out of 10 runs).

Table 3 shows the best result achieved (using both normalizations and the synthetic samples) for C_{6class} and C_{bin}. As it can be seen, the binary classifier approach increases the accuracy. It happens because in this approach the hit can be achieved six times

Table 3
Accuracy for both classifiers using all processing steps and the synthetic samples for six expressions on the CK+ database.

Classifier	Angry (%)	Disgust (%)	Fear (%)	Happy (%)	Sad (%)	Surprise (%)
$C_{6classE}$	93.33	100.00	96.00	98.55	84.52	99.20
C_{binE}	98.27	99.37	99.24	99.68	98.17	98.81
Average of C_{6class}: 96.76% ± 0.07						
Average of C_{bin}: 98.92% ± 0.02						

(one for each classifier), instead of using just one classifier, where each sample has just one chance to be properly classified. The binary classifier approach was employed to allow a fair comparison with some methods in the literature that just report these results. We think that the six-class classifier (C_{6class}) is a fairer evaluation method. However, the C_{bin} classification approach is a useful method when we are interested in just one expression. The standard deviation reported in the table is based on the runs of all subjects.

The training parameter values that achieve the results shown in Table 3 are shown in Table 4. The same parameter values are used in the experiments on other databases.

Using the result shown in Table 3, the confusion matrix shown in Table 5 was created for the six-class classifier.

Based on the results of the six-class classifier, we can note that the disgust, happy and surprise expressions achieve an accuracy rate higher than 98%. While the angry and fear expression were about 93% and 96%, respectively. The sad expression achieves the smallest recognition rate, with only 84.52%. Looking at the confusion matrix, the sad expression was confused in the majority of the time with the surprise expression. This shows that the features of these two expressions are not well separated in the pixel space, i.e. they are very similar to each other in some cases. Fig. 11 shows some examples of the misclassification.

Fig. 12 shows an illustration of the learned kernels and the generated maps for each convolution layer. In the first convolution layer, the input image is processed by the 32 learned kernels and generates 32 output maps. In the second convolution layer, the 64 learned kernels are used to generate new maps for each one of the 32 maps of the previous layer. The kernels shown in Fig. 12 were learned in the training using the CK+ database for the six basic expressions. As it can be seen in the figure, after the second convolutional layer, the generated maps are focused on regions near the eyes, mouth and nose. Indeed, these regions are more critical for facial expression analysis, as suggested by psychological studies [71].

Instead of recognizing only six expressions, we can also recognize the neutral expression, which results in a classifier that recognizes seven expressions. The result of the seven expressions classifier to the CK+ database, using the same methodology of the six-class is shown in Table 6. The standard deviation reported in the table is between the runs of all subjects.

As it can be seen, for the binary classifier approach, we have a slight decrease in accuracy, from 98.90% to 98.80%. On the other hand, in the seven-class classifier the decrease was larger, from 96.7% to 95.7%. It happens because, in the seven-class classifier approach, one more output was included in the network. On the other hand, in the binary-class approach, one new classifier was inserted, keeping the others unchanged. The confusion matrix for the seven expressions is shown in Table 7.

Table 4
Training parameters.

Parameter	Value
Momentum	0.95
Learning rate	0.01
Epochs	10,000
Loss function	*SoftmaxWithLoss*
Gaussian standard deviation	3
Synthetic samples amount	70

Table 5
Confusion matrix using both normalizations and synthetic samples for six expressions on the CK+ database.

	Angry	Disgust	Fear	Happy	Sad	Surprise
Angry	126	6	2	0	1	0
Disgust	0	177	0	0	0	0
Fear	0	0	72	0	3	0
Happy	3	0	0	204	0	0
Sad	3	0	1	0	71	9
Surprise	1	0	1	0	0	247

BU-3DFE database experiment. This experiment follows the same approach as the CK+ database, the only difference is that in the BU-3DFE database the groups have about only eight subjects. The results of this experiment is computed as an average of the 64 runs (8 folds * 8 subjects per group * 1 best configuration out of 10 runs). The result for six and seven expressions for both classifiers is shown in Table 8. The standard deviation reported in the table is between the runs of all subjects.

As it can be seen, the accuracy for the BU-3DFE decreased compared with the CK+ database. One possible reason is that this database has more subjects from different ethnicities and light conditions, and is smaller than the CK+. In addition, we have an increase in the accuracy for the C_{bin} classifier on seven expressions, whereas we have a decrease in accuracy for the C_{nclass} classifier on seven expressions. The confusion matrices for this experiment, on six and seven expressions, can be seen in Table 13 and in Table 14, respectively, in the appendices (Appendix A).

The BU-3DFE database contains 3D models of faces in the six basic expressions and the neutral expressions. Commonly, this database is used for 3D reconstruction and facial expression recognition with 3D data. However, in this work, just the 2D image of the subjects are used to recognize the expressions. There are other works in the literature that presents higher facial expression recognition accuracies, but using 3D information to infer the expressions [72–74].

A better evaluation of the proposed method in real environments is the cross-database experiments, i.e. train the method with one database and test with another (in this case the BU-3DFE).

To perform the cross-database experiment, seven groups of the CK+ database were used to train the network and one was used to be the validation set (to choose the best network weights based on the best epoch presentation order of the training set). The BU-3DFE was used to test the network. The training was done eight times, each one with a different validation set. We ran each configuration (training set plus validation set) 10 times, each one with a different presenting order in the training samples. The result in the cross-database experiment is computed as an average of the 8 runs to consider different combinations of training and validation sets (8 folds * 1 best configuration out of 10 runs) showing all the BU-3DFE images to test the network. The results are shown in Table 9.

The confusion matrices for this experiment, on six and seven expressions, can be seen in Table 15 and in Table 16 respectively, in the appendices (Appendix A).

JAFFE database experiment. This experiment follows a slightly different approach from the CK+ and BU-3DFE experiments in regards to the number of groups. The JAFFE database contains

Fig. 11. In (1), the expected expression was sad, but the method returned fear. In (2), the expected expression was angry, but the method returned fear. In (3), the expected expression was sad, but the method returned angry. In (4), the expected expression was angry, but the method returned sad.

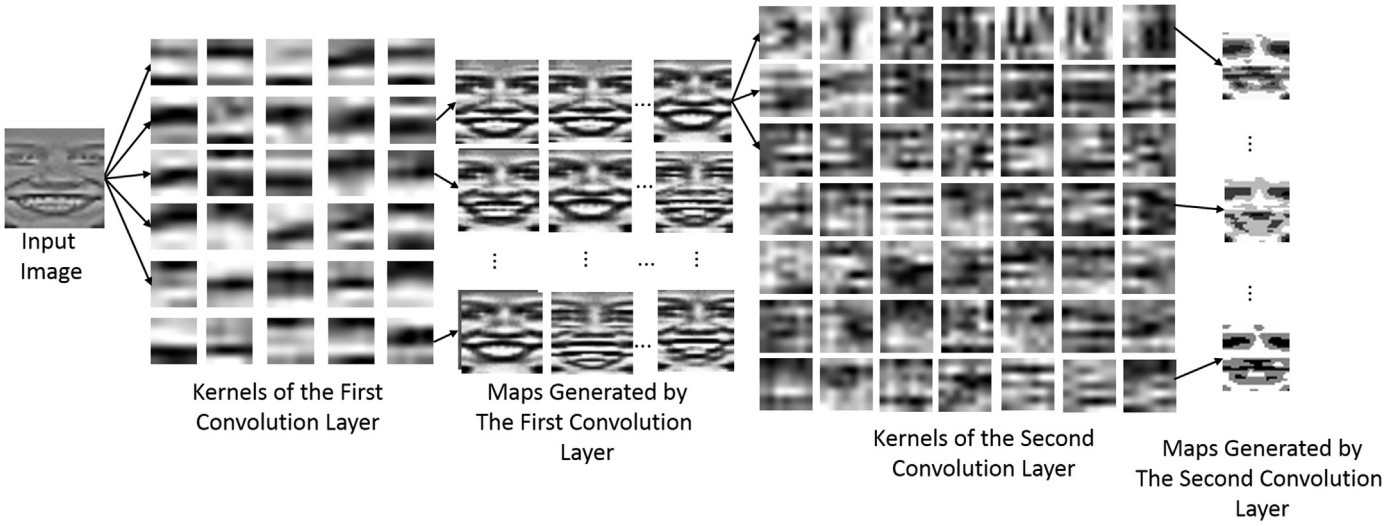


Fig. 12. Illustration of the learned kernels and the generated maps for each convolution layer. In the first convolution layer, the input image is processed by the 32 learned kernels and generates 32 output maps. In the second convolution layer, the 64 learned kernels are used to generate new maps for each one of the 32 maps of the previous layer. The sub-sampling layers are not represented in this image. Only a subset of the 32 kernels for the first layer and of the 64 kernels for the second layer are shown. The generated maps were equalized to allow for a better visualization.

Table 6
Accuracy for both classifiers using all processing steps and the synthetic samples for seven expressions.

Classifier	**Neutral** (%)	**Angry** (%)	**Disgust** (%)	**Fear** (%)	**Happy** (%)	**Sad** (%)	**Surprise** (%)
$C_{7classE}$	95.15	91.11	99.44	92.00	100.0	82.14	98.80
C_{binE}	97.49	97.82	99.76	99.11	99.76	98.79	98.87
Average of C_{7class}: **95.79**% $\pm$ **0.06**							
Average of C_{bin}: **98.80**% $\pm$ **0.01**							

Table 7
Confusion matrix using both normalizations and synthetic samples for seven expressions on the CK+ database.

	Neutral	Angry	Disgust	Fear	Happy	Sad	Surprise
Neutral	294	11	1	1	0	0	2
Angry	8	123	1	0	3	0	0
Disgust	0	1	176	0	0	0	0
Fear	6	0	0	69	0	0	0
Happy	0	0	0	0	207	0	0
Sad	0	3	0	3	0	69	9
Surprise	2	0	0	1	0	0	246

images from only 10 subjects, therefore, as done by other works in the literature [7,8], the images were separated in 10 groups, each one with just one subject. The test was carried out using a 10-fold cross validation, the training group contains eight subjects, the validation group one subject and the testing group one subject. The result of this experiment is computed as an average of the 10

Table 8
Accuracy using six and seven (six basic plus neutral) expressions for the BU-3DFE and JAFFE databases.

Train	**Test**	**Classifier**	**6-expressions**	**7-expressions**
BU-3DFE	BU-3DFE	C_{nclass}	72.89% $\pm$ 0.05	71.62% $\pm$ 0.04
BU-3DFE	BU-3DFE	C_{bin}	90.96% $\pm$ 0.01	91.89% $\pm$ 0.01
JAFFE	JAFFE	C_{nclass}	53.44% $\pm$ 0.15	53.57% $\pm$ 0.13
JAFFE	JAFFE	C_{bin}	84.48% $\pm$ 0.05	86.74% $\pm$ 0.03

Table 9
Cross-database experiment for the BU-3DFE and JAFFE databases.

Train	**Test**	**Classifier**	**6-expressions** (%)	**7-expressions** (%)
CK+	BU-3DFE	C_{nclass}	45.91	42.25
CK+	BU-3DFE	C_{bin}	81.97	83.50
CK+	JAFFE	C_{nclass}	38.80	37.36
CK+	JAFFE	C_{bin}	79.60	82.10

runs (10 folds * 1 subjects per group * 1 best configuration out of 10 runs). The result for six and seven expressions for both classifiers is shown in Table 8. The standard deviation reported in the table is between the runs of all subjects.

As it can be seen in Table 8, compared with the CK+ and BU-3DFE results, the accuracy decreased considerably. It also happens in other works [7,8,44], motivated mainly due to the small database. The required data amount that methods for facial action (or expression) recognition need in the training phase to achieve a good accuracy is studied by Girard et al. [53]. In one of their experiments, the recognition rate varies from 63% to 94% as the training set increased from 8 subjects to 64 subjects. Once the JAFFE dataset contains images from only 10 subjects, it is unfair to compare its result with the CK+ or the BU3DFE datasets that contain, respectively, 100 subjects and 64 subjects.

This problem of small amount of data is more emphasized in the technique used in this work. Convolutional Neural Networks, generally, requires a big amount of data to adjust its parameters. Therefore, there are other approaches in the literature that do not require a so high amount of data and consequently achieve better results than the method presented here [44,8]. The confusion matrices for this experiment, on six and seven expressions, can be seen in Table 17 and in Table 18, respectively, in the appendices (Appendix A).

The cross-database experiment was also performed to the JAFFE database. In this experiment the network is trained and validated only on the CK+ database and the tests were carried out on the JAFFE database. This experiment follows the same approach described for the BU-3DFE. The result in the cross-database experiment is computed as an average of the 8 runs to consider different combinations of training and validation sets (8 folds * 1 best configuration out of 10 runs) showing all the JAFFE images to test the network. The results for this experiment are shown in Table 9.

One motivation for the small accuracy reported in Table 9 is the cultural differences over the training subjects and the testing subjects. While in the BU-3DFE we have some subjects of the same ethnicity of the CK+ database, the same does not happen with the JAFFE database. The confusion matrices for this experiment, on six and seven expressions, can be seen in Table 19 and in Table 20, respectively, in the appendices (Appendix A).

4.5. Comparisons

Table 10 summarizes all results presented in this section, showing the evolution of the proposed method by changing the image pre-processing steps.

As it can be seen in Table 10, the best performance was achieved using both normalization procedures and the synthetic samples (as it can be seen in the row highlighted in bold). This table just summarizes the experiments performed to choose the best configuration to the proposed classifier, using a simplified training methodology.

Table 10
Preprocessing comparison.

Preprocessing	Accuracy (%)
None	53.57
Cropping	71.67
Rotation correction	61.55
Spatial normalization	87.86
Intensity normalization	57.00
Spatial and intensity normalization	86.67
Spatial normalization and synthetic samples	89.11
Spatial and intensity normalization and synthetic samples	**89.79**

Table 11
Comparison for the CK+ database.

Method	Classifier	6-expressions	7-expressions
Fan and Tjahjadi [16][a]	C_{nclass}	83.70	–
	C_{bin}	–	–
Zavaschi et al. [44]	C_{nclass}	–	88.90
	C_{bin}	–	–
Rivera et al. [43]	C_{nclass}	–	89.30
	C_{bin}	–	–
Gu et al. [67]	C_{nclass}	91.51	–
	C_{bin}	–	–
Zhong et al. [68]	C_{nclass}	93.30	–
	C_{bin}	–	–
AUDN [12]	C_{nclass}	93.70	–
	C_{bin}	–	–
Liu et al. [76][a]	C_{nclass}	94.19	–
	C_{bin}	–	–
Lee et al. [69][a]	C_{nclass}	94.90	–
	C_{bin}	–	–
LBP+SVM [8]	C_{nclass}	95.10	91.40
	C_{bin}	–	–
BDBN [7]	C_{nclass}	–	–
	C_{bin}	96.70	–
IntraFace [70][a]	C_{nclass}	96.40	–
	C_{bin}	–	–
Proposed	C_{nclass}	**96.76**	**95.75**
	C_{bin}	**98.92**	**98.80**

[a] The authors also recognize the contempt expression. They were not placed in the 7-expression column because our 7-expression recognition system is based on the six basic expression plus the neural expression (not the contempt expression).

Table 11 shows the results of the CK+ database using the full validation methodology (with training, validation and test sets) explained before. In this table, a comparison between the proposed method and other methods in the literature that use the same experimental methodology (i.e. perform a *k*-fold cross-validation and guarantee that the same subject is not in the training and testing groups) is presented.

As it can be seen in Table 11, the proposed method achieves competitive results in the CK+ database for all experiment configurations (as it can be seen in the row highlighted in bold). Besides, the training and recognition times are also smaller than the others. The total time to train the system is with the CK+ is about 20 min. The whole experimentation time including all the *k*-fold configurations of the proposed method was 32 h, and the recognition is real time (only 0.01 s for each image), almost 100 images can be processed per second. In comparison with Liu et al. [7], their training took eight days and the recognition was about 0.21 s per image. Note that we disregard the hardware used for computing the time. Shan et al. [8] and Zhon et al. [68] did not report the training and recognition time. The method proposed by Lee et al. [69] requires manual image preprocessing before the facial expression recognition.

It is important to note that although there are some other methods in the literature that report accuracies higher than ours

(98.92%), the results are not comparable. As discussed before, a fair evaluation method for facial expression recognition should guarantee that images of a subject are not present in both training and testing sets at the same time. Accuracy achieved without this constraint overcomes the presented result. Some of these methods randomly select data to the folds [39–42]. Some other do not mention if this constraint was kept in the images chosen for each fold [77,78,10] therefore they were assumed to have an overlap. There are also methods which select only a subset of the database (or a subset of the six basic expressions) to evaluate the accuracy [79–81,13]. Rivera et al. [43] also reported an accuracy of 99.50% in the CK [82] database, which is a previous version of the CK+ [4] database with less subjects and image sequences. However, as it can be seen in Table 11, the results with CK+ tend to be worse than the ones with the CK.

Some methods in the literature [8,7,67] also performed the cross-database experiment in the JAFFE database (training in the CK+ database and testing in the JAFFE). A comparison of this work with these methods, for the cross-database experiment, is shown in Table 12. In our literature review, we did not find any work using a cross-database evaluation with the BU-3DFE for expression recognition. The results achieved by the proposed method in the cross-database experiments with the JAFFE database are highlighted in bold.

Comparing the presented method with Shan et al. [8] our accuracy was about 4% smaller using 7 expressions and the C_{nclass} classifier. They did not report the result of the six basic expressions. On the other hand, compared with *Liu et al.* [7], using the binary classifier approach the proposed method significantly increases the accuracy, from 68.0% to 82.0%. Ali et al. [13] also perform a cross-database validation in their work, achieving an accuracy of 48.67% training in the RAFD database and testing with the JAFFE database. Unfortunately, we cannot compare with our results due to the discrepancy in the experimental methodology. The first is the database, in our case the training was carried out with the CK+ database, while theirs with the RAFD database. Secondly, they recognize only five expressions (anger, happy, surprise, sad and fear).

4.6. Limitations

As discussed before, the presented method needs the locations of each eye for the image pre-processing steps. The eye detection can be easily included in the system adopting methods available in the literature [59,61], while still keeping whole method (including the eyes detection) in real time. In addition, as shown in Table 1, the accuracy of some expressions, e.g. sad, was about 84%, whereas the accuracy of the whole method was about 96%. This suggests that the variation between these classes are not enough to separate them. One approach to address this problem is to create a specialized classifier for those expressions, to be used as a second classifier. Another limitation of the presented method is the controlled environment of the input images with the frontal face of the subjects. All these limitations, could be addressed with a large set of training data, which will allow for a deeper network, capable of handling such constraints.

Table 12
Comparison for the JAFFE cross-database experiment.

Method	Classifier	6-expressions	7-expressions
LBP+SVM [8]	C_{nclass}	–	41.30
	C_{bin}	–	–
BDBN [7]	C_{nclass}	–	–
	C_{bin}	–	68.00
Gu et al. [67]	C_{nclass}	55.87	–
	C_{bin}	–	–
Proposed	C_{nclass}	**38.80**	**37.36**
	C_{bin}	**79.60**	**82.10**

5. Conclusion

In this paper, we propose a facial expression recognition system that uses a combination of standard methods, like Convolutional Neural Network and specific image pre-processing steps. Experiments showed that the combination of the normalization procedures improve significantly the method's accuracy. As shown in the results, in comparison with the recent methods in the literature, that use the same facial expression database and experimental methodology, our method achieves competitive results and presents a simpler solution. In addition, it takes less time to train and its recognition is performed in real time. Finally, the cross-database experiments show that the proposed approach also works in unknown environments, where the testing image acquisition conditions and subjects vary from the training images, but still has room left for improvement.

As explained in Section 1, the use of Convolutional Neural Networks aims to decrease the need for hand-coded features. Its input can be raw images, instead of an already selected set of features. It happens because this neural network model is able to learn the set of features that best models the desired classification. To perform such learning, Convolutional Neural Networks need a large amount of data, that we do not have. This is a constraint of deep architectures, motivated by the large amount of parameters that need adjustment during training. To address this problem (our limited data), the pre-processing operations were applied to the images, in order to decrease the variations between images and to select a subset of the features to be learned, which reduces the need for a large amount of data. If we had a better set of images, with more variation and more samples (millions), these pre-processing operations could not be necessary to achieve the reported accuracy and even the cross-database validation could be improved.

Preliminary experiments were performed with deeper architectures, trained with a big amount of data. In these experiments, a deep Convolutional Neural Network composed by 38 layers and trained with about 982,800 images from 2662 subjects, proposed by Parkhi et al. [83] to recognize faces, was briefly studied. The already trained model was used as a pre-trained feature extractor plugged as input of a simple two-layered neural network trained with the CK+ database. In this experiment, no preprocessing operation was applied. Despite the experiment simplicity, the results achieved were promising and even increase the accuracy achieved in the cross-database experiments (reported in Section 4), with the cost of decreasing the accuracy in the same database experiments. These results indicate that a deep learning approach of such type can be a better way to produce a discriminative model for facial expression recognition, allowing it to work in uncontrolled scenarios, which is one of the current challenges in this field.

As future work, the application of this feature extraction method will be investigated in other problems. In addition, we want to investigate other learning methods in order to increase the method robustness in unknown environments (e.g. with varying light conditions, culture and others). Also, more tests will be performed using the face descriptor proposed by Parkhi et al. [83], using fine adjustment techniques, which aims to tune an already trained deep neural network in order to focus on more specific features (in our case, expressions).

Conflict of interest

We wish to confirm that there are no known conflicts of interest associated with this work.

Acknowledgment

We would like to thank Universidade Federal do Espírito Santos – UFES (project SIEEPEF, 5911/2015), Fundação de Amparo Pesquisa do Espírito Santo – FAPES (grants 65883632/14, 53631242/11, and 60902841/13), Coordenação de Aperfeiçoamento de Pessoal de Nível Superior – CAPES (grant 11012/13-7 and scholarship) and Conselho Nacional de Desenvolvimento Científico e Tecnológico – CNPq (grants 552630/2011-0 and 312786/2013-1).

Appendix A

See Tables 13–20.

Table 13
Confusion matrix for six expressions on the BU-3DFE database.

	Angry	Disgust	Fear	Happy	Sad	Surprise
Angry	109	26	7	2	10	3
Disgust	15	12	14	11	0	7
Fear	12	21	61	10	9	19
Happy	0	6	6	159	3	0
Sad	17	3	14	5	124	5
Surprise	7	6	6	5	11	134

Table 14
Confusion matrix for seven expressions on the BU-3DFE database.

	Neutral	Angry	Disgust	Fear	Happy	Sad	Surprise
Neutral	191	11	4	5	1	14	6
Angry	22	93	24	3	0	15	0
Disgust	0	9	117	20	6	0	7
Fear	21	3	23	47	6	10	22
Happy	5	0	6	2	160	0	1
Sad	28	10	0	8	3	113	6
Surprise	13	0	6	7	6	5	132

Table 15
Confusion matrix for six expressions on the BU-3DFE database in the cross-database experiment.

	Angry	Disgust	Fear	Happy	Sad	Surprise
Angry	55	30	0	0	23	49
Disgust	12	57	9	6	5	71
Fear	4	3	13	12	24	76
Happy	6	8	5	111	22	22
Sad	6	0	3	3	51	105
Surprise	5	0	0	3	7	154

Table 16
Confusion matrix for seven expressions on the BU-3DFE database in the cross-database experiment.

	Neutral	Angry	Disgust	Fear	Happy	Sad	Surprise
Neutral	116	1	0	4	0	24	87
Angry	77	28	25	0	0	5	22
Disgust	23	8	62	3	9	10	44
Fear	29	0	3	7	11	25	57
Happy	36	3	8	2	100	14	11
Sad	21	7	0	3	3	54	80
Surprise	1	3	2	0	3	5	155

Table 17
Confusion matrix for six expressions on the JAFFE database.

	Angry	Disgust	Fear	Happy	Sad	Surprise
Angry	168	32	26	10	32	2
Disgust	79	129	21	1	30	1
Fear	38	17	98	6	72	57
Happy	9	6	6	224	19	24
Sad	53	26	42	14	123	21
Surprise	7	1	61	17	21	163

Table 18
Confusion matrix for seven expressions on the JAFFE database.

	Neutral	Angry	Disgust	Fear	Happy	Sad	Surprise
Neutral	156	33	3	4	22	8	35
Angry	6	175	17	28	2	39	3
Disgust	5	84	118	12	0	41	1
Fear	24	30	0	103	0	80	43
Happy	46	12	1	3	210	6	10
Sad	8	55	27	42	12	116	19
Surprise	66	2	0	36	11	6	149

Table 19
Confusion matrix for six expressions on the JAFFE database in the cross-database experiment.

	Angry	Disgust	Fear	Happy	Sad	Surprise
Angry	2	2	2	2	11	11
Disgust	1	3	0	0	7	18
Fear	1	0	2	0	9	20
Happy	1	0	2	18	11	0
Sad	0	2	0	1	18	10
Surprise	1	0	0	1	3	25

Table 20
Confusion matrix for seven expressions on the JAFFE database in the cross-database experiment.

	Neutral	Angry	Disgust	Fear	Happy	Sad	Surprise
Neutral	4	0	0	0	0	20	5
Angry	5	4	2	0	3	10	6
Disgust	3	2	4	0	1	7	12
Fear	0	0	0	3	0	9	20
Happy	2	2	0	2	17	8	1
Sad	1	2	2	0	1	18	7
Surprise	0	1	0	0	1	2	26

References

[1] Y. Wu, H. Liu, H. Zha, Modeling facial expression space for recognition, in: 2005 IEEE/RSJ International Conference on Intelligent Robots and Systems, 2005 (IROS 2005), 2005, pp. 1968–1973.

[2] C. Darwin, The Expression of the Emotions in Man and Animals, CreateSpace Independent Publishing Platform, 2012.

[3] S.Z. Li, A.K. Jain, Handbook of Face Recognition, Springer Science & Business Media, Secaucus, NJ, USA, 2011.

[4] P. Lucey, J. Cohn, T. Kanade, J. Saragih, Z. Ambadar, I. Matthews, The extended Cohn–Kanade dataset (CK+): a complete dataset for action unit and emotion-specified expression, in: 2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), 2010, pp. 94–101.

[5] M. Lyons, J. Budynek, S. Akamatsu, Automatic classification of single facial images, IEEE Trans. Pattern Anal. Mach. Intell. 21 (12) (1999) 1357–1362.

[6] L. Yin, X. Wei, Y. Sun, J. Wang, M. Rosato, A 3d facial expression database for facial behavior research, in: 7th International Conference on Automatic Face and Gesture Recognition (FGR06), Institute of Electrical & Electronics Engineers (IEEE), Southampton, UK, 2006.

[7] P. Liu, S. Han, Z. Meng, Y. Tong, Facial expression recognition via a boosted deep belief network, in: 2014 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2014, pp. 1805–1812.

[8] C. Shan, S. Gong, P.W. McOwan, Facial expression recognition based on local binary patterns: a comprehensive study, Image Vis. Comput. 27 (6) (2009) 803–816.

[9] W. Liu, C. Song, Y. Wang, Facial expression recognition based on discriminative dictionary learning, in: 2012 21st International Conference on Pattern Recognition (ICPR), 2012, pp. 1839–1842.

[10] I. Song, H.-J. Kim, P.B. Jeon, Deep learning for real-time robust facial expression recognition on a smartphone, in: International Conference on Consumer Electronics (ICCE), Institute of Electrical & Electronics Engineers (IEEE), Las Vegas, NV, USA, 2014.

[11] P. Burkert, F. Trier, M.Z. Afzal, A. Dengel, M. Liwicki, Dexpression: Deep Convolutional Neural Network for Expression Recognition, CoRR abs/1509.05371 (URL ⟨http://arxiv.org/abs/1509.05371⟩).

[12] M. Liu, S. Li, S. Shan, X. Chen, Au-inspired deep networks for facial expression feature learning, Neurocomputing 159 (2015) 126–136, http://dx.doi.org/10.1016/j.neucom.2015.02.011.

[13] G. Ali, M.A. Iqbal, T.-S. Choi, Boosted NNE collections for multicultural facial expression recognition, Pattern Recognit. 55 (2016) 14–27, http://dx.doi.org/10.1016/j.patcog.2016.01.032.

[14] Y.-H. Byeon, K.-C. Kwak, Facial expression recognition using 3d convolutional neural network. International Journal of Advanced Computer Science and Applications(IJACSA), 5 (2014).

[15] J.-J.J. Lien, T. Kanade, J. Cohn, C. Li, Detection, tracking, and classification of action units in facial expression, J. Robot. Auton. Syst. 31(3), 2000, 131-146.

[16] X. Fan, T. Tjahjadi, A spatial–temporal framework based on histogram of gradients and optical flow for facial expression recognition in video sequences, Pattern Recognit. 48 (11) (2015) 3407–3416.

[17] W. Zhang, Y. Zhang, L. Ma, J. Guan, S. Gong, Multimodal learning for facial expression recognition, Pattern Recognit. 48 (10) (2015) 3191–3202.

[18] C.-R. Chen, W.-S. Wong, C.-T. Chiu, A 0.64 mm real-time cascade face detection design based on reduced two-field extraction, IEEE Trans. Very Large Scale Integr. (VLSI) Syst. 19 (11) (2011) 1937–1948.

[19] C. Garcia, M. Delakis, Convolutional face finder: a neural architecture for fast and robust face detection, IEEE Trans. Pattern Anal. Mach. Intell. 26 (11) (2004) 1408–1423, http://dx.doi.org/10.1109/TPAMI.2004.97.

[20] Z. Zhang, D. Yi, Z. Lei, S. Li, Regularized transfer boosting for face detection across spectrum, IEEE Signal Process. Lett. 19 (3) (2012) 131–134.

[21] M. Bartlett, G. Littlewort, M. Frank, C. Lainscsek, I. Fasel, J. Movellan, Recognizing facial expression: machine learning and application to spontaneous behavior, in: IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2005 (CVPR 2005), vol. 2, 2005, pp. 568–573.

[22] P. Liu, M. Reale, L. Yin, 3d head pose estimation based on scene flow and generic head model, in: 2012 IEEE International Conference on Multimedia and Expo (ICME), 2012, pp. 794–799.

[23] W.W. Kim, S. Park, J. Hwang, S. Lee, Automatic head pose estimation from a single camera using projective geometry, in: 2011 8th International Conference on Information, Communications and Signal Processing (ICICS), 2011, pp. 1–5.

[24] M. Demirkus, D. Precup, J. Clark, T. Arbel, Multi-layer temporal graphical model for head pose estimation in real-world videos, in: 2014 IEEE International Conference on Image Processing (ICIP), 2014, pp. 3392–3396.

[25] Z. Zhang, M. Lyons, M. Schuster, S. Akamatsu, Comparison between geometry-based and gabor-wavelets-based facial expression recognition using multi-layer perceptron, in: 1998 Proceedings of the Third IEEE International Conference on Automatic Face and Gesture Recognition, 1998, pp. 454–459.

[26] P. Yang, Q. Liu, D. Metaxas, Boosting coded dynamic features for facial action units and facial expression recognition, in: IEEE Conference on Computer Vision and Pattern Recognition, 2007 (CVPR'07), 2007, pp. 1–6.

[27] S. Jain, C. Hu, J. Aggarwal, Facial expression recognition with temporal modeling of shapes, in: 2011 IEEE International Conference on Computer Vision Workshops (ICCV Workshops), 2011, pp. 1642–1649, http://dx.doi.org/10.1109/ICCVW.2011.6130446.

[28] Y. Lin, M. Song, D.T.P. Quynh, Y. He, C. Chen, Sparse coding for flexible, robust 3d facial-expression synthesis, IEEE Comput. Graph. Appl. 32 (2) (2012) 76–88.

[29] S. Rifai, Y. Bengio, A. Courville, P. Vincent, M. Mirza, Disentangling factors of variation for facial expression recognition, in: A. Fitzgibbon, S. Lazebnik, P. Perona, Y. Sato, C. Schmid (Eds.), Computer Vision – ECCV 2012, Lecture Notes in Computer Science, vol. 7577, Springer, Berlin Heidelberg, 2012, pp. 808–822.

[30] B. Fasel, Robust face analysis using convolutional neural networks, in: Proceedings of the 16th International Conference on Pattern Recognition, 2002, vol. 2, 2002, pp. 40–43.

[31] F. Beat, Head-pose invariant facial expression recognition using convolutional neural networks, in: Proceedings of the Fourth IEEE International Conference on Multimodal Interfaces, 2002, 2002, pp. 529–534.

[32] M. Matsugu, K. Mori, Y. Mitari, Y. Kaneda, Subject independent facial expression recognition with robust face detection using a convolutional neural network, Neural Netw.: Off. J. Int. Neural Netw. Soc. 16 (5) (2003) 555–559.

[33] Y. Bengio, Y. LeCun, Scaling learning algorithms towards AI, in: L. Bottou, O. Chapelle, D. DeCoste, J. Weston (Eds.), Large-Scale Kernel Machines, MIT Press, Cambridge, Massachusetts, USA, 2007 (URL ⟨http://yann.lecun.com/exdb/publis/pdf/bengio-lecun-07.pdf⟩).

[34] P.E. Utgoff, D.J. Stracuzzi, Many-layered learning, Neural Comput. 14 (10) (2002) 2497–2529.

[35] Y. Bengio, I.J. Goodfellow, A. Courville, Deep Learning, MIT Press, Cambridge, Massachusetts, USA, 2015.

[36] Y. Lecun, L. Bottou, Y. Bengio, P. Haffner, Gradient-based Learn. Appl. Doc. Recognit. 86 (11) (1998) 2278–2324, http://dx.doi.org/10.1109/5.726791.

[37] D.C. Cirean, U. Meier, J. Masci, L.M. Gambardella, J. Schmidhuber, Flexible, high performance convolutional neural networks for image classification, in: Proceedings of the Twenty-Second International Joint Conference on Artificial Intelligence (IJCAI'11), vol. 2, AAAI Press, Barcelona, Catalonia, Spain, 2011, pp. 1237–1242.

[38] P. Zhao-yi, W. Zhi-qiang, Z. Yu, Application of mean shift algorithm in real-time facial expression recognition, in: International Symposium on Computer Network and Multimedia Technology, 2009 (CNMT 2009), 2009, pp. 1–4.

[39] H.Y. Patil, A.G. Kothari, K.M. Bhurchandi, Expression invariant face recognition using local binary patterns and contourlet transform, Opt.-Int. J. Light Electron Opt. 127 (5) (2016) 2670–2678, http://dx.doi.org/10.1016/j.ijleo.2015.11.187.

[40] J.Y.R. Cornejo, H. Pedrini, F. Florez-Revuelta, Facial expression recognition with occlusions based on geometric representation, in: Progress in Pattern Recognition, Image Analysis, Computer Vision, and Applications: Proceedings of the 20th Iberoamerican Congress (CIARP 2015), Montevideo, Uruguay, November 9–12, 2015, Springer International Publishing, Cham, 2015, pp. 263–270.

[41] Z. Wang, Q. Ruan, G. An, Facial expression recognition using sparse local fisher discriminant analysis, Neurocomputing 174 (Part B) (2016) 756–766, http://dx.doi.org/10.1016/j.neucom.2015.09.083.

[42] S. Arivazhagan, R.A. Priyadharshini, S. Sowmiya, Facial expression recognition based on local directional number pattern and anfis classifier, in: 2014 International Conference on Communication and Network Technologies (ICCNT), 2014, pp. 62–67 (http://dx.doi.org/10.1109/CNT.2014.7062726).

[43] A.R. Rivera, J.R. Castillo, O.O. Chae, Local directional number pattern for face analysis: face and expression recognition, IEEE Trans. Image Process. 22 (5) (2013) 1740–1752.

[44] T.H. Zavaschi, A.S. Britto, L.E. Oliveira, A.L. Koerich, Fusion of feature sets and classifiers for facial expression recognition, Expert Syst. Appl. 40 (2) (2013) 646–655, http://dx.doi.org/10.1016/j.eswa.2012.07.074.

[45] A.T. Lopes, E. de Aguiar, T.O. Santos, A facial expression recognition system using convolutional networks, in: 2015 28th SIBGRAPI Conference on Graphics, Patterns and Images, Institute of Electrical & Electronics Engineers (IEEE), Salvador, Bahia, Brasil, 2015.

[46] S. Demyanov, J. Bailey, R. Kotagiri, C. Leckie, Invariant Backpropagation: How To Train a Transformation-Invariant Neural Network (arXiv:1502.04434[cs, stat]).

[47] Y. Jia, E. Shelhamer, J. Donahue, S. Karayev, J. Long, R. Girshick, S. Guadarrama, T. Darrell, Caffe: Convolutional Architecture for Fast Feature Embedding (arXiv:1408.5093).

[48] C.-D. Caleanu, Face expression recognition: a brief overview of the last decade, in: 2013 IEEE 8th International Symposium on Applied Computational Intelligence and Informatics (SACI), 2013, pp. 157–161.

[49] M.K.A.E. Meguid, M.D. Levine, Fully automated recognition of spontaneous facial expressions in videos using random forest classifiers, IEEE Trans. Affect. Comput. 5 (2) (2014) 141–154, http://dx.doi.org/10.1109/TAFFC.2014.2317711.

[50] C. Turan, K. M. Lam, Region-based feature fusion for facial-expression recognition, in: 2014 IEEE International Conference on Image Processing (ICIP), 2014, pp. 5966–5970 (http://dx.doi.org/10.1109/ICIP.2014.7026204).

[51] M. Lyons, S. Akamatsu, M. Kamachi, J. Gyoba, Coding facial expressions with gabor wavelets, in: Proceedings of the Third IEEE International Conference on Automatic Face and Gesture Recognition, 1998, 1998, pp. 200–205.

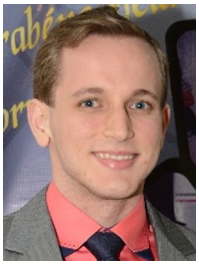
[52] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, R. Salakhutdinov, Dropout: a simple way to prevent neural networks from overfitting, J. Mach. Learn. Res. 15 (1) (2014) 1929–1958.

[53] J.M. Girard, J.F. Cohn, L.A. Jeni, S. Lucey, F.D. la Torre, How much training data for facial action unit detection?, in: 2015 11th IEEE International Conference and Workshops on Automatic Face and Gesture Recognition (FG), vol. 1, 2015, pp. 1–8 (http://dx.doi.org/10.1109/FG.2015.7163106).

[54] M. Valstar, M. Pantic, Induced disgust, happiness and surprise: an addition to the mmi facial expression database, in: Proceedings of the 3rd International Workshop on EMOTION (Satellite of LREC): Corpora for Research on Emotion and Affect, 2010, p. 65.

[55] A. Dhall, R. Goecke, S. Lucey, T. Gedeon, Static facial expression analysis in tough conditions: data, evaluation protocol and benchmark, in: 2011 IEEE

International Conference on Computer Vision Workshops (ICCV Workshops), IEEE, Barcelona, Catalonia, Spain, 2011, pp. 2106–2112.
[56] P. Simard, D. Steinkraus, J.C. Platt, Best practices for convolutional neural networks applied to visual document analysis, in: 2003 Proceedings of the Seventh International Conference on Document Analysis and Recognition, 2003, pp. 958–963.
[57] J.-I. Choi, C.-W. La, P.-K. Rhee, Y.-L. Bae, Face and eye location algorithms for visual user interface, in: Proceedings of First Signal Processing Society Workshop on Multimedia Signal Processing, Institute of Electrical & Electronics Engineers (IEEE), Princeton, NJ, USA, 1997.
[58] G. Li, X. Cai, X. Li, Y. Liu, An efficient face normalization algorithm based on eyes detection, in: 2006 IEEE/RSJ International Conference on Intelligent Robots and Systems, Institute of Electrical & Electronics Engineers (IEEE), Beijing, China, 2006.
[59] J.M. Saragih, S. Lucey, J.F. Cohn, Deformable model fitting by regularized landmark mean-shift, Int. J. Comput. Vision. 91 (2) (2010) 200–215.
[60] A. Asthana, S. Zafeiriou, S. Cheng, M. Pantic, Robust discriminative response map fitting with constrained local models, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2013, pp. 3444–3451.
[61] S. Cheng, A. Asthana, S. Zafeiriou, J. Shen, M. Pantic, Real-time generic face tracking in the wild with cuda, in: Proceedings of the 5th ACM Multimedia Systems Conference, ACM, Singapore, Singapore 2014, pp. 148–151.
[62] B.A. Wandell, Foundations of Vision, 1st ed., Sinauer Associates Inc, Sunderland, Mass, 1995.
[63] L. Bottou, Stochastic Gradient Descent Tricks, Springer, New York, NY, USA. 2012.
[64] X. Glorot, Y. Bengio, Understanding the difficulty of training deep feedforward neural networks, in: Proceedings of the International Conference on Artificial Intelligence and Statistics (AISTATS10), Society for Artificial Intelligence and Statistics, Sardinia, Italy, 2010.
[65] X. Glorot, A. Bordes, Y. Bengio, Deep sparse rectifier neural networks, in: G.J. Gordon, D.B. Dunson (Eds.), Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics (AISTATS-11), vol. 15, 2011, pp. 315–323.
[66] P. Ekman, W. Friesen, Facial Action Coding System: A Technique for the Measurement of Facial Movement, Consulting Psychologists Press, Palo Alto, 1978.
[67] W. Gu, C. Xiang, Y. Venkatesh, D. Huang, H. Lin, Facial expression recognition using radial encoding of local gabor features and classifier synthesis, Pattern Recognit. 45 (1) (2012) 80–91, http://dx.doi.org/10.1016/j.patcog.2011.05.006.
[68] L. Zhong, Q. Liu, P. Yang, B. Liu, J. Huang, D. Metaxas, Learning active facial patches for expression analysis, in: 2012 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2012, pp. 2562–2569.
[69] S.H. Lee, W.J. Baddar, Y.M. Ro, Collaborative expression representation using peak expression and intra class variation face images for practical subject-independent emotion recognition in videos, Pattern Recognit. 54 (2016) 52–67, http://dx.doi.org/10.1016/j.patcog.2015.12.016.
[70] F.D. la Torre, W.S. Chu, X. Xiong, F. Vicente, X. Ding, J. Cohn, Intraface, in: 2015 11th IEEE International Conference and Workshops on Automatic Face and Gesture Recognition (FG), vol. 1, 2015, pp. 1–8 (http://dx.doi.org/10.1109/FG.2015.7163082).
[71] J. Cohn A. Zlochower, A Computerized Analysis of Facial Expression: Feasibility of Automated Discrimination, vol. 2. American Psychological Society, 1995, p. 6.
[72] M. Xue, A. Mian, W. Liu, L. Li, Fully automatic 3d facial expression recognition using local depth features, in: IEEE Winter Conference on Applications of Computer Vision, 2014, pp. 1096–1103 (http://dx.doi.org/10.1109/WACV.2014.6835736).
[73] T. Sha, M. Song, J. Bu, C. Chen, D. Tao, Feature level analysis for 3d facial expression recognition, Neurocomputing 74 (12–13) (2011) 2135–2141, http://dx.doi.org/10.1016/j.neucom.2011.01.008.
[74] A. Maalej, B.B. Amor, M. Daoudi, A. Srivastava, S. Berretti, Shape analysis of local facial patches for 3d facial expression recognition, Pattern Recognit. 44 (8) (2011) 1581–1589, http://dx.doi.org/10.1016/j.patcog.2011.02.012.
[76] M. Liu, S. Shan, R. Wang, X. Chen, Learning expressionlets on spatio-temporal manifold for dynamic facial expression recognition, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2014, pp. 1749–1756.
[77] D. Mery, K. Bowyer, Automatic facial attribute analysis via adaptive sparse representation of random patches, Pattern Recognit. Lett. 68 (Part 2) (2015) 260–269 (Special Issue on "Soft Biometrics", .
[78] M.H. Siddiqi, R. Ali, A.M. Khan, Y.T. Park, S. Lee, Human facial expression recognition using stepwise linear discriminant analysis and hidden conditional random fields, IEEE Trans. Image Process. 24 (4) (2015) 1386–1398, http://dx.doi.org/10.1109/TIP.2015.2405346.
[79] A. Zafer, R. Nawaz, J. Iqbal, Face recognition with expression variation via robust ncc, in: 2013 IEEE 9th International Conference on Emerging Technologies (ICET), 2013, pp. 1–5 (http://dx.doi.org/10.1109/ICET.2013.6743520).
[80] M.H. Siddiqi, R. Ali, A.M. Khan, E.S. Kim, G.J. Kim, S. Lee, Facial expression recognition using active contour-based face detection, facial movement-based feature extraction, and non-linear feature selection, Multimed. Syst. 21 (6) (2014) 541–555, http://dx.doi.org/10.1007/s00530-014-0400-2.
[81] M.H. Siddiqi, R. Ali, M. Idris, A.M. Khan, E.S. Kim, M.C. Whang, S. Lee, Human facial expression recognition using curvelet feature extraction and normalized mutual information feature selection, Multimed. Tools Appl. 75 (2) (2014) 935–959, http://dx.doi.org/10.1007/s11042-014-2333-3.
[82] T. Kanade, Y. Tian, J.F. Cohn, Comprehensive database for facial expression analysis, in: Proceedings of the Fourth IEEE International Conference on Automatic Face and Gesture Recognition 2000 (FG'00), IEEE Computer Society, Washington, DC, USA, 2000, p. 46.
[83] O.M. Parkhi, A. Vedaldi, A. Zisserman, Deep face recognition, in: British Machine Vision Conference, 2015, 46-53.



André Teixeira Lopes was born in Cachoeiro de Itapemirim, ES, Brazil, on March 19, 1992. He received the B. Sc. degree in computer science in 2013 from the Universidade Federal do Espírito Santo (UFES). He received the M.Sc. degree in computer science from the same university in 2016. Currently, he is a Ph.D. student and a member of the Laboratório de Computação de Alto Desempenho (LCAD – High Performance Computing Laboratory), both at UFES, in Vitória, ES, Brazil. His research interests include the following topics computer vision, image processing, machine learning and computer graphics.

Edilson de Aguiar was born in Vila Velha, ES, Brazil, on June 11, 1979. In March 2002, he received the B.Sc. degree in computer engineering from the Universidade Federal do Espírito Santo (UFES), in Vitória, ES, Brazil. He received the M.Sc. degree in computer science in December 2003 and the Ph.D. degree in computer science in December 2008, both from the Saarland University and the Max-Planck Institute for Computer Science, in Saarbrücken, Saarland, Germany. After that he worked as researcher from 2009 to 2010 at the Disney Research laboratory in Pittsburgh, USA. Since then, he has been with the Departamento de Computação e Eletrônica of UFES, in São Mateus, ES, Brazil, where he is an adjunct professor and researcher at the Laboratório de Computação de Alto Desempenho (LCAD – High Performance Computing Laboratory). His research interests are in the areas of computer graphics, computer vision, image processing and robotics. He has been involved in research projects financed through Brazilian research agencies, such as State of Espírito Santo Research Foundation (Fundação de Apoio a Pesquisa do Estado do Espírito Santo – FAPES). He has also been in the program committee and organizing committee of national and international conferences in computer science.

Alberto F. De Souza was born in Cachoeiro de Itapemirim, ES, Brazil, on October 27, 1963. He received the B. Eng. (Cum Laude) degree in electronics engineering and M.Sc. in systems engineering in computer science from the Universidade Federal do Rio de Janeiro (COPPE/UFRJ), in Rio de Janeiro, RJ, Brazil, in 1988 and 1993, respectively; and Doctor of Philosophy (Ph.D.) in computer science from the University College London, in London, United Kingdom, in 1999. He is a professor of computer science and coordinator of the Laboratório de Computação de Alto Desempenho (LCAD – High Performance Computing Laboratory) at the Universidade Federal do Espírito Santo (UFES), in Vitória, ES, Brazil. He has authored/co-authored one USA patent and over 90 publications. He has edited proceedings of four conferences (two IEEE sponsored conferences), is a standing member of the Steering Committee of the International Conference in Computer Architecture and High Performance Computing (SBAC-PAD), senior member of the IEEE, and comendador of the order of Rubem Braga.

Thiago Oliveira-Santos was born in Vitória, ES, Brazil, on December 13, 1979. In 2004, he received the B.Sc. degree in computer engineering from the Universidade Federal do Espírito Santo (UFES), in Vitória, ES, Brazil. He received the M.Sc. degree in computer science from the same university in 2006. In 2011, he received a Ph. D. degree in biomedical engineering from the University of Bern in Switzerland, where he also worked as a post-doctoral researcher until 2013. Since then, he has been working as an adjunct professor at the Department of Computer Science of UFES in Vitoria, ES, Brazil. His research activities are performed at the Laboratório de Computação de Alto Desempenho (LCAD – High Performance Computing Laboratory) and include the following topics computer vision, image processing, computer graphics, and robotics.